การทดสอบอัตโนมัติสำหรับระบบฟันด์คอนเน็กต์

Automated Test for FundConnext System

พงษ์ศิริ ศิริพงษ์

สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COOPERATIVE EDUCATION

Automated Test for FundConnext System

PONGSIRI SIRIPONG

A COOPERATIVE EDUCATION SUBMITTED IN PARTIAL

FULFILLMENT OF THE REQUIREMENT FOR

THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)

DEPARTMENT OF COMPUTER SCIENCE, SCHOOL OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2022

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	การทดสอบอัตโนมัติสำหรับระบบฟันด์คอนเน็กต์ Automated Test for FundConnext System
ชื่อนักศึกษา	นายพงษ์ศิริ ศิริพงษ์ 62050194
ปริญญา	วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2565
อาจารย์ที่ปรึกษา	ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์

คณะวิทยาศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้สหกิจศึกษานี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) ประจำปีการศึกษา 2565

คณะกรรมการสอบ	ลายมือชื่อ
ดร.อัคเดช อุดมชัยพร ประธานกรรมการ	
ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์ กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	การทดสอบอัตโนมัติสำหรับระบบฟันด์คอนเน็กต์ Automated Test for FundConnext System
ชื่อนักศึกษา	นายพงษ์ศิริ ศิริพงษ์ 62050194
ปริญญา	วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2565
อาจารย์ที่ปรึกษา	ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์

บทคัดย่อ

สหกิจศึกษานี้เกี่ยวกับการทดสอบอัตโนมัติให้กับซอฟต์แวร์ของระบบฟันด์คอนเน็กต์ซึ่งเป็นระบบที่ตลาดหลักทรัพย์ใช้ในการจัดการข้อมูลของบริษัทหลักทรัพย์และบริษัทจัดการลงทุนเพื่อช่วยในการลดปัญหาการเก็บข้อมูลที่ซับซ้อนและสร้างความสะดวกในการทำรายการระหว่างบริษัทหลักทรัพย์และบริษัทจัดการลงทุน งานที่รับผิดชอบคือการทดสอบอัตโนมัติของโค้ดที่เกี่ยวกับข้อมูลนำเข้า ข้อมูลที่แสดงในตาราง ไฟล์และเนื้อหาของไฟล์ที่ระบบสร้าง และผลลัพธ์ที่ได้จากการติดต่อกับฟังก์ชันเอพีไอ ซึ่งขอบเขตของงานยังรวมไปถึงการวิเคราะห์ความต้องการและจัดเตรียมข้อมูลก่อนนำข้อมูลไปทำการทดสอบ นอกจากนี้งานที่ได้รับมอบหมายยังรวมไปถึงการทดสอบการทำงานแบบแมนนวล และการบำรุงรักษาสคริปต์การทดสอบอัตโนมัติ เครื่องมือที่ใช้ในการทดสอบคือไซเปรสและโรบอตเฟรมเวิร์ค

คำสำคัญ : การทดสอบซอฟต์แวร์อัตโนมัติ, โรบอตเฟรมเวิร์ค, ไซเปรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Automated Test for FundConnext System
Students	Mr. Pongsiri Siripong Student ID 62050194
Degree	Bachelor of Science (Computer Science)
Department	Computer Science
Faculty	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2022
Advisor	Rungrat Wiangsripanawan, Ph.D

Abstract

This cooperative study involves doing automated software testing for programs or codes in FundConnext system, a system used by the Stock Exchange of Thailand to serve as a central system for managing data for the Securities Company (SA) and Asset Management Company (AMC) in order to reduce the complexity of data collection and facilitate transactions between the Securities Company and Asset Management Company. The responsible task is to conduct the automated testing of the code related to the following data: input data, data displayed in the table, data in the files generated by the system and the returned data from API functions. In each test, test requirement analysis and test data preparation are also done by the author. The assigned tasks also include functional manual tests and automated test script maintenance. Cypress and Robot framework are used as the tools for the testing.

Keywords : Automated Software Testing, Robot Framework, Cypress

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการสหกิจศึกษานี้เป็นการศึกษาการทดสอบอัตโนมัติสำหรับระบบฟันด์คอนเน็กต์ของบริษัทตลาดหลักทรัพย์แห่งประเทศไทยสามารถสำเร็จลุล่วงไปได้ด้วยดีเพราะความกรุณาจาก นายภูริเดช แก้ววิเศษ (พี่เลี้ยง) และสมาชิกในทีมฟันด์คอนเน็กต์ ที่คอยให้คำปรึกษา คำแนะนำ และข้อคิดเห็นที่เป็นประโยชน์แก่ผู้จัดทำสำหรับข้อผิดพลาดต่างๆเพื่อให้ผู้จัดทำได้เรียนรู้ และแก้ไขพัฒนาต่อไป ผู้จัดทำขอกราบขอบพระคุณในการเป็นที่ปรึกษาและให้การช่วยเหลือตลอดมา

ขอขอบพระคุณ ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์ ที่แนะนำแนวทางในการแก้ไขปัญหาและให้ความรู้ต่างๆ แก่ผู้จัดทำทั้งเป็นอาจารย์ที่ปรึกษาในการทำโครงการสหกิจศึกษาครั้งนี้

สุดท้ายนี้ขอขอบพระคุณบิดา มารดา และสมาชิกครอบครัวทุกคน ผู้ซึ่งให้การสนับสนุนและให้กำลังใจ รวมถึงเพื่อนๆทุกคนที่คอยให้คำปรึกษาในเรื่องต่าง ๆ เป็นอย่างดีเสมอ

นายพงษ์ศิริ ศิริพงษ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก
บทคัดย่อภาษาอังกฤษ ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ฉ
สารบัญรูป ..... ซ
คำย่อ/สัญลักษณ์ ..... ฌ
บทที่ 1 บทนำ 1
1.1 ความเป็นมาและความสำคัญ 1
1.2 วัตถุประสงค์ของสหกิจศึกษา 2
1.3 ขอบเขตของสหกิจศึกษา 2
1.4 ประโยชน์ที่คาดว่าจะได้รับ 2
1.5 ขั้นตอนในการดำเนินงาน 3
บทที่ 2 ทฤษฎีและเครื่องมือที่เกี่ยวข้อง 4
2.1 Agile and Scrum 4
2.2 การทดสอบซอฟต์แวร์ 6
2.2.1 วัตถุประสงค์ของการทดสอบ 6
2.2.2 กระบวนการทดสอบ 6
2.2.3 กระบวนการทดสอบในระบบฟันด์คอนเน็กต์ 7
2.3 Manual Test 8
2.4 Automated Test 8
2.5 Test automation maintenance 8
2.6 เครื่องมือที่ใช้ 10
2.6.1 Postman 10
2.6.2 Cypress 10
บทที่ 3 วิธีการดำเนินงานสหกิจศึกษา 12
3.1 กระบวนการดำเนินงาน 12
3.2 วิธีการดำเนินงาน 14
3.2.1 Functional Test โดยทดสอบในรูปแบบ Manual Test 15
3.2.2 Automated Test แบบ Input data 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.2.3 Automated Test แบบ Look and Filed 44
3.2.4 Automated Test แบบ Look and Filed (ตาราง) 55
3.2.5 Automated Test แบบ Validate File 70
3.2.6 Automated Test API 82
3.3 วิธีการบำรุงและดูแลรักษา 120
3.3.1 ข้อมูลสำหรับการทดสอบ 120
3.3.2 Flankly Test 120
3.3.3 กระทบจากการเพิ่มเติมหรือแก้ไขระบบ 121
3.3.4 Localhost ผ่านแต่ Serverไม่ผ่าน 121
3.4 การย้าย Automated Test Tools 121
บทที่ 4 ผลการดำเนินงานสหกิจศึกษาและการอภิปรายผล 122
4.1 ผลการทดสอบ Functional Test แบบ Manual Test 122
4.2 ผลการทดสอบ Automated Test แบบ Input Data 152
4.3 ผลการทดสอบ Automated Test แบบ Look and Field 162
4.4 ผลการทดสอบ Automated Test แบบ Look and Field (ตาราง) 167
4.5 ผลการทดสอบ Automated Test แบบ Validate File 179
4.6 ผลการทดสอบ Automated Test API 184
บทที่ 5 สรุปผลการดำเนินงานสหกิจศึกษาและข้อเสนอแนะ 212
เอกสารอ้างอิง 213
ภาคผนวก ก 215
ภาคผนวก ข 219
ภาคผนวก ค 220

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

3.1 Test Script การทดสอบ Functional Test แบบ Manual Test ตัวอย่างที่ 1 15
3.2 Test Script การทดสอบ Functional Test แบบ Manual Test ตัวอย่างที่ 2 19
3.3 Test Script การทดสอบ Automated Functional Test แบบ input Data ตัวอย่างที่ 1 30
3.4 Test Script การทดสอบ Automated Functional Test แบบ input Data ตัวอย่างที่ 2 38
3.5 Test Script การทดสอบ Automated Functional Test แบบ Look and Field ตัวอย่างที่ 1 47
3.6 Test Script การทดสอบ Automated Functional Test แบบ Look and Field ตัวอย่างที่ 2 52
3.7 Test Script การทดสอบ Automated Functional Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 1 59
3.8 Test Script การทดสอบ Automated Functional Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 63
3.9 Test Script การทดสอบ Automated Functional Test แบบ Validate File ตัวอย่างที่ 1 73
3.10 Test Script การทดสอบ Automated Functional Test แบบ Validate File ตัวอย่างที่ 2 77
3.11 Test Script การทดสอบ Automated Test API ตัวอย่างที่ 1 82
3.12 Test Script การทดสอบ Automated Test API ตัวอย่างที่ 2 91
3.13 Test Script การทดสอบ Automated Test API ตัวอย่างที่ 3 98
3.14 ตารางเปรียบเทียบคำสั่งการเขียน Automated Test ของ Robot และ Cypress 121
4.1 ผลการทดสอบ Functional Test แบบ Manual Test ตัวอย่างที่ 1 122
4.2 ผลการทดสอบ Functional Test แบบ Manual Test ตัวอย่างที่ 2 134
4.3 ผลการทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 152
4.4 ผลการทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 2 160
4.5 ผลการทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 1 162
4.6 ผลการทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 2 165
4.7 ผลการทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 1 167
4.8 ผลการทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 171

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

4.9 ผลการทดสอบ Automated Test แบบ Validate File ตัวอย่างที่ 1 179
4.10 ผลการทดสอบ Automated Test แบบ Validate File ตัวอย่างที่ 2 182
4.11 ผลการทดสอบ Automated Test API ตัวอย่างที่ 1 184
4.12 ผลการทดสอบ Automated Test API ตัวอย่างที่ 2 192
4.13 ผลการทดสอบ Automated Test API ตัวอย่างที่ 3 197

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	หน้าจอ UI การรัน Cypress.io (1)	11
2.2	หน้าจอ UI การรัน Cypress.io (2)	11
3.1	Diagram แสดงขั้นตอนการทดสอบระบบของทีมฟันด์คอนเน็กต์	12
3.2	ตัวอย่างไฟล์ที่ใช้ในการ Upload Point Code	18
3.3	หน้าจอเมนู Code Management ของ SA	18
3.4	หน้าจอเมนู Code Management ของ TSD	19
3.5	หน้าจอเมนู Change Unitholder to Another Account (1)	25
3.6	หน้าจอเมนู Change Unitholder to Another Account (2)	26
3.7	หน้าจอสำหรับกรอกข้อมูล Account	26
3.8	ช่องรับข้อมูลแบบธรรมดา	27
3.9	ช่องรับข้อมูลแบบ DropDown	28
3.10	ช่องรับข้อมูลแบบตัวเลือก	28
3.11	ช่องรับข้อมูลแบบ Checkbox	29
3.12	ช่องรับข้อมูลแบบ Search Option	29
3.13	ตัวอย่างการใส่ข้อมูล Personal Information	33
3.14	ตัวอย่าง API ในการลบข้อมูลบัญชี Individual	33
3.15	หน้าจอเมนูเปิดบัญชี Individual Background Tab	34
3.16	ตัวอย่างการเข้าสู่ระบบ FundConnext	34
3.17	ตัวอย่างการเรียกใช้ API ในการลบข้อมูลบัญชี Individual	34
3.18	Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 1	35
3.19	Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 2	35
3.20	Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 3	35
3.21	Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 4	35
3.22	Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 5	36
3.23	Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 6	36
3.24	Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 7	36
3.25	Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 8	37
3.26	Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 9	37
3.27	Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 10	37
3.28	Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 11	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.29	Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 12	38
3.30	Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 13	38
3.31	หน้าจอเมนูเปิดบัญชี ByWhom ผ่านบัญชี Juristic (1)	39
3.32	หน้าจอเมนูเปิดบัญชี ByWhom ผ่านบัญชี Juristic (2)	40
3.33	ตัวอย่าง API ในการลบข้อมูลบัญชี Juristic	40
3.34	ตัวอย่างการเรียกใช้ API ในการลบข้อมูลบัญชี Individual	40
3.35	Script ในการทำ Automated Test input Data ตัวอย่างที่ 2 กรณีที่ 1	41
3.36	Script ในการทำ Automated Test input Data ตัวอย่างที่ 2 กรณีที่ 2	41
3.37	Script ในการทำ Automated Test input Data ตัวอย่างที่ 2 กรณีที่ 3 (1)	41
3.38	Script ในการทำ Automated Test input Data ตัวอย่างที่ 2 กรณีที่ 3 (2)	42
3.39	Script ในการทำ Automated Test input Data ตัวอย่างที่ 2 กรณีที่ 3 (3)	42
3.40	Script ในการทำ Automated Test input Data ตัวอย่างที่ 2 กรณีที่ 3 (4)	42
3.41	Script ในการทำ Automated Test input Data ตัวอย่างที่ 2 กรณีที่ 3 (5)	43
3.42	ช่องรับข้อมูลแบบธรรมดา	44
3.43	ช่องรับข้อมูลแบบ Dropdown	45
3.44	ช่องรับข้อมูลแบบตัวเลือก	45
3.45	ช่องรับข้อมูลแบบ Checkbox	46
3.46	ช่องรับข้อมูลที่ไม่ทำงาน	46
3.47	ตัวอย่างหน้าจอ Complete	47
3.48	หน้าจอเมนู Account Tab	48
3.49	Script ในการทำ Automated Test look and Field ฟังก์ชันการตรวจสอบ Account	49
3.50	Script ในการทำ Automated Test look and Field คำสั่งการตรวจสอบ Account	49
3.51	Script ในการทำ Automated Test look and Field ตัวอย่างที่ 1 กรณีที่ 1	50
3.52	Script ในการทำ Automated Test look and Field ตัวอย่างที่ 1 กรณีที่ 2	50
3.53	Script ในการทำ Automated Test look and Field ตัวอย่างที่ 1 กรณีที่ 3	51
3.54	Script ในการทำ Automated Test look and Field ตัวอย่างที่ 1 กรณีที่ 4	51
3.55	หน้าจอเมนู SA Profile (1)	52
3.56	หน้าจอเมนู SA Profile (2)	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.57 Script ในการทำ Automated Test look and Field ฟังก์ชันตรวจสอบ SA Profile.... 53
3.58 Script ในการทำ Automated Test look and Field ตัวอย่างที่ 2 กรณีที่ 1 54
3.59 Script ในการทำ Automated Test look and Field ตัวอย่างที่ 2 กรณีที่ 2 54
3.60 ฟังก์ชันการ Build Column 55
3.61 ฟังก์ชันการ Filter 56
3.62 ฟังก์ชันการ Sort 56
3.63 ฟังก์ชันการ Get Row / Get Column 57
3.64 ฟังก์ชันการ Clear 57
3.65 ฟังก์ชันการ Filter / Clear / Sort 58
3.66 หน้าจอแสดงข้อมูล Failed XWT Order 57
3.67 หน้าจอเมนู Fail XWT Order (1) 61
3.68 หน้าจอเมนู Fail XWT Order (2) 61
3.69 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 1 กรณีที่ 1 .. 62
3.70 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 1 กรณีที่ 2 .. 62
3.71 หน้าจอเมนู Balance Inquiry (1) 64
3.72 หน้าจอเมนู Balance Inquiry (2) 65
3.73 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 1 .. 65
3.74 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 2 .. 65
3.75 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 3 .. 66
3.76 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 4 .. 66
3.77 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 5 .. 67
3.78 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 6 .. 67
3.79 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 7 .. 68
3.80 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 8 .. 68
3.81 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 9 .. 68
3.82 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 10 69
3.83 ฟังก์ชันการ Upload แบบเตรียมไฟล์ 70
3.84 ฟังก์ชันการ Upload แบบเขียนไฟล์ 71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.85 ฟังก์ชันการ Compare text File 71
3.86 ฟังก์ชันการ Compare CSV File 72
3.87 ฟังก์ชันการ Compare Excel File 72
3.88 หน้าจอ Plugin สำหรับ Cypress 72
3.89 ตัวอย่างไฟล์ทดสอบ Upload more than 80 Char 74
3.90 หน้าจอเมนู Upload Attachment (1) 75
3.91 หน้าจอเมนู Upload Attachment (2) 75
3.92 Script ในการทำ Automated Validate File ตัวอย่างที่ 1 กรณีที่ 1 76
3.93 Script ในการทำ Automated Validate File ตัวอย่างที่ 1 กรณีที่ 2 76
3.94 ตัวอย่างไฟล์ทดสอบ Compare Balance Inquiry (CSV) 78
3.95 หน้าจอเมนู Balance Inquiry (CSV) 78
3.96 Script ในการทำ Automated Validate File ตัวอย่างที่ 2 เตรียมการ 78
3.97 Script ในการทำ Automated Validate File ตัวอย่างที่ 2 รายละเอียดไฟล์ Balance.. 79
3.98 Script ในการทำ Automated Validate File ตัวอย่างที่ 2 การ Insert log upload 79
3.99 Script ในการทำ Automated Validate File ตัวอย่างที่ 2 ตรวจสอบไฟล์ CSV 79
3.100 Script ในการทำ Automated Validate File ตัวอย่างที่ 2 กรณีที่ 1 80
3.101 Script ในการทำ Automated Validate File ตัวอย่างที่ 2 กรณีที่ 2 80
3.102 Script ในการทำ Automated Validate File ตัวอย่างที่ 2 กรณีที่ 3 81
3.103 Script ในการทำ Automated Validate File ตัวอย่างที่ 2 กรณีที่ 4 81
3.104 ข้อมูล Fail XWT Order ที่เตรียมก่อนทดสอบ API 84
3.105 ตัวอย่าง API get Failed XWT Order 84
3.106 Script ในการทำ Automated Test API ตัวอย่างที่ 1 เรียกใช้ API 85
3.107 Script ในการทำ Automated Test API ตัวอย่างที่ 1 เตรียมการ 85
3.108 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 1 85
3.109 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 2 86
3.110 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 3 87
3.111 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 4 88
3.112 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 5 88
3.113 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 6 89
3.114 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 7 89

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.115 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 8 90
3.116 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 9 90
3.117 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 10 90
3.118 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 11 91
3.119 Script ในการทำ Automated Test API ตัวอย่างที่ 2 เรียกใช้ API Create Order 92
3.120 Script ในการทำ Automated Test API ตัวอย่างที่ 2 Create Order 93
3.121 Script ในการทำ Automated Test API ตัวอย่างที่ 2 เรียกใช้ API Delete Order 93
3.122 Script ในการทำ Automated Test API ตัวอย่างที่ 2 Delete Order 94
3.123 ตัวอย่าง API Approved CrossAmcSwitching 94
3.124 Script ในการทำ Automated Test API ตัวอย่างที่ 2 กรณีที่ 1 95
3.125 Script ในการทำ Automated Test API ตัวอย่างที่ 2 กรณีที่ 2 95
3.126 Script ในการทำ Automated Test API ตัวอย่างที่ 2 กรณีที่ 3 96
3.127 Script ในการทำ Automated Test API ตัวอย่างที่ 2 กรณีที่ 4 96
3.128 Script ในการทำ Automated Test API ตัวอย่างที่ 2 กรณีที่ 5 97
3.129 Script ในการทำ Automated Test API ตัวอย่างที่ 2 กรณีที่ 6 97
3.130 Script ในการทำ Automated Test API ตัวอย่างที่ 2 กรณีที่ 7 98
3.131 ตัวอย่าง Reset Unitholder Bank Account 102
3.132 ตัวอย่าง API Update Unitholder bank 102
3.133 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 1 103
3.134 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 2 104
3.135 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 3 105
3.136 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 4 106
3.137 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 5 107
3.138 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 6 108
3.139 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 7 109
3.140 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 8 110
3.141 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 9 110
3.142 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 10 111
3.143 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 11 113
3.144 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 12 114

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.145 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 13 115
3.146 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 14 116
3.147 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 15 117
3.148 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 16 118
3.149 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 17 119
4.1 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 1 125
4.2 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 2 125
4.3 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 3 126
4.4 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 4 126
4.5 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 5 127
4.6 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 6 127
4.7 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 7 128
4.8 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 8 128
4.9 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 9 129
4.10 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 10 129
4.11 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 11 130
4.12 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 12 130
4.13 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 13 130
4.14 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 14 131
4.15 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 15 (1) 131
4.16 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 15 (2) 131
4.17 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 16 132
4.18 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 17 132
4.19 ตัวอย่างไฟล์ผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 17 132
4.20 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 18 133
4.21 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 19 133
4.22 หน้าจอสำหรับกรอก Change Unitholder to Another Account 140
4.23 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 1 140
4.24 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 2 141
4.25 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 3 141

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

4.26 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 4 142
4.27 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 5 142
4.28 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 6 143
4.29 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 7 143
4.30 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 8 144
4.31 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 9 144
4.32 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 10 145
4.33 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 11 145
4.34 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 12 146
4.35 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 13 146
4.36 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 14 147
4.37 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 15 147
4.38 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 16 148
4.39 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 17 148
4.40 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 18 149
4.41 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 19 149
4.42 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 20 150
4.43 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 21 150
4.44 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 22 151
4.45 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 23 151
4.46 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 24 152
4.47 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 155
4.48 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 1 156
4.49 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 2 156
4.50 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 3 156
4.51 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 4 157

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

4.52 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 5 157
4.53 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 6 158
4.54 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 7 158
4.55 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 8 158
4.56 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 9 159
4.57 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 10 159
4.58 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 11 159
4.59 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 12 159
4.60 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 13 160
4.61 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 2 161
4.62 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 2 กรณีที่ 1 161
4.63 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 2 กรณีที่ 2 162
4.64 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 2 กรณีที่ 3 162
4.65 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 1 163
4.66 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 1 กรณีที่ 1 164
4.67 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 1 กรณีที่ 2 164

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

4.68 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 1 กรณีที่ 3 164
4.69 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 1 กรณีที่ 4 165
4.70 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 2..... 166
4.71 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 2 กรณีที่ 1 166
4.72 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 2 กรณีที่ 2 167
4.73 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 1 169
4.74 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 1 กรณีที่ 1 (1) 170
4.75 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 1 กรณีที่ 1 (2) 170
4.76 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 1 กรณีที่ 2 171
4.77 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 173
4.78 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 1 173
4.79 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 2 174
4.80 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 3 174
4.81 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 4 (1) 175
4.82 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 4 (2) 175

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

4.83 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 5 176
4.84 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 6 176
4.85 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 7 177
4.86 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 8 177
4.87 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 9 178
4.88 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 10 178
4.89 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Validate File ตัวอย่างที่ 1 180
4.90 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Validate File ตัวอย่างที่ 1 กรณีที่ 1 180
4.91 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Validate File ตัวอย่างที่ 1 กรณีที่ 2 181
4.92 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Validate File ตัวอย่างที่ 1 กรณีที่ 3 181
4.93 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Validate File ตัวอย่างที่ 1 กรณีที่ 4 182
4.94 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Validate File ตัวอย่างที่ 2 183
4.95 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 186
4.96 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 1 186
4.97 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 2 187
4.98 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 3 187
4.99 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 4 188
4.100 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 5 188
4.101 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 6 189
4.102 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 7 189

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

4.103 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 8 190
4.104 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 9 190
4.105 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 10 191
4.106 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 11 191
4.107 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 2 193
4.108 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 2 กรณีที่ 1 194
4.109 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 2 กรณีที่ 2 194
4.110 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 2 กรณีที่ 3 195
4.111 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 2 กรณีที่ 4 195
4.112 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 2 กรณีที่ 5 196
4.113 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 2 กรณีที่ 6 196
4.114 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 2 กรณีที่ 7 197
4.115 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 201
4.116 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 1 202
4.117 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 2 202
4.118 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 3 203
4.119 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 4 203
4.120 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 5 204
4.121 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 6 204
4.122 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 7 205
4.123 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 8 205
4.124 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 9 206
4.125 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 10 206
4.126 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 11 207
4.127 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 12 207
4.128 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 13 208
4.129 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 14 208
4.130 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 15 209
4.131 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 16 209
4.132 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 17 210

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำย่อ/สัญลักษณ์

คำย่อ/สัญลักษณ์	คำอธิบาย
AMC	Asset Management Company (AMC) บริษัทหลักทรัพย์จัดการลงทุน เป็นบริษัทที่ลงทุนในกองทุนรวม โดยนำเงินจากลูกค้าแล้วนำมาลงทุนที่แตกต่างกันไป เช่น หุ้นพันธบัตร อสังหาริมทรัพย์ ห้างหุ้นส่วนจำกัด,และอื่น ๆ โดย AMC จะจัดการป้องกันความเสี่ยงและแผนบำนาญ เพื่อให้บริการแก่นักลงทุนรายย่อยได้ดียิ่งขึ้น
API	เป็นตัวกลางที่ค่อยรับคำสั่งต่างๆ ประมวลผล และการส่งข้อมูลกลับคืนไปยังผู้ส่งแบบอัตโนมัติ
Branch	สาขา
CSV	Text File สำหรับเก็บข้อมูลแบบตาราง โดยใช้จุลภาค (,) แบ่งข้อมูลในแต่ละหลัก (Column) และใช้การเว้นบรรทัดแทนการแบ่งแถว (Row)
CrossAmcSwitching	การสับเปลี่ยนกองทุนข้าม บลจ.
HQ	สำนักงานใหญ่
OMN	Omnibus การไม่เปิดเผยรายละเอียดของเจ้าของบัญชี
POINT Code	Codeที่สามารถใช้สำหรับการซื้อกองทุน
RED	Redemption การขายกองทุน
SA	Selling Agent (SA) บริษัทนายหน้าซื้อขายหน่วยลงทุนให้บริการด้านกองทุนรวม เป็นนายหน้าในการซื้อขายกองทุนให้แก่นักลงทุน
SEG	Segregate การเปิดเผยรายละเอียดของเจ้าของบัญชี
SUB	Subscription การซื้อกองทุน
TXT	ไฟล์ Text
UH	Unitholder หน่วยลงทุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

ในปัจจุบันระบบซอฟต์แวร์ต่างๆนั้นเข้ามาช่วยอำนวยความสะดวกในการใช้ชีวิตให้กับคนทั่วไปทุกคน ไม่ว่าจะเกี่ยวข้องกับการใช้ชีวิตในประจำวันต่างๆ เช่น สั่งอาหาร การเดินทาง จองที่พัก ใช้เพื่อคลายเครียดหรือนำมาใช้ในการประกอบอาชีพก็ตาม ที่ระบบซอฟต์แวร์นั้นสามารถเข้ามาเป็นส่วนหนึ่งในชีวิตประจำวันมากขึ้นนั้น เนื่องจากว่าระบบซอฟต์แวร์ต่างๆได้ถูกพัฒนาขึ้นอย่างก้าวกระโดด ซึ่งปัจจัยที่ทำให้ซอฟต์แวร์มีการพัฒนาได้อย่างก้าวกระโดดมีเหตุผลมากมายไม่ว่าจะเป็นระบบมีการประมวลผลที่มีความเร็วมากขึ้น การจัดเก็บข้อมูลได้มากขึ้น ความสามารถในการส่งข้อมูลในปริมาณมากขึ้นแต่ว่าใช้เวลาในการส่งที่น้อยลง รวมทั้งการรับส่งข้อมูลที่เกี่ยวกับธุรกรรมทางการเงินในรูปแบบต่างๆ

ดังนั้นระบบซอฟต์แวร์จึงถูกเปลี่ยนความสำคัญจากสิ่งที่เข้ามาช่วยอำนวยความสะดวกมาเป็นสิ่งที่สำคัญในชีวิตประจำวันและนั้นจึงเป็นสาเหตุที่หน่วยงานต่างๆให้ความสนใจในการพัฒนาซอฟต์แวร์ใหม่ๆเพื่อตอบสนองความต้องการของผู้ใช้งานให้มากขึ้น หนึ่งในนั้นคือซอฟต์แวร์ที่เกี่ยวข้องกับธุรกรรมทางการเงินเหตุนี้จึงได้เกิดแนวคิด ขั้นตอน กระบวนการ หรือรูปแบบต่างๆที่นำเข้ามาปรับใช้ในกระบวนการพัฒนาซอฟต์แวร์ ซึ่งการ “ทดสอบคุณภาพของซอฟต์แวร์” เป็นส่วนหนึ่งที่มีความสำคัญมากในกระบวนการพัฒนาซอฟต์แวร์เช่นกันเพราะหลังจากที่ซอฟต์แวร์มีการนำไปใช้งานจริงนั้นไม่สามารถเกิดปัญหาที่กระทบต่อการใช้งานของผู้ใช้งานได้ ซึ่งการที่จะทำให้ประสิทธิภาพของซอฟต์แวร์ก่อนนำไปใช้จริงมีปัญหาน้อยที่สุดเท่าที่จะเป็นไปได้ ทำให้เกิดกระบวนการทดสอบซอฟต์แวร์ขึ้นซึ่งกระบวนการหลักๆนั้นถูกเรียกว่า Manual Test แต่ว่าการทดสอบแบบ Manual Test ในบางครั้งอาจจะเกิดข้อผิดพลาดขึ้นจากการเพิ่มหรือแก้ไขงานบางส่วน หรือมีการใช้เวลาและบุคลากรในการทดสอบที่มากเกินความจำเป็น

ทำให้ในปัจจุบันนั้นมีการนำ Automated Test Tools เข้ามาเพื่อเป็นตัวช่วยใช้ในการตรวจสอบอีกทาง และลดความผิดพลาดเมื่อมีการเพิ่มหรือแก้ไขงานบางส่วนเพิ่มเติม รวมทั้ง Automated Test ยังสามารถลดระยะเวลาที่ใช้ในการทดสอบระบบได้อีกด้วย การทำ Automated Test นั้นจะถูกจัดทำขึ้นมาตาม Test Script ที่วางไว้แล้วนำมาเขียนเป็น Code เพื่อใช้ในการทดสอบระบบ ซึ่งในปัจจุบันก็มีหลากหลาย Tools หรือ Framework ต่างๆที่ให้เลือกใช้ในการทำ Automated Test เช่น Robot Framework , Cypress เป็นต้น โดยตัวอย่างที่จะนำมาแสดงส่วนใหญ่จะนำมาจากระบบฟันด์คอนเน็กต์ที่เป็นระบบซอฟต์แวร์ที่เกี่ยวกับธุรกรรมทางการเงินที่อยู่ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบของระบบส่วนกลางในการซื้อขาย สับเปลี่ยนกองทุน ซึ่งการทดสอบระบบของฟันด์คอนเน็กต์ได้เลือกใช้เครื่องมือในการทำ Automated Test 2 ตัว คือ Cypress และ Robot Framework

ระบบฟันด์คอนเน็กต์เป็นเว็บแอปพลิเคชั่นที่ทางตลาดหลักทรัพย์จัดทำขึ้นมาเพื่อที่ใช้เป็นระบบส่วนกลางในการจัดการข้อมูลบริษัทหลักทรัพย์ (SA) และบริษัทจัดการการลงทุน (AMC) ตั้งแต่กระบวนการซื้อขายกองทุน การเปิดบัญชี การจัดตั้งกองทุน การทำรายการซื้อขาย การรับส่งรายการการจากบริษัทหลักทรัพย์ (SA) ไปยังบริษัทจัดการการลงทุน (AMC) ทั้งหมดก็เพื่อลดปัญหาการเก็บข้อมูลที่ซับซ้อน และสร้างความสะดวกในการทำรายการระหว่างบริษัทหลักทรัพย์ (SA) และบริษัทจัดการการลงทุน (AMC) โครงการฟันด์คอนเน็กต์เป็นระบบซอฟต์แวร์ขนาดใหญ่ และเกี่ยวข้องกับเรื่องการเงิน จึงจำเป็นที่จะต้องมีการตรวจสอบคุณภาพของซอฟต์แวร์เพื่อให้ซอฟต์แวร์เกิดข้อผิดพลาดน้อยที่สุด

1.2 วัตถุประสงค์ของสหกิจศึกษา

1) เพื่อสามารถวิเคราะห์ Requirement ที่ได้รับมา
2) เพื่อเข้าใจในการเขียน Test Script ก่อนนำไป Manual Test และ Automated Test
3) เพื่อทดสอบระบบฟันด์คอนเน็กต์ ในรูปแบบ Manual Test และ Automated Test
4) เตรียมข้อมูลต่างๆก่อนเริ่มทำการทดสอบระบบ
5) Maintenance Automated Test Script ในเครื่อง Server ของระบบเพื่อควบคุมคุณภาพของระบบฟันด์คอนเน็กต์

1.3 ขอบเขตของสหกิจศึกษา

1) วิเคราะห์ Requirement ที่ได้รับมาจาก Project Owner
2) จัดทำ Test Script ตาม Requirement ที่ได้รับมอบหมายมาในแต่ละ Sprint
3) นำ Test Script มาจัดการทำ Manual Test และ Automated Test
4) สามารถเข้าใจ และเขียน Automated Test โดยใช้ Cypress
5) ปรับปรุง และ รักษา Automated Test Script

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) ระบบฟันด์คอนเน็กต์มีการทำงานที่ถูกต้องตาม Requirement ที่ได้รับมอบหมายมา
2) Automated Test ที่จัดทำสามารถทดสอบระบบได้อย่างถูกต้องโดยไม่เกิดปัญหา
3) สามารถรักษาคุณภาพการทำงานของระบบฟันด์คอนเน็กต์ได้
4) ควบคุมคุณภาพการทำงานของ Automated Test

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ขั้นตอนในการดำเนินงาน

ผู้จัดทำได้รับมอบหมายในหน้าที่ Tester ภายในทีม จึงต้องมีการเรียนรู้ระบบการทำงานภายในทีมฟันด์คอนเน็กต์ก่อนเริ่มต้นทำงาน โดยระบบการทำงานจะเป็นในรูปแบบของ Scrum ที่จะแบ่งการทำงานเป็น Sprint โดย Sprint ละ 2 อาทิตย์ รวมถึงในทุกๆวันก่อนที่จะเริ่มการทำงานจะมีการพูดคุยเพื่อสรุปงานที่ตนเองได้ทำไปเมื่อวานกับสิ่งที่จะทำภายในวันนี้กับสมาชิกภายในทีมซึ่งกระบวนการการทำงานใน Sprint เป็นดังนี้

1) Sprint Planning

   เป็นการประชุมในทุกๆครั้งที่เริ่มในแต่ละ Sprint โดยที่ Project Owner จะอธิบายรายละเอียดเนื้อหาในแต่ละ Requirement หลังจากนั้นสมาชิกภายในทีมฝ่าย IT จะทำการประเมิณราคาขนาดของงาน ในแต่ละ Requirement และวางแผนสิ่งที่จะต้องภายใน Sprint

2) Refinement

   ขั้นตอนในการทำงานจะทำในรูปแบบเดียวกับ Sprint Planning แต่จุดประสงค์นั้นต่างออกไปนั้นก็คือการเลือก Card ที่จะนำไปทำใน Sprint ถัดไปเพื่อที่จะประเมินภาพรวมของงานก่อนที่จะลงมือทำจริง รวมถึงจะทราบใน Sprint ถัดไปนั้นควรจะนำ Card ใดเข้ามาทำบ้าง

3) Sprint Review

   นำเสนองานที่ทำภายใน Sprint ให้กับ Project Owner ว่าระบบสามารถทำงานได้ตรงตาม Requirement ที่ต้องการหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและเครื่องมือที่เกี่ยวข้อง

จากที่กล่าวมาในบทที่ 1 ระบบฟันด์คอนเน็กต์จำเป็นต้องมีการทดสอบระบบทั้งในรูปแบบ Manual Test และ Automated Test ซึ่งการทำให้การทดสอบระบบมีประสิทธิภาพนั้นจึงมีการนำแนวคิด และทฤษฎี รวมถึงเครื่องมือต่างๆที่นำเข้ามาเพื่อช่วยให้การทดสอบระบบมีประสิทธิภาพเพิ่มมากขึ้น

2.1 Agile and Scrum

การพัฒนาระบบซอฟต์แวร์ในสมัยอดีตองค์กรส่วนใหญ่จะใช้รูปแบบการทำงานแบบ Waterfall Process ที่จะมีการวางแผนก่อนเริ่มโครงการ ทั้งระยะเวลาในการพัฒนา เงิน จำนวนคน และอุปกรณ์ต่างๆ ตั้งแต่ต้นจนจบโครงการ ซึ่งนั้นทำให้เป็นเรื่องยากที่จะสามารถวางแผนในทุกส่วนให้ลงตัวและแม่นยำทั้งในเรื่องของงบประมาณและเวลาที่อาจจะมีการเพิ่มขึ้นระหว่างการพัฒนา รวมถึงการทราบข้อผิดพลาดนั้นช้า เนื่องจากว่าจะมีการทดสอบซอฟต์แวร์อยู่ในขั้นตอนของ Test ซึ่งเป็นส่วนที่การออกแบบ และพัฒนาใกล้ที่จะเสร็จสมบูรณ์แล้ว หากพบข้อผิดพลาดไม่ว่าจะเกิดจากการเข้าใจ Requirement ผิดพลาดหรือมีการเปลี่ยนแปลง Requirement ระหว่างพัฒนาซึ่งทำให้แก้ไขได้ยากหรือบางครั้งอาจจะถึงขั้นต้องทำใหม่ทั้งหมด

Agile Methodology จึงเป็นอีกแนวคิดในการพัฒนาซอฟต์แวร์ในองค์กรในปัจจุบันมักจะนำมาใช้ภายในทีมพัฒนาซอฟต์แวร์เพื่อตอบสนองการพัฒนาซอฟต์แวร์ในปัจจุบันที่เน้นตรวจสอบข้อผิดพลาดรวดเร็วนำไปแก้ไขทันทีและมีการสอบถาม Requirement กับลูกค้าอย่างสม่ำเสมอเพื่อลดข้อผิดพลาดที่อาจเกิดขึ้นจากการเข้าใจ Reqirement ไม่ตรงกัน ซึ่ง Agile เน้นการวางแผนระยะสั้น การสื่อสารภายในทีมกับลูกค้า การพบข้อผิดพลาดให้เร็ว การทำงานในระยะเวลาที่สั้นลง เช่น การวางแผนและทำงานในระยะเวลา 2 อาทิตย์ หรือทุกเดือนแล้วส่งมอบให้ลูกค้าในทันทีไม่เหมือนกับการทำงานแบบ Waterfall Process ที่จะมีการวางแผนทั้งหมดก่อนเริ่มการพัฒนาซอฟต์แวร์และส่งมอบให้ลูกค้าทีเดียวเมื่องานเสร็จสิ้น ซึ่งการทำงานแบบ Agile ไม่ได้มีการระบุขั้นตอนหรือกฎไว้อย่างชัดเจน โดยจะปรับเปลี่ยนไปตามการทำงานในแต่ละองค์กร

Scrum เป็นวิธีการทำงานที่มักนำมาใช้คู่กับ Agile โดยมีจุดประสงค์หลักๆ เพื่อลดความซ้ำซ้อนของปัญหา ลดขั้นตอนระหว่างการทำงาน และสามารถรับการเปลี่ยนแปลงที่เกิดขึ้นได้บ่อยๆ การทำงานแบบ Scrum จะมีตำแหน่งหลักๆ 3 ตำแหน่ง ได้แก่

- Project Owner : มีหน้าในการเจรจากับลูกค้าเพื่อรับความต้องการของลูกค้าแล้วนำมาวิเคราะห์แล้วนำมาเสนอให้กับสมาชิกภายในทีม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Scrum Master : ผู้ที่คอยประเมินปัญหาที่เกิดขึ้นระหว่างพัฒนาซอฟต์แวร์และคอยแก้ไขปัญหาเหล่านั้น
- Team : สมาชิกทุกคนที่อยู่ในทีมซึ่งมีความถนัดแตกต่างกันไป เช่น Dev, UX/UI Design, Tester เป็นต้น

Scrum เน้นการนำความรู้จากประสบการณ์จริงมาพัฒนาการทำงานในทีมปัจจุบันให้ดียิ่งขึ้น ประกอบด้วย 3 ส่วน ดังนี้

- ความโปร่งใส (Transparency) : ทุกคนภายในทีมต้องเห็นภาพชัดเจนและเข้าใจตรงกัน
- การตรวจสอบ (Inspection) : นำผลลัพธ์ที่ได้มาตรวจสอบว่าได้ตามเป้าหมายที่ต้องการหรือไม่
- การปรับเปลี่ยน (Adoption) : การปรับเปลี่ยนการดำเนินงานเมื่อผลลัพธ์ไม่เป็นไปตามที่ต้องการ

สำหรับทีมฟันด์คอนเน็กต์ใช้วิธีการแบ่งงานเป็น Sprint Sprintละ 2 อาทิตย์ โดยมีขั้นตอนในแต่ละ Sprint ดังนี้

1. Sprint Planning ขั้นตอนในวันแรกที่เริ่มต้น Sprint โดยจะแบ่งเป็น

   Planning Part 1 Project Owner จะกำหนดและอธิบายรายละเอียดงานของแต่ละการ์ดที่จะนำเข้ามาทำใน Sprint ซึ่งแต่ละการ์ดก็คือ Requirement ที่ Project Owner ได้รับมาจากลูกค้า ซึ่งจะมีไฟล์เอกสารประกอบเพื่ออธิบายรายละเอียดของแต่ละการ์ดให้กับสมาชิกภายในทีม

   Planning Part 2 สมาชิกฝ่าย IT จะประเมินขนาดของงาน ว่าสามารถทำได้ทั้งหมดกี่การ์ดในแต่ละ Sprint ซึ่งในขั้นตอนนี้สามารถเพิ่มลดการ์ดที่จะนำเข้ามาทำได้ตามความเหมาะสม รวมถึงทำการวางแผนสิ่งที่จะต้องทำในการ์ดต่างๆ
2. Process in Sprint สมาชิกในละคนทำหน้าที่ของตนเองที่ได้รับใน Sprint
3. Refinement ขั้นตอนที่ทำในวันแรกของสัปดาห์ที่ 2 ของ Sprint โดยมีลักษณะคล้ายกับ Sprint Planning แต่จุดประสงค์คือพูดคุยเกียวกับ Requirement ที่จะนำเข้ามาทำใน Sprint ถัดไป
4. Sprint Review ขั้นตอนที่ทำในวันสุดท้ายของ Sprint โดยเป็นการที่ฝ่าย IT จะทำการนำเสนองานในแต่ละการ์ดกับ Project Owner เพื่อตรวจสอบว่างานที่ได้รับมาได้ผลลัพธ์ตรงตามที่ต้องการหรือไม่
5. Retro ขั้นตอนที่ทำหลังจาก Sprint Review โดยจะเป็นการให้สมาชิกแต่ละคนภายในทีมพูดคุยกันเกี่ยวกับภาพรวมทั้งหมดที่เกิดขึ้นใน Sprint โดยมีหัวข้อ 4 ข้อ
   - ขอ 3 คำ 3 คำที่พูดพึงภาพรวมของ Sprint
   - Good เรื่องดีๆที่เกิดขึ้นใน Sprint

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Bad ปัญหาที่ส่งผลกระทบกับการทำงาน
- Try นำปัญหาที่เกิดขึ้นมาช่วยกันหาทางแก้ไขเพื่อไม่ให้เกิดขึ้นใน Sprint ถัดๆไป

6. Standup Meeting การประชุมประจำวันในช่วงก่อนเริ่มทำงานแต่ละวันเป็นการแจ้งความคืบหน้างานของตนเองที่ทำให้แก่สมาชิกในทีม โดยประกอบไปด้วย เมื่อวานทำอะไร ปัญหาที่เกิดขึ้น วันนี้จะทำอะไร เพื่อเมื่อมีปัญหาเกิดขึ้นสมาชิกภายในทีมจะได้สามารถช่วยเหลือได้ก่อนที่จะจบ Sprint

2.2 การทดสอบซอฟต์แวร์

การทดสอบซอฟต์แวร์ คือ กระบวนการหนึ่งในการพัฒนาซอฟต์แวร์ซึ่งเป็นกระบวนที่ค้นหาข้อผิดพลาดและส่วนที่ขาดหายไปของระบบที่ทำการทดสอบเพื่อนำจุดที่ผิดพลาดหรือตกหล่นมาแก้ไขก่อนส่งมอบให้กับลูกค้ารวมถึงเป็นการเพิ่มประสิทธิภาพในกับซอฟต์แวร์มากยิ่งขึ้น

2.2.1 วัตถุประสงค์ของการทดสอบ

Software Testing คือ กระบวนหนึ่งในการพัฒนาซอฟต์แวร์ซึ่งเป็นการทดสอบซอฟต์แวร์ว่าซอฟต์แวร์นั้นสามารถทำงานได้ถูกต้องที่คาดหวังไว้หรือเป็นไปตามข้อกำหนดทางธุรกิจ นอกจากนี้เป็นกระบวนการที่ใช้ค้นหาและระบุข้อผิดพลาดเพื่อนำจุดผิดพลาดนั้นมาทำการแก้ไข เพื่อลดข้อผิดพลาดก่อนที่จะส่งมอบซอฟต์แวร์ และเพิ่มประสิทธิภาพการทำงานใหม่มากยิ่งขึ้น ในกระบวนการทดสอบซอฟต์แวร์จำเป็นต้องวางแผนการทดสอบให้ครอบคลุม จึงต้องมีมุมมองในการทดสอบดังนี้

1. การทดสอบให้ซอฟต์แวร์ทำงานได้ในสถานการณ์ปกติ
2. การทดสอบให้ซอฟต์แวร์ทำงานไม่ได้ เช่น พยายามให้ระบบเกิดข้อผิดพลาด เพื่อหาจุดผิดพลาดในจุดนั้น
3. การทดสอบเพื่อลดความเสี่ยงที่ระบบจะทำงานล้มเหลว

2.2.2 กระบวนการทดสอบ

กระบวนการทดสอบซอฟต์แวร์สามารถแบ่งเป็นขั้นตอนพื้นฐานดังต่อไปนี้

1. Planning and Control การวางแผนและขั้นตอนในการทดสอบเกี่ยวกับการสร้างเอกสารที่อธิบายแนวทางและวัตถุประสงค์ของการทดสอบ โดยระบุเงื่อนไขและเกณฑ์การทดสอบให้ชัดเจน รวมถึงการวางแผนในการทดสอบแต่ละขั้นดังนี้
    - Unit Testing เป็นการทดสอบหลังจากที่ผู้พัฒนาโปรแกรมได้เขียนโปรแกรมเสร็จสิ้น โดยจะเป็นการตรวจสอบหาข้อบกพร่องและโครงสร้างของโปรแกรมว่าทำงานได้ถูกต้องหรือไม่ การทำ Unit Testing เป็นวิธีการนึงที่ช่วยการตรวจสอบการทำงานจุดเล็กว่าทำงานได้ถูกต้องหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Integration Testing เป็นการทดสอบการทำงานของโปรแกรมทั้งหมด โดยจะเป็นการทดสอบแต่ละฟังก์ชันว่าสามารถทำงานได้หรือไม่และทำงานร่วมกับฟังก์ชันอื่นๆได้หรือไม่
- System Testing เป็นการทดสอบการทำงานภาพรวมของระบบว่าสามารถตอบสนองความต้องการทั้งการทำงานของฟังก์ชันและประสิทธิภาพว่าสอดคล้องกับความต้องการของลูกค้าหรือไม่
- User Acceptance Testing เป็นกระบวนการทำสอบระบบการนำไปใช้งานจริง เปรียบเทียบการทำงานและประสิทธิภาพของระบบ โดยอิงจาก Requirement ที่ได้รับมอบหมายและเป็นไปตามข้อกำหนดทางธุรกิจนั้นจริงๆ ซึ่งผู้ใช้งานจริงจะเข้ามามีส่วนร่วมในการทดสอบก่อนนำระบบไปใช้งานจริงโดยจะจัดเตรียม Environment ต่างๆ ให้ใกล้เคียงกับการใช้งานจริงมากที่สุด

2. Analysis and Design การวิเคราะห์การทดสอบและการออกแบบการทดสอบเป็นขั้นตอนในการ ทบทวนเกณฑ์การทดสอบ การเขียนกรณีทดสอบ การออกแบบสภาพแวดล้อมในการทดสอบรวมถึงระบุเครื่องมือต่างๆในการทดสอบ
3. Implementation and Execution การดำเนินการทดสอบจริงบนระบบ โดยผ่านคอมพิวเตอร์ตั้งแต่การทดสอบด้วยตนเองหรือใช้เครื่องมืออัตโนมัติตามกรณีทดสอบที่ได้ออกแบบไว้ในขั้นต้นเพื่อบันทึกผลการทดสอบผ่าน หรือไม่ผ่าน
4. Evaluating exit criteria and Reporting การประเมินการหยุดทดสอบเป็นกระบวนที่กำหนดว่าเมื่อใดควรหยุดทดสอบขึ้นอยู่กับแต่ละฟังก์ชันและความเสี่ยงที่เกิดขึ้นซึ่งจะแตกต่างกันออกไปในแต่ละโครงการและสุดท้ายเป็นการเขียนรายงานสรุปผลการทดสอบว่าเป็นอย่างไร

2.2.3 กระบวนการทดสอบในระบบฟันด์คอนเน็กต์

วิธีการทดสอบซอฟต์แวร์เป็นขั้นตอนหนึ่งในการพัฒนาซอฟต์แวร์ที่ใช้รองรับการทำงานของระบบซึ่งวิธีการทดสอบไม่จำเป็นต้องทำตามทฤษฎีพื้นฐานที่กล่าวข้างต้นทั้งหมดสามารถนำมาปรับให้เหมาะสมกับแต่ละระบบได้ สำหรับโครงการฟันด์คอนเน็กต์ได้แบ่งการทดสอบดังนี้

การทดสอบขั้น Unit Testing จะทำในช่วงที่ของการพัฒนาซอฟต์แวร์ซึ่งผู้พัฒนาซอฟต์แวร์ในส่วนนั้นๆจะเป็นผู้รับผิดชอบในการทดสอบ

การทดสอบขั้น Integration Testing และการทดสอบขั้น System Testing จะทดสอบเมื่อผู้พัฒนาซอฟต์แวร์ทำงานเสร็จสิ้นแล้วนักทดสอบระบบจะเป็นผู้รับผิดชอบในการทดสอบทั้งสองส่วนนี้ โดยจะมีขั้นตอนในการทดสอบเป็น Manual Test ตาม Test Script เมื่อพบข้อผิดพลาดจะนำไปแจ้งผู้พัฒนาซอฟต์แวร์ในส่วนนั้นทำงานแก้ไขทัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบขั้น User Acceptance Testing เป็นการทดสอบระบบก่อนใช้งานจริง โดยออกแบบการทำ Automated Test ให้ทำงานในนอกจากนี้ก็จะมีลูกค้าที่เข้ามาในการทดสอบส่วนนี้ด้วยเช่นกัน

2.3 Manual Test

การทดสอบระบบโดยดำเนินการที่ผู้ทดสอบทำการ Test ตาม Test Script ที่ออกแบบไว้เพื่อหาข้อผิดพลาดและผลกระทบที่เกิดขึ้นจากการพัฒนาระบบให้ได้มากที่สุด ซึ่งการทดสอบก็จะแตกต่างๆกันไปขึ้นอยู่กับ Requirement และความสามารถต่างๆในแต่ละระบบ โดย Manual Test มีข้อดีข้อเสียดังต่อไปนี้

ข้อดี คือ มองเห็นจุดผิดพลาดได้อย่างชัดเจน คุ้มค่ามากกว่าหากเป็นงานระยะสั้น ประหยัดเงินทุนมากกว่าการทำ Automated Test

ข้อเสีย คือ ใช้เวลาในการ Regression test นานหากเป็นระบบงานที่ใหญ่ เกิดข้อผิดพลาดอันมาจากการเพิ่มหรือแก้ไขงานบางส่วน

2.4 Automated Test

การทดสอบระบบอัตโนมัติเป็นการทดสอบระบบโดยใช้เครื่องมือ Automated Tools สำหรับเพื่อให้เครื่องมือนั้นทดสอบด้วยตนเอง แต่จะต้องมีการเขียนโค้ดเพื่อทำ Script รวมถึงมีการจัดเตรียมข้อมูลต่างๆไว้ให้เครื่องมือนั้น Run Test ในการทดสอบ โดยมีข้อดีข้อเสียดังต่อไปนี้

ข้อดี คือ ลดระยะเวลาในการ Regression test ลดข้อผิดพลาดที่เกิดขึ้นจากการพัฒนาซอฟต์แวร์ คุ้มค่าในระบบงานใหญ่ที่ใช้ในระยะยาว

ข้อเสีย คือ มีค่าใช้จ่ายในการทำมากกว่า Manual Test เพราะต้องมีการ Maintenance Automated Test Script อย่างสม่ำเสมอ

สำหรับโครงการฟันด์คอนเน็กต์ได้ใช้ Automated Test Tools อย่าง Cypress และ Robot Framework ในการทำ Automated Test ซึ่งใช้ Visual Studio ในการเขียน Automated Test Script ขึ้นมานั้นเอง

2.5 Test automation maintenance

การดูแลรักษา Automated Test Script เป็นกระบวนการที่สำคัญในการพัฒนา การบำรุงรักษาสคริปต์การทดสอบอัตโนมัติ ด้วยวิธีนี้เราสามารถมั่นใจได้ว่าสคริปต์ทดสอบอัตโนมัติยังสามารถทำงานได้อย่างถูกต้องตามคาดหวัง และสามารถครอบคลุมทุกรายละเอียดของแอปพลิเคชันหรือซอฟต์แวร์ที่เรากำลังทดสอบได้ ต่อไปนี้คือขั้นตอนในการดูแลรักษา Automated Test Script

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ตรวจสอบความถูกต้องของสคริปต์ เมื่อมีการเปลี่ยนแปลงในซอฟต์แวร์หรือส่วนประกอบอื่นๆ ที่ส่งผลกระทบต่อการทำงานของสคริปต์ทดสอบ ควรตรวจสอบสคริปต์ทดสอบว่ายังสามารถทำงานได้อย่างถูกต้องหรือไม่ โดยใช้การรันสคริปต์ทดสอบและตรวจสอบผลลัพธ์ที่ได้ หากพบข้อผิดพลาด ให้แก้ไขสคริปต์ทดสอบให้ถูกต้องตามการเปลี่ยนแปลงที่เกิดขึ้น
2. การบันทึกข้อผิดพลาด (Logging) เพื่อให้ง่ายต่อการตรวจสอบและวิเคราะห์ข้อผิดพลาดที่เกิดขึ้นในกระบวนการทดสอบ ควรมีการบันทึกข้อผิดพลาดในระหว่างการรันสคริปต์ทดสอบ โดยการใช้เครื่องมือหรือการเขียนโค้ดในสคริปต์ทดสอบเพื่อบันทึกข้อผิดพลาด เช่น ข้อความผิดพลาด ข้อมูลแสดงสถานะ เป็นต้น
3. การจัดการข้อมูลทดสอบ (Test Data Management) การดูแลรักษาสคริปต์ทดสอบอัตโนมัติเชื่อมโยงกับข้อมูลทดสอบ เราควรมีการจัดการข้อมูลทดสอบอย่างรอบคอบ การเตรียมข้อมูลทดสอบที่เหมาะสมก่อนการรันสคริปต์ทดสอบ และตรวจสอบว่าสคริปต์สามารถใช้ข้อมูลทดสอบนั้นได้เสมอ
4. การทำงานร่วมกันในทีม (Collaboration) หากมีทีมผู้พัฒนาและดูแลรักษาสคริปต์ทดสอบอัตโนมัติ ควรมีการทำงานร่วมกันอย่างมีประสิทธิภาพ ทำความเข้าใจกันเกี่ยวกับวัตถุประสงค์ของสคริปต์ และมีการแบ่งหน้าที่และความรับผิดชอบที่ชัดเจน การใช้เครื่องมือหรือแพลตฟอร์มที่ช่วยในการจัดการสคริปต์และการแจ้งเตือนการทำงานเป็นต้น
5. การรักษาและอัปเดตสคริปต์ สำหรับสคริปต์ทดสอบที่ใช้ในการทดสอบเวอร์ชันก่อนหน้า ควรทำการรักษาและอัปเดตเพื่อให้สอดคล้องกับการเปลี่ยนแปลงที่เกิดขึ้นในระบบหรือซอฟต์แวร์ปัจจุบัน นอกจากนี้ยังควรมีการรักษาความเข้ากันได้กับเครื่องมือที่ใช้ในการทดสอบ เพื่อปรับปรุงสคริปต์และเพิ่มประสิทธิภาพในการทดสอบ
6. การทดสอบแบบอัตโนมัติของสคริปต์การทดสอบ เพื่อให้แน่ใจว่าสคริปต์ทดสอบยังคงทำงานได้อย่างถูกต้อง ควรมีการรันสคริปต์ทดสอบให้เป็นประจำ โดยอาจใช้เครื่องมือหรือเฟรมเวิร์กสำหรับการทดสอบอัตโนมัติเพื่อตรวจสอบสถานะและผลลัพธ์ของการทดสอบ

การดูแลรักษา Automated Test Script เป็นกระบวนการต่อเนื่องที่ต้องทำเป็นระยะเวลานาน โดยควรมีการตรวจสอบและปรับปรุงเพื่อให้สคริปต์ทดสอบอัตโนมัติสามารถทำงานได้อย่างเหมาะสมตามการเปลี่ยนแปลงที่เกิดขึ้นในระบบหรือซอฟต์แวร์ นอกจากนี้ยังควรใช้เครื่องมือหรือแพลตฟอร์มที่ช่วยในกระบวนการดูแลรักษาเพื่อเพิ่มประสิทธิภาพและความเป็นระเบียบในการทดสอบอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 เครื่องมือที่ใช้

2.6.1 Postman

Postman เป็นเครื่องมือที่ไว้ใช้สำหรับ API Developers และเป็น API Test Tools ในการส่ง Request ไปยัง HTTP/s และตรวจสอบ Response ที่ได้รับและดูผลลัพธ์เพื่อทดสอบซอฟต์แวร์

Postman มี Method หลัก 5 ประเภทดังนี้

1. GET เป็นการส่ง Request เพื่อขอข้อมูลโดยตรง
2. PUT เป็นการอัพเดตข้อมูลไปยังตำแหน่งที่มีอยู่ โดยระบุข้อมูลทั้งหมดเพื่ออัพเดต
3. POST เป็นการส่งข้อมูลเพื่อสร้างข้อมูลชุดใหม่ขึ้นมา
4. PATCH เป็นการอัพเดตข้อมูลที่มีอยู่แล้ว โดยการส่งข้อมูลที่ระบุตำแหน่งที่ต้องการเปลี่ยนแปลงได้ทันที
5. DELETE เป็นการลบข้อมูล

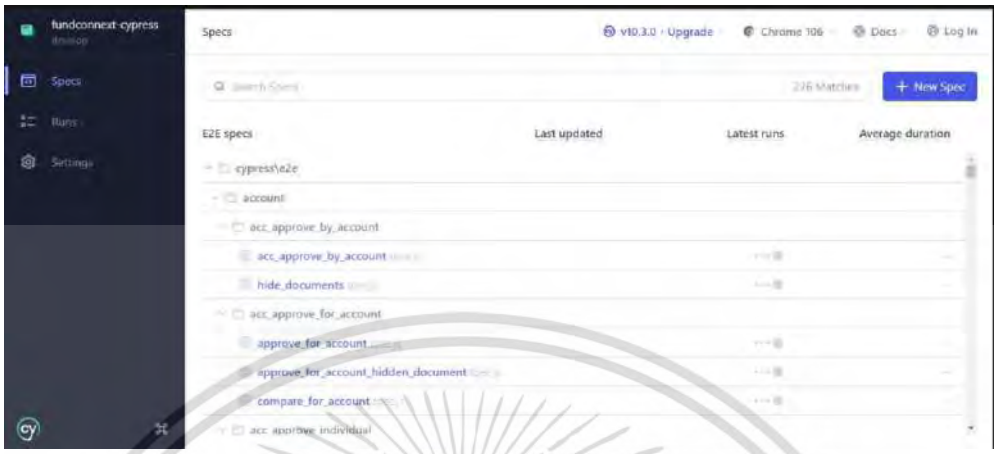
2.6.2 Cypress

Test automation framework โดยถูกพัฒนามาจากภาษา JavaScript ซึ่งรองรับการทดสอบทั้งบน Google Chrome, Mozilla Firefox และ Browser อื่นๆ ซึ่งโครงสร้างหลักใน Cypress มีดังนี้

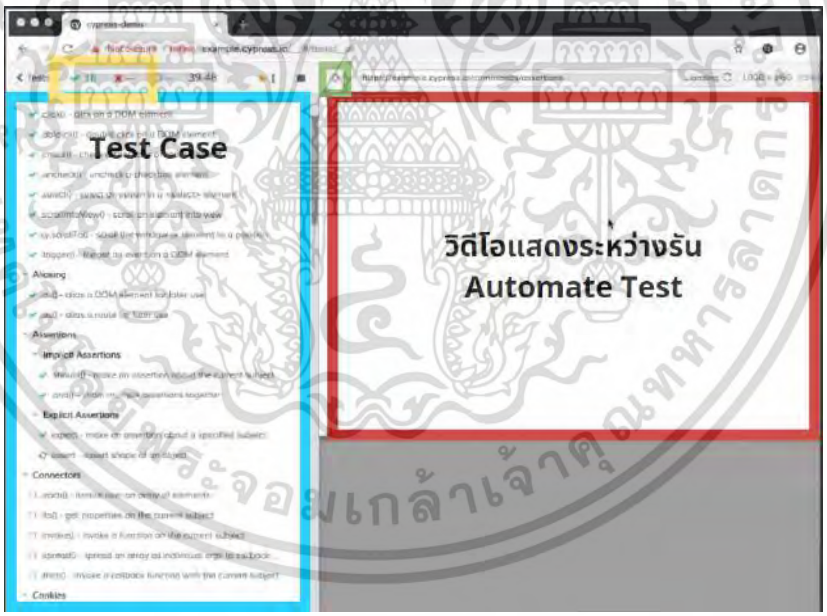
1. import เป็นการเรียกใช้ฟังก์ชันต่างๆ ที่จะนำเข้าใช้ใน Test case นั้น
2. describe คำอธิบายรายละเอียดโดยรวมของ Automated Test นั้น
3. it เป็นส่วนในการเขียน Test case ที่เราต้องการการนำชุดคำสั่งต่างๆมาร้อยเรียงกันเป็นขั้นตอนในการทดสอบในแต่ละ Test case โดยจะมีคำสั่งพื้นฐานดังนี้
   visit คือ คำสั่งที่ใช้ในการเข้าถึงเว็บไซต์ต่างๆที่ต้องการทดสอบ
   get คือ คำสั่งในการระบุ Element หรือตำแหน่งต่างๆ
   xpath คือ คำสั่งในการระบุ Element แต่มีรูปแบบการระบุตำแหน่งต่างกับ Get
   click คือ คำสั่งคลิกที่ต้องการ
   type คือ คำสั่งกรอกข้อมูลลงใน input ที่ต้องการ
   clear คือ คำสั่งที่ใช้ลบข้อมูลใน input ต่างๆ
   select คือ คำสั่งที่ใช้เลือก Dropdown
   contain คือ คำสั่งในการตรวจสอบ Text ใน Element
   should คือ คำสั่งที่ใช้ในการยืนยัน เช่น emtry value text
4. before/after ส่วนที่ต้องการให้มีการทำงานก่อนจะเริ่มหรือหลังจบการทำงานของ Test case เช่น คำสั่ง Query การเรียกใช้ API เพื่อใช้ในการเตรียม Data หรือทำงานบางอย่างก่อนหรือหลังการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยตัว Cypress จะมีหน้าต่าง UI เข้ามาที่ช่วยเมื่อทำการทดสอบ โดยมีหน้าจอดังนี้



รูปที่ 2.1 หน้าจอ UI การรัน Cypress.io (1)



รูปที่ 2.2 หน้าจอ UI การรัน Cypress.io (2)

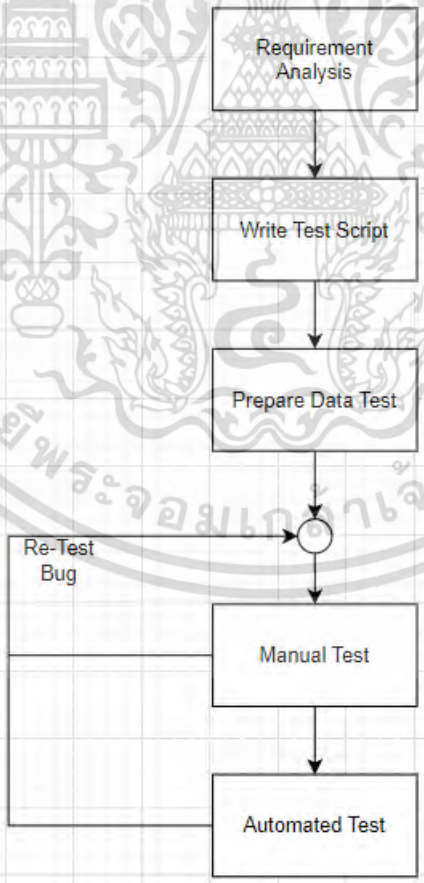
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการดำเนินงานสหกิจศึกษา

สหกิจศึกษาภายใต้หัวข้อการทดสอบอัตโนมัติสำหรับระบบฟันด์คอนเน็กต์ ผู้จัดทำได้รับมอบหมายในหน้าที่ Tester ภายในทีม ซึ่งมีหน้าที่ตั้งแต่การวิเคราะห์ความต้องการก่อนนำมาจัดทำกรณีทดสอบต่างๆ และการเตรียมข้อมูลเพื่อนำไปทดสอบระบบทั้งในรูปแบบ Manual Test และ Automated Test หลังจากการทดสอบเสร็จก็จะทำการบำรุงรักษา Automated Test Script ที่จัดทำให้สามารถทำงานได้ถูกต้องอย่างสม่ำเสมอ ซึ่งภายในทีมของระบบฟันด์คอนเน็กต์จะมีกระบวนการในการทดสอบซอฟต์แวร์ตามขั้นตอนต่อไปนี้

3.1 กระบวนการดำเนินงาน

กระบวนการทดสอบซอฟต์แวร์ในการทดสอบระบบฟันด์คอนเน็กต์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.1 Diagram แสดงขั้นตอนการทดสอบระบบของทีมฟันด์คอนเน็กต์

กระบวนการทดสอบซอฟต์แวร์ในการทดสอบระบบฟันด์คอนเน็กต์

1. Requirement Analysis คือ การวิเคราะห์ Requirement ที่ได้รับมาจาก Project Owner โดยในแต่ละ Requirement จะมีเอกสารแนบประกอบสำหรับนำไปใช้วิเคราะห์ เพื่อนำมาจัดเตรียม Test Script, Data Test, รวมถึงเครื่องมือที่จำเป็นจะต้องใช้ในการทดสอบระบบ
2. Write Test Script เป็นการเขียนกรณีทดสอบ (Test Case) โดยอ้างอิงจาก Requirement ที่กำหนดไว้ในแต่ละการ์ด โดยจะจัดทำผ่าน Google Sheet เป็นรูปแบบตาราง เพื่อนำกรณีทดสอบเหล่านั้นมาใช้ตรวจสอบการทำงานของระบบให้ถูกต้อง
3. Prepare Data Test การจัดเตรียมข้อมูลสำหรับใช้ในการทดสอบของแต่ละ Test Caseต่างๆ ในขั้นตอนนี้จะต้องอาศัยการเขียน SQL ประกอบเพื่อใช้ในการเตรียมรวมถึงตรวจสอบข้อมูลต่างๆให้ถูกต้องตาม Test Script
4. Manual Test การทดสอบระบบโดยทดสอบตาม Test Script ที่จัดเตรียมไว้ และพยายามหาข้อผิดพลาด และผลกระทบที่อาจเกิดขึ้นให้ได้มากที่สุด
5. Automated Test การทดสอบระบบโดยใช้เครื่องมืออัตโนมัติ (Automated Tools) แต่จะต้องมีการจัดทำ Data รวมถึงเขียนโค้ดสำหรับทำ Automated Test Script โดยระบบฟันด์คอนเน็กต์จะแยกDatabase ไว้สำหรับDataในการทำ Automated Test เฉพาะ เนื่องจากต้องให้การทดสอบใกล้เคียงกับการทำงานของระบบจริงมากที่สุด ซึ่งการจัดทำ Automated Testอาจจะมีการใช้ Test Script เดียวกับการทำ Manual Test หรือแตกต่างกันออกไปในแต่ละกรณี โดย Tools ที่ใช้ในการทำมีดังนี้ Cypress และ Robot Framework
6. Re-Test Bug กระบวนการจัดการเมื่อทดสอบระบบแล้วพบข้อผิดพลาด กระบวนการจัดการเมื่อทดสอบระบบแล้วพบข้อผิดพลาดจากการทำงานของระบบ โดยจะมีขั้นตอนในการจัดการดังนี้
    - 6.1 Manual Test หรือ Automated Test เพื่อใช้ในการตรวจสอบข้อผิดพลาดที่เกิดขึ้นจากการ ทำงานของระบบ
    - 6.2 Defect Found เมื่อพบข้อผิดพลาดจากการทำงานของระบบจะนำไปยืนยันกับสมาชิกในทีมว่าเป็นข้อผิดพลาดแบบใด
    - 6.3 Error or Missing Requirement การยืนยันกับสมาชิกภายในทีมว่าข้อผิดพลาดเกิดจากกรณีใด โดยจะแบ่งเป็น 2 กรณี คือ ข้อผิดพลาดจากการพัฒนาระบบ หรือข้อผิดพลาดจากเข้าใจ Requirement ไม่ตรงกัน
    - 6.4 Open Card Bug หลังจากได้รับการยืนยันว่าเป็นข้อผิดพลาดแบบใดก็จะทำการ Open Card Bug ที่มีข้อมูลเกี่ยวกับรายละเอียดของปัญหาที่เกิดขึ้น โดยในกรณีที่เกิดจากข้อผิดพลาดจากการพัฒนาระบบก็สามารถที่จะไปแจ้งทีม Developerที่รับผิดชอบงานนั้นๆ แต่ว่าหากเป็นกรณีที่เกิดจากความเข้าใจใน Requirement ที่ไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรงกัน จะต้องมีการพูดคุยกับ Project Ownerและสมาชิกในทีมเพื่อทำความเข้าใจใน Requirementนั้นอีกครั้ง

6.5 Fix Bug ทีม Developer นำ Card Bug ไปทำการแก้ไขให้ดูต้องตาม Requirement

6.6 หลังจากนั้น Tester จะทำการทดสอบหลังจากการแก้ไขอีกครั้ง

3.2 วิธีการดำเนินงาน

การทดสอบระบบสามารถแบ่งได้เป็น 6 ส่วนหลักดังนี้

1. Functional Test โดยทดสอบในรูปแบบ Manual Test ประกอบไปด้วย
   - ตัวอย่างที่ 1 เมนู Upload Point Code
   - ตัวอย่างที่ 2 เมนู Change Unitholder Account
2. Automated Test แบบ Input data ประกอบไปด้วย
   - ตัวอย่างที่ 1 เมนูเปิดบัญชีประเภท Individual
   - ตัวอย่างที่ 2 เมนูเปิดบัญชีประเภท By
3. Automated Test แบบ Look and Filed ประกอบไปด้วย
   - ตัวอย่างที่ 1 แสดง Account แบบ Hierarchy
   - ตัวอย่างที่ 2 เมนู SA Profile
4. Automated Test แบบ Look and Filed (ตาราง) ประกอบไปด้วย
   - ตัวอย่างที่ 1 เมนู Failed XWT Order
   - ตัวอย่างที่ 2 เมนู Account Balance Inquiry
5. Automated Test แบบ Validate File ประกอบด้วย
   - ตัวอย่างที่ 1 เมนู Upload Attachment
   - ตัวอย่างที่ 2 เมนู Account Balance Inquiry (CSV)
6. Automated Test API ประกอบด้วย
   - ตัวอย่างที่ 1 API Get Failed XWT Order
   - ตัวอย่างที่ 2 API Approve Failed XWT Order
   - ตัวอย่างที่ 3 API Update Bank Account

ซึ่งจะมีการอธิบายรายละเอียดการทดสอบ กรณีทดสอบ และAutomated Test Script ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 Functional Test โดยทดสอบในรูปแบบ Manual Test

การทดสอบแบบ Manual Test นั้นมีการทดสอบที่แตกต่างกันออกไปในแต่ละระบบงาน ซึ่งจะเป็นการเข้าไปใช้งานฟังก์ชันต่างๆแล้วทำการทดสอบตาม Test Script ที่ออกแบบไว้เพื่อตรวจสอบผลลัพธ์ที่ได้ว่าถูกต้องตาม Requirement หรือไม่

ตัวอย่างที่ 1 การทดสอบ Functional Test แบบ Manual Test

Requirement : เมนู Upload Point Code สำหรับรายการ Point SA Upload voucher code ได้ผ่านหน้าจอเท่านั้นตาม format (Code, Amount, Expiry Date) validate platform / issuer ให้สอดคล้องกับ code , expiry date ต้องมากกว่าวันที่ปัจจุบัน, sa ต้องมี permission ในการ add platform / issue มีหน้า list code และลบได้เฉพาะที่ยังไม่ถูกใช้งาน (status ไม่เท่ากับ redeem) ถ้ามี code อยู่แล้ว แต่ upload เข้ามาใหม่ จะเข้าไม่ได้ขึ้น error ว่ามี code อยู่แล้ว (duplicate error) ถ้าต้องการแก้ไข ให้ลบ และ insert เข้ามาใหม่ มีปุ่ม download excel ออกไปได้ ออกตามที่ filter ของ header IT Support การ config มีหน้าจอให้ Oper ดูการ config ดูว่า SA ไหนใช้ Platform หรือ Issuer อะไรบ้าง อยู่ในหน้า File Maintenance -> Code Management สร้าง table mapping point convertor (Point platform) กับ point issuer กับ Point code (voucher code)

ตารางที่ 3.1 Test Script การทดสอบ Functional Test แบบ Manual Test ตัวอย่างที่ 1

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result
Code Management - SA						
1	Web	SA	Application Management	Code Management	- Upload Code(ก่อนเวลา 16.00) ตรวจสอบ Error	- Uploadไม่สำเร็จ - แสดงข้อความ “Must be uploaded after 16.00”
2	Web	SA	Application Management	Code Management	- Upload Code(หลังเวลา 16.00)	- Upload Code ได้สำเร็จ
3	Web	SA	Application Manage	Code Management	- Upload Code ซ้ำที่มีในระบบแล้ว	- Uploadไม่สำเร็จ - แสดงข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			ment			“Duplicate Code”
4	Web	SA	Application Management	Code Management	- Upload Code ที่ไม่ได้ Config Code Mappingไว้	- Uploadไม่สำเร็จ - แสดงข้อความ “Incorrect Code Format”
5	Web	SA	Application Management	Code Management	- Upload Code ไม่ครบ 9 ตัวอักษร	- Uploadไม่สำเร็จ - แสดงข้อความ “Incorrect Code Format”
6	Web	SA	Application Management	Code Management	- Upload Code เกิน 9 ตัวอักษร	- Uploadไม่สำเร็จ - แสดงข้อความ “Incorrect Code Format”
7	Web	SA	Application Management	Code Management	- Upload Code อักษรตัวเล็ก	- Uploadไม่สำเร็จ - แสดง Point_code เป็นอักษรตัวใหญ่
8	Web	SA	Application Management	Code Management	- Upload Code ที่หมดอายุไปแล้ว	- Uploadไม่สำเร็จ - แสดงข้อความ “Incorrect Expiry Date”
9	Web	SA	Application Management	Code Management	- Upload Code ที่Expiry Date ตรงกับวันที่Upload	- Upload Code ได้สำเร็จ
10	Web	SA	Application Management	Code Management	- Upload Code ที่มีคนละPrefix-Codeในไฟล์เดียวกัน	- Upload Code ได้สำเร็จ
11	Web	SA	Application Management	Code Management	- ตรวจสอบหัวข้อตารางที่แสดงบนเมนู Code Management	- แสดงหัวข้อตารางได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12	Web	SA	Application Management	Code Management	- ตรวจสอบข้อมูล Status เมื่อนำPoint Codeไปใช้ทำรายการ	- StatusของPoint Code นั้นถูกปรับเป็น Redeemed ได้ถูกต้อง
13	Web	SA	Application Management	Code Management	- ตรวจสอบข้อมูล Redeemed Data เมื่อนำ Point Code ไปทำรายการ	- แสดง Redeemed Data ในวันที่ Code ใช้งานได้ถูกต้อง
14	Web	SA	Application Management	Code Management	- ตรวจสอบ Checkbox สำหรับลบข้อมูล Point Code	- แสดง Checkbox หน้าPoint Codeที่มี Status เป็น Available และ Expired - ไม่แสดง Checkbox หน้าPoint Codeที่มี Status เป็น Redeemed
15	Web	SA	Application Management	Code Management	- ตรวจสอบการลบข้อมูล Point Code	- สามารถลบ Point Codeที่ต้องการได้ถูกต้อง - แสดง Pop up เพื่อ Confirm การ Delete Point Code
16	Web	SA	Application Management	Code Management	- ตรวจสอบModal แสดงผล Point Code Mapping	- กดปุ่ม Point Code แล้วแสดง Modal ข้อมูล Point Code Mapping ได้ถูกต้อง
17	Web	SA	Application Management	Code Management	- ตรวจสอบFile Excel เมื่อ Filterข้อมูลในตารางแล้วทำการ Download File	- แสดงข้อมูลใน File Excel เฉพาะที่ Filter ได้ถูกต้อง
Code Management - TSD						
18	Web	TSD	File	Code	- ตรวจสอบหัวข้อ	- แสดงหัวข้อตารางได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

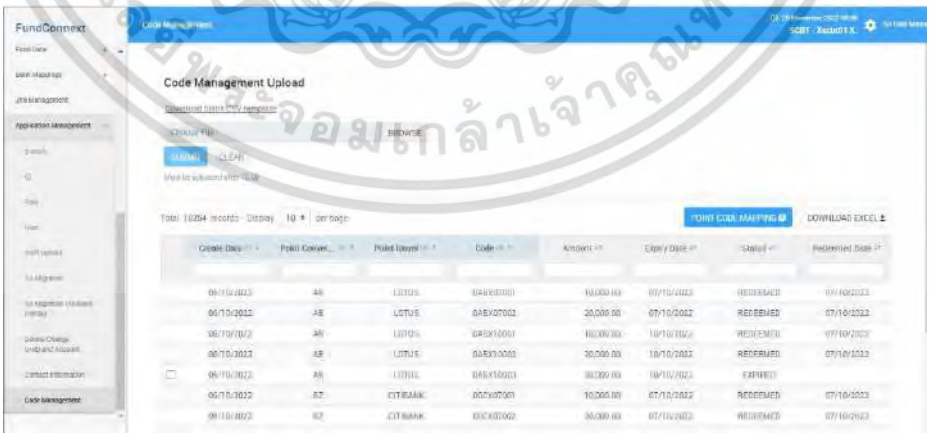
			Manage ment	Manage ment	ตารางที่แสดงบนเมนู Code Management ของ TSD	ถูกต้อง
19	Web	TSD	File Manage ment	Code Manage ment	- ตรวจสอบการแสดงผลข้อมูลในตารางเมนู Code Management ของ TSD	- แสดงข้อมูลในตารางได้ถูกต้อง

ขั้นตอนการทดสอบ

เตรียมไฟล์สำหรับการ Upload Point Code

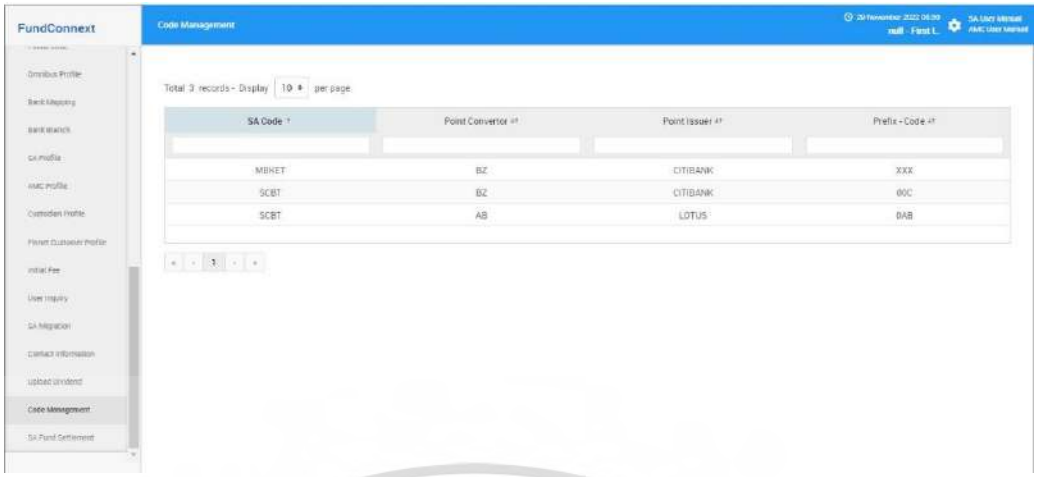
รูปที่ 3.2 ตัวอย่างไฟล์ที่ใช้ในการ Upload Point Code

Manual Test ตามกรณีทดสอบที่ได้ออกแบบไว้บนหน้าจอ ดังรูป 3.3 และ 3.4



รูปที่ 3.3 หน้าจอเมนู Code Management ของ SA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 หน้าจอเมนู Code Management ของ TSD

ตัวอย่างที่ 2 การทดสอบ Functional Test แบบ Manual Test

Requirement : เมนูสำหรับปรับเปลี่ยน Change Unitholder to Another Account ของแต่ละ Account ซึ่งจะสามารถย้ายได้เมื่อ Account ต้นทาง และAccount ปลายทางไม่ติดล็อก การ Change Unitholder to Another Account เมื่อกดสัญลักษณ์ที่ทำการ Change Unitholder to Another Account จะแสดง Popupเพื่อใส่ข้อมูล Accountปลายทางรวมถึงใส่User/Password และเหตุผลของผู้ที่มีสิทธิในการ Change Unitholder to Another Account และมีการเก็บLog การ Change Unitholder to Another Account ด้วยทุกครั้ง

ตารางที่ 3.2 Test Script การทดสอบ Functional Test แบบ Manual Test ตัวอย่างที่ 2

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result
SA Migrate UH						
1	Web	SA	Application Management	Change Unitholder to Another Account	- SA Migrate UH : SEG เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางไม่ติด Lock	- สามารถ Change Unitholder to Another Account ได้ถูกต้อง - มีการเก็บ Log - Orderทั้งหมดถูก Updateให้อยู่ภายใต้ Accountใหม่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2	Web	SA	Application Management	Change Unitholder to Another Account	- SA Migrate UH : OMN เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางไม่ติด Lock	- สามารถ Change Unitholder to Another Account ได้ถูกต้อง - มีการเก็บ Log - Orderทั้งหมดถูก Updateให้อยู่ภายใต้ Accountใหม่ถูกต้อง
3	Web	SA	Application Management	Change Unitholder to Another Account	- SA Migrate UH : SEG เข้าระบบโดยที่ Account ต้นทางติด Lock Account ปลายทางไม่ติด Lock	- Cannot change Account No. due to pending account Status.
4	Web	SA	Application Management	Change Unitholder to Another Account	- SA Migrate UH : OMN เข้าระบบโดยที่ Account ต้นทางติด Lock Account ปลายทางไม่ติด Lock	- Cannot change Account No. due to pending account Status.
5	Web	SA	Application Management	Change Unitholder to Another Account	- SA Migrate UH : SEG เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางติด Lock	- Cannot change Account No. due to pending account Status.
6	Web	SA	Application Management	Change Unitholder to Another Account	- SA Migrate UH : OMN เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทาง	- Cannot change Account No. due to pending account Status.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

					ติด Lock	
SA Send Order						
7	Web	SA	Application Management	Change Unitholder to Another Account	- SA Send Order : SEG เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางไม่ติด Lock	- สามารถ Change Unitholder to Another Account ได้ถูกต้อง - มีการเก็บ Log - Orderทั้งหมดถูก Updateให้อยู่ภายใต้ Accountใหม่ถูกต้อง
8	Web	SA	Application Management	Change Unitholder to Another Account	- SA Send Order : OMN เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางไม่ติด Lock	- สามารถ Change Unitholder to Another Account ได้ถูกต้อง - มีการเก็บ Log - Orderทั้งหมดถูก Updateให้อยู่ภายใต้ Accountใหม่ถูกต้อง
9	Web	SA	Application Management	Change Unitholder to Another Account	- SA Send Order : SEG เข้าระบบโดยที่ Account ต้นทางติด Lock Account ปลายทางติดไม่ Lock	- Cannot change Account No. due to pending account Status.
10	Web	SA	Application Management	Change Unitholder to Another Account	- SA Send Order : OMN เข้าระบบโดยที่ Account ต้นทางติด Lock Account ปลายทางติดไม่ Lock	- Cannot change Account No. due to pending account Status.
11	Web	SA	Application	Change Unitholder	- SA Send Order :	- Cannot change Account No. due to

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			Management	to Another Account	SEG เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางติด Lock	pending account Status.
12	Web	SA	Application Management	Change Unitholder to Another Account	- SA Send Order : OMN เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางติด Lock	- Cannot change Account No. due to pending account Status.
SA MoveSink						
13	Web	SA	Application Management	Change Unitholder to Another Account	- SA MoveSink : SEG เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางไม่ติด Lock	- สามารถ Change Unitholder to Another Account ได้ถูกต้อง - มีการเก็บ Log - Orderทั้งหมดถูก Updateให้อยู่ภายใต้ Accountใหม่ถูกต้อง
14	Web	SA	Application Management	Change Unitholder to Another Account	- SA MoveSink : OMN เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางไม่ติด Lock	- สามารถ Change Unitholder to Another Account ได้ถูกต้อง - มีการเก็บ Log - Orderทั้งหมดถูก Updateให้อยู่ภายใต้ Accountใหม่ถูกต้อง
15	Web	SA	Application Management	Change Unitholder to Another Account	- SA MoveSink : SEG เข้าระบบโดยที่ Account ต้นทางติด	- Cannot change Account No. due to pending account Status.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

					Lock Account ปลายทางติดไม่ Lock	
16	Web	SA	Application Management	Change Unitholder to Another Account	- SA MoveSink : OMN เข้าระบบโดยที่ Account ต้นทางติด Lock Account ปลายทางติดไม่ Lock	- Cannot change Account No. due to pending account Status.
17	Web	SA	Application Management	Change Unitholder to Another Account	- SA MoveSink : SEG เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางติด Lock	- Cannot change Account No. due to pending account Status.
18	Web	SA	Application Management	Change Unitholder to Another Account	- SA MoveSink : OMN เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางติด Lock	- Cannot change Account No. due to pending account Status.
SA Request API UH						
19	Web	SA	Application Management	Change Unitholder to Another Account	- SA Request API UH : SEG เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางไม่ติด Lock	- สามารถ Change Unitholder to Another Account ได้ถูกต้อง - มีการเก็บ Log - Orderทั้งหมดถูก Updateให้อยู่ภายใต้ Accountใหม่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

20	Web	SA	Application Management	Change Unitholder to Another Account	- SA Request API UH : OMNเข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางไม่ติด Lock	- สามารถ Change Unitholder to Another Account ได้ถูกต้อง - มีการเก็บ Log - Orderทั้งหมดถูก Updateให้อยู่ภายใต้ Accountใหม่ถูกต้อง
21	Web	SA	Application Management	Change Unitholder to Another Account	- SA Request API UH : SEG เข้าระบบโดยที่ Account ต้นทางติด Lock Account ปลายทางติดไม่ Lock	- Cannot change Account No. due to pending account Status.
22	Web	SA	Application Management	Change Unitholder to Another Account	- SA Request API UH : OMN เข้าระบบโดยที่ Account ต้นทางติด Lock Account ปลายทางติดไม่ Lock	- Cannot change Account No. due to pending account Status.
23	Web	SA	Application Management	Change Unitholder to Another Account	- SA Request API UH : SEG เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางติด Lock	- Cannot change Account No. due to pending account Status.
24	Web	SA	Application Management	Change Unitholder to Another Account	- SA Request API UH : OMNเข้าระบบโดยที่ Account ต้นทางไม่	- Cannot change Account No. due to pending account Status.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

					ติด Lock Account ปลายทาง ติด Lock	

ขั้นตอนการทดสอบ

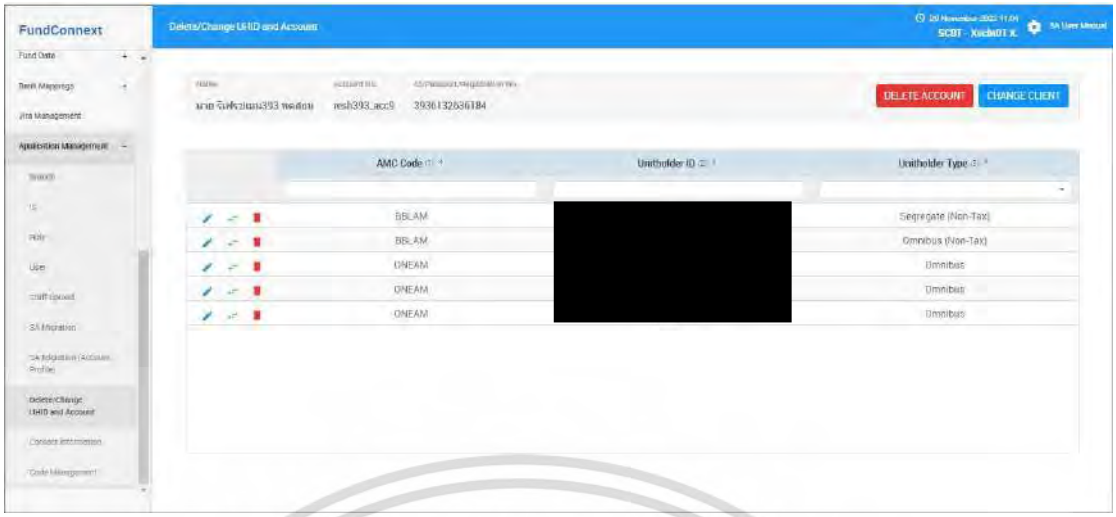
เตรียมข้อมูล Unitholder SEG และ Unitholder SEG OMN โดยวิธีดังนี้

- Migrate
- Send Order
- Movesink
- Request API

Manual Test ตามกรณีทดสอบที่ได้ออกแบบไว้

รูปที่ 3.5 หน้าจอเมนู Change Unitholder to Another Account (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 หน้าจอเมนู Change Unitholder to Another Account (2)

รูปที่ 3.7 หน้าจอสำหรับกรอกข้อมูล Account

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 Automated Test แบบ Input data

การทดสอบประเภท Functional Test แบบ Input data เป็นการทดสอบที่คำนึงถึงการใส่ข้อมูลตาม Business Flow ในระบบเพื่อดูผลลัพธ์ในกรณีต่างๆทั้งข้อมูลที่ถูกต้องและไม่ถูกต้อง โดยต้องแสดงผลลัพธ์ตรงกับ Requirement ได้ถูกต้อง

การทำ Test Script และเตรียม Data Test แบบเป็น 2 กรณีหลัก คือ

1. กรณีการทำ Test Script และเตรียม Data Test ให้ระบบสามารถทำงานได้ถูกต้อง และเกิดผลลัพธ์ที่ถูกต้อง
2. กรณีการทำ Test Script และเตรียม Data Test ให้ระบบทำงานผิดพลาด และเกิดผลลัพธ์จากการผิดพลาดได้ถูกต้อง

การทดสอบแบบ Automated Test เป็นการทดสอบการกรอกข้อมูล input ต่างๆและดูผลลัพธ์ที่ได้จากหน้าจอโดยคำสั่งส่วนใหญ่ที่ใช้ทดสอบแบบ Input data คือคำสั่ง Type(), Select() ในระบบฟันด์คอนเน็กต์มีช่องการรับ Input ดังนี้

ช่องรับ Input แบบทั่วไป โดยวิธีการกรอกข้อมูลในช่องรับข้อมูลประเภทนี้จะมีการใช้คำสั่ง

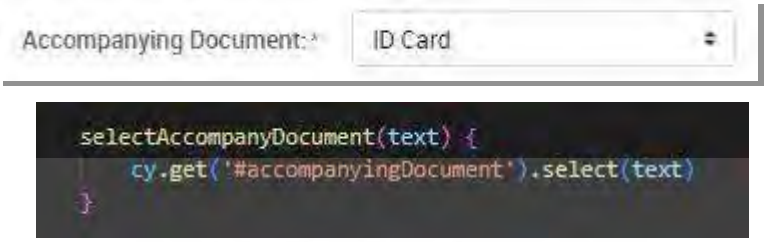
Clear() : ลบข้อมูลทั้งหมดที่อยู่ในช่องรับข้อมูล เพื่อลดปัญหาDataผิดพลาดเวลาทำการทดสอบซ้ำ

Type() : การกรอกข้อมูลที่ต้องการลงไปในช่องรับข้อมูลที่กำหนด

รูปที่ 3.8 ช่องรับข้อมูลแบบธรรมดา

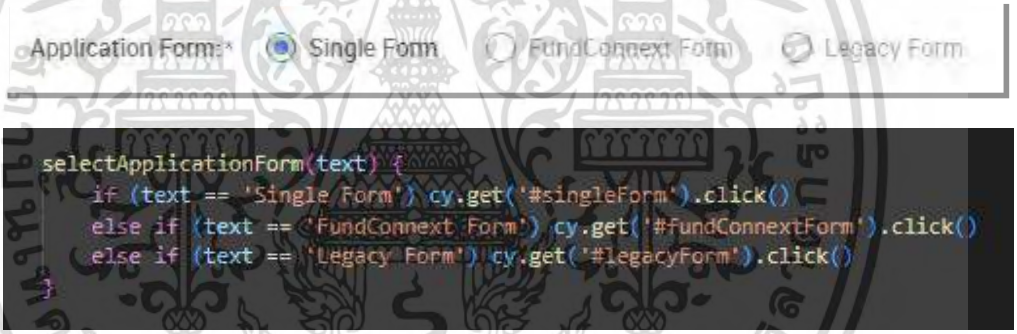
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่องรับ Input แบบ Dropdown โดยการเลือกข้อมูลใน Dropdown จะเป็นการใช้คำสั่ง Select() เพื่อเลือกตัวเลือกใน Dropdown โดยใส่ค่าที่ต้องการเลือกใน Dropdownนั้นๆ ดังรูป 3.9



รูปที่ 3.9 ช่องรับข้อมูลแบบ DropDown

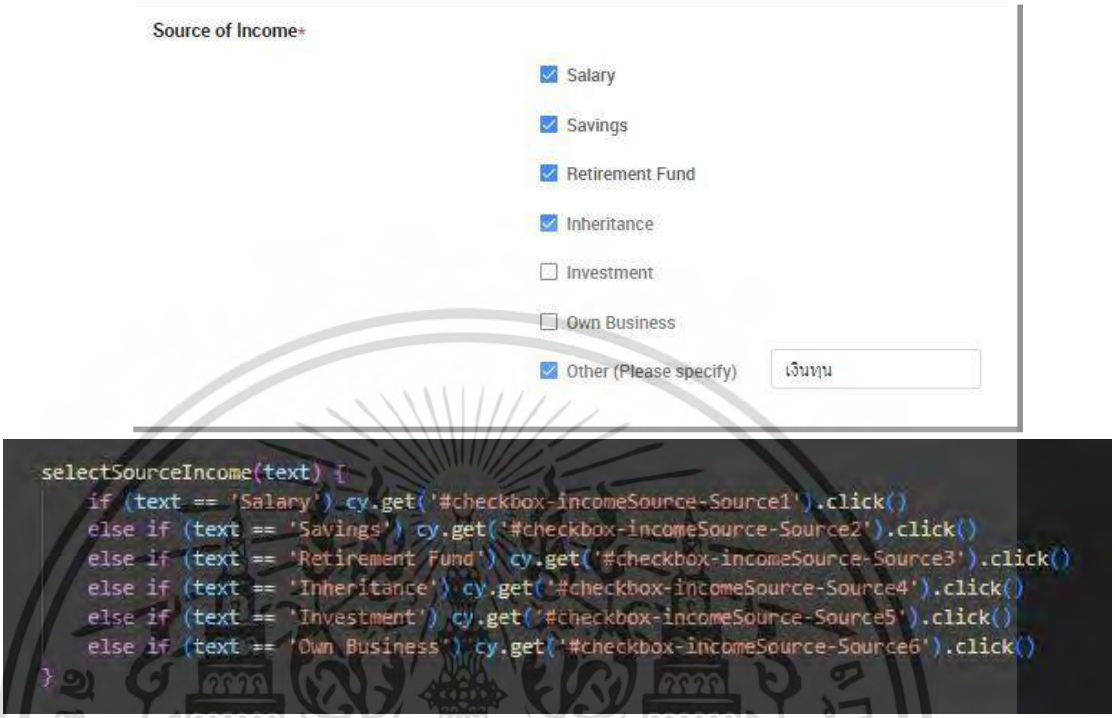
ช่องรับ Input แบบตัวเลือก เลือกตัวเลือกอย่างในอย่างหนึ่งบนหน้าจอ ส่วนใหญ่จะเป็นการใช้คำสั่ง Click() เพื่อเลือกตัวเลือกที่ต้องการ ดังรูป 3.10



รูปที่ 3.10 ช่องรับข้อมูลแบบตัวเลือก

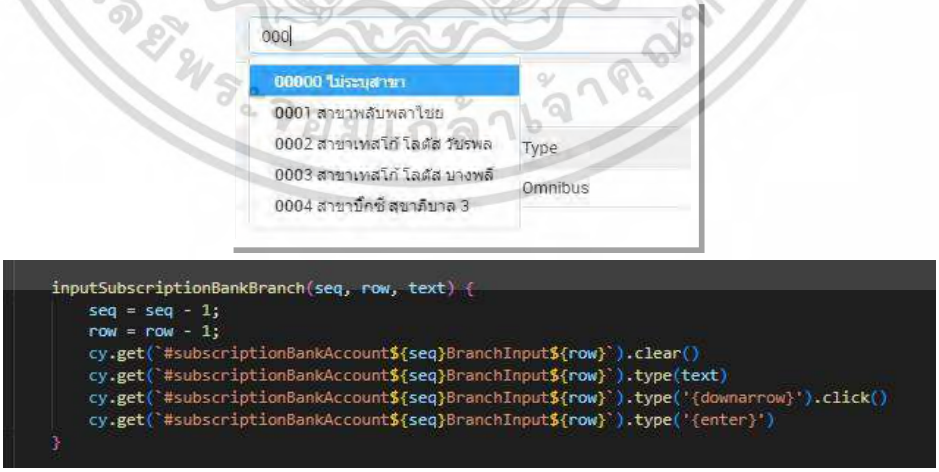
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่องรับ Input แบบตัวเลือกเลือกตัวเลือกอย่างในอย่างหนึ่งหรือมากกว่าบนหน้าจอ ส่วนใหญ่จะเป็นการใช้คำสั่ง Click() เพื่อเลือกตัวเลือกที่ต้องการ ดังรูป 3.11



รูปที่ 3.11 ช่องรับข้อมูลแบบ Checkbox

ช่องรับข้อมูลแบบSearch Option เป็นช่องรับข้อมูลที่จะแสดงตัวเลือกให้เลือกเมื่อทำการค้นหาจะมีการเพิ่มคำสั่ง {downarrow} , {enter} ดังรูป 3.12



รูปที่ 3.12 ช่องรับข้อมูลแบบ Search Option

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 1 การทดสอบ Automated Functional Test แบบ input Data

Requirement : SAเปิดบัญชีลูกค้าประเภท Individual ไม่สามารถเปิดบัญชีได้หากไม่มีการกรอกข้อมูล Background Information ในFieldต่อไปนี้

- Occuoation
- Business Type
- Workplace Information
- Source income
- Monthly income
- Fx risk
- Derivatives risk
- Suitability form
- ValnerableDescription
- Fatca form

โดยจะแสดงข้อความ Required Field บริเวณด้านล่างช่องรับข้อมูลหากไม่กรอกข้อมูลก่อนบันทึก

ตารางที่ 3.3 Test Script การทดสอบ Automated Functional Test แบบ input Data ตัวอย่างที่ 1

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result
1	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึกBackground เมื่อไม่กรอกข้อมูล Occupation	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง
2	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึกBackground เมื่อไม่กรอกข้อมูล Occupation Other	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

						Required Field ได้ถูกต้อง
3	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึกBackground เมื่อไม่กรอกข้อมูล Business Type	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง
4	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึกBackground เมื่อไม่กรอกข้อมูล Business Type Other	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง
5	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึกBackground เมื่อไม่กรอกข้อมูล Workplace information	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง
6	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึกBackground เมื่อไม่กรอกข้อมูล Source of income	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง
7	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึกBackground เมื่อไม่กรอกข้อมูล monthly income	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง
8	Web	SA	Account Manage	Open	- ตรวจสอบการบันทึกBackground	- ไม่สามารถบันทึกข้อมูล Backgroundได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			ment	Account	เมื่อไม่กรอกข้อมูล fx risk	ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง
9	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึกBackground เมื่อไม่กรอกข้อมูล derivatives risk	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง
10	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึกBackground เมื่อไม่กรอกข้อมูล suitability form	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง
11	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึกBackground เมื่อไม่กรอกข้อมูล vulnerable detail	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง
12	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึกBackground เมื่อไม่กรอกข้อมูล fatca form	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง
13	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึกBackground เมื่อกรอกข้อมูลครบ	- สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทดสอบ

เตรียมการเปิดบัญชีประเภท Individual ข้อมูล Personal Information ข้อมูล Address Information และข้อมูลสำหรับ Background Information

```
Personalinformation.selectCardType('Passport')
Personalinformation.selectPassportCountry('Angola')
Personalinformation.inputCardNo('[redacted]')
Personalinformation.inputExpiryDate('10/10/2030')
Personalinformation.selectApplicationForm('Single Form')
Personalinformation.inputApplicationReviewDate('12121919')
Personalinformation.selectTitle('Other')
Personalinformation.inputTitleOther('พลตรี')
Personalinformation.inputThaiName('[redacted]')
Personalinformation.inputThaiSurname('[redacted]')
Personalinformation.inputEngName('[redacted]')
Personalinformation.inputEngSurname('[redacted]')
Personalinformation.inputBirthDate('12121990')
Personalinformation.selectNationality('Thai')
Personalinformation.inputMobilePhone('0123456789')
Personalinformation.selectMaritalStatus('Single')
Personalinformation.clickSaveTab1()
Addressinformation.checkAddressTabSelected()
```

รูปที่ 3.13 ตัวอย่างการใส่ข้อมูล Personal Information

API ในการลบข้อมูลบัญชี Individual โดยใช้ Card Type และ Card No.

```
Cypress.Commands.add('deleteIndividualInvestorByPassport', (cardType, cardNo) => {
    cy.request({
        method: 'GET',
        url:
            urldeleteIndividualInvestorByPassport +
            'cardNo=' +
            cardNo +
            '&cardType=' +
            cardType,
    }).then(response => {
        expect(response).property('status').to.equal(200)
        const apiGet = response.body
        cy.log(apiGet)
    });
});
```

รูปที่ 3.14 ตัวอย่าง API ในการลบข้อมูลบัญชี Individual

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างหน้าจอเมนูเปิดบัญชี

รูปที่ 3.15 หน้าจอเมนูเปิดบัญชี Individual Background Tab

ขั้นตอนการเข้าสู่ระบบ FundConnext

```
loginFundConnext.goToFundConnext()
loginFundConnext.typeUsername('robot_sa1_mkt1')
loginFundConnext.clickLoginButton()
groupMenu.clickMenu('menuAccountManagement')
subMenu.clickMenu('menuOpenAccount')
investorType.clickIndividualAccount()
```

รูปที่ 3.16 ตัวอย่างการเข้าสู่ระบบ FundConnext

ลบข้อมูลจากการทดสอบรอบก่อนหน้าด้วยการใส่ไว้ภายใต้ Before เพื่อให้ทำงานก่อนเริ่มกรณีทดสอบ

```
before(() => {
    cy.deleteIndividualInvestorByPassport('PASSPORT', '[redacted]')
})
```

รูปที่ 3.17 ตัวอย่างการเรียกใช้ API ในการลบข้อมูลบัญชี Individual

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 1 ไม่สามารถบันทึก Background Information เมื่อไม่กรอกข้อมูล Occupation

```
Open Cypress | Set ".only"
it('Cannot save background information when not select Occupation', () => {
    Backgroundinformation.clickSaveTab3()
    Backgroundinformation.shouldShowOccupationErrorMessage()
})
```

รูปที่ 3.18 Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 1

กรณีทดสอบที่ 2 ไม่สามารถบันทึก Background Information เมื่อไม่กรอกข้อมูล Occupation Other

```
Open Cypress | Set ".only"
it('Cannot save background information when not input occupation other', () => {
    Backgroundinformation.selectOccupation('Other')
    Backgroundinformation.clickSaveTab3()
    Backgroundinformation.shouldShowOtherOccupationErrorMessage()
})
```

รูปที่ 3.19 Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 2

กรณีทดสอบที่ 3 ไม่สามารถบันทึก Background Information เมื่อไม่กรอกข้อมูล Business Type

```
Open Cypress | Set ".only"
it('Cannot save background information when not select business type when select occupation that require business',
    Backgroundinformation.inputOccupationOther('Occupation')
    Backgroundinformation.clickSaveTab3()
    Backgroundinformation.shouldShowBusinessErrorMessage()
})
```

รูปที่ 3.20 Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 3

กรณีทดสอบที่ 4 ไม่สามารถบันทึก Background Information เมื่อไม่กรอกข้อมูล Business Type Other

```
Open Cypress | Set ".only"
it('Cannot save background information when not input business type other', () => {
    Backgroundinformation.selectBusinessTypeOther()
    Backgroundinformation.clickSaveTab3()
    Backgroundinformation.shouldShowOtherBusinessErrorMessage()
})
```

รูปที่ 3.21 Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 5 ไม่สามารถบันทึก Background Information เมื่อไม่กรอกข้อมูล Workplace

```
Open Cypress | Set ".only"
it('Cannot save background information when not workplace address
    Backgroundinformation.inputBusinessTypeOther('Business Type')
    Backgroundinformation.clickSaveTab3()
    Backgroundinformation.shouldShowWorkplaceErrorMessage()
})
```

รูปที่ 3.22 Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 5

กรณีทดสอบที่ 6 ไม่สามารถบันทึก Background Information เมื่อไม่กรอกข้อมูล Income

```
Open Cypress | Set ".only"
it('Cannot save background information when not select source of income', () => {
    Backgroundinformation.inputWorkplaceName('The Stock Exchange of Thailand')
    Backgroundinformation.inputWorkplaceAddressNo('93')
    Backgroundinformation.inputWorkplaceAddressMooNo('8')
    Backgroundinformation.inputWorkplaceAddressBuildingMooban('A')
    Backgroundinformation.inputWorkplaceAddressRoomNo('10')
    Backgroundinformation.inputWorkplaceAddressFloor('11')
    Backgroundinformation.inputWorkplaceAddressSoi('9')
    Backgroundinformation.inputWorkplaceAddressRoad('Rama IV')
    Backgroundinformation.selectWorkplaceAddressCountry('Thailand')
    Backgroundinformation.inputWorkplaceAddressPostalCodeWithDistrict('53180','นาซม')
    Backgroundinformation.inputWorkplaceAddressPosition('JuniorMan')
    Backgroundinformation.clickSaveTab3()
    Backgroundinformation.shouldShowSourceOfIncomeErrorMessage()
})
```

รูปที่ 3.23 Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 6

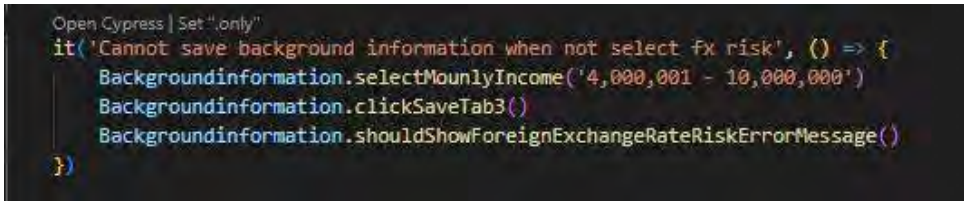
กรณีทดสอบที่ 7 ไม่สามารถบันทึก Background Information เมื่อไม่กรอกข้อมูล Monthly income

```
Open Cypress | Set ".only"
it('Cannot save background information when not select monthly income', () => {
    Backgroundinformation.selectSourceOfIncomeOther()
    Backgroundinformation.inputSourceOfIncomeOther('Income Source')
    Backgroundinformation.clickSaveTab3()
    Backgroundinformation.shouldShowMonthlyIncomeErrorMessage()
})
```

รูปที่ 3.24 Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 7

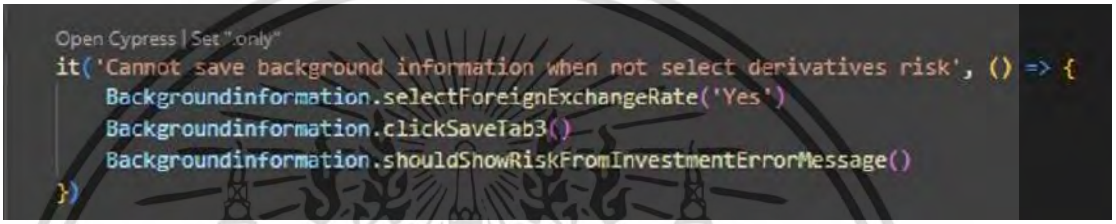
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 8 ไม่สามารถบันทึก Background Information เมื่อไม่กรอกข้อมูล fx risk



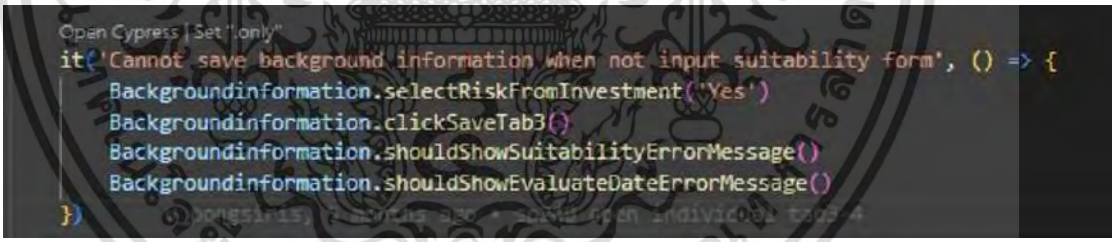
รูปที่ 3.25 Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 8

กรณีทดสอบที่ 9 ไม่สามารถบันทึก Background Information เมื่อไม่กรอกข้อมูล derivatives risk



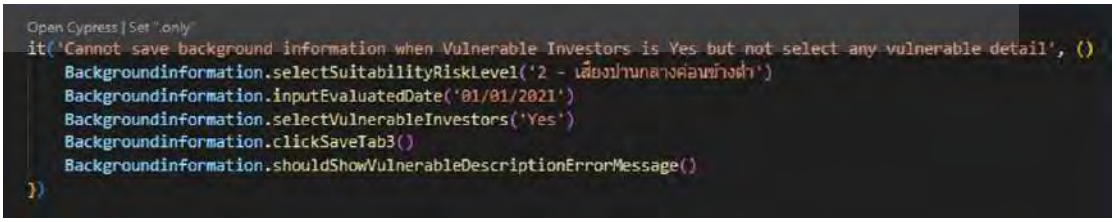
รูปที่ 3.26 Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 9

กรณีทดสอบที่ 10 ไม่สามารถบันทึก Background Information เมื่อไม่กรอกข้อมูล suitability form



รูปที่ 3.27 Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 10

กรณีทดสอบที่ 11 ไม่สามารถบันทึก Background Information เมื่อไม่กรอกข้อมูล Vulnerable



รูปที่ 3.28 Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 12 ไม่สามารถบันทึก Background Information เมื่อไม่กรอกข้อมูล fatca form

```
Open Cypress | Set ".only"
it('Cannot save background information when not input fatca form', () => {
    Backgroundinformation.selecVulnerableDescription('ผู้ที่มีอายุตั้งแต่ 60 ปีขึ้นไป')
    Backgroundinformation.clickSaveTab3()
    Backgroundinformation.shouldShowFatcaErrorMessage()
    Backgroundinformation.shouldShowFatcaDateErrorMessage()
})
```

รูปที่ 3.29 Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 12

กรณีทดสอบที่ 13 สามารถบันทึก Background Information ได้ เมื่อกรอกข้อมูลครบถ้วน

```
Open Cypress | Set ".only"
it('Can save background information when not input Asset Value (Baht),Referral and Accepted by', () => {
    Backgroundinformation.selectFatca('Yes')
    Backgroundinformation.inputFatcaDeclarationDate('01/01/2021')
    Backgroundinformation.clickSaveTab3()
    Accountinformation.checkAccountTabSelected()
})
```

รูปที่ 3.30 Script ในการทำ Automated Test input Data ตัวอย่างที่ 1 กรณีที่ 13

ตัวอย่างที่ 2 การทดสอบ Automated Functional Test แบบ input Data

Requirement : SAเปิดบัญชีลูกค้าประเภท By Account สามารถเปิดบัญชีและเรียกดูข้อมูลหลังมีการบันทึกได้ ซึ่งการเปิดบัญชี By Account จะต้องใช้บัญชีประเภท Juristicที่ไม่ติดแม่กุญแจเท่านั้น หากกรอกบัญชีที่ติดแม่กุญแจจะแสดงข้อความ Error “ID inactive. Please approve individual/juristic profile.” และกรณีที่กรอกบัญชีที่ไม่มีในระบบจะขึ้นแจ้งเตือนว่า “Please create new individual/juristic profile”

ตารางที่ 3.4 Test Script การทดสอบ Automated Functional Test แบบ input Data ตัวอย่างที่ 2

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result
1	Web	SA	Account Management	Open Account	- ตรวจสอบError Message เมื่อใส่ Card Noที่ไม่มีในระบบ	- แสดงError Message “Please create new individual/juristic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

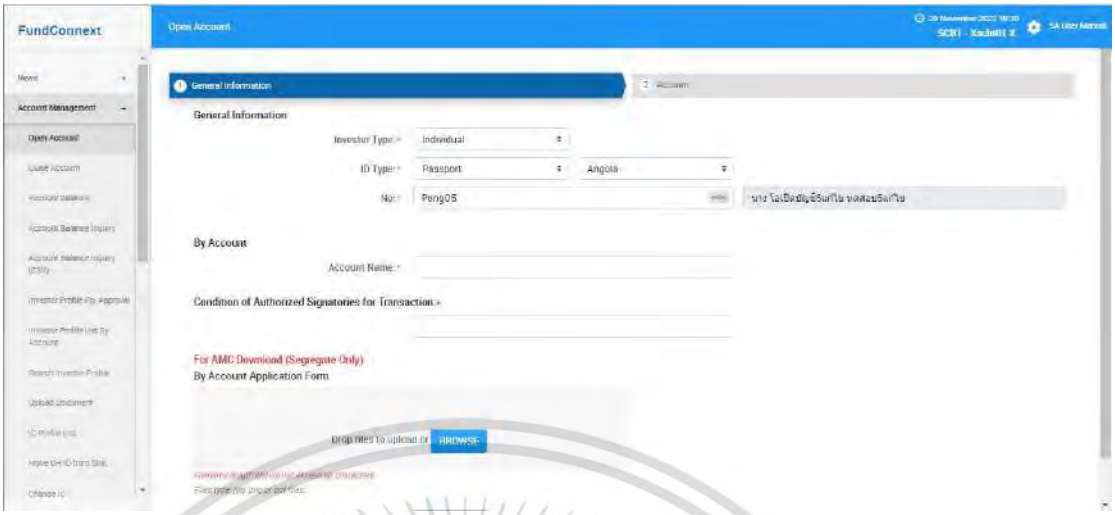
						profile”ได้ถูกต้อง
2	Web	SA	Account Management	Open Account	- ตรวจสอบError Message เมื่อใส่บัญชีJuristicที่ติดแม่กุญแจ	- แสดงError Message “ID inactive. Please approve individual/juristic profile”ได้ถูกต้อง
3	Web	SA	Account Management	Open Account	- ตรวจสอบการเปิดบัญชีBy Account เมื่อใส่Card No บัญชีJuristicที่ไม่ติดแม่กุญแจ	- สามารถเปิดบัญชี By Account ได้ถูกต้อง
4	Web	SA	Account Management	Open Account	- ตรวจสอบข้อมูลหลังจากเปิดบัญชี By Account	- สามารถเปิดบัญชี By Account ได้ถูกต้อง - แสดงข้อมูลที่เปิดบัญชีได้ถูกต้อง

ขั้นตอนการทดสอบ

เตรียมข้อมูลสำหรับการเปิดบัญชีประเภท Juristic ข้อมูล Personal information ข้อมูล Address Information และ Background information เพื่อนำมาเปิดบัญชี By Account และข้อมูลสำหรับเปิดบัญชี By Account

รูปที่ 3.31 หน้าจอเมนูเปิดบัญชี ByWhom ผ่านบัญชี Juristic (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.32 หน้าจอเมนูเปิดบัญชี ByWhom ผ่านบัญชี Juristic (2)

API สำหรับลบข้อมูลบัญชี Juristic โดยรับค่า JurRegisNo และ saCode

```
const urlDeleteJuristicAccountInvestor = '                    n/deleteInvestor?'
// Delete Juristic Account Investor
Cypress.Commands.add('deleteJuristicAccountInvestor', (jurRegisNo,saCode) => {
    cy.request({
        method: 'GET',
        url:
        urlDeleteJuristicAccountInvestor +
            'jurRegisNo=' +
            jurRegisNo +
            '&saCode=' +
            saCode,
    }).then(response => {
        expect(response).property('status').to.equal(200)
        const apiGet = response.body
        cy.log(apiGet)
    });
});
```

รูปที่ 3.33 ตัวอย่าง API ในการลบข้อมูลบัญชี Juristic

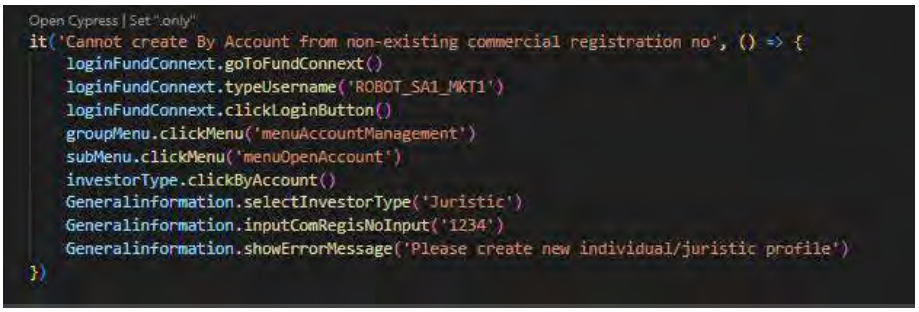
เรียกใช้ API ในการลบข้อมูลบัญชี Juristic เพื่อลบข้อมูลจากการทดสอบรอบก่อนหน้าด้วยการใส่ไว้ภายใต้ Before เพื่อให้ทำงานก่อนเริ่มกรณีทดสอบ

```
before(() => {
    cy.deleteJuristicAccountInvestor('ROBOTJUR_BYWH', 'ROBOTSA1')
})
```

รูปที่ 3.34 ตัวอย่างการเรียกใช้ API ในการลบข้อมูลบัญชี Individual

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

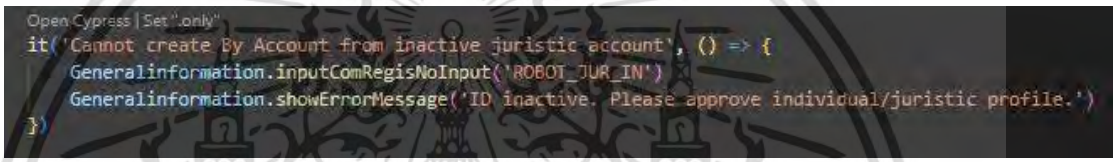
กรณีทดสอบที่ 1 ไม่สามารถเปิดบัญชี By Whom ได้หากใส่เลขบัญชี Juristic ที่ไม่มีในระบบ



```
Open Cypress | Set ".only"
it('Cannot create By Account from non-existing commercial registration no', () => {
    loginFundConnext.goToFundConnext()
    loginFundConnext.typeUsername('ROBOT_SA1_MKT1')
    loginFundConnext.clickLoginButton()
    groupMenu.clickMenu('menuAccountManagement')
    subMenu.clickMenu('menuOpenAccount')
    investorType.clickByAccount()
    Generalinformation.selectInvestorType('Juristic')
    Generalinformation.inputComRegisNoInput('1234')
    Generalinformation.showErrorMessage('Please create new individual/juristic profile')
})
```

รูปที่ 3.35 Script ในการทำ Automated Test input Data ตัวอย่างที่ 2 กรณีที่ 1

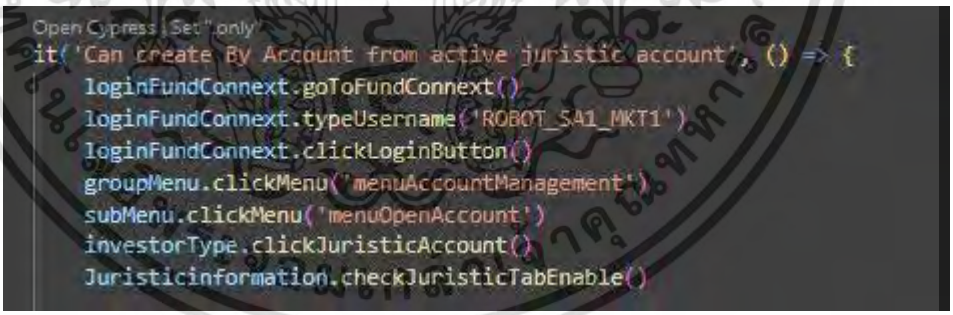
กรณีทดสอบที่ 2 ไม่สามารถเปิดบัญชี By Whom ได้หากใส่เลขบัญชี Juristic ที่ติดแม่กุญแจ (ยังไม่ได้Approve)



```
Open Cypress | Set ".only"
it('Cannot create By Account from inactive juristic account', () => {
    Generalinformation.inputComRegisNoInput('ROBOT_JUR_IN')
    Generalinformation.showErrorMessage('ID inactive. Please approve individual/juristic profile.')
})
```

รูปที่ 3.36 Script ในการทำ Automated Test input Data ตัวอย่างที่ 2 กรณีที่ 2

กรณีทดสอบที่ 3 สามารถเปิดบัญชี By Whom ได้หากใส่เลขบัญชี Juristic มีในระบบและไม่ติดแม่กุญแจ (เข้าสู้หน้าเปิดบัญชีJuristic)



```
Open Cypress | Set ".only"
it('Can create By Account from active juristic account', () => {
    loginFundConnext.goToFundConnext()
    loginFundConnext.typeUsername('ROBOT_SA1_MKT1')
    loginFundConnext.clickLoginButton()
    groupMenu.clickMenu('menuAccountManagement')
    subMenu.clickMenu('menuOpenAccount')
    investorType.clickJuristicAccount()
    Juristicinformation.checkJuristicTabEnable()
```

รูปที่ 3.37 Script ในการทำ Automated Test input Data ตัวอย่างที่ 2 กรณีที่ 3 (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 3 สามารถเปิดบัญชี By Whom ได้หากใส่เลขบัญชี Juristic มีในระบบและไม่ติดแม่กุญแจ (กรอกข้อมูล Juristic)

```
Juristicinformation.inputCommercialRegistrationNo('ROBOTJUR_BYWH')
Juristicinformation.inputJuristicThaiName('JUR_BY_ACC')
Juristicinformation.selectApplicationForm('Single Form')
Juristicinformation.inputApplicationReviewDate('01/01/2018')
Juristicinformation.selectRegisCountry('Thailand')
Juristicinformation.inputJuristicEngName('[redacted]')
Juristicinformation.inputTaxId('1234567890123')
Juristicinformation.selectBranch('Head Office')
Juristicinformation.inputAnnualIncome('123456789')
Juristicinformation.selectJuristicType('Juristic (Thailand)')
Juristicinformation.selectTaxExempt('Tax Exempt on Dividend and Capital Gain')
Juristicinformation.selectBusinessType('Antique Trading')
Juristicinformation.inputAssetValue('11223344.55')
Juristicinformation.inputShareholderEquity('660011.22')
Juristicinformation.selectSourceOfInvestmentFund('Donation')
Juristicinformation.selectSourceOfInvestmentFund('Loan')
Juristicinformation.selectCountrySource('Thailand')
Juristicinformation.selectCountry('Thailand')
Juristicinformation.inputPostalCodeWithDistrict('45000', 'หนองแวงควง')
Juristicinformation.inputAddressNo('8/3')
Juristicinformation.inputTelephone('022292272')
```

รูปที่ 3.38 Script ในการทำ Automated Test input Data ตัวอย่างที่ 2 กรณีที่ 3 (2)

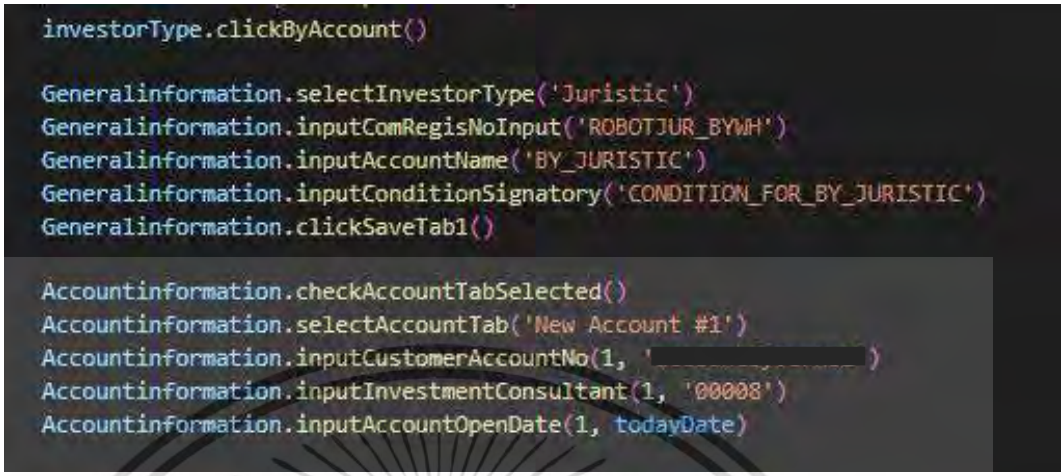
กรณีทดสอบที่ 3 สามารถเปิดบัญชี By Whom ได้หากใส่เลขบัญชี Juristic มีในระบบและไม่ติดแม่กุญแจ (Approve บัญชี Juristic)

```
loginFundConnext.goToFundConnext()
loginFundConnext.typeUsername('ROBOT_SA1_ACAPR')
loginFundConnext.clickLoginButton()
groupMenu.clickMenu('menuAccountManagement')
subMenu.clickMenu('menuInvestorProfileApproval')
menuInvestorProfileForApproval.filterInvestorName('JUR_BY_ACC')
menuInvestorProfileForApproval.clickNameLinkToOpenInvestorProfilePreviewPage()
menuInvestorProfileForApproval.clickApproveTabName('0')
```

รูปที่ 3.39 Script ในการทำ Automated Test input Data ตัวอย่างที่ 2 กรณีที่ 3 (3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 3 สามารถเปิดบัญชี By Whom ได้หากใส่เลขบัญชี Juristic มีในระบบและไม่ติดแม่กุญแจ (กรอกข้อมูลบัญชี By Whom)



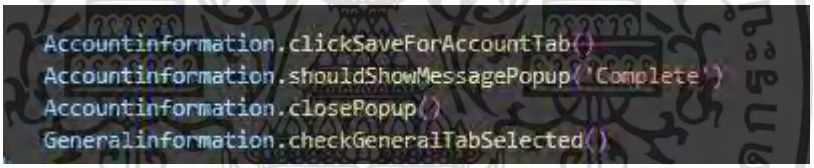
```
investorType.clickByAccount()

Generalinformation.selectInvestorType('Juristic')
Generalinformation.inputComRegisNoInput('ROBOTJUR_BYWH')
Generalinformation.inputAccountName('BY_JURISTIC')
Generalinformation.inputConditionSignatory('CONDITION_FOR_BY_JURISTIC')
Generalinformation.clickSaveTab1()

Accountinformation.checkAccountTabSelected()
Accountinformation.selectAccountTab('New Account #1')
Accountinformation.inputCustomerAccountNo(1, '                    ')
Accountinformation.inputInvestmentConsultant(1, '00008')
Accountinformation.inputAccountOpenDate(1, todayDate)
```

รูปที่ 3.40 Script ในการทำ Automated Test input Data ตัวอย่างที่ 2 กรณีที่ 3 (4)

กรณีทดสอบที่ 3 สามารถเปิดบัญชี By Whom ได้หากใส่เลขบัญชี Juristic มีในระบบและไม่ติดแม่กุญแจ (กรอกข้อมูลบัญชี By Whom)



```
Accountinformation.clickSaveForAccountTab()
Accountinformation.shouldShowMessagePopup('Complete')
Accountinformation.closePopup()
Generalinformation.checkGeneralTabSelected()
```

รูปที่ 3.41 Script ในการทำ Automated Test input Data ตัวอย่างที่ 2 กรณีที่ 3 (5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 Automated Test แบบ Look and Filed

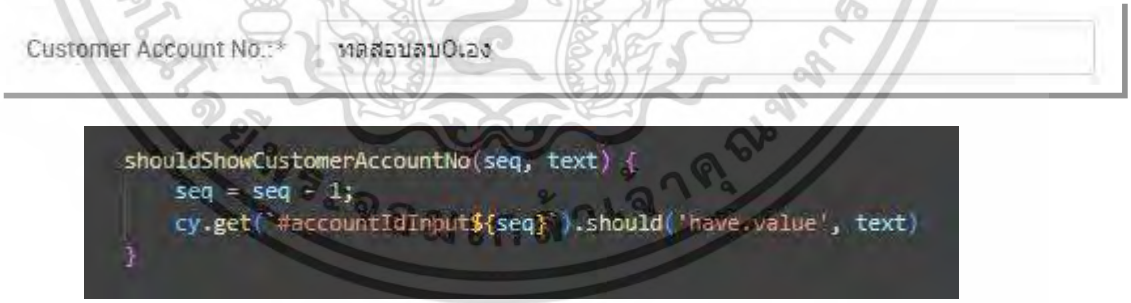
การทดสอบประเภท Functional Test แบบ Look and Filed โดย Look เป็นการตรวจสอบข้อมูลที่แสดงบนหน้าจอว่า แสดงข้อมูลถูกต้องและตรงกับฐานข้อมูล หน้าจอไม่นำข้อมูลจากส่วนอื่นมาแสดง ส่วน Field เป็นการกรองข้อมูลที่จะนำมาแสดงหลักจากผู้ใช้งานใส่ข้อมูลต่างๆ โดยเลือกผ่านตัวเลือกในหน้าจอต่างๆ

การทำ Test Script และเตรียม Data Test แบบเป็น 2 กรณีหลัก คือ

1. กรณีการทำ Test Script และเตรียม Data Test สำหรับ Look ต้องแสดงหน้าจอและข้อมูลในฐานข้อมูลที่ตรงกัน
2. กรณีการทำ Test Script และเตรียม Data Test สำหรับ Field การแสดงหน้าจอหลังจากการกรองข้อมูลผ่านหน้าจอต่างๆทั้งตัวเลือกผ่านหน้าจอต้องแสดงข้อมูลที่ถูกต้อง และไม่แสดงบนหน้าจอหากไม่มีข้อมูลอยู่ในฐานข้อมูล

การทดสอบแบบ Automated Test Look and Field เป็นการตรวจสอบข้อมูลที่แสดงหน้าจอได้แสดงถูกต้องหรือแสดงข้อผิดพลาดตามที่ต้องการหรือไม่ หลังจากมีการบันทึกข้อมูล โดยคำสั่งส่วนใหญ่ที่ใช้ทดสอบแบบ Look and Field คือ คำสั่ง should(), contain()

ตรวจสอบการแสดงผลข้อมูลเมื่อทำการกรอกข้อมูลลงในช่องรับข้อมูล หรือข้อความที่แสดงบนหน้าจอจะเป็นการใช้คำสั่ง Should(have.value, Expect value) ดังรูป 3.42



รูปที่ 3.42 ช่องรับข้อมูลแบบธรรมดา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรวจสอบการแสดงผลช่องรับ Input แบบdropdown เมื่อทำการเลือกตัวเลือกใน Dropdown แล้วใช้คำสั่ง contains(Expect value) ดังรูป 3.43

รูปที่ 3.43 ช่องรับข้อมูลแบบ Dropdown

ตรวจสอบการแสดงผลช่องรับ Input ตัวเลือก เมื่อทำการเลือกตัวเลือกใดตัวเลือกหนึ่ง จะใช้คำสั่ง should(have.class, mat-radio-checked) โดยขึ้นอยู่กับแต่ละหน้าจอ และความต้องการ ดังรูป 3.44

รูปที่ 3.44 ช่องรับข้อมูลแบบตัวเลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรวจสอบการแสดงผลช่องรับ Input แบบCheckbox เมื่อทำการเลือกตัวเลือดใดตัวเลือกหนึ่งหรือมากกว่าหนึ่งตัวเลือก จะใช้คำสั่ง should(have.class, mat-checkbox-checked) โดยขึ้นอยู่กับแต่ละหน้าจอและความต้องการ ดังรูป 3.45

```
shouldShowOccupationOther() {
    cy.get('#checkbox-occupation-170-0').should('have.class', 'mat-checkbox-checked')
}
```

รูปที่ 3.45 ช่องรับข้อมูลแบบ Checkbox

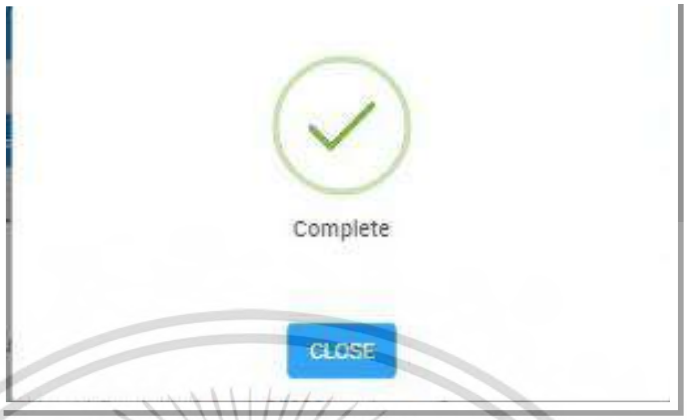
ตรวจสอบการทำงานของปุ่มหรือช่องรับข้อมูลต่างๆ โดยใช้คำสั่งshould(be.enabled) และ should(be.disabled) เพื่อทดสอบการทำงานของปุ่มหรือช่องรับข้อมูล ดังรูป 3.46

```
searchButtonDisabled() {
    cy.get('#searchBtn').should('be.disabled')
}
```

รูปที่ 3.46 ช่องรับข้อมูลที่ไม่ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรวจสอบการแสดงผลของModal เช่น Success, Complete, Error, ข้อความเพิ่มเติมที่แสดงในModal รวมถึงModalอื่นๆ ดังรูป 3.47




```
showSuccessModal() {
  cy.get('.modal-dialog .modal-title.text-center').should('contain', 'Success').wait(100);
  cy.get('.modal-dialog #closeBtn').click()
}

showErrorModal(title, message) {
  cy.get('#error-title').should('contain', title);
  cy.get('#error-message p').should('contain', message);
  cy.wait(500);
  cy.get('#closeBtn').click()
}
```

รูปที่ 3.47 ตัวอย่างหน้าจอ Complete

ตัวอย่างที่ 1 การทดสอบ Automated Functional Test แบบ Look and Field

Requirement : แก้ไขเมนู Open Account จะแสดงเป็นแบบ Hierarchy โดยจะแสดง Tab Account ตาม Invesment Consultant ที่สังกัดอยู่ภายใต้สาขาและทีมนั้นๆ ซึ่งจะต้องมีการทดสอบในบัญชีทุกประเภท เช่น Individual, Juristic, Joint, By Whom, For Whom

ตารางที่ 3.5 Test Script การทดสอบ Automated Functional Test แบบ Look and Field ตัวอย่างที่ 1

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result
1	Web	SA	Account Management	Open Account	- ตรวจสอบการแสดง Account	- แสดง Account ทั้งหมดได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

					ที่หน้าจอเมื่อ Login : HQ	
2	Web	SA	Account Management	Open Account	- ตรวจสอบการแสดง Account ที่หน้าจอเมื่อ Login : B1	- แสดง Account ใน Branch1 เท่านั้น
3	Web	SA	Account Management	Open Account	- ตรวจสอบการแสดง Account ที่หน้าจอเมื่อ Login : B1T1 แล้วกด ที่Account Tab	- แสดง Account ใน Branch1 Team1 เท่านั้น
4	Web	SA	Account Management	Open Account	- ตรวจสอบการแสดง Account ที่หน้าจอเมื่อ Login : B1T1 แล้วกดSaveเพื่อ เพื่อไปยัง Account Tab	- แสดง Account ใน Branch1 Team1 เท่านั้น

ขั้นตอนการทดสอบ

เตรียมข้อมูล Account ไว้ภายใต้ Branch ต่างๆ

รูปที่ 3.48 หน้าจอเมนู Account Tab

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันที่ใช้ในการตรวจสอบการแสดง Account ที่อยู่ภายใต้ Branch และ Team

```
function showAllaccount() {
    Accountinformation.shouldShowAccount('_ACCB1T1A1') //B1T1
    Accountinformation.shouldShowAccount('_ACCB2T2A1') //B2T2
    Accountinformation.shouldShowAccount('1_ACCB1T1_ICB2') //B2
    Accountinformation.shouldShowAccount('1_ACCB1T1A1') //B1T1
    Accountinformation.shouldShowAccount('1_ACCB1T2A1') //B1T2
    Accountinformation.shouldShowAccount('1_ACCB2') //B2
    Accountinformation.shouldShowAccount('1_ACCB2T1A1') //B2T1
    Accountinformation.shouldShowAccount('1_ACCB2T2A1') //B2T2
    Accountinformation.shouldShowAccount('OMN_B1T1A1') //B1T1
}

function showBranch1_account() {
    Accountinformation.shouldShowAccount('_ACCB1T1A1') //B1T1
    Accountinformation.shouldShowAccount('1_ACCB1T1A1') //B1T1
    Accountinformation.shouldShowAccount('1_ACCB1T2A1') //B1T2
    Accountinformation.shouldShowAccount('OMN_B1T1A1') //B1T1
}

function showBranch1_Team1_account() {
    Accountinformation.shouldShowAccount('_ACCB1T1A1') //B1T1
    Accountinformation.shouldShowAccount('1_ACCB1T1A1') //B1T1
    Accountinformation.shouldShowAccount('OMN_B1T1A1') //B1T1
}
```

รูปที่ 3.49 Script ในการทำ Automated Test look and Field ฟังก์ชันการตรวจสอบ Account

โดยใช้คำสั่งดังรูป 3.50 ในการตรวจสอบการแสดง Account ที่อยู่ภายใต้ Branch และ Team

```
shouldShowAccount(name) {
    cy.contains(name).should('exist')
}
```

รูปที่ 3.50 Script ในการทำ Automated Test look and Field คำสั่งการตรวจสอบ Account

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 1 เมื่อเข้าสู่ระบบด้วย User ที่เป็น HQ สามารถเห็น Account ทั้งหมดภายใต้ SA ได้

```
Open Cypress | Set "only"
it('User Head Quater can see all account of branch and team correctly', () => {
    // Ref#3640
    loginFundConnext.goToFundConnext()
    loginFundConnext.typeUsername('ROBOT_HIER')
    loginFundConnext.clickLoginButton()
    groupMenu.clickMenu('menuAccountManagement')
    subMenu.clickMenu('menuOpenAccount')
    investortype.clickIndividualAccount()

    Personalinformation.selectCardType('Passport')
    Personalinformation.selectPassportCountry('Angola')
    Personalinformation.inputCardNo('ROBOTB1T1')
    // กรณีรันแล้วตายบนserver ยังไม่ได้จัดการเรื่องเพิ่มwait เนื่องจากบางครั้งโหลดข้อมูลไม่ทัน
    Accountinformation.clickAccountTab()
    showAllaccount()
})
```

รูปที่ 3.51 Script ในการทำ Automated Test look and Field ตัวอย่างที่ 1 กรณีที่ 1

กรณีทดสอบที่ 2 เมื่อเข้าสู่ระบบด้วย User ระดับBranch สามารถเห็นAccountทั้งหมดภายใต้ Branch เท่านั้น

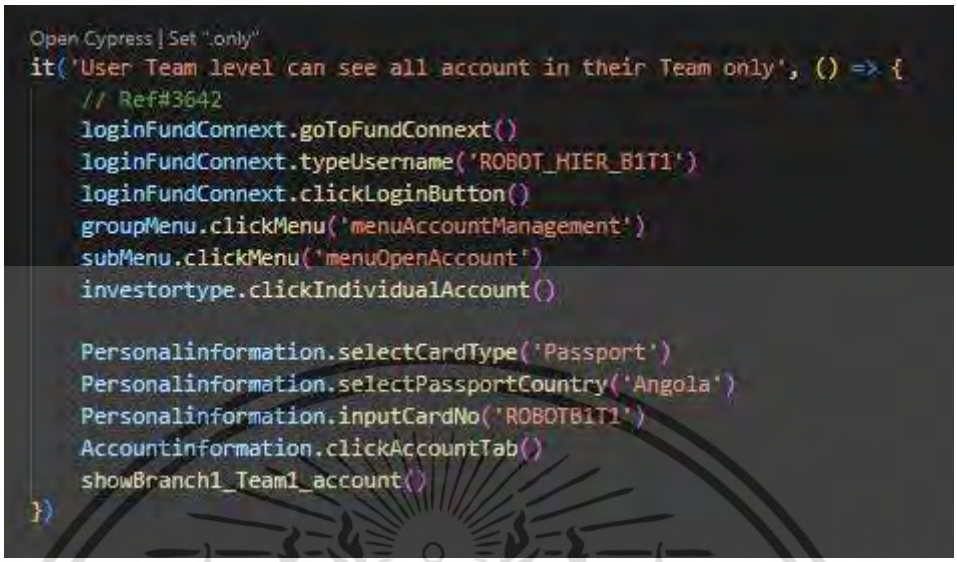
```
Open Cypress | Set "only"
it('User Branch level can see all account in same branch', () => {
    // Ref#3641
    loginFundConnext.goToFundConnext()
    loginFundConnext.typeUsername('ROBOT_HIER_B1')
    loginFundConnext.clickLoginButton()
    groupMenu.clickMenu('menuAccountManagement')
    subMenu.clickMenu('menuOpenAccount')
    investortype.clickIndividualAccount()

    Personalinformation.selectCardType('Passport')
    Personalinformation.selectPassportCountry('Angola')
    Personalinformation.inputCardNo('ROBOTB1T1')
    Accountinformation.clickAccountTab()
    showBranch1_account()
})
```

รูปที่ 3.52 Script ในการทำ Automated Test look and Field ตัวอย่างที่ 1 กรณีที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 3 เมื่อเข้าสู่ระบบด้วย User ระดับTeam สามารถเห็น Account ทั้งหมดภายใต้ Team เท่านั้น



```
Open Cypress | Set ".only"
it('User Team level can see all account in their Team only', () => {
    // Ref#3642
    loginFundConnext.goToFundConnext()
    loginFundConnext.typeUsername('ROBOT_HIER_B1T1')
    loginFundConnext.clickLoginButton()
    groupMenu.clickMenu('menuAccountManagement')
    subMenu.clickMenu('menuOpenAccount')
    investortype.clickIndividualAccount()

    Personalinformation.selectCardType('Passport')
    Personalinformation.selectPassportCountry('Angola')
    Personalinformation.inputCardNo('ROBOTB1T1')
    Accountinformation.clickAccountTab()
    showBranch1_Team1_account()
})
```

รูปที่ 3.53 Script ในการทำ Automated Test look and Field ตัวอย่างที่ 1 กรณีที่ 3

กรณีทดสอบที่ 4 เมื่อเข้าสู่ระบบด้วย User ระดับ Team สามารถเห็น Account ทั้งหมดภายใต้ Team เท่านั้น เมื่อทำการบันทึกข้อมูล

```
Open Cypress | Set ".only"
it('User Team level can see all account in their Team only when click save to change tab', () => {
    // Ref#3642
    loginFundConnext.goToFundConnext()
    loginFundConnext.typeUsername('ROBOT_HIER_B1T1')
    loginFundConnext.clickLoginButton()
    groupMenu.clickMenu('menuAccountManagement')
    subMenu.clickMenu('menuOpenAccount')
    investortype.clickIndividualAccount()

    Personalinformation.selectCardType('Passport')
    Personalinformation.selectPassportCountry('Angola')
    Personalinformation.inputCardNo('ROBOTB1T1')
    Personalinformation.clickSaveTab1()
    Addressinformation.clickSaveTab2()
    Backgroundinformation.clickSaveTab3()
    showBranch1_Team1_account()
})
```

รูปที่ 3.54 Script ในการทำ Automated Test look and Field ตัวอย่างที่ 1 กรณีที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2 การทดสอบ Automated Functional Test แบบ Look and Field

Requirement : หน้าจอของ TSD ที่ใช้ในการแสดงผลสิทธิการเข้าการใช้งานต่างๆของแต่ละ SA ซึ่งสามารถค้นหาได้มากสุดครั้งละ 3 SA

ตารางที่ 3.6 Test Script การทดสอบ Automated Functional Test แบบ Look and Field ตัวอย่างที่ 2

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result
1	Web	TSD	File Maintenance	SA Profile	- TSD สามารถค้นหาข้อมูลของ SA ได้ครั้งละ 2 SA	- แสดงข้อมูลของ SA ทั้งสองได้ถูกต้อง
2	Web	TSD	File Maintenance	SA Profile	- TSD สามารถค้นหาข้อมูลของ SA ได้ครั้งละ 3 SA	- แสดงข้อมูลของ SA ทั้งสามได้ถูกต้อง

ขั้นตอนการทดสอบ

เตรียมสิทธิในการเข้าถึงการใช้งานต่างๆใน SA ที่เราต้องการตรวจสอบบนหน้าจอ SA Profile

รูปที่ 3.55 หน้าจอเมนู SA Profile (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.56 หน้าจอเมนู SA Profile (2)

ฟังก์ชันที่ใช้ในการตรวจสอบการแสดงข้อมูลในหน้าจอ SA Profile

```
showDataInTableResultSAColumn1(rowName, text) {
    const row = this.getRowSequence(rowName);
    cy.xpath(`                                          tr[${row}] //td[2]`).should('have.text', text)
}

showDataInTableResultSAColumn2(rowName, text) {
    const row = this.getRowSequence(rowName);
    cy.xpath(`                                          tr[${row}] //td[3]`).should('have.text', text)
}

showDataInTableResultSAColumn3(rowName, text) {
    const row = this.getRowSequence(rowName);
    cy.xpath(`                                          tr[${row}] //td[4]`).should('have.text', text)
}
```

```
getRowSequence(rowName) {
    let row = 0;
    switch (rowName) {
        case 'SA Code':
            row = 1;
            break;
        case 'English Name':
            row = 2;
            break;
        case 'Passthrough':
            row = 17;
            break;
        case 'Manage data access by branch':
            row = 39;
            break;
        case 'Use Calculate FSP015':
            row = 43;
            break;
        case 'PND Sender ID':
            row = 53;
            break;
        case 'PND Branch no':
            row = 54;
            break;
        case 'PND Branch Type':
            row = 55;
            break;
```

รูปที่ 3.57 Script ในการทำ Automated Test look and Field ฟังก์ชันตรวจสอบ SA Profile

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 1 TSD สามารถค้นหา SA Profile 2 SAพร้อมกันและแสดงผลได้ถูกต้อง

```
Open Cypress | Set ".only"
it('TSD can search selling agent profile with 2 selling agent', () => {
    loginFundConnext.goToFundConnext()
    loginFundConnext.typeUsername('ROBOT_TSDOPER')
    loginFundConnext.clickLoginButton()
    groupMenu.clickMenu('menuFileMaintenance')
    subMenu.clickMenu('menuSaProfile')
    saProfile.showPage('SA Profile')

    saProfile.selectSellingAgent(1, 'ROBOTSA1')
    saProfile.selectSellingAgent(2, 'ROBOTSA2')
    saProfile.clickSearchButton()

    saProfile.showDataInTableResultSAColumn1('SA Code', 'ROBOTSA1')
    saProfile.showDataInTableResultSAColumn2('SA Code', 'ROBOTSA2')
    saProfile.showDataInTableResultSAColumn3('SA Code', '')

    saProfile.showDataInTableResultSAColumn1('English Name', 'Robot SA1 Normal')
    saProfile.showDataInTableResultSAColumn2('English Name', 'Robot SA2 Passthrough')
    saProfile.showDataInTableResultSAColumn3('English Name', '')

    saProfile.showDataInTableResultSAColumn1('Passthrough', 'N')
    saProfile.showDataInTableResultSAColumn2('Passthrough', 'Y')
    saProfile.showDataInTableResultSAColumn3('Passthrough', '')

    saProfile.showDataInTableResultSAColumn1('Shift Extra holiday', 'Y')
    saProfile.showDataInTableResultSAColumn2('Shift Extra holiday', 'N')
    saProfile.showDataInTableResultSAColumn3('Shift Extra holiday', '')
```

รูปที่ 3.58 Script ในการทำ Automated Test look and Field ตัวอย่างที่ 2 กรณีที่ 1

กรณีทดสอบที่ 2 TSD สามารถค้นหา SA Profile 3 SAพร้อมกันและแสดงผลได้ถูกต้อง

```
Open Cypress | Set ".only"
it('TSD can search selling agent profile with 3 selling agent', () => {
    saProfile.selectSellingAgent(1, 'ROBOTSA1')
    saProfile.selectSellingAgent(2, 'ROBOTSA2')
    saProfile.selectSellingAgent(3, 'ROBOTSA3')
    saProfile.clickSearchButton()

    saProfile.showDataInTableResultSAColumn1('SA Code', 'ROBOTSA1')
    saProfile.showDataInTableResultSAColumn2('SA Code', 'ROBOTSA2')
    saProfile.showDataInTableResultSAColumn3('SA Code', 'ROBOTSA3')

    saProfile.showDataInTableResultSAColumn1('English Name', 'Robot SA1 Normal')
    saProfile.showDataInTableResultSAColumn2('English Name', 'Robot SA2 Passthrough')
    saProfile.showDataInTableResultSAColumn3('English Name', 'Robot SA3 TALIS')

    saProfile.showDataInTableResultSAColumn1('Shift Extra holiday', 'Y')
    saProfile.showDataInTableResultSAColumn2('Shift Extra holiday', 'N')
    saProfile.showDataInTableResultSAColumn3('Shift Extra holiday', 'N')

    saProfile.showDataInTableResultSAColumn1('Use ETF Transfer', 'Y')
    saProfile.showDataInTableResultSAColumn2('Use ETF Transfer', 'N')
    saProfile.showDataInTableResultSAColumn3('Use ETF Transfer', 'N')

    saProfile.showDataInTableResultSAColumn1('Confirmation Note Text Version', 'V1.0')
    saProfile.showDataInTableResultSAColumn2('Confirmation Note Text Version', 'V1.0')
    saProfile.showDataInTableResultSAColumn3('Confirmation Note Text Version', 'V1.0')
})
```

รูปที่ 3.59 Script ในการทำ Automated Test look and Field ตัวอย่างที่ 2 กรณีที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 Automated Test แบบ Look and Filed (ตาราง)

การทดสอบประเภท Functional Test แบบ Look and Filed โดย Look เป็นการตรวจสอบข้อมูลที่แสดงบนหน้าจอว่าแสดงข้อมูลถูกต้องและตรงกับฐานข้อมูล หน้าจอไม่นำข้อมูลจากส่วนอื่นมาแสดง ส่วน Field เป็นการกรองข้อมูลที่จะนำมาแสดงหลักจากผู้ใช้งานใส่ข้อมูลต่างๆ ซึ่งในระบบฟันด์คอนเน็กต์มีหลายหน้าจอและฟังก์ชัน ที่เกี่ยวกับ Look and Filed (Table) โดยเป็นการตรวจสอบข้อมูลภายในตารางหลักจากผู้ใช้งานทำการกรอง เรียงว่าข้อมูลนั้นแสดงได้ถูกต้องหรือไม่

การทำ Test Script และเตรียม Data Test แบบเป็น 2 กรณีหลัก คือ

1. กรณีการทำ Test Script และเตรียม Data Test สำหรับ Look ต้องแสดงหน้าจอและข้อมูลในฐานข้อมูลที่ตรงกัน
2. กรณีการทำ Test Script และเตรียม Data Test สำหรับ Field การแสดงหน้าจอหลังจากการกรอง เรียงข้อมูลในตารางนั้นต้องแสดงข้อมูลที่ถูกต้อง และไม่แสดงบนหน้าจอหากไม่มีข้อมูลอยู่ในฐานข้อมูล

การทดสอบแบบ Automated Test Look and Field เป็นการตรวจสอบข้อมูลที่แสดงหน้าจอได้แสดงถูกต้องหรือแสดงข้อผิดพลาดตามที่ต้องการหรือไม่ หลังจากมีการบันทึกข้อมูล โดยคำสั่งส่วนใหญ่ที่ใช้ทดสอบแบบ Look and Field คือ คำสั่ง should(), contain() ซึ่งการทดสอบข้อมูลในตารางนั้นก็มีข้อแตกต่างกันออกไปทำให้มีรูปแบบในการเขียนAutomated Test แตกต่างออกไปซึ่งคำสั่งฟังก์ชันพื้นฐานในการตรวจสอบตารางมีอยู่ 3 ส่วนหลัก คือ Filter, Sort, CheckData

การสร้างตัวแปรเพื่อกำหนด Column ในตาราง

```
export class ColumnMethodBuilder {
    #parentId
    #colId
    #hasFilter
    #isCheckbox
    #isSelectOption
    constructor(parentId) {
        this.#parentId = parentId;
    }

    build() {
        return new ColumnMethod(this.#parentId, this.#colId, this.#hasFilter, this.#isCheckbox, this.#isSelectOption);
    }

class failXWTorderColumns {
    constructor(parentId) {
        this.paymentDate = new ColumnMethodBuilder(parentId).colId('paymentDate').build();
        this.xwtReferenceNo = new ColumnMethodBuilder(parentId).colId('xwtReferenceNo').build();
        this.accountNo = new ColumnMethodBuilder(parentId).colId('accountNo').build();
        this.counterUnitholderNo = new ColumnMethodBuilder(parentId).colId('counterUnitholderNo').build();
        this.customerName = new ColumnMethodBuilder(parentId).colId('customerName').build();
        this.counterFundCode = new ColumnMethodBuilder(parentId).colId('counterFundCode').build();
        this.amount = new ColumnMethodBuilder(parentId).colId('amount').build();
        this.reason = new ColumnMethodBuilder(parentId).colId('reason').build();
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.60 ฟังก์ชันการ Build Column

คำสั่งในการFilter ข้อมูลในตาราง

```
by: (value) => {
    if (!this.#hasFilter) return;
    cy.wait(500)
    if (this.#isSelectOption) {
        cy.xpath(`${this.#aggridLocator}//*[@col-id="${this.#colId}"]`).invoke('attr', 'aria-colindex')
            .then(colIndex => {
                cy.xpath(this.#aggridLocator)
                    .find(`.ag-header-row-column-filter`)
                    .find(`.ag-header-cell[aria-colindex="${colIndex}"]`)
                    .find('#select-filter')
                    .select(value);
            })
    } else {
        cy.xpath(`${this.#aggridLocator}//*[@col-id="${this.#colId}"]`).invoke('attr', 'aria-colindex')
            .then(colIndex => {
                cy.xpath(this.#aggridLocator)
                    .find(`.ag-header-row-column-filter`)
                    .find(`.ag-header-cell[aria-colindex="${colIndex}"]`)
                    .find("input").clear({timeout: 500}).type(value, {timeout: 500}).type('{enter}', {timeout: 500});
            })
    }
    cy.wait(500);
}
```

รูปที่ 3.61 ฟังก์ชันการ Filter

คำสั่งในการ Sort ข้อมูลในตาราง

```
sortMethod = {
    asc: () => {
        let sortLocation = `${this.#aggridLocator}//div[@aria-rowindex="1"]//*[@col-id="${this.#colId}"]`;
        cy.wait(300)
        cy.xpath(sortLocation).find('.ag-sort-ascending-icon').as('ascendingIcon');
        cy.xpath(sortLocation).find('.ag-sort-none-icon').as('noneSortIcon');
        cy.then(function () {
            const isAscHidden = this.ascendingIcon.hasClass('ag-hidden')
            const isNoneSort = !this.noneSortIcon.hasClass('ag-hidden')
            if (isNoneSort) {
                cy.xpath(sortLocation, {timeout: 300}).click({timeout: 300});
            } else if (isAscHidden) {
                cy.xpath(sortLocation).click();
                cy.xpath(sortLocation).click();
            }
        })
    },
    desc: () => {
        let sortLocation = `${this.#aggridLocator}//div[@aria-rowindex="1"]//*[@col-id="${this.#colId}"]`;
        cy.wait(300)
        cy.xpath(sortLocation).find('.ag-sort-descending-icon').as('descendingIcon');
        cy.xpath(sortLocation).find('.ag-sort-none-icon').as('noneSortIcon');
        cy.then(function () {
            const isDescHidden = this.descendingIcon.hasClass('ag-hidden')
            const isNoneSort = !this.noneSortIcon.hasClass('ag-hidden')
            if (isNoneSort) {
                cy.xpath(sortLocation, {timeout: 300}).click({timeout: 300});
                cy.xpath(sortLocation, {timeout: 300}).click({timeout: 300});
            } else if (isDescHidden) {
                cy.xpath(sortLocation).click();
            }
        })
    }
}
```

รูปที่ 3.62 ฟังก์ชันการ Sort

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การดึงข้อมูลในตารางผ่าน Row และ Column

```
row (rowIndex) {
    let location = `${this.#aggridLocator}//*[@row-index="${rowIndex}"]//*[@col-id="${this.#colId}"]$
    cy.wait(200)
    return cy.xpath(location, {timeout: 200})
}

    get column () {
        cy.wait(500)
        return cy.xpath(`${this.#aggridLocator}//*[@col-id="${this.#colId}"]`)
    }
```

รูปที่ 3.63 ฟังก์ชันการ Get Row / Get Column

คำสั่งในการ Clear ข้อมูล Filter

```
filterMethod = {
    clear: () => {
        if (!this.#hasFilter) return;
        cy.wait(500)
        if (this.#isSelectOption) {
            cy.xpath(`${this.#aggridLocator}//*[@col-id="${this.#colId}"]`).invoke('attr', 'aria-colindex')
                .then(colIndex => {
                    cy.xpath(this.#aggridLocator)
                        .find(`.ag-header-row-column-filter`)
                        .find(`.ag-header-cell[aria-colindex="${colIndex}"]`)
                        .find('#select-filter')
                        .select("");
                })
        } else {
            cy.xpath(`${this.#aggridLocator}//*[@col-id="${this.#colId}"]`)
                .invoke('attr', 'aria-colindex')
                .then(colIndex => {
                    cy.xpath(this.#aggridLocator)
                        .find(`.ag-header-row-column-filter`)
                        .find(`.ag-header-cell[aria-colindex="${colIndex}"]`)
                        .find("input").clear({timeout: 200}).type('{enter}', {timeout: 200});
                })
        }
        cy.wait(500)
```

รูปที่ 3.64 ฟังก์ชันการ Clear

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการเขียนฟังก์ชันในการทดสอบตารางอีกรูปแบบหนึ่ง

```
filter(tableName, colName, text) {
    const path = `                    -${tableName}"]                    /input[@aria-label="${colName} Filter Inp
    cy.xpath(path).clear({ force: true })
    cy.xpath(path).type(text)
    cy.xpath(path).tab()
    cy.wait(1000)
}

clear(tableName, colName) {
    const path = `                    -${tableName}"]                    /input[@aria-label="${colName} Filter Inp
    cy.xpath(path).clear({ force: true })
    cy.wait(1000)
}

sort(tableName, colName) {
    const path = `                    -${tableName}"]                    /*[@col-id="${colName}"]`
    cy.xpath(path).click()
    cy.wait(1000)
}
```

รูปที่ 3.65 ฟังก์ชันการ Filter / Clear / Sort

ตัวอย่างที่ 1 การทดสอบ Automated Functional Test แบบ Look and Field (ตาราง)

Requirement : หน้าจอสำหรับSA ที่แสดงผล Error สำหรับรายการ Switchingข้าม บลจ.

โดยสามารถ Inquiry ได้จาก Payment Date (วันที่ทำรายขาขาย) ซึ่งจะแสดงเป็นตารางดังนี้

รูปที่ 3.66 หน้าจอแสดงข้อมูล Failed XWT Order

ซึ่งให้จัดเรียงรายการโดย Payment Date จากน้อยไปมาก และ XWT Ref No.จากน้อยไปมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.7 Test Script การทดสอบ Automated Functional Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 1

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result
1	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบการแสดงผลหน้าจอ Failed Switching Order	- แสดงช่องรับข้อมูล Payment Date From/To ได้ถูกต้อง
2	Web	SA	Order Management	Failed Switching Order	- กรอกข้อมูลเฉพาะ Payment Date From	- ไม่สามารถค้นหาข้อมูลได้
3	Web	SA	Order Management	Failed Switching Order	- กรอกข้อมูลเฉพาะ Payment Date To	- ไม่สามารถค้นหาข้อมูลได้
4	Web	SA	Order Management	Failed Switching Order	- แสดงหัวข้อตารางหน้าจอ Failed Switching Order	- แสดงหัวข้อตารางได้ถูกต้อง
5	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบข้อมูล Field : PaymentDate(RED) เมื่อ Gen รายการ Sub ไม่สำเร็จ	- แสดง Payment Date(RED) เป็น yyyymmdd ได้ถูกต้อง
6	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบข้อมูล Field : XWT Ref No. เมื่อ Gen รายการ Sub ไม่สำเร็จ	- แสดง XWT Ref No. เป็น xxxxxxxx ได้ถูกต้อง
7	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบข้อมูล Field : Account No เมื่อ Gen รายการ Sub ไม่	- แสดง Account เป็น Accountที่ทำรายการได้ถูกต้อง

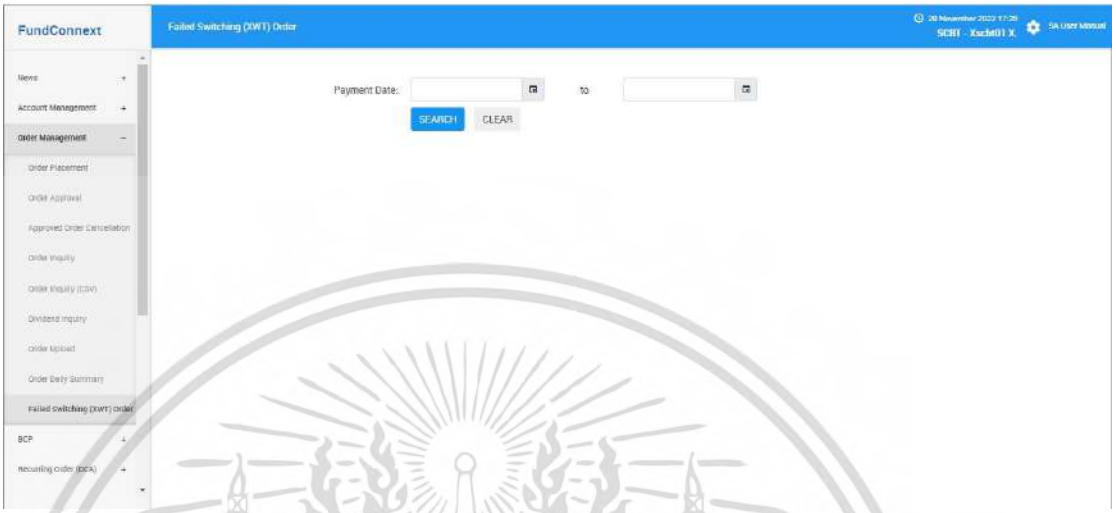
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

					สำเร็จ	
8	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบข้อมูล Field : Counter Unitholder ID เมื่อ Gen รายการ Sub ไม่สำเร็จ	- แสดง Counter Unitholder ID เป็น Unitholder ที่ทำรายการขาซื้อได้ถูกต้อง
9	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบข้อมูล Field : Customer Name เมื่อ Gen รายการ Sub ไม่สำเร็จ	- แสดง Customer Name เป็นรายชื่อลูกค้าที่ทำรายการได้ถูกต้อง
10	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบข้อมูล Field : Counter fund Code เมื่อ Gen รายการ Sub ไม่สำเร็จ	- แสดง Counter fund Code เป็น FundCodeที่ทำรายการขาซื้อได้ถูกต้อง
11	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบข้อมูล Field : Amount เมื่อ Gen รายการ Sub ไม่สำเร็จ	- แสดง Amount เป็น ยอดเงินของรายการได้ถูกต้อง
12	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบข้อมูล Default Sort	- แสดง Default Sort ดังนี้ Payment Date น้อยไปมาก XWT Ref No น้อยไปมาก
13	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบการ Filter และ Sort ข้อมูล	- สามารถ Filter และSort ข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทดสอบ

ทำรายการ Cross AMC Switching ทำรายการ Allot Gen Sub Order ไม่สำเร็จ เพื่อให้หน้าจอแสดงรายการ Cross AMC Switching ที่ไม่สำเร็จ

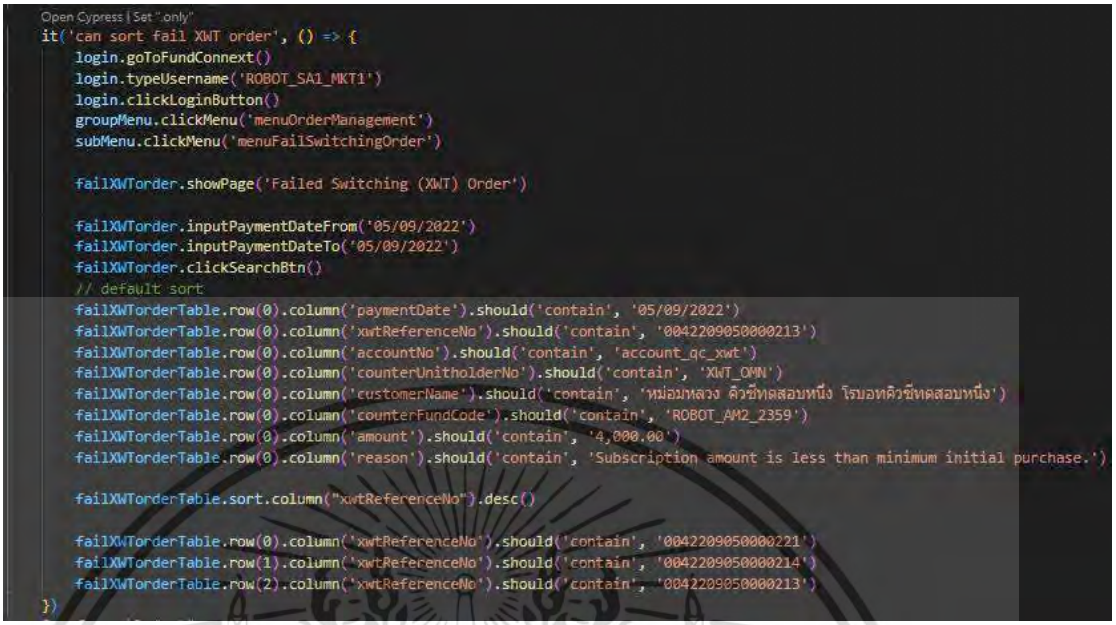


รูปที่ 3.67 หน้าจอเมนู Fail XWT Order (1)

รูปที่ 3.68 หน้าจอเมนู Fail XWT Order (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 1 ตรวจสอบหน้าจอ Failed XWT Order ทั้ง Default Sort และการ Sort



```
Open Cypress | Set ".only"
it('can sort fail XWT order', () => {
    login.goToFundConnext()
    login.typeUsername('ROBOT_SA1_MKT1')
    login.clickLoginButton()
    groupMenu.clickMenu('menuOrderManagement')
    subMenu.clickMenu('menuFailSwitchingOrder')

    failXWTorder.showPage('Failed Switching (XWT) Order')

    failXWTorder.inputPaymentDateFrom('05/09/2022')
    failXWTorder.inputPaymentDateTo('05/09/2022')
    failXWTorder.clickSearchBtn()
    // default sort
    failXWTorderTable.row(0).column('paymentDate').should('contain', '05/09/2022')
    failXWTorderTable.row(0).column('xwtReferenceNo').should('contain', '0042209050000213')
    failXWTorderTable.row(0).column('accountNo').should('contain', 'account_qc_xwt')
    failXWTorderTable.row(0).column('counterUnitholderNo').should('contain', 'XWT_OMN')
    failXWTorderTable.row(0).column('customerName').should('contain', 'หม่อมหลวง คิวซีทดสอบหนึ่ง โรบอทคิวซีทดสอบหนึ่ง')
    failXWTorderTable.row(0).column('counterFundCode').should('contain', 'ROBOT_AM2_2359')
    failXWTorderTable.row(0).column('amount').should('contain', '4,000.00')
    failXWTorderTable.row(0).column('reason').should('contain', 'Subscription amount is less than minimum initial purchase.')

    failXWTorderTable.sort.column("xwtReferenceNo").desc()

    failXWTorderTable.row(0).column('xwtReferenceNo').should('contain', '0042209050000221')
    failXWTorderTable.row(1).column('xwtReferenceNo').should('contain', '0042209050000214')
    failXWTorderTable.row(2).column('xwtReferenceNo').should('contain', '0042209050000213')
})
```

รูปที่ 3.69 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 1 กรณีที่ 1

กรณีทดสอบที่ 2 ตรวจสอบการ Filter หน้าจอ Fail XWT Order

```
Open Cypress | Set ".only"
it('can search and filter fail XWT order', () => {
    failXWTorderTable.filter.column('xwtReferenceNo').by('0042209050000214')
    failXWTorderTable.row(0).column('xwtReferenceNo').should('contain', '0042209050000214')

    failXWTorderTable.filter.clear('xwtReferenceNo')

    failXWTorderTable.filter.column('reason').by('Subscription amount is less than minimum initial purchase.')
    failXWTorderTable.row(0).column('reason').should('contain', 'Subscription amount is less than minimum initial purchase.')
    failXWTorderTable.row(1).column('reason').should('contain', 'Subscription amount is less than minimum initial purchase.')

    failXWTorderTable.filter.clear('reason')
})
```

รูปที่ 3.70 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 1 กรณีที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2 การทดสอบ Automated Functional Test แบบ Look and Field (ตาราง)

Requirement : หน้าจอ Account Balance Inquiry หน้าจอที่ใช้ในการกรอกข้อมูล Account Balance ให้แสดงออกมาในรูปแบบของตารางรวมถึงสามารถทำการ Download ออกมาในรูปแบบของ Excel ซึ่งจะมี Criteria ที่ใช้ในการ Inquiry แตกต่างกันออกไปแบบ Hierachry

ตารางที่ 3.8 Test Script การทดสอบ Automated Functional Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result
1	Web	SA	Account Management	Account Balance Inquiry	- ตรวจสอบหัวข้อตารางเมนู Account Balance Inquiry	- แสดงหัวข้อตารางได้ถูกต้อง
2	Web	SA	Account Management	Account Balance Inquiry	- ตรวจสอบ Criteria เมื่อ Login : HQ	- แสดง Criteria Branch Input Team Input IC Input Investor Input
3	Web	SA	Account Management	Account Balance Inquiry	- ตรวจสอบข้อมูลในตารางเมื่อ Login : HQ ไม่ Criteria	- แสดงข้อมูล Account Balanceทั้งหมดในSA ได้ถูกต้อง
4	Web	SA	Account Management	Account Balance Inquiry	- ตรวจสอบข้อมูลในตารางเมื่อ Login : HQ Criteria : Branch/Team/IC	- แสดงข้อมูล Account Balanceตาม Criteria ได้ถูกต้อง
5	Web	SA	Account Management	Account Balance Inquiry	- ตรวจสอบ Criteria เมื่อ Login : Branch	- แสดง Criteria Team Input IC Input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช

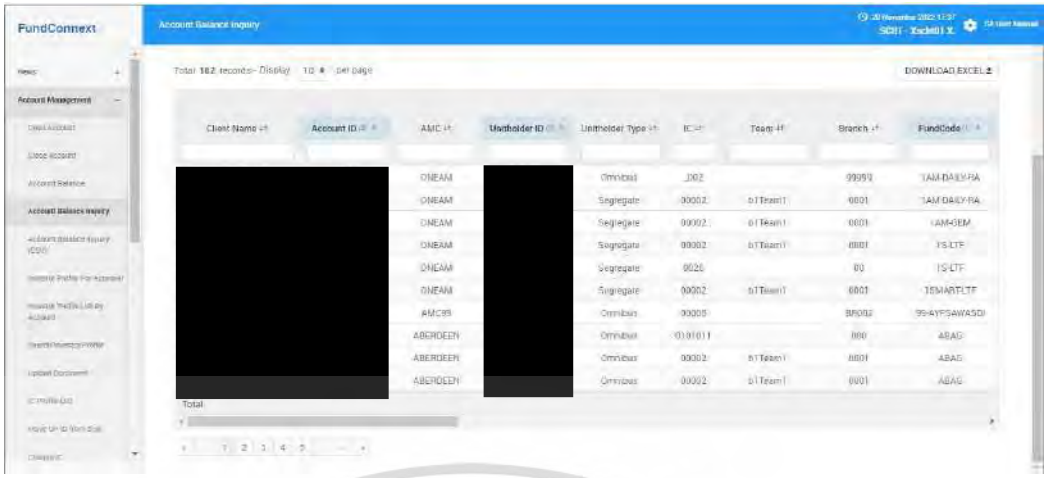
						Investor Input
6	Web	SA	Account Management	Account Balance Inquiry	- ตรวจสอบข้อมูลในตารางเมื่อ Login : Branch ไม่ Criteria	- แสดงข้อมูล Account Balanceทั้งหมดในBranch เท่านั้น
7	Web	SA	Account Management	Account Balance Inquiry	- ตรวจสอบ Criteria เมื่อ Login : Team	- แสดง Criteria IC Input Investor Input
8	Web	SA	Account Management	Account Balance Inquiry	- ตรวจสอบข้อมูลในตารางเมื่อ Login : Branch ไม่ Criteria	- แสดงข้อมูล Account Balanceทั้งหมดในTeam เท่านั้น
9	Web	SA	Account Management	Account Balance Inquiry	- ตรวจสอบ Criteria เมื่อ Login : Staff	- แสดง Criteria Investor Input
10	Web	SA	Account Management	Account Balance Inquiry	- ตรวจสอบ Criteria Fund Code	- แสดง Criteria FundCode ในทุกระดับของUser

ขั้นตอนการทดสอบ

เตรียมข้อมูล Account Balance โดยการ Upload File Unitholder Balance

รูปที่ 3.71 หน้าจอเมนู Balance Inquiry (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.72 หน้าจอเมนู Balance Inquiry (2)

กรณีทดสอบที่ 1 ตรวจสอบหัวข้อของตารางหน้าจอ Balance Inquiry

```
Open Cypress | Set ".only"
it('Account balance inquiry must show header label correctly', () => {
    balanceInquiry.clickSearchBtn()
    balanceInquiry.checkHeaderTableBalanceInquiry()
    balanceInquiry.clickClearBtn()
})
```

รูปที่ 3.73 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 1

กรณีทดสอบที่ 2 ตรวจสอบ Criteria เมื่อ Login ด้วย HQ

```
Open Cypress | Set ".only"
it('SA Hier HQ can access menu Account balance inquiry', () => {
    // Ref#697
    loginFundConnext.goToFundConnext()
    loginFundConnext.typeUsername('ROBOT_HIER')
    loginFundConnext.clickLoginButton()
    groupMenu.clickMenu('menuAccountManagement')
    subMenu.clickMenu('menuAccountBalanceInquiry')
    balanceInquiry.showPage('Account Balance Inquiry')
    balanceInquiry.showHQcriteria()
})
```

รูปที่ 3.74 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 3 ตรวจสอบข้อมูลในตาราง เมื่อ Login ด้วย HQ

```
Open Cypress | Set ".only"
it('SA Hier HQ can see all Account balance data in SA when search all criteria', () => {
    balanceInquiry.clickSearchBtn()
    // B1T1
    balanceInquiryTable.filter.column('branch').by('B001')
    balanceInquiryTable.filter.column('team').by('Team 1')
    balanceInquiryTable.row(0).column('clientName').should('contain', 'นายโรบอททีมหนึ่ง สาขาหนึ่ง')
    balanceInquiryTable.row(0).column('accountId').should('contain', '_ACCB1T1A1')
    balanceInquiryTable.row(0).column('ic').should('contain', 'ICB001T1')
    balanceInquiryTable.row(0).column('team').should('contain', 'Team 1')
    balanceInquiryTable.row(0).column('branch').should('contain', 'B001')
    // B1T2
    balanceInquiryTable.filter.column('branch').by('B001')
    balanceInquiryTable.filter.column('team').by('Team 2')
    balanceInquiryTable.row(0).column('clientName').should('contain', 'นายโรบอททีมสอง สาขาหนึ่ง')
    balanceInquiryTable.row(0).column('accountId').should('contain', '_ACCB1T2A1')
    balanceInquiryTable.row(0).column('ic').should('contain', 'ICB001T2')
    balanceInquiryTable.row(0).column('team').should('contain', 'Team 2')
    balanceInquiryTable.row(0).column('branch').should('contain', 'B001')
    // B2T1
    balanceInquiryTable.filter.column('branch').by('B002')
    balanceInquiryTable.filter.column('team').by('Team 1')
    balanceInquiryTable.row(0).column('clientName').should('contain', 'นายโรบอทสาขาสอง ทีมหนึ่ง')
    balanceInquiryTable.row(0).column('accountId').should('contain', '_ACCB2T1A1')
    balanceInquiryTable.row(0).column('ic').should('contain', 'ICB002T1')
    balanceInquiryTable.row(0).column('team').should('contain', 'Team 1')
    balanceInquiryTable.row(0).column('branch').should('contain', 'B002')
```

รูปที่ 3.75 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 3

กรณีทดสอบที่ 4 ตรวจสอบข้อมูลในตาราง เมื่อ Criteria ด้วยตัวเลือกต่างๆ

```
Open Cypress | Set ".only"
it('SA Hier HQ can search all criteria and show account balance correctly', () => {
    balanceInquiry.inputBranchCode('B001')
    balanceInquiry.clickSearchBtn()
    balanceInquiryTable.row(0).column('clientName').should('contain', 'นายโรบอททีมหนึ่ง สาขาหนึ่ง')
    balanceInquiryTable.row(0).column('ic').should('contain', 'ICB001T1')
    balanceInquiryTable.row(0).column('team').should('contain', 'Team 1')
    balanceInquiryTable.row(0).column('branch').should('contain', 'B001')

    balanceInquiry.inputTeam('Team 2')
    balanceInquiry.clickSearchBtn()
    balanceInquiryTable.row(0).column('clientName').should('contain', 'นายโรบอททีมสอง สาขาหนึ่ง')
    balanceInquiryTable.row(0).column('ic').should('contain', 'ICB001T2')
    balanceInquiryTable.row(0).column('team').should('contain', 'Team 2')
    balanceInquiryTable.row(0).column('branch').should('contain', 'B001')
    balanceInquiry.clickClearBtn()

    balanceInquiry.inputClientName('โรบอททีมหนึ่ง สาขาหนึ่ง')
    balanceInquiry.clickSearchBtn()
    balanceInquiryTable.row(0).column('clientName').should('contain', 'นายโรบอททีมหนึ่ง สาขาหนึ่ง')
    balanceInquiryTable.row(0).column('ic').should('contain', 'ICB001T1')
    balanceInquiryTable.row(0).column('team').should('contain', 'Team 1')
    balanceInquiryTable.row(0).column('branch').should('contain', 'B001')
    balanceInquiry.clickClearBtn()
})
```

รูปที่ 3.76 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 5 ตรวจสอบ Criteria เมื่อ Login ด้วย Branch

```
Open Cypress | Set ".only"
it('SA Hier Branch can search with team ic and client', () => {
    // Ref#699
    loginFundConnext.goToFundConnext()
    loginFundConnext.typeUsername('RB_SAHIER_BRAN2')
    loginFundConnext.clickLoginButton()
    groupMenu.clickMenu('menuAccountManagement')
    subMenu.clickMenu('menuAccountBalanceInquiry')
    balanceInquiry.showPage('Account Balance Inquiry')
    balanceInquiry.showBranchcriteria()
})
```

รูปที่ 3.77 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 5

กรณีทดสอบที่ 6 ตรวจสอบข้อมูลในตาราง เมื่อ Login ด้วย Branch

```
Open Cypress | Set ".only"
it('SA Hier Branch can see Only Account balance data in their branch when search all criteria', () => {
    balanceInquiry.clickSearchBtn()
    balanceInquiryTable.filter.column('branch').by('B001')
    cy.contains('No Data')

    balanceInquiryTable.filter.column('branch').by('B002')
    balanceInquiryTable.filter.column('team').by('Team 1')
    balanceInquiryTable.row(0).column('clientName').should('contain', 'นายโรบอทสาขาสอง ทีมหนึ่ง')
    balanceInquiryTable.row(0).column('accountId').should('contain', '_ACCB2T1A1')
    balanceInquiryTable.row(0).column('ic').should('contain', 'ICB002T1')
    balanceInquiryTable.row(0).column('team').should('contain', 'Team 1')
    balanceInquiryTable.row(0).column('branch').should('contain', 'B002')

    balanceInquiryTable.filter.column('branch').by('B002')
    balanceInquiryTable.filter.column('team').by('Team 2')
    balanceInquiryTable.row(0).column('clientName').should('contain', 'นายโรบอททีมหนึ่ง สาขาหนึ่ง')
    balanceInquiryTable.row(0).column('accountId').should('contain', '_ACCB2T2A1')
    balanceInquiryTable.row(0).column('ic').should('contain', 'ICB002T2')
    balanceInquiryTable.row(0).column('team').should('contain', 'Team 2')
    balanceInquiryTable.row(0).column('branch').should('contain', 'B002')
})
```

รูปที่ 3.78 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 7 ตรวจสอบ Criteria เมื่อ Login ด้วย Team

```
Open Cypress | Set ".only"
it('SA Hier Team can search with ic and client', () => {
    // Ref#700
    loginFundConnext.goToFundConnext()
    loginFundConnext.typeUsername('ROBOT_HIER_B1T1')
    loginFundConnext.clickLoginButton()
    groupMenu.clickMenu('menuAccountManagement')
    subMenu.clickMenu('menuAccountBalanceInquiry')
    balanceInquiry.showPage('Account Balance Inquiry')
    balanceInquiry.showTeamcriteria()
})
```

รูปที่ 3.79 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 7

กรณีทดสอบที่ 8 ตรวจสอบข้อมูลในตาราง เมื่อ Login ด้วย Team

```
Open Cypress | Set ".only"
it('SA Hier Team can see Only Account balance data in their Team when search all criteria', () => {
    balanceInquiry.clickSearchBtn()
    balanceInquiryTable.filter.column('branch').by('B001')
    balanceInquiryTable.filter.column('team').by('Team 2')
    cy.contains('No Data')

    balanceInquiryTable.filter.column('branch').by('B001')
    balanceInquiryTable.filter.column('team').by('Team 1')
    balanceInquiryTable.row(0).column('clientName').should('contain', 'นายโรบอททีมหนึ่ง สาขาหนึ่ง')
    balanceInquiryTable.row(0).column('accountId').should('contain', '_ACCB1T1A1')
    balanceInquiryTable.row(0).column('ic').should('contain', 'ICB001T1')
    balanceInquiryTable.row(0).column('team').should('contain', 'Team 1')
    balanceInquiryTable.row(0).column('branch').should('contain', 'B001')

    balanceInquiryTable.filter.column('branch').by('B002')
    balanceInquiryTable.filter.column('team').by('Team 1')
    cy.contains('No Data')
})
```

รูปที่ 3.80 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 8

กรณีทดสอบที่ 9 ตรวจสอบ Criteria เมื่อ Login ด้วย Team

```
Open Cypress | Set ".only"
it('SA Hier Staff can search with client', () => {
    // Ref#701
    loginFundConnext.goToFundConnext()
    loginFundConnext.typeUsername('_SaBranch2')
    loginFundConnext.clickLoginButton()
    groupMenu.clickMenu('menuAccountManagement')
    subMenu.clickMenu('menuAccountBalanceInquiry')
    balanceInquiry.showPage('Account Balance Inquiry')
    balanceInquiry.showStaffcriteria()
})
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.81 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 9

กรณีทดสอบที่ 10 ตรวจสอบ Criteria FundCode

```
Open Cypress | Set ".only"
it('SA Hier can search with fundCode', () => {
    loginFundConnext.goToFundConnext()
    loginFundConnext.typeUsername('ROBOT_HIER')
    loginFundConnext.clickLoginButton()
    groupMenu.clickMenu('menuAccountManagement')
    subMenu.clickMenu('menuAccountBalanceInquiry')
    balanceInquiry.showPage('Account Balance Inquiry')
    balanceInquiry.inputFundCode('ROBOT_AM1_2100')
    balanceInquiry.clickSearchBtn()
    balanceInquiryTable.row(0).column('clientName').should('contain', 'นายโรบอททีมหนึ่ง สาขาหนึ่ง')
    balanceInquiryTable.row(0).column('ic').should('contain', 'IC8001T1')
    balanceInquiryTable.row(0).column('team').should('contain', 'Team 1')
    balanceInquiryTable.row(0).column('branch').should('contain', 'B001')
})
```

รูปที่ 3.82 Script ในการทำ Automated Test look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 Automated Test แบบ Validate File

การทดสอบประเภท Functional Test แบบ Validate File เป็นการตรวจสอบผลลัพธ์ที่ได้เมื่อทำการUpload File และตรวจสอบชื่อไฟล์ ตรวจสอบข้อมูลภายในไฟล์ว่าถูกต้องหรือไม่ เมื่อ Download File จากระบบซึ่งในระบบฟันด์คอนเน็กต์มีไฟล์ 3ประเภทหลักๆคือ Txt File, CSV File, Excel File

การทำ Test Script และเตรียม Data Test แบบเป็น 2 กรณีหลัก คือ

1. กรณีที่ทำ Test Script และเตรียม Data Test สำหรับ Upload File มีการเตรียมชื่อไฟล์ ข้อมูลภายในไฟล์ หรือการเขียนไฟล์ขึ้นใหม่ เพื่อใช้ในการ Upload ทั้งกรณีที่ถูกต้องตามเงื่อนไข และผิดเงื่อนไขที่กำหนดไว้และดูผลลัพธ์ที่ได้จากการ Upload
2. กรณีที่ทำ Test Script และเตรียม Data Test สำหรับ Download File มีการเตรียมไฟล์ที่ถูกต้องเพื่อใช้สำหรับเปรียบเทียบกับไฟล์ที่ได้รับมาจากระบบ

การทำ Automated Test ในส่วนของการ Upload File จะเขียนในรูปแบบการตรวจสอบ Toast Messageที่แสดงผ่านหน้าจอรวมถึงบางครั้งก็ตรวจสอบข้อมูลที่จะต้องแสดงในระบบทั้งนี้การเตรียมไฟล์ในการ Upload จะมีรูปแบบแตกต่างกันออกไปในไฟล์แต่ละประเภท มีการเตรียมไฟล์ไว้เพื่อUpload การเขียนไฟล์ขึ้นมาใหม่ ส่วนในด้านของการ Downloadนั้นจะมีการเตรียมข้อมูลไฟล์ที่ถูกต้องเพื่อใช้ในการเปรียบเทียบเมื่อ Dowload File นั้นมาจากระบบ

โดยรูปแบบการจัดเตรียมไฟล์สำหรับการ Upload สามารถทำได้ 2 วิธีดังนี้

1. เตรียมไฟล์ไว้ก่อนทำการ Upload โดยใช้คำสั่ง attachFile ดังรูปที่ 3.83

```
uploadCustomerTxt(filePath) {
    console.log(filePath)
    cy.get('#customerTextInputFileInput').attachFile(filePath, { allowEmpty: true })
}

// Ref#24
uploadOmnibusProfile.uploadCustomerTxt(filepath + 'valid_omn_customer_profile_robotsa1_ver1.txt')
```

รูปที่ 3.83 ฟังก์ชันการ Upload แบบเตรียมไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เขียนไฟล์ขึ้นมาใหม่แล้วทำการ Upload เช่น ชื่อไฟล์ เนื้อหาในไฟล์

```
uploadOrder(content, fileName) {
    const filePath = `${fixtures}/upload_red_order_transaction/${fileName}.txt`

    cy.writeFile(filePath, content)
    cy.get('#order_transFileInput').attachFile(filePath)
}
// Ref#3503
const fileName = `${today}_ROBOTSA1_ORDERTRANSACTION5`
const fileContent = `${today}|ROBOTSA1|1|V2.0\n$1234567890|${today}100000|RBPAYTOINVESTOR|ROBOTAM
// KTB
uploadOrderTransactionsRed.uploadOrder(fileContent, fileName)
uploadOrderTransactionsRed.clickUpload('#order_transUploadButton')
```

รูปที่ 3.84 ฟังก์ชันการ Upload แบบเขียนไฟล์

การ Compare text File โดยใช้คำสั่ง readFile ในการอ่านข้อมูลของไฟล์ที่ถูกต้องกับไฟล์ที่ได้รับมาจากระบบ

```
// Compare File
Cypress.Commands.add('compareTextFile', (correctFile, actualFile) => {
    cy.readFile(`${correctFile}`).then(text => {
        // const fileContent = Fy(text)
        if (true) return cy.log(`expect ${correctFile} text size ${text.length} (will compare with ${actualFile})`).then(() => {
        return text
    }).as('correctFile')

    cy.readFile(`${actualFile}`).then(text => {
        // const fileContent2 = JSON.stringify(text)
        if (true) return cy.log(`actual ${actualFile} text size ${text.length}`).then(() => { return text })
        return text
    }).as('actualFile')

    cy.then(function () {
        cy.wrap(this.correctFile).should('eq', this.actualFile)
    })
})
```

```
cy.compareTextFile(expected, actual);
```

รูปที่ 3.85 ฟังก์ชันการ Compare text File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การ Compare CSV File โดยใช้คำสั่ง readFile ในการอ่านข้อมูลของไฟล์ที่ถูกต้องกับไฟล์ที่ได้รับมาจากระบบ

```
validateAccountBalanceInquiryCsv(fileName, correctFileName) {
    const downloadsFolder = Cypress.config('downloadsFolder');

    cy.wait(1000)
    cy.task('downloads', downloadsFolder).then((files) => {
        const downloadFileName = files.find(file => file.includes(fileName))

        var csvDownloaded = path.join(downloadsFolder, downloadFileName)
        var csvUploaded = `${integrationFolder}/account/acc_balance_inquiry_csv/expected/${correctFileName}`

        cy.readFile(csvUploaded, 'utf-8').then((uploadTxt) => {
            cy.readFile(csvDownloaded, 'utf-8', { timeout: 10000 }).then((downloadTxt) => {
            expect(downloadTxt).to.equal(uploadTxt);
            });
        });
    })
}
balanceInquiryCSV.validateAccountBalanceInquiryCsv(fileName, 'ACC_BALANCE_INQUIRY_HIER_HQ.csv')
```

รูปที่ 3.86 ฟังก์ชันการ Compare CSV File

การ Compare Excel File โดยใช้คำสั่ง task ในการเรียกใช้ plugin compareExcel

```
cy.task('compareExcel', {
    file1: expected,
    file2: actual
});
```

รูปที่ 3.87 ฟังก์ชันการ Compare Excel File

โดยการติดตั้ง Pluginต่างๆเพื่อนำมาใช้งาน เช่น ฟังก์ชัน compareExcel, ClearFile เป็นต้น สามารถเขียนได้ ดังรูปที่ 3.88

```
module.exports = (on, config) => {
    on('task', {
        readXlsx: readXlsx.read,
        compareExcel,
        compareExcelHeader,
        clearFileDownloads,
        clearFile: path => {
            return clearFile(path);
        },
        executeQuery: query => {
            return executeQuery(query);
        },
        downloads: (downloadspath) => {
            return fs.readdirSync(downloadspath)
        },
        zipFileAsBase64,
        unzipping: unzipping.unzip,
        unzipSimple: unzipping.unzipSimple,
        csvToJson
    });
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.88 หน้าจอ Plugin สำหรับ Cypress

ตัวอย่างที่ 1 การทดสอบ Automated Functional Test แบบ Validate File

Requirement : แก้ไขการ Upload Attachment ผ่านหน้าจอให้รองรับชื่อไฟล์ที่ยาวไม่เกิน 80 ตัวอักษรทั้งตัวอักษรภาษาไทย อังกฤษและอักขระพิเศษ โดยเมื่อความยาวเกินจะฟ้อง “Filename length should not exceed 80 Characters” รวมถึงเพิ่มข้อความบริเวณใต้กล่องUpload Attachement ต่างๆว่า “Filename length should not exceed 80 Characters”ในกรณีนี้จะต้องมีการทดสอบในบัญชีทุกประเภท เช่น Individual, Juristic, Joint, By Whom, For Whom และหน้าจอ Upload Document ด้วย

ตารางที่ 3.9 Test Script การทดสอบ Automated Functional Test แบบ Validate File ตัวอย่างที่ 1

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result
1	Web	SA	Open Account	Individual	- ตรวจสอบข้อความใต้กล่อง Upload Attachment ใน Tab4(Account Tab) Tab5(Personal Document)	- แสดงข้อความสีแดง “Filename length should not exceed 80 Characters” ทุกกล่อง
2	Web	SA	Open Account	Individual	- Upload Attachment Filename > 80 ทุกกล่องใน Account Tab	- ไม่สามารถ Upload ไฟล์ได้ถูกต้อง - แสดง Popup “Filename length should not exceed 80 Characters”ได้ถูกต้อง
3	Web	SA	Open Account	Individual	- Upload Attachment Filename <= 80 ทุกกล่องใน Account Tab	- สามารถ Upload ไฟล์ได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4	Web	SA	Open Account	Individual	- Upload Attachment Filename > 80 ทุกกล่องใน Personal Document Tab	- ไม่สามารถ Upload ไฟล์ได้ถูกต้อง - แสดง Popup “Filename length should not exceed 80 Characters”ได้ถูกต้อง
5	Web	SA	Open Account	Individual	- Upload Attachment Filename <= 80 ทุกกล่องใน Personal Document Tab	- สามารถ Upload ไฟล์ได้ถูกต้อง

ขั้นตอนการทดสอบ

เตรียมไฟล์สำหรับ Upload attachment ทั้งหมด 6 ไฟล์ คือ

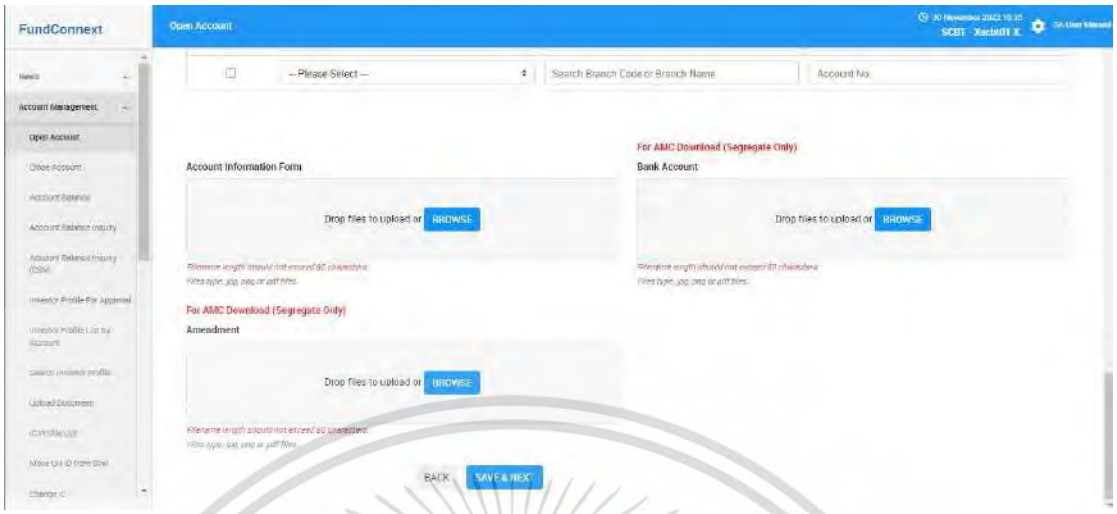
- File Thai name length มากกว่า 80 Char
- File Eng name length มากกว่า 80 Char
- File Mix name length มากกว่า 80 Char
- File Thai name length น้อยกว่า 80 Char
- File Eng name length น้อยกว่า 80 Char
- File Mix name length น้อยกว่า 80 Char

ดังรูปที่ 3.89 ที่จะประกอบไปด้วย 3 ไฟล์ ที่มีสกุลไฟล์แตกต่างกัน

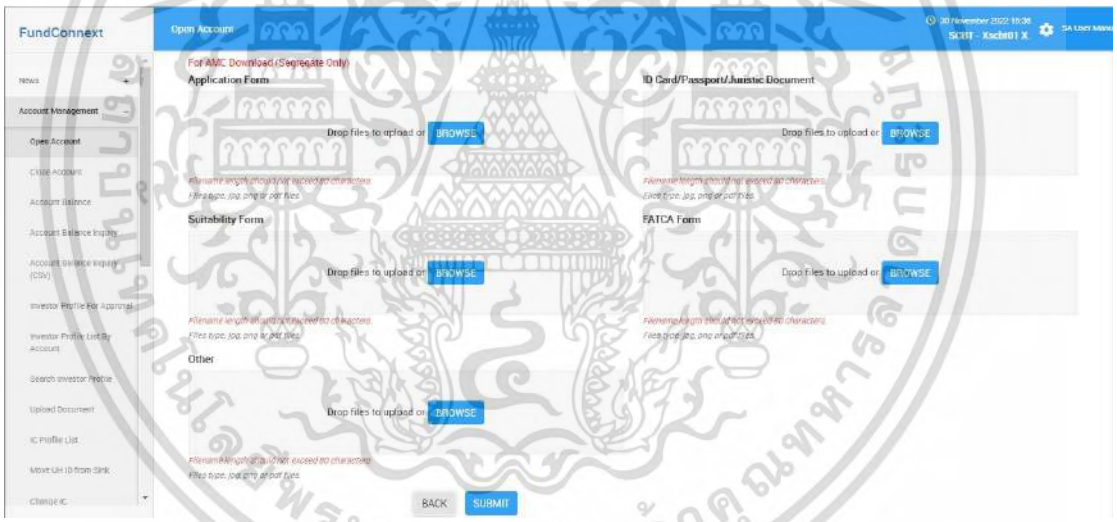
รูปที่ 3.89 ตัวอย่างไฟล์ทดสอบ Upload more than 80 Char

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างหน้าจอเมนู Upload Attachment



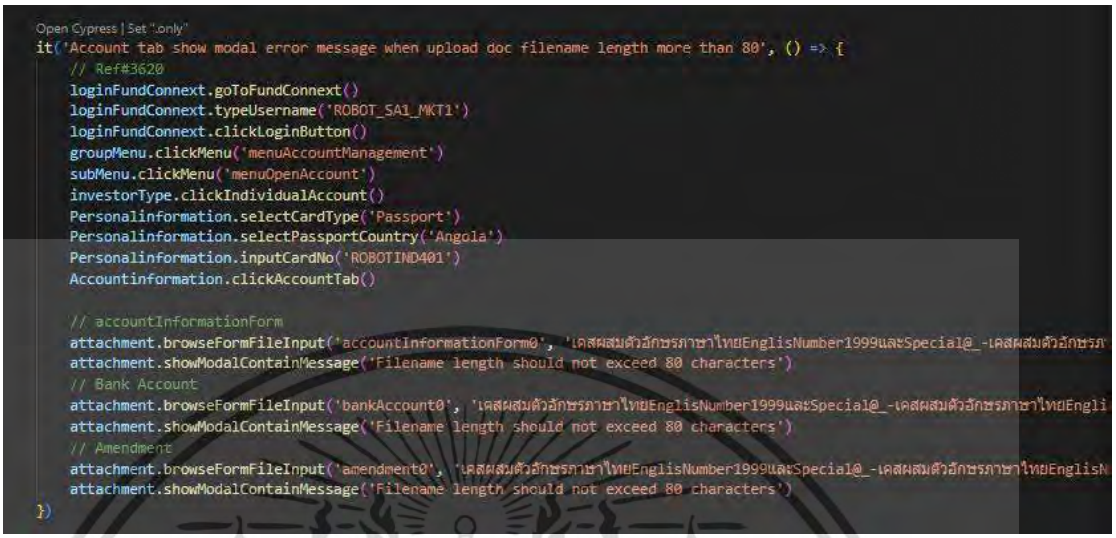
รูปที่ 3.90 หน้าจอเมนู Upload Attachment (1)



รูปที่ 3.91 หน้าจอเมนู Upload Attachment (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 1 ตรวจสอบModal เมื่อ Upload Attachment ชื่อมากกว่า 80 ตัวอักษรใน Account Tab



```
Open Cypress | Set "only"
it('Account tab show modal error message when upload doc filename length more than 80', () => {
    // Ref#3620
    loginFundConnext.goToFundConnext()
    loginFundConnext.typeUsername('ROBOT_SA1_MKT1')
    loginFundConnext.clickLoginButton()
    groupMenu.clickMenu('menuAccountManagement')
    subMenu.clickMenu('menuOpenAccount')
    investorType.clickIndividualAccount()
    Personalinformation.selectCardType('Passport')
    Personalinformation.selectPassportCountry('Angola')
    Personalinformation.inputCardNo('ROBOTIND401')
    Accountinformation.clickAccountTab()

    // accountInformationForm
    attachment.browseFormFileInput('accountInformationForm0', 'เคสผสมตัวอักษรภาษาไทยEnglisNumber1999และSpecial@_-เคสผสมตัวอักษรภ
    attachment.showModalContainMessage('Filename length should not exceed 80 characters')
    // Bank Account
    attachment.browseFormFileInput('bankAccount0', 'เคสผสมตัวอักษรภาษาไทยEnglisNumber1999และSpecial@_-เคสผสมตัวอักษรภาษาไทยEngli
    attachment.showModalContainMessage('Filename length should not exceed 80 characters')
    // Amendment
    attachment.browseFormFileInput('amendment0', 'เคสผสมตัวอักษรภาษาไทยEnglisNumber1999และSpecial@_-เคสผสมตัวอักษรภาษาไทยEnglisN
    attachment.showModalContainMessage('Filename length should not exceed 80 characters')
})
```

รูปที่ 3.92 Script ในการทำ Automated Validate File ตัวอย่างที่ 1 กรณีที่ 1

กรณีทดสอบที่ 2 ตรวจสอบModal เมื่อ Upload Attachment ชื่อมากกว่า 80 ตัวอักษรใน PersonalDocumentTab

```
Open Cypress | Set "only"
it('PersonalDoc tab show modal error message when upload doc filename length more than 80', () => {
    // Ref#3621
    PersonalDocumentinformation.clickPersonalDocumentTab()

    // Application Form
    attachment.browseFormFileInput('applicationForm', 'เคสผสมตัวอักษรภาษาไทยEnglisNumber1999และSpecial@_-เคสผสมตัวอักษรภาษาไทยEn
    attachment.showModalContainMessage('Filename length should not exceed 80 characters')
    // ID Card/Passport/Juristic Document
    attachment.browseFormFileInput('idCardDocument', 'เคสผสมตัวอักษรภาษาไทยEnglisNumber1999และSpecial@_-เคสผสมตัวอักษรภาษาไทยEng
    attachment.showModalContainMessage('Filename length should not exceed 80 characters')
    // Suitability Form
    attachment.browseFormFileInput('suitabilityForm', 'เคสผสมตัวอักษรภาษาไทยEnglisNumber1999และSpecial@_-เคสผสมตัวอักษรภาษาไทยEn
    attachment.showModalContainMessage('Filename length should not exceed 80 characters')
    // FATCA Form
    attachment.browseFormFileInput('fatcaForm', 'เคสผสมตัวอักษรภาษาไทยEnglisNumber1999และSpecial@_-เคสผสมตัวอักษรภาษาไทยEnglisNu
    attachment.showModalContainMessage('Filename length should not exceed 80 characters')
    // Other
    attachment.browseFormFileInput('others', 'เคสผสมตัวอักษรภาษาไทยEnglisNumber1999และSpecial@_-เคสผสมตัวอักษรภาษาไทยEnglisNumbe
    attachment.showModalContainMessage('Filename length should not exceed 80 characters')
})
```

รูปที่ 3.93 Script ในการทำ Automated Validate File ตัวอย่างที่ 1 กรณีที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2 การทดสอบ Automated Functional Test แบบ Validate File

Requirement : เมนู Account Balance Inquiry (CSV) ใช้ในการดึงข้อมูล Account Balance ออกมาในรูปแบบของไฟล์ CSV โดยระบบจะให้ Download เฉพาะวันปัจจุบันเมื่อกดปุ่ม Generate CSV จะทำการ Generate link ขึ้นมาเพื่อใช้ในการDownload

ตารางที่ 3.10 Test Script การทดสอบ Automated Functional Test แบบ Validate File ตัวอย่างที่ 2

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result
1	Web	SA	Account Management	Balance Inquiry CSV	- ตรวจสอบไฟล์ที่ได้รับเมื่อ Generate CSV โดนไม่ใส่ Criteria	- แสดงไฟล์ที่ได้รับมาถูกต้อง
2	Web	SA	Account Management	Balance Inquiry CSV	- ตรวจสอบไฟล์ที่ได้รับเมื่อ Generate CSV โดน Criteria : Branch	- แสดงไฟล์ที่ได้รับมาถูกต้อง
3	Web	SA	Account Management	Balance Inquiry CSV	- ตรวจสอบไฟล์ที่ได้รับเมื่อ Generate CSV โดน Criteria : Investment Consultant	- แสดงไฟล์ที่ได้รับมาถูกต้อง
4	Web	SA	Account Management	Balance Inquiry CSV	- ตรวจสอบไฟล์ที่ได้รับเมื่อ Generate CSV โดน Criteria : Client Name	- แสดงไฟล์ที่ได้รับมาถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

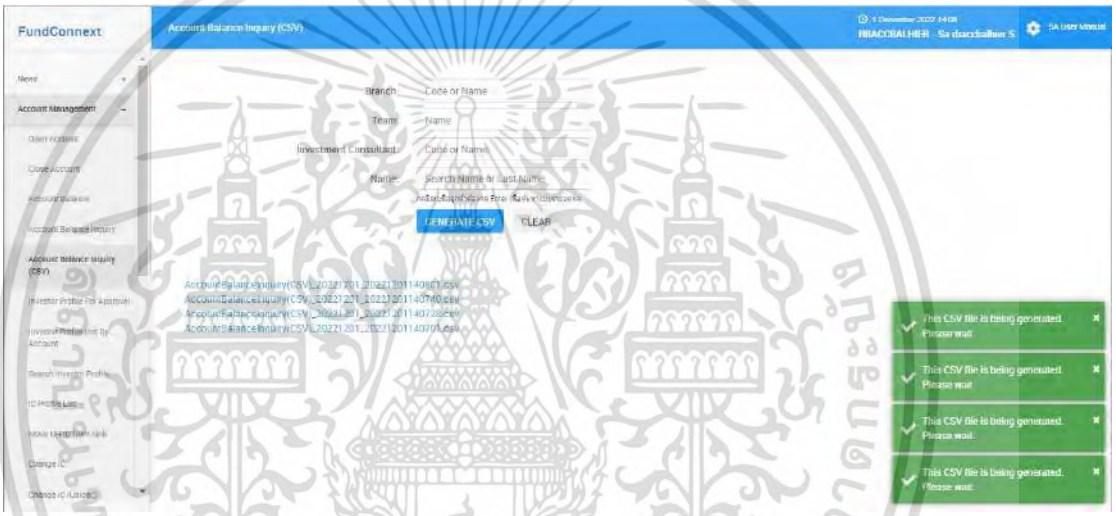
ขั้นตอนการทดสอบ

เตรียมไฟล์ Account Balance inquiry (CSV) ที่มีข้อมูลถูกต้องเพื่อนำไปเปรียบเทียบ

```
v_pongsiris, 2 months ago | 1 author (v_pongsiris)
Account ID,AMC,Unitholder ID,Unitholder Type,IC,Team,Branch,Fund Code,Available Balance Unit,Available Balance Amount,Outstanding
Barabeer,RBAMFORACCBAL,DMYW8UWZ2F1R3,Segregate,00001,,BKK,RB_TEST_ACC_BAL,10000.1234,1000.12,10000.123400,1000.12,0.000000,0.00,0.
Barabeer,RBAMFORACCBAL,DMYW8UWZ2F1R3,Segregate,00001,,BKK,RB_TEST_ACC_BAL2,50000,5000,50000.000000,5000.00,0.000000,0.00,0.000000,
RBACCBAL-0001,RBAMFORACCBAL,DMYW8UWY17IV8,Omnibus,00001,,BKK,RB_TEST_ACC_BAL,10000.1234,1000.12,10000.123400,1000.12,0.000000,0.0
RBACCBAL-0001,RBAMFORACCBAL,DMYW8UWY17IV8,Omnibus,00001,,BKK,RB_TEST_ACC_BAL2,50000,5000,50000.000000,5000.00,0.000000,0.00,0.0000
RBACCBAL-0001,RBAMFORACCBAL,DMYW8UWZ2F1R1,Segregate,00001,,BKK,RB_TEST_ACC_BAL,10000.1234,1000.12,10000.123400,1000.12,0.000000,0.
RBACCBAL-0001,RBAMFORACCBAL,DMYW8UWZ2F1R1,Segregate,00001,,BKK,RB_TEST_ACC_BAL2,50000,5000,50000.000000,5000.00,0.000000,0.00,0.0
```

รูปที่ 3.94 ตัวอย่างไฟล์ทดสอบ Compare Balance Inquiry (CSV)

ตัวอย่างหน้าจอเมนู Balance Inquiry (CSV)



รูปที่ 3.95 หน้าจอเมนู Balance Inquiry (CSV)

คำสั่งภายใน Before สำหรับเตรียมการ Upload Balance รวมถึง Upload Log Balance ก่อนการ Generate Balance inquiry CSV

```
    loginFundConnext.goToFundConnext()
    loginFundConnext.typeUsername('RBAMFORACCBAL')
    loginFundConnext.clickLoginButton()
    groupMenu.clickMenu('menuUpload')
    subMenu.clickMenu('menuUploadEndOfDay')
    // Test Environment : UNITHOLDER_BALANCE_UPLOAD_CUTOFF_TIME = 17:35 , Prod : 15:35
    if (`${HH}.${MM}` < 17.35) {
        insertFilelogBefore1735()
        amcUhBalance.uploadUnitholderBalanceFileTxt(content, 'RBAMFORACCBAL_UNITHOLDERBALANCE', previousDate)
    } else {
        insertFilelogAfter1735()
        amcUhBalance.uploadUnitholderBalanceFileTxt(content, 'RBAMFORACCBAL_UNITHOLDERBALANCE', todayDate)
    }
    amcUhBalance.clickUpload('#unitholderBalanceUploadButton')
    amcUhBalance.toastSuccess('uploaded successfully.')
    amcUhBalance.clickSuccessToast()
})
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้นำไปใช้ในการศึกษาต่อ ไม่อนุญาตให้นำไปใช้

รูปที่ 3.96 Script ในการทำ Automated Validate File ตัวอย่างที่ 2 เตรียมการ

```
        const content = `|RBAMFORACCBAL|16
RBACCBALNORMAL||DMYW8UWY17IV8|RB_TEST_ACC_BAL|10000.1234|1000.12|10000.1234|1000.12|0.0000|0.00|0.0000|10.3851|13.1859|20211004
RBACCBALNORMAL||DMYW8UWY17IV8|RB_TEST_ACC_BAL2|50000|5000|50000|5000|0.0000|0.00|0.0000|10.3851|10.1111|20211004
RBACCBALNORMAL||DMYW8UWZ2F1R1|RB_TEST_ACC_BAL|10000.1234|1000.12|10000.1234|1000.12|0.0000|0.00|0.0000|10.3851|13.1859|20211004
RBACCBALNORMAL||DMYW8UWZ2F1R1|RB_TEST_ACC_BAL2|50000|5000|50000|5000|0.0000|0.00|0.0000|10.3851|10.1111|20211004
RBACCBALNORMAL||DMYW8UWZ2F1R3|RB_TEST_ACC_BAL|10000.1234|1000.12|10000.1234|1000.12|0.0000|0.00|0.0000|10.3851|13.1859|20211004
RBACCBALNORMAL||DMYW8UWZ2F1R3|RB_TEST_ACC_BAL2|50000|5000|50000|5000|0.0000|0.00|0.0000|10.3851|10.1111|20211004
RBACCBALHIER||DMYW8UWZ2F1RD|RB_TEST_ACC_BAL|10000.1234|1000.12|10000.1234|1000.12|0.0000|0.00|0.0000|10.3851|13.1859|20211004
RBACCBALHIER||DMYW8UWZ2F1RD|RB_TEST_ACC_BAL2|50000|5000|50000|5000|0.0000|0.00|0.0000|10.3851|10.1111|20211004
RBACCBALHIER||DMYW8UWZ2F1RF|RB_TEST_ACC_BAL|10000.1234|1000.12|10000.1234|1000.12|0.0000|0.00|0.0000|10.3851|13.1859|20211004
RBACCBALHIER||DMYW8UWZ2F1RF|RB_TEST_ACC_BAL2|50000|5000|50000|5000|0.0000|0.00|0.0000|10.3851|10.1111|20211004
RBACCBALHIER||DMYW8UWZ2F1RN|RB_TEST_ACC_BAL|10000.1234|1000.12|10000.1234|1000.12|0.0000|0.00|0.0000|10.3851|13.1859|20211004
RBACCBALHIER||DMYW8UWZ2F1RN|RB_TEST_ACC_BAL2|50000|5000|50000|5000|0.0000|0.00|0.0000|10.3851|10.1111|20211004
RBACCBALHIER||DMYW8UWZ2F1R9|RB_TEST_ACC_BAL|10000.1234|1000.12|10000.1234|1000.12|0.0000|0.00|0.0000|10.3851|13.1859|20211004
RBACCBALHIER||DMYW8UWZ2F1R9|RB_TEST_ACC_BAL2|50000|5000|50000|5000|0.0000|0.00|0.0000|10.3851|10.1111|20211004
RBACCBALHIER||DMYW8UWZ2F1RP|RB_TEST_ACC_BAL|10000.1234|1000.12|10000.1234|1000.12|0.0000|0.00|0.0000|10.3851|13.1859|20211004
RBACCBALHIER||DMYW8UWZ2F1RP|RB_TEST_ACC_BAL2|50000|5000|50000|5000|0.0000|0.00|0.0000|10.3851|10.1111|20211004`
```

รูปที่ 3.97 Script ในการทำ Automated Validate File ตัวอย่างที่ 2 รายละเอียดไฟล์ Balance

```
function insertFilelogBefore1735() {
    cy.task('executeQuery', `INSERT INTO amc_upload_file_log
    (version, amc_id, business_date, description, end_upload, file_type, start_upload, status, upload_file_name, username,
        remote_addr, server_addr)
    VALUES (0, 10148, '${year}-${month}-${previousDay}', 'SUCCESS', '${year}-${month}-${previousDay} ${HH}:${MM}:${SS}',
    'ALLOTTEDTRANSACTIONS', '${year}-${month}-${previousDay} ${HH}:${MM}:${SS}',
    'COMPLETE', 'RBAMFORACCBAL', 'RBAMFORACCBAL', null, null)`)
}
```

รูปที่ 3.98 Script ในการทำ Automated Validate File ตัวอย่างที่ 2 การ insert log upload

```
validateAccountBalanceInquiryCsv(fileName, correctFileName) {
    const downloadsFolder = Cypress.config('downloadsFolder');

    cy.wait(1000)
    cy.task('downloads', downloadsFolder).then((files) => {
        const downloadFileName = files.find(file => file.includes(fileName))

        var csvDownloaded = path.join(downloadsFolder, downloadFileName)
        var csvUploaded = `${integrationFolder}/account/acc_balance_inquiry_csv/expected/${correctFileName}`

        cy.readFile(csvUploaded, 'utf-8').then((uploadTxt) => {
            cy.readFile(csvDownloaded, 'utf-8', { timeout: 10000 }).then((downloadTxt) => {
            expect(downloadTxt).to.equal(uploadTxt);
            });
        });
    })
}
```

รูปที่ 3.99 Script ในการทำ Automated Validate File ตัวอย่างที่ 2 ตรวจสอบไฟล์ CSV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 1 ตรวจสอบไฟล์ที่ได้รับเมื่อไม่ Criteria

```
Open Cypress | Set ".only"
it('SA can generate csv without input criteria', () => {
    // Ref#3107
    balanceInquiryCSV.showHQcriteria()
    balanceInquiryCSV.clickGenerateBtn()

    const todayDateHour = DateTime.local().toFormat("yyyyMMddHHmm")
    const fileName = `AccountBalanceInquiry(CSV)_${todayDate}_${todayDateHour}`;

    balanceInquiryCSV.toastLatestSuccess('This CSV file is being generated.  Please wait.');
    // wait for CSV generate success
    cy.wait(12000)
    cy.reload();

    balanceInquiryCSV.clickDownloadCsv(fileName)
    balanceInquiryCSV.validateAccountBalanceInquiryCsv(fileName, 'ACC_BALANCE_INQUIRY_CORRECT.csv')
})
```

รูปที่ 3.100 Script ในการทำ Automated Validate File ตัวอย่างที่ 2 กรณีที่ 1

กรณีทดสอบที่ 2 ตรวจสอบไฟล์ที่ได้รับเมื่อ Criteria ด้วย Branch

```
Open Cypress | Set ".only"
it('SA can generate csv with Branch', () => {
    // Ref#3107
    balanceInquiryCSV.clickClearBtn()
    balanceInquiryCSV.inputBranchCode('BKK')
    balanceInquiryCSV.clickGenerateBtn()

    const todayDateHour = DateTime.local().toFormat("yyyyMMddHHmm")
    const fileName = `AccountBalanceInquiry(CSV)_${todayDate}_${todayDateHour}`;

    balanceInquiryCSV.toastLatestSuccess('This CSV file is being generated.  Please wait.');
    // wait for CSV generate success
    cy.wait(12000)
    cy.reload();

    balanceInquiryCSV.clickDownloadCsv(fileName)
    balanceInquiryCSV.validateAccountBalanceInquiryCsv(fileName, 'ACC_BALANCE_INQUIRY_BY_BRANCH.csv')
})
```

รูปที่ 3.101 Script ในการทำ Automated Validate File ตัวอย่างที่ 2 กรณีที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 3 ตรวจสอบไฟล์ที่ได้รับเมื่อ Criteria ด้วย IC

```
Open Cypress | Set "only"
it('SA can generate csv with Investment Consultant', () => {
    // Ref#3107
    balanceInquiryCSV.clickClearBtn()
    balanceInquiryCSV.inputIC('000')
    balanceInquiryCSV.clickGenerateBtn()

    const todayDateHour = DateTime.local().toFormat("yyyyMMddHHmm")
    const fileName = `AccountBalanceInquiry(CSV)_${todayDate}_${todayDateHour}`;

    balanceInquiryCSV.toastLatestSuccess('This CSV file is being generated.  Please wait.');
    // wait for CSV generate success
    cy.wait(12000)
    cy.reload();

    balanceInquiryCSV.clickDownloadCsv(fileName)
    balanceInquiryCSV.validateAccountBalanceInquiryCsv(fileName, 'ACC_BALANCE_INQUIRY_BY_IC.csv')
})
```

รูปที่ 3.102 Script ในการทำ Automated Validate File ตัวอย่างที่ 2 กรณีที่ 3

กรณีทดสอบที่ 4 ตรวจสอบไฟล์ที่ได้รับเมื่อ Criteria ด้วย Client Name

```
Open Cypress | Set "only"
it('SA can generate csv with Client Name', () => {
    // Ref#3107
    balanceInquiryCSV.clickClearBtn()
    balanceInquiryCSV.inputClientName('TEST')
    balanceInquiryCSV.clickGenerateBtn()

    const todayDateHour = DateTime.local().toFormat("yyyyMMddHHmm")
    const fileName = `AccountBalanceInquiry(CSV)_${todayDate}_${todayDateHour}`;

    balanceInquiryCSV.toastLatestSuccess('This CSV file is being generated.  Please wait.');
    // wait for CSV generate success
    cy.wait(12000)
    cy.reload();

    balanceInquiryCSV.clickDownloadCsv(fileName)
    balanceInquiryCSV.validateAccountBalanceInquiryCsv(fileName, 'ACC_BALANCE_INQUIRY_BY_CLIENT.csv')
})
```

รูปที่ 3.103 Script ในการทำ Automated Validate File ตัวอย่างที่ 2 กรณีที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 Automated Test API

การทดสอบประเภท API จะเป็นการทดสอบเกี่ยวกับผลลัพธ์ที่กลับมาหลังจากการ Request API ทั้งข้อมูลตอบกลับที่ส่งข้อมูลไปถูกต้อง และข้อมูลตอบกลับที่ส่งข้อมูลที่ผิดพลาด

การทำ Test Script และเตรียม Data Test แบบเป็น 2 กรณีหลัก คือ

1. กรณีที่ทำ Test Script และเตรียม Data Test ทำรายการที่ต้องการทดสอบที่ถูกต้อง จะแสดงข้อมูลตอบกลับมาที่ถูกต้อง
2. กรณีที่ทำ Test Script และเตรียม Data Test ทำรายการที่ต้องการทดสอบที่ไม่ถูกต้อง จะไม่แสดงข้อมูลตอบกลับมาและจะมีข้อความการผิดพลาดตอบกลับ

การทำ Automated Test จะมีการเขียนฟังก์ชัน เพื่อใช้ในการ Request API ซึ่งจะมีรูปแบบกันแตกต่างออกไปในแต่ละ API ทั้งการส่งแบบ Parameter , Body เป็นต้น แล้วนำผลตอบรับที่ได้มาเปรียบเทียบกับผลลัพธ์ที่ต้องการเพื่อทำการทดสอบ

ตัวอย่างที่ 1 การทดสอบ Automated Test API

Requirement : เพิ่ม API สำหรับแสดง Log Error สำหรับรายการ Switching ข้ามบลจ. โดยสามารถระบุ Account No, Payment Date To/From, XWT Reference No โดยหากไม่ระบุ Payment Date จะแสดงข้อมูลรายการย้อนหลังแค่ 1ปี

ตารางที่ 3.11 Test Script การทดสอบ Automated Test API ตัวอย่างที่ 1

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result
1	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อไม่ใส่ token	- ไม่สามารถทำรายการได้
2	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อใส่ Account No	- แสดงรายการที่ Gen Sub ไม่สำเร็จของ account No นั้น
3	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อใส่ Payment	- แสดงรายการที่ Gen Sub ไม่สำเร็จของ Payment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

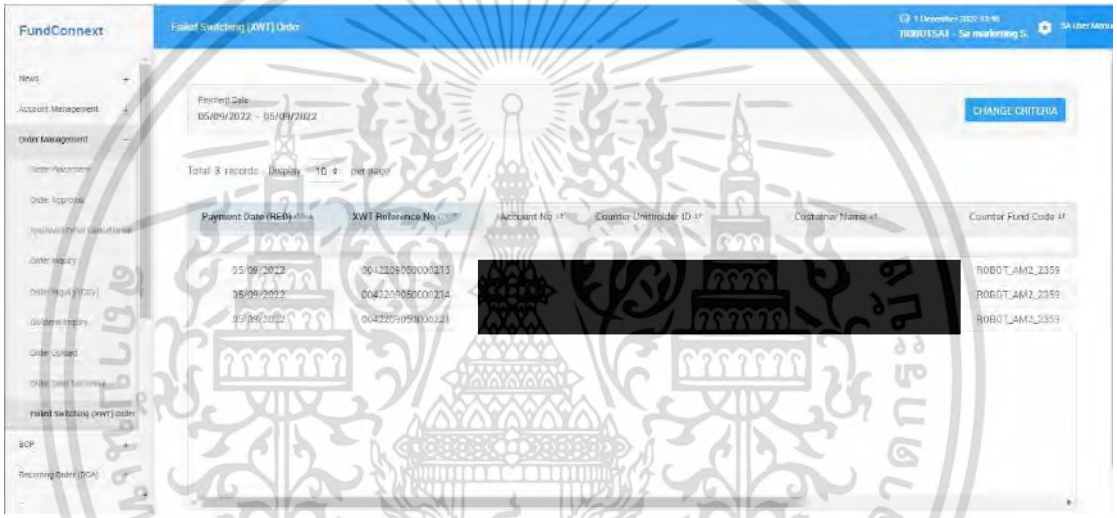
					Date from/to	Date นั้น
4	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อใส่ XWT Ref No.	- แสดงรายการที่ Gen Sub ไม่สำเร็จของ XWT Ref No. นั้น
5	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อใส่ Account No และ Payment Date from/to	- แสดงรายการที่ Gen Sub ไม่สำเร็จของAccount No และPayment Date ที่ระบุนั้น
6	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อใส่ Account No และ XWT Ref No	- แสดงรายการที่ Gen Sub ไม่สำเร็จของAccount No และXWT Ref Noที่ระบุนั้น
7	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อใส่ Payment Date to/from และ XWT Ref No	- แสดงรายการที่ Gen Sub ไม่สำเร็จของ Payment Date และXWT Ref Noที่ระบุนั้น
8	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อใส่ Account No, Payment Date to/from และ XWT Ref No	- แสดงรายการที่ Gen Sub ไม่สำเร็จของ Account No, Payment Date และXWT Ref Noที่ระบุนั้น
9	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อCriteria รายการที่ไม่มีในระบบ	- ไม่แสดงรายการได้ถูกต้อง
10	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อใส่ Payment Date to	- ไม่แสดงรายการได้ถูกต้อง - แสดง Error payment Date from is required

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

						field
11	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อใส่ Payment Date from	- ไม่แสดงรายการได้ถูกต้อง - แสดง Error payment Date to is required field

ขั้นตอนการทดสอบ

เตรียมข้อมูล Fail XWT Order



รูปที่ 3.104 ข้อมูล Fail XWT Order ที่เตรียมก่อนทดสอบ API

ตัวอย่าง API get Failed XWT Order ในเครื่องมือ Postman

รูปที่ 3.105 ตัวอย่าง API get Failed XWT Order

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน การเรียกใช้ API get Failed XWT Order เพื่อนำ response ที่ได้รับมาตรวจสอบ

```
const callApiGetfailXWTorder = (criteria) => {
    const options = criteria ? '?' : ''
    let params = Object.entries(criteria).map(([key, val]) => `${key}=${val}`).join('&')
    let request = {
        method: 'GET',
        url: `/api/crossAmcSwitching/fail${options}${params}`,
        failOnStatusCode: false
    }
    cy.request(request).as('response')
}
```

รูปที่ 3.106 Script ในการทำ Automated Test API ตัวอย่างที่ 1 เรียกใช้ API

คำสั่งภายใต้ before ทำการส่ง API Login ก่อนเรียกใช้ API get Failed XWT Order

```
beforeEach(() => {
    cy.apiLogin('ROBOT_SA1_API1')
})
```

รูปที่ 3.107 Script ในการทำ Automated Test API ตัวอย่างที่ 1 เตรียมการ

กรณีทดสอบที่ 1 เรียกใช้ API get Failed XWT Order โดยไม่Login

```
Open Cypress | Set ".only"
it('Cannot get fail XWT order without login', () => {
    // Ref # 3597
    cy.clearTokenApi()
    callApiGetfailXWTorder({})
    cy.get('@response').should((res) => {
        expect(res.status).equal(401)
        expect(res.body.errMsg).to.deep.equal({
            "code": "E000",
            "message": "Unauthorized access"
        })
    })
});
```

รูปที่ 3.108 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 2 เรียกใช้ API get Failed XWT Order โดยค้นหาด้วย Account No

```
it('Can list fail XWT order by account No', () => {
    // Ref # 3598
    callApiGetfailXWTorder({ accountNo: 'account_qc_xwt'})
    cy.get('@response').should((res) => {
        expect(res.status).equal(200)
        expect(res.body).to.deep.equal([
            {
                "paymentDate": "20220905",
                "xwtReferenceNo": "0042209050000213",
                "accountNo": "account_qc_xwt",
                "counterUnitholderNo": "XWT_OMN",
                "counterFundCode": "ROBOT_AM2_2359",
                "amount": "4000.00",
                "reason": "Subscription amount is less than minimum initial purchase.",
                "customerName": "หม่อมหลวง คิวบีทดสอบหนึ่ง โรบอทคิวบีทดสอบหนึ่ง"
            },
            {
                "paymentDate": "20220905",
                "xwtReferenceNo": "0042209050000214",
                "accountNo": "account_qc_xwt",
                "counterUnitholderNo": "XWT_OMN",
                "counterFundCode": "ROBOT_AM2_2359",
                "amount": "4000.00",
                "reason": "Subscription amount is less than minimum initial purchase.",
                "customerName": "หม่อมหลวง คิวบีทดสอบหนึ่ง โรบอทคิวบีทดสอบหนึ่ง"
            },
            {
                "paymentDate": "20220905",
                "xwtReferenceNo": "0042209050000221",
                "accountNo": "account_qc_xwt",
                "counterUnitholderNo": "XWT_OMN",
                "counterFundCode": "ROBOT_AM2_2359",
                "amount": "4000.00",
                "reason": "Subscription amount is less than minimum purchase.",
                "customerName": "หม่อมหลวง คิวบีทดสอบหนึ่ง โรบอทคิวบีทดสอบหนึ่ง"
            }
        ])
    })
});
```

รูปที่ 3.109 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 3 เรียกใช้ API get Failed XWT Order โดยค้นหาด้วย paymentDate

```
it('Can list fail XWT order by paymentDate', () => {
    // Ref # 3599
    callApiGetfailXWTorder({ paymentDateFrom: '20220905', paymentDateTo: '20220905'})
    cy.get('@response').should((res) => {
        expect(res.status).equal(200)
        expect(res.body).to.deep.equal([
            {
                "paymentDate": "20220905",
                "xwtReferenceNo": "0042209050000213",
                "accountNo": "account_qc_xwt",
                "counterUnitholderNo": "XWT_OMN",
                "counterFundCode": "ROBOT_AM2_2359",
                "amount": "4000.00",
                "reason": "Subscription amount is less than minimum initial purchase.",
                "customerName": "หม่อมหลวง คิวซีทดสอบหนึ่ง โรบอทคิวซีทดสอบหนึ่ง"
            },
            {
                "paymentDate": "20220905",
                "xwtReferenceNo": "0042209050000214",
                "accountNo": "account_qc_xwt",
                "counterUnitholderNo": "XWT_OMN",
                "counterFundCode": "ROBOT_AM2_2359",
                "amount": "4000.00",
                "reason": "Subscription amount is less than minimum initial purchase.",
                "customerName": "หม่อมหลวง คิวซีทดสอบหนึ่ง โรบอทคิวซีทดสอบหนึ่ง"
            },
            {
                "paymentDate": "20220905",
                "xwtReferenceNo": "0042209050000221",
                "accountNo": "account_qc_xwt",
                "counterUnitholderNo": "XWT_OMN",
                "counterFundCode": "ROBOT_AM2_2359",
                "amount": "4000.00",
                "reason": "Subscription amount is less than minimum purchase.",
                "customerName": "หม่อมหลวง คิวซีทดสอบหนึ่ง โรบอทคิวซีทดสอบหนึ่ง"
            }
        ])
    })
});
```

รูปที่ 3.110 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 4 เรียกใช้ API get Failed XWT Order โดยค้นหาด้วย XWT Ref No.

```
Open Cypress | Set ".only"
it('Can list fail XWT order by XWT Reference No', () => {
    // Ref # 3600
    callApiGetfailXWTorder({xwtReferenceNo: '0042209050000221' })
    cy.get('@response').should((res) => {
        expect(res.status).equal(200)
        expect(res.body).to.deep.equal([
            {
                "paymentDate": "20220905",
                "xwtReferenceNo": "0042209050000221",
                "accountNo": "account_qc_xwt",
                "counterUnitholderNo": "XWT_OMN",
                "counterFundCode": "ROBOT_AM2_2359",
                "amount": "4000.00",
                "reason": "Subscription amount is less than minimum purchase.",
                "customerName": "หม่อมหลวง คิวซีทดสอบหนึ่ง โรบอทคิวซีทดสอบหนึ่ง"
            }
        ])
    })
});
```

รูปที่ 3.111 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 4

กรณีทดสอบที่ 5 เรียกใช้ API get Failed XWT Order โดยค้นหาด้วย account No และ Payment Date

```
Open Cypress | Set ".only"
it('Can list fail XWT order by account No and paymentDate', () => {
    // Ref # 3601
    callApiGetfailXWTorder({ paymentDateFrom: '20220905', paymentDateTo: '20220905', accountNo: 'account_qc_xwt'})
    cy.get('@response').should((res) => {
        expect(res.status).equal(200)
        expect(res.body).to.deep.equal([
            {
                "paymentDate": "20220905",
                "xwtReferenceNo": "0042209050000213",
                "accountNo": "account_qc_xwt",
                "counterUnitholderNo": "XWT_OMN",
                "counterFundCode": "ROBOT_AM2_2359",
                "amount": "4000.00",
                "reason": "Subscription amount is less than minimum initial purchase.",
                "customerName": "หม่อมหลวง คิวซีทดสอบหนึ่ง โรบอทคิวซีทดสอบหนึ่ง"
            },
            {
                "paymentDate": "20220905",
                "xwtReferenceNo": "0042209050000214",
                "accountNo": "account_qc_xwt",
                "counterUnitholderNo": "XWT_OMN",
                "counterFundCode": "ROBOT_AM2_2359",
                "amount": "4000.00",
                "reason": "Subscription amount is less than minimum initial purchase.",
                "customerName": "หม่อมหลวง คิวซีทดสอบหนึ่ง โรบอทคิวซีทดสอบหนึ่ง"
            },
            {
                "paymentDate": "20220905",
                "xwtReferenceNo": "0042209050000221",
                "accountNo": "account_qc_xwt",
                "counterUnitholderNo": "XWT_OMN",
                "counterFundCode": "ROBOT_AM2_2359",
                "amount": "4000.00",
                "reason": "Subscription amount is less than minimum purchase.",
                "customerName": "หม่อมหลวง คิวซีทดสอบหนึ่ง โรบอทคิวซีทดสอบหนึ่ง"
            }
        ])
    })
```

รูปที่ 3.112 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 6 เรียกใช้ API get Failed XWT Order โดยค้นหาด้วย account No และ XWT Ref No.

```
Open Cypress | Set "only"
it('Can list fail XWT order by account No and xwtReferenceNo', () => {
    // Ref # 3602
    callApiGetfailXWTorder({ accountNo: 'account_qc_xwt', xwtReferenceNo: '0042209050000221' })
    cy.get('@response').should((res) => {
        expect(res.status).equal(200)
        expect(res.body).to.deep.equal([
            {
                "paymentDate": "20220905",
                "xwtReferenceNo": "0042209050000221",
                "accountNo": "account_qc_xwt",
                "counterUnitholderNo": "XWT_OMN",
                "counterFundCode": "ROBOT_AM2_2359",
                "amount": "4000.00",
                "reason": "Subscription amount is less than minimum purchase.",
                "customerName": "หม่อมหลวง คิวซีทดสอบหนึ่ง โรบอทคิวซีทดสอบหนึ่ง"
            }
        ])
    })
});
```

รูปที่ 3.113 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 6

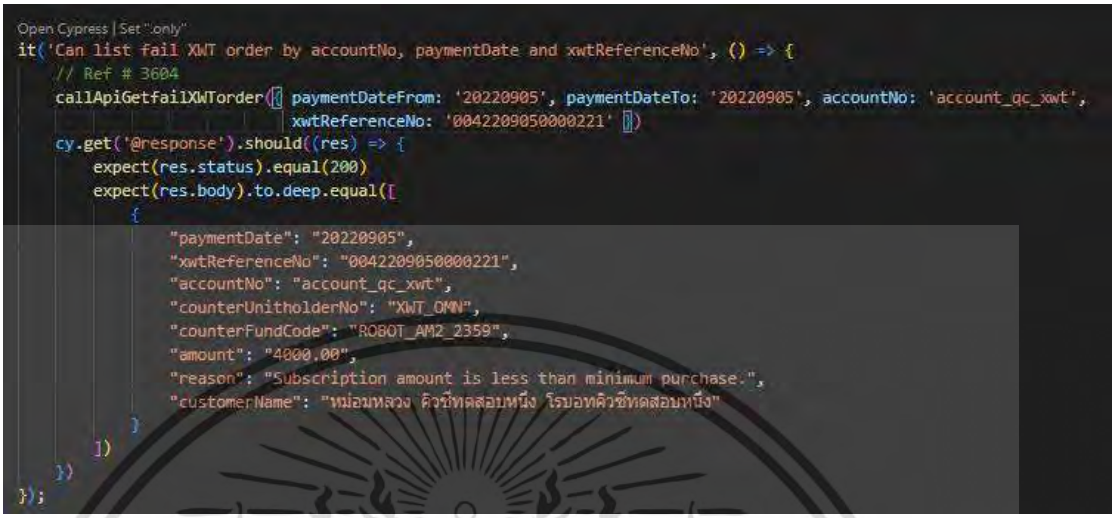
กรณีทดสอบที่ 7 เรียกใช้ API get Failed XWT Order โดยค้นหาด้วย Payment Date และ XWT Ref No

```
Open Cypress | Set "only"
it('Can list fail XWT order by paymentDate and xwtReferenceNo', () => {
    // Ref # 3603
    callApiGetfailXWTorder({ paymentDateFrom: '20220905', paymentDateTo: '20220905', xwtReferenceNo: '0042209050000221'
    cy.get('@response').should((res) => {
        expect(res.status).equal(200)
        expect(res.body).to.deep.equal([
            {
                "paymentDate": "20220905",
                "xwtReferenceNo": "0042209050000221",
                "accountNo": "account_qc_xwt",
                "counterUnitholderNo": "XWT_OMN",
                "counterFundCode": "ROBOT_AM2_2359",
                "amount": "4000.00",
                "reason": "Subscription amount is less than minimum purchase.",
                "customerName": "หม่อมหลวง คิวซีทดสอบหนึ่ง โรบอทคิวซีทดสอบหนึ่ง"
            }
        ])
    })
});
```

รูปที่ 3.114 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

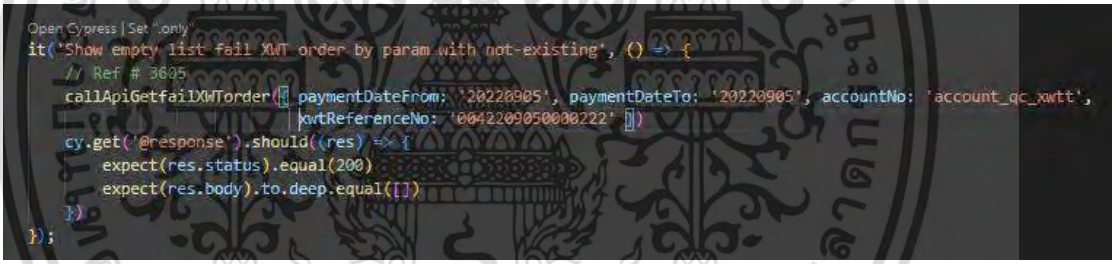
กรณีทดสอบที่ 8 เรียกใช้ API get Failed XWT Order โดยค้นหาด้วย Payment Date, account No. และ XWT Ref No



```
Open Cypress | Set ".only"
it('Can list fail XWT order by accountNo, paymentDate and xwtReferenceNo', () => {
    // Ref # 3604
    callApiGetfailXWTorder({ paymentDateFrom: '20220905', paymentDateTo: '20220905', accountNo: 'account_qc_xwt',
                             xwtReferenceNo: '0042209050000221' })
    cy.get('@response').should((res) => {
        expect(res.status).equal(200)
        expect(res.body).to.deep.equal([
            {
                "paymentDate": "20220905",
                "xwtReferenceNo": "0042209050000221",
                "accountNo": "account_qc_xwt",
                "counterUnitholderNo": "XWT_OMN",
                "counterFundCode": "ROBOT_AM2_2359",
                "amount": "4000.00",
                "reason": "Subscription amount is less than minimum purchase.",
                "customerName": "หม่อมหลวง คิวซีทดสอบหนึ่ง โรบอทคิวซีทดสอบหนึ่ง"
            }
        ])
    })
});
```

รูปที่ 3.115 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 8

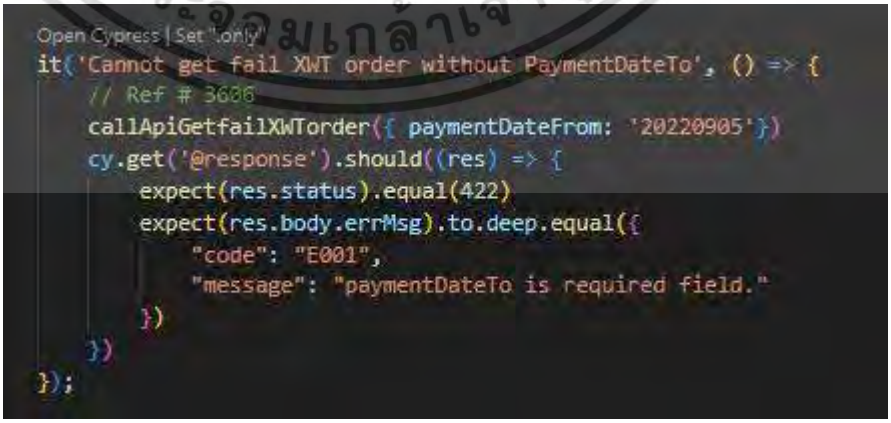
กรณีทดสอบที่ 9 เรียกใช้ API get Failed XWT Order โดยค้นหาข้อมูลที่ไม่มีในระบบ



```
Open Cypress | Set ".only"
it('Show empty list fail XWT order by param with not-existing', () => {
    // Ref # 3605
    callApiGetfailXWTorder({ paymentDateFrom: '20220905', paymentDateTo: '20220905', accountNo: 'account_qc_xwtt',
                             xwtReferenceNo: '0042209050000222' })
    cy.get('@response').should((res) => {
        expect(res.status).equal(200)
        expect(res.body).to.deep.equal([])
    })
});
```

รูปที่ 3.116 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 9

กรณีทดสอบที่ 10 เรียกใช้ API get Failed XWT Order โดยไม่ใส่ Payment Date To



```
Open Cypress | Set ".only"
it('Cannot get fail XWT order without PaymentDateTo', () => {
    // Ref # 3606
    callApiGetfailXWTorder({ paymentDateFrom: '20220905'})
    cy.get('@response').should((res) => {
        expect(res.status).equal(422)
        expect(res.body.errMsg).to.deep.equal({
            "code": "E001",
            "message": "paymentDateTo is required field."
        })
    })
});
```

รูปที่ 3.117 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 11 เรียกใช้ API get Failed XWT Order โดยไม่ใส่ Payment Date From

```
Open Cypress | Set ".only"
it('Cannot get fail XWT order without PaymentDateFrom', () => {
    // Ref # 3607
    callApiGetfailXWTorder({ paymentDateTo: '20220905'})
    cy.get('@response').should((res) => {
        expect(res.status).equal(422)
        expect(res.body.errMsg).to.deep.equal({
            "code": "E001",
            "message": "paymentDateFrom is required field."
        })
    })
});
```

รูปที่ 3.118 Script ในการทำ Automated Test API ตัวอย่างที่ 1 กรณีที่ 11

ตัวอย่างที่ 2 การทดสอบ Automated Test API

Requirement : API สำหรับใช้ในการ Approved รายการ Cross AMC Switching

ตารางที่ 3.12 Test Script การทดสอบ Automated Test API ตัวอย่างที่ 2

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result
1	API	SA	Approve cross amc switching		- Send API Approve โดยใส่ Transaction ID ที่ไม่มีในระบบ	- แสดงข้อความ “Transaction ID cannot be found”
2	API	SA	Approve cross amc switching		- Send API Approve โดยใส่ Transaction ID ที่มี Status Submit	- สามารถApprove รายการดังกล่าวได้
3	API	SA	Approve cross amc switching		- Send API Approve โดยไม่ใส่ Status	- แสดงข้อความ “Status is required field” - ไม่สามารถ Approve รายการดังกล่าวได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4	API	SA	Approve cross amc switching		- Send API Approve โดยใส่ Status ไม่ใช่ Approve, Cancel	- แสดงข้อความ “Invalid field format(Status)” - ไม่สามารถ Approve รายการดังกล่าวได้
5	API	SA	Approve cross amc switching		- Send API Approve โดย Login : SA passthrough	- ไม่สามารถ Approve รายการดังกล่าวได้
6	API	SA	Approve cross amc switching		- Send API Approve โดยไม่ Login	- ไม่สามารถ Approve รายการดังกล่าวได้
7	API	SA	Approve cross amc switching		- Send API Approve โดยใส่ Transaction ID ที่มี Status Approved	- แสดงข้อความ “Transaction is already approved” - ไม่สามารถ Approve รายการดังกล่าวได้

ขั้นตอนการทดสอบ

สร้างรายการ CrossAmcSwitching ผ่าน API

```
before(() => {
    resetFundBeforeTest()
    cy.apiLogin('ROBOT_SA1_API1');
    createCrossAmcSwitchingOrder().as('transactionId');
});
```

รูปที่ 3.119 Script ในการทำ Automated Test API ตัวอย่างที่ 2 เรียกใช้ API Create Order

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน Call API Create CrossAmcSwitching โดยมีข้อมูล ดังรูปที่ 120

```
function createCrossAmcSwitchingOrder() {
    const crossAmcSwitchingOrderBody = {
        "saOrderReferenceNo": `${todayDate + (Math.random() * 100000000).toFixed(0)}`,
        "transactionDateTime": DateTime.local().startOf('day').toFormat("yyyyMMddHHmmss"),
        "saCode": "ROBOTSA1",
        "accountId": "[redacted]",
        "unitholderId"[redacted],
        "fundCode": "ROBOT_AM14_2100",
        "redemptionType": "AMT",
        "amount": 5555,
        "counterUnitholderId": "XWT_OMN",
        "counterFundCode": "ROBOT_AM2_2359",
        "effectiveDate": todayDate,
        "overrideRiskProfile": "Y",
        "overrideFxRisk": "Y",
        "paymentType": "COL_SA",
        "collateralAccount": "1234567890",
        "channel": "MOB",
        "iclicense": "00108",
        "branchNo": "",
        "forceEntry": "N",
        "sellAllUnitFlag": "N",
        "autoRedeemFundCode": null,
        "status": "SUBMITTED"
    };

    return cy.request({
        method: 'POST',
        url: `api/crossAmcSwitching`,
        body: crossAmcSwitchingOrderBody
    }).its('body.transactionId');
}
```

รูปที่ 3.120 Script ในการทำ Automated Test API ตัวอย่างที่ 2 Create Order

การลบรายการ CrossAmcSwitching หลังจบการทดสอบ เพื่อลดความทับซ้อนของข้อมูล

```
after(() => {
    cy.get('@transactionId').then(transactionId => {
        teardownCrossAmcSwitchingOrder(transactionId);
    });
});
```

รูปที่ 3.121 Script ในการทำ Automated Test API ตัวอย่างที่ 2 เรียกใช้ API Delete Order

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน Delete CrossAmcSwitching โดยส่งค่า transactionId ที่ต้องลบออกจากระบบ

```
function teardownCrossAmcSwitchingOrder(transactionId) {
    cy.request({
        method: 'GET',
        url: `[redacted]/deleteFundOrder?code=${transactionId}`,
        failOnStatusCode: false,
    })
}
```

รูปที่ 3.122 Script ในการทำ Automated Test API ตัวอย่างที่ 2 Delete Order

ตัวอย่าง API Approved CrossAmcSwitching ในเครื่องมือ Postman

รูปที่ 3.123 ตัวอย่าง API Approved CrossAmcSwitching

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 1 ไม่สามารถ Approveรายการที่ไม่มีอยู่ในระบบได้

```
Open Cypress | Set ".only"
it('Cannot approve cross amc switching order with non-existing transaction id', () => {
    //ref#3495
    cy.request({
        method: 'PATCH',
        url:                    /crossAmcSwitching/notExistingId',
        failOnStatusCode: false,
        body: { "status": "Approve" }
    }).then(response => {
        expect(response.status).to.equal(422);
        expect(response.body).to.deep.equal({
            errMsg: {
                "code": "E107",
                "message": "Transaction ID cannot be found."
            }
        });
    });
});
```

รูปที่ 3.124 Script ในการทำ Automated Test API ตัวอย่างที่ 2 กรณีที่ 1

กรณีทดสอบที่ 2 สามารถ Approveรายการด้วยการใส่ Status เป็น Approve

```
Open Cypress | Set ".only"
it('Can approve cross amc switching order with status = Approve', () => {
    //ref#3496
    cy.get('@transactionId').then(transactionId => {
        cy.request({
            method: 'PATCH',
            url:                    `/api/crossAmcSwitching/${transactionId}`,
            body: { "status": "Approve" }
        }).then(response => {
            expect(response.status).to.equal(200);
        });
    });
});
```

รูปที่ 3.125 Script ในการทำ Automated Test API ตัวอย่างที่ 2 กรณีที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 3 ไม่สามารถ Approveรายการหากไม่ใส่ข้อมูล Status

```
Open Cypress | Set ".only"
it('Cannot approve cross amc switching order without field status', () => {
    // ref#3497
    cy.get('@transactionId').then(transactionId => {
        cy.request({
            method: 'PATCH',
            [redacted]`/api/crossAmcSwitching/${transactionId}`,
            failOnStatusCode: false,
            body: { }
        }).then(response => {
            expect(response.status).to.equal(422);
            expect(response.body).to.deep.equal({
                errMsg: {
                    "code": "E001",
                    "message": "status is required field."
                }
            });
        });
    });
});
```

รูปที่ 3.126 Script ในการทำ Automated Test API ตัวอย่างที่ 2 กรณีที่ 3

กรณีทดสอบที่ 4 ไม่สามารถ Approveรายการหากใส่ข้อมูล Status ไม่ถูกต้อง

```
Open Cypress | Set ".only"
it('Cannot approve cross amc switching order with invalid status field', () => {
    // ref#3498
    cy.get('@transactionId').then(transactionId => {
        cy.request({
            method: 'PATCH',
            [redacted]`/api/crossAmcSwitching/${transactionId}`,
            failOnStatusCode: false,
            body: { "status": "invalid" }
        }).then(response => {
            expect(response.status).to.equal(422);
            expect(response.body).to.deep.equal({
                errMsg: {
                    "code": "E002",
                    "message": "Invalid field format (status)."
                }
            });
        });
    });
});
```

รูปที่ 3.127 Script ในการทำ Automated Test API ตัวอย่างที่ 2 กรณีที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 5 ไม่สามารถ ApproveรายการหากLogin SA ที่เป็นpassthrough

```
Open Cypress | Set ".only"
it('SA passthrough cannot approve cross amc switching order', () => {
    //ref#3499
    cy.apiLogin('ROBOT_SA2_API1');

    cy.get('@transactionId').then(transactionId => {
        cy.request({
            method: 'PATCH',
            [redacted]`/api/crossAmcSwitching/${transactionId}`,
            failOnStatusCode: false,
            body: { "status": "Approve" }
        }).then(response => {
            expect(response.status).to.equal(401);
            expect(response.body).to.deep.equal({
                errMsg: {
                    "code": "E000",
                    "message": "Unauthorized access"
                }
            });
        });
    });
});
```

รูปที่ 3.128 Script ในการทำ Automated Test API ตัวอย่างที่ 2 กรณีที่ 5

กรณีทดสอบที่ 6 ไม่สามารถ Approve รายการหากไม่ Login SA

```
Open Cypress | Set ".only"
it('Cannot approve cross amc switching order without login', () => {
    //ref#3500
    cy.clearTokenApi();

    cy.get('@transactionId').then(transactionId => {
        cy.request({
            method: 'PATCH',
            [redacted] + `/api/crossAmcSwitching/${transactionId}`,
            failOnStatusCode: false,
            body: { "status": "Approve" }
        }).then(response => {
            expect(response.status).to.equal(401);
            expect(response.body).to.deep.equal({
                errMsg: {
                    "code": "E000",
                    "message": "Unauthorized access"
                }
            });
        });
    });
});
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.129 Script ในการทำ Automated Test API ตัวอย่างที่ 2 กรณีที่ 6

กรณีทดสอบที่ 7 ไม่สามารถ Approve รายการได้หาก Status เป็น Approved

```
Open Cypress | Set ".only"
it('Cannot approve cross amc switching order with already approved transaction id', () => {
    // ref#3501
    cy.apiLogin('ROBOT_SA1_API1');

    cy.get('@transactionId').then(transactionId => {
        cy.request({
            method: 'PATCH',
            [redacted] `/api/crossAmcSwitching/${transactionId}`,
            failOnStatusCode: false,
            body: { "status": "Approve" }
        }).then(response => {
            expect(response.status).to.equal(422);
            expect(response.body).to.deep.equal({
                errMsg: {
                    "code": "E106",
                    "message": "Transaction is already approved."
                }
            });
        });
    });
});
```

รูปที่ 3.130 Script ในการทำ Automated Test API ตัวอย่างที่ 2 กรณีที่ 7

ตัวอย่างที่ 3 การทดสอบ Automated Test API

Requirement : API สำหรับ Update Unitholder Bank SUB/RED ของ SA Normal ระบบจะให้ updateข้อมูล เฉพาะลูกค้าที่เป็นประเภท Segregate เท่านั้น และไม่สามารถ Update บัญชีลูกค้าที่ติดแม่กุญแจได้

ตารางที่ 3.13 Test Script การทดสอบ Automated Test API ตัวอย่างที่ 3

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result
1	API	SA	Update bank unitholder		- Send API โดย Unitholder Type : OMN	- ไม่สามารถ Update bank ได้ - แสดงข้อความ”Cannot update bank Account due to this unitholder type is Omnibus Account”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2	API	SA	Update bank unitholder		- Send API โดย Update Bank SUB/RED ครบทั้ง 5 ธนาคาร	-สามารถ Update bank ได้
3	API	SA	Update bank unitholder		- Send API โดย Update Bank เฉพาะ SUB	- ไม่สามารถ Update bank ได้ - แสดงข้อความ”redemptionBank is required field”
4	API	SA	Update bank unitholder		- Send API โดย Update Bank เฉพาะ RED	- ไม่สามารถ Update bank ได้ - แสดงข้อความ”subscriptionBank is required field”
5	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูล BankAccountNo (SUB) น้อยกว่า 10 char	- ไม่สามารถ Update bank ได้ - แสดงข้อความ”Invalid field format(subscriptionBankAccount)”
6	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูล BankAccount No (RED) น้อยกว่า 10 char	- ไม่สามารถ Update bank ได้ - แสดงข้อความ”Invalid field format(redemptionBankAccount)”
7	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูลเลือก Bank SCB และ BankAccount	- ไม่สามารถ Update bank ได้ - แสดงข้อความ”Invalid

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

					No (SUB) ไม่ถูกต้องตาม Format SCB	field format(Bank Account No)”
8	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูลเลือก Bank SCB และ BankAccount No (RED) ไม่ถูกต้องตาม Format SCB	- ไม่สามารถ Update bank ได้ - แสดงข้อความ”Invalid field format(Bank Account No)”
9	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูลเลือก Default Bank(SUB) มากกว่า 1 Bank	- ไม่สามารถ Update bank ได้ - แสดงข้อความ”subscriptionBankAccount.default is invalid”
10	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูลเลือก Default Bank(RED) มากกว่า 1 Bank	- ไม่สามารถ Update bank ได้ - แสดงข้อความ”redemptionBankAccount.default is invalid”
11	API	SA	Update bank unitholder		- Send API โดยข้อมูล Bank (SUB) มากกว่า 5 ธนาคาร	- ไม่สามารถ Update bank ได้ - แสดงข้อความ” subscriptionBankAccount.limitBanks is invalid”
12	API	SA	Update bank unitholder		- Send API โดยข้อมูล Bank (RED) มากกว่า 5 ธนาคาร	- ไม่สามารถ Update bank ได้ - แสดงข้อความ” redemptionBankAccount.limitBanks is invalid”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูล Bank Code(SUB) ที่ไม่มีในระบบ	- ไม่สามารถ Update bank ได้ - แสดงข้อความ” subscriptionBankAccount.BankCode is invalid”
14	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูล Bank Code(RED) ที่ไม่มีในระบบ	- ไม่สามารถ Update bank ได้ - แสดงข้อความ” redemptionBankAccount.BankCode is invalid”
15	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูล Bank Branch(SUB) ที่ไม่มีในระบบ	- ไม่สามารถ Update bank ได้ - แสดงข้อความ” subscriptionBankAccount.BankBranch is invalid”
16	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูล Bank Branch(RED) ที่ไม่มีในระบบ	- ไม่สามารถ Update bank ได้ - แสดงข้อความ” redemptionBankAccount.BankBranch is invalid”
17	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูล Unitholder ที่ไม่มีในระบบ	- ไม่สามารถ Update bank ได้ - แสดงข้อความ”Unitholder ID cannot be found”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทดสอบ

ฟังก์ชันการ Reset Unitholder Bank Account ก่อนทำงานทดสอบในแต่ละรอบ

```
beforeEach(() => {
    cy.apiLogin('ROBOT_SA1_API1');
    resetAccountRobot();
})

function resetAccountRobot() {
    return cy.request({
        method: 'GET',
        url: '                resetAccountRobotForTestApiUpdateUnitholder',
        failOnStatusCode: false,
    }, {forceResetCache: true});
}
```

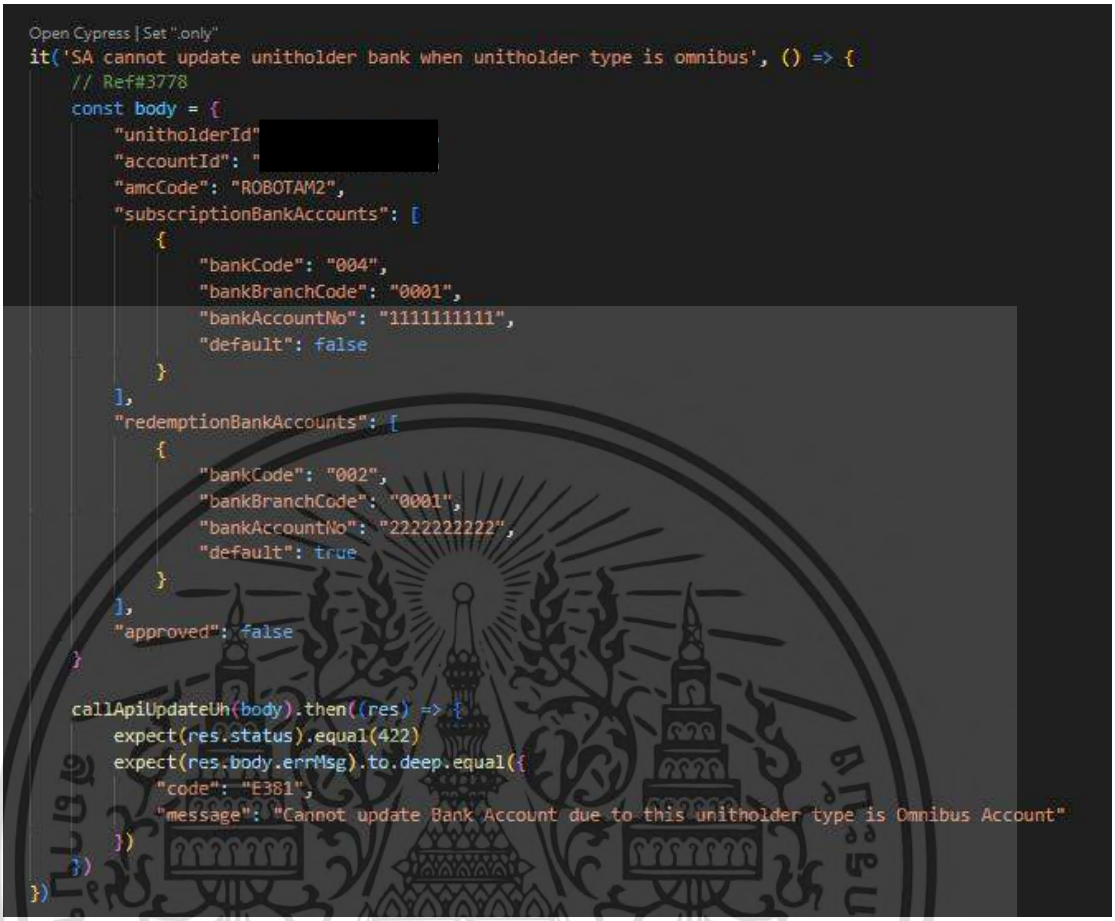
รูปที่ 3.131 ตัวอย่าง Reset Unitholder Bank Account

ตัวอย่าง API Update Unitholder bank ในเครื่องมือ Postman

รูปที่ 3.132 ตัวอย่าง API Update Unitholder bank

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 1 ไม่สามารถUpdate Unitholder Bankได้หากUnitholderTypeเป็น OMN



```
Open Cypress | Set ".only"
it('SA cannot update unitholder bank when unitholder type is omnibus', () => {
    // Ref#3778
    const body = {
        "unitholderId"
        "accountId": "
        "amcCode": "ROBOTAM2",
        "subscriptionBankAccounts": [
            {
                "bankCode": "004",
                "bankBranchCode": "0001",
                "bankAccountNo": "1111111111",
                "default": false
            }
        ],
        "redemptionBankAccounts": [
            {
                "bankCode": "002",
                "bankBranchCode": "0001",
                "bankAccountNo": "2222222222",
                "default": true
            }
        ],
        "approved": false
    }

    callApiUpdateUh(body).then((res) => {
        expect(res.status).equal(422)
        expect(res.body.errMsg).to.deep.equal({
            "code": "E381",
            "message": "Cannot update Bank Account due to this unitholder type is Omnibus Account"
        })
    })
})
```

รูปที่ 3.133 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 2 สามารถUpdate Unitholder Bankได้หากUnitholderTypeเป็น SEG

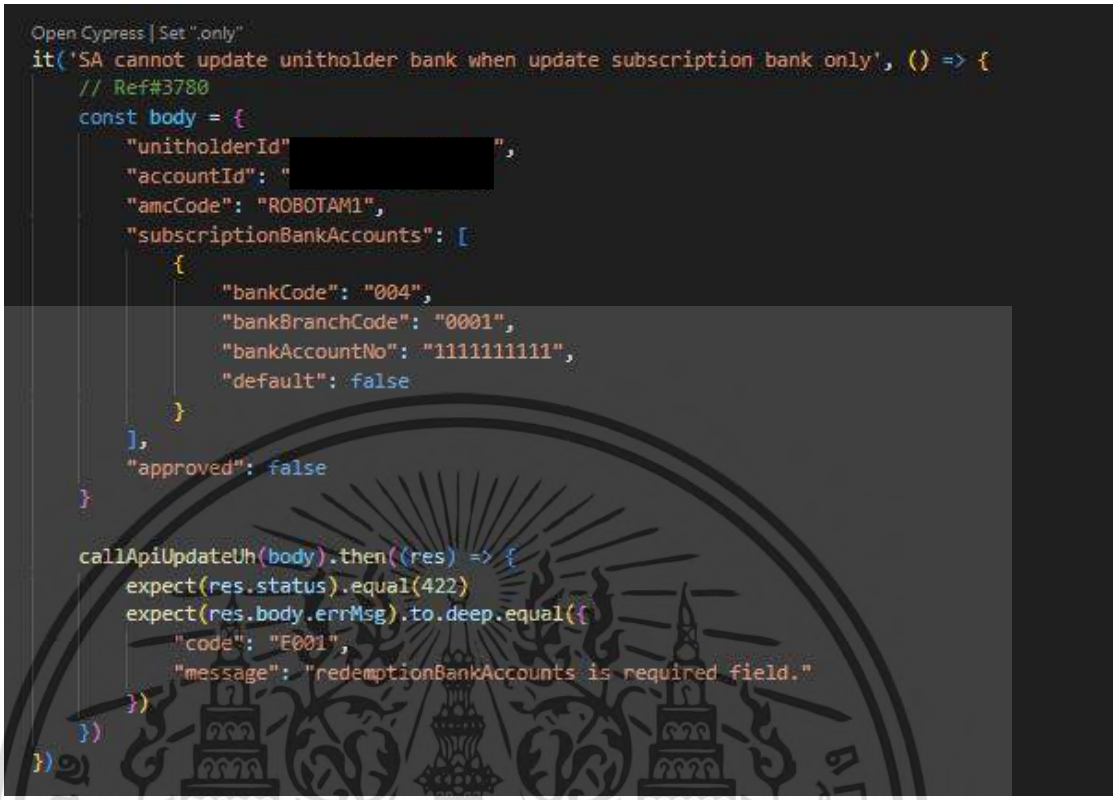
```
Open Cypress | Set ".only"
it('SA can update unitholder bank when unitholder type is segregate and input all data', () => {
    // Ref#3779
    const body = {
        "unitholderId": "[redacted]",
        "accountId": "[redacted]",
        "amcCode": "ROBOTAM1",
        "subscriptionBankAccounts": [
            {
                "bankCode": "014",
                "bankBranchCode": "0001",
                "bankAccountNo": "1111111114",
                "default": true
            },
            {
                "bankCode": "004",
                "bankBranchCode": "00000",
                "bankAccountNo": "1111111111",
                "default": false
            },
            {
                "bankCode": "006",
                "bankBranchCode": "00000",
                "bankAccountNo": "2222222222",
                "default": false
            },
            {
                "bankCode": "008",
                "bankBranchCode": "00000",
                "bankAccountNo": "3333333333",
                "default": false
            },
            {
                "bankCode": "010",
                "bankBranchCode": "00000",
                "bankAccountNo": "4444444444",
                "default": false
            }
        ]
    }

callApiUpdateUh(body).then((res) => {
    expect(res.status).equal(200)
})
```

รูปที่ 3.134 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 3 ไม่สามารถUpdate Unitholder Bankได้หากใส่เฉพาะข้อมูล Subscription Bank



```
Open Cypress | Set ".only"
it('SA cannot update unitholder bank when update subscription bank only', () => {
    // Ref#3780
    const body = {
        "unitholderId"[redacted],
        "accountId": "[redacted]
        "amcCode": "ROBOTAM1",
        "subscriptionBankAccounts": [
            {
                "bankCode": "004",
                "bankBranchCode": "0001",
                "bankAccountNo": "1111111111",
                "default": false
            }
        ],
        "approved": false
    }

    callApiUpdateUh(body).then((res) => {
        expect(res.status).equal(422)
        expect(res.body.errMsg).to.deep.equal({
            "code": "E001",
            "message": "redemptionBankAccounts is required field."
        })
    })
})
```

รูปที่ 3.135 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 4 ไม่สามารถ Update Unitholder Bankได้หากใส่เฉพาะข้อมูล Redemption Bank

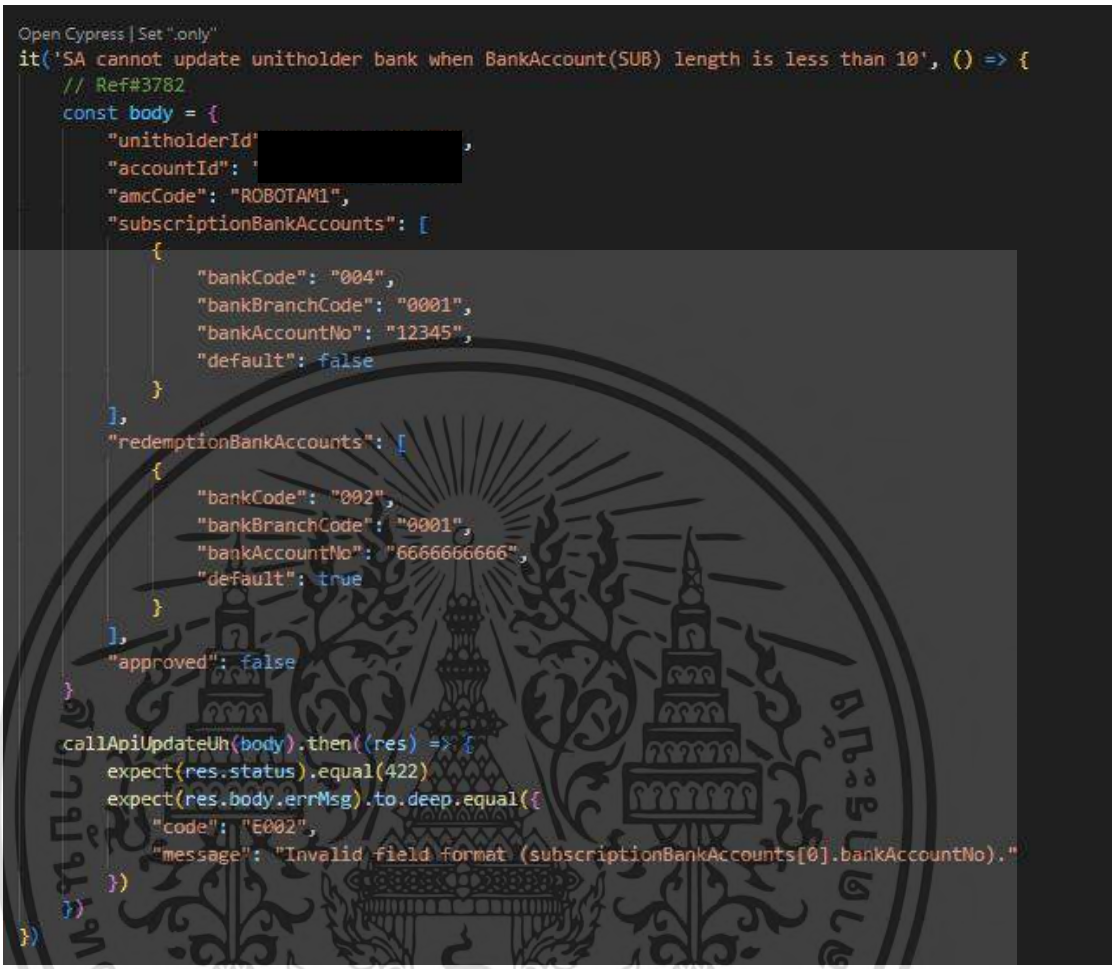
```
Open Cypress | Set ".only"
it('SA cannot update unitholder bank when update redemption bank only', () => {
    // Ref#3781
    const body = {
        "unitholderId": [redacted],
        "accountId": "[redacted]",
        "amcCode": "ROBOTAM1",
        "redemptionBankAccounts": [
            {
                "bankCode": "002",
                "bankBranchCode": "0001",
                "bankAccountNo": "2222222222",
                "default": true
            }
        ],
        "approved": false
    }

    callApiUpdateUh(body).then((res) => {
        expect(res.status).equal(422)
        expect(res.body.errMsg).to.deep.equal({
            "code": "E001",
            "message": "subscriptionBankAccounts is required field."
        })
    })
})
```

รูปที่ 3.136 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 5 ไม่สามารถUpdate Unitholder Bankได้หากใส่ข้อมูล Sub bankAccount No. น้อยกว่า 10หลัก

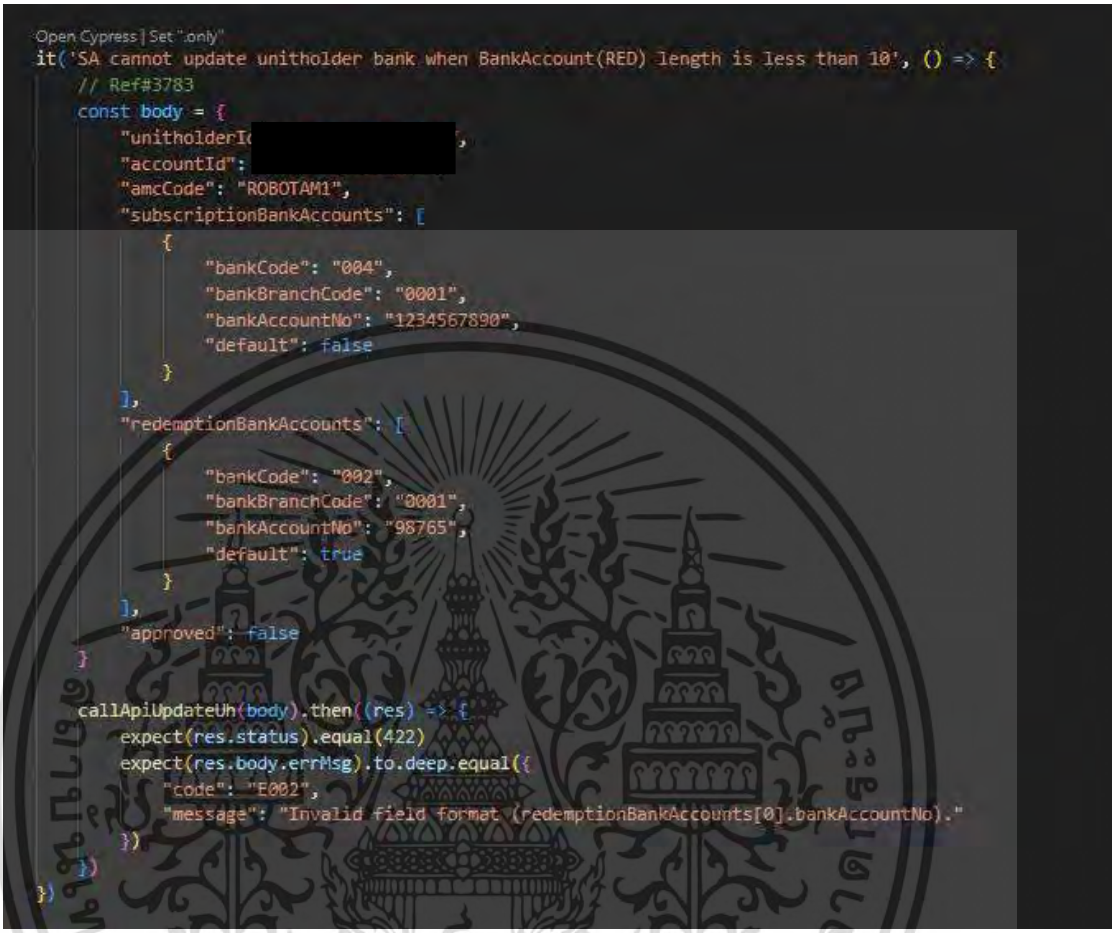


```
Open Cypress | Set ".only"
it('SA cannot update unitholder bank when BankAccount(SUB) length is less than 10', () => {
    // Ref#3782
    const body = {
        "unitholderId": [redacted],
        "accountId": [redacted],
        "amcCode": "ROBOTAM1",
        "subscriptionBankAccounts": [
            {
                "bankCode": "004",
                "bankBranchCode": "0001",
                "bankAccountNo": "12345",
                "default": false
            }
        ],
        "redemptionBankAccounts": [
            {
                "bankCode": "002",
                "bankBranchCode": "0001",
                "bankAccountNo": "6666666666",
                "default": true
            }
        ],
        "approved": false
    }
    callApiUpdateUh(body).then((res) => {
        expect(res.status).equal(422)
        expect(res.body.errMsg).to.deep.equal({
            "code": "E002",
            "message": "Invalid field format (subscriptionBankAccounts[0].bankAccountNo)."
        })
    })
})
```

รูปที่ 3.137 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 6 ไม่สามารถUpdate Unitholder Bankได้หากใส่ข้อมูล Red bankAccount No. น้อยกว่า 10หลัก



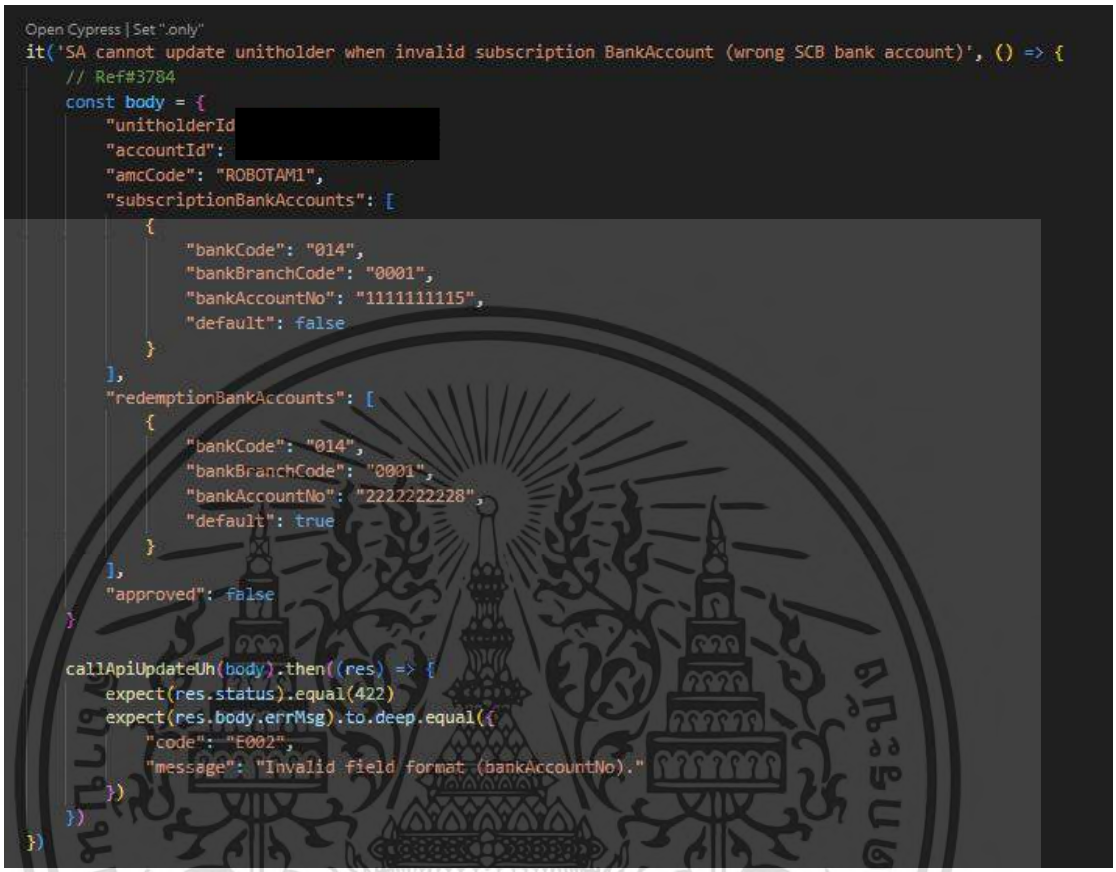
```
Open Cypress | Set ".only"
it('SA cannot update unitholder bank when BankAccount(RED) length is less than 10', () => {
    // Ref#3783
    const body = {
        "unitholderI[redacted],
        "accountId":
        "amcCode": "ROBOTAM1",
        "subscriptionBankAccounts": [
            {
                "bankCode": "004",
                "bankBranchCode": "0001",
                "bankAccountNo": "1234567890",
                "default": false
            }
        ],
        "redemptionBankAccounts": [
            {
                "bankCode": "002",
                "bankBranchCode": "0001",
                "bankAccountNo": "98765",
                "default": true
            }
        ],
        "approved": false
    }

    callApiUpdateUh(body).then((res) => {
        expect(res.status).equal(422)
        expect(res.body.errMsg).to.deep.equal({
            "code": "E002",
            "message": "Invalid field format (redemptionBankAccounts[0].bankAccountNo)."
        })
    })
})
```

รูปที่ 3.138 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 7 ไม่สามารถUpdate Unitholder Bankได้หากใส่ข้อมูล Sub Bank Code เป็น SCB และbankAccount No. ไม่ถูกต้องตามFormat



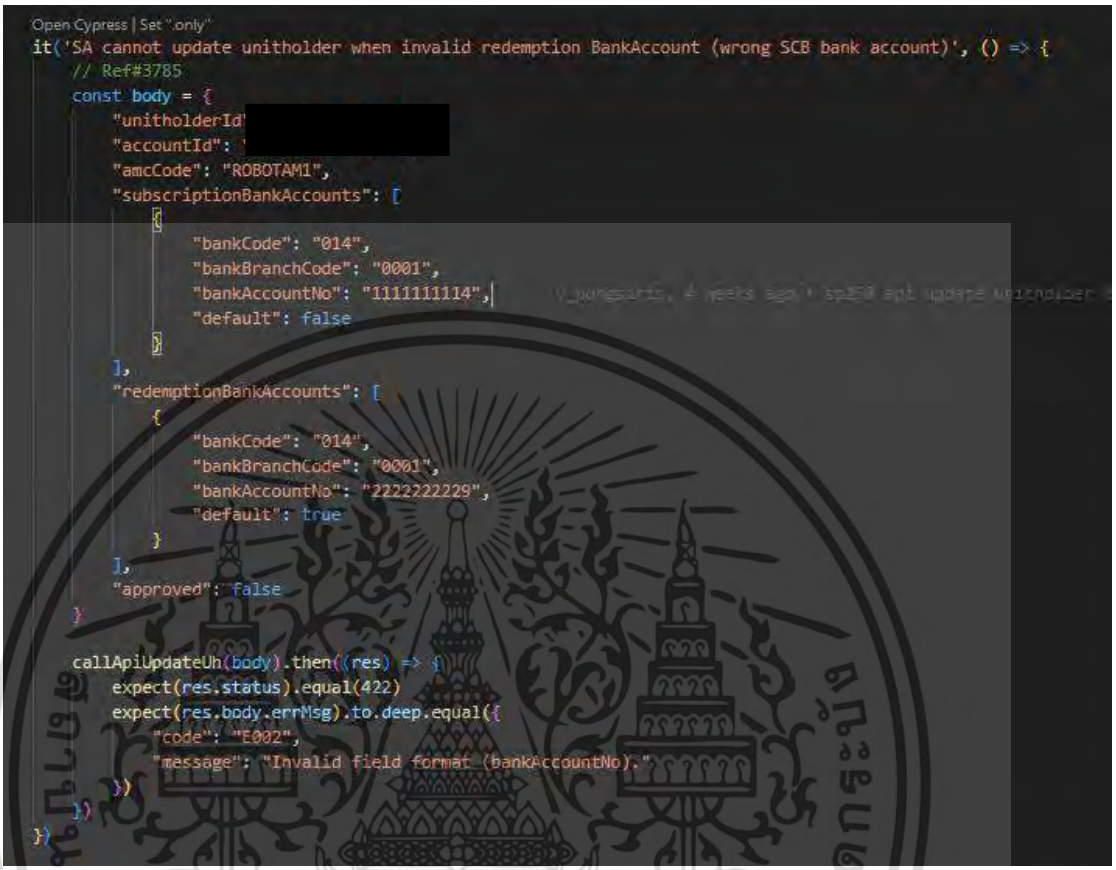
```
Open Cypress | Set ".only"
it('SA cannot update unitholder when invalid subscription BankAccount (wrong SCB bank account)', () => {
    // Ref#3784
    const body = {
        "unitholderId
        "accountId":
        "amcCode": "ROBOTAM1",
        "subscriptionBankAccounts": [
            {
                "bankCode": "014",
                "bankBranchCode": "0001",
                "bankAccountNo": "1111111115",
                "default": false
            }
        ],
        "redemptionBankAccounts": [
            {
                "bankCode": "014",
                "bankBranchCode": "0001",
                "bankAccountNo": "2222222228",
                "default": true
            }
        ],
        "approved": false
    }

    callApiUpdateUh(body).then((res) => {
        expect(res.status).equal(422)
        expect(res.body.errMsg).to.deep.equal({
            "code": "E002",
            "message": "Invalid field format (bankAccountNo)."
        })
    })
})
```

รูปที่ 3.139 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 8 ไม่สามารถUpdate Unitholder Bankได้หากใส่ข้อมูล Red Bank Code เป็น SCB และbankAccount No. ไม่ถูกต้องตามFormat



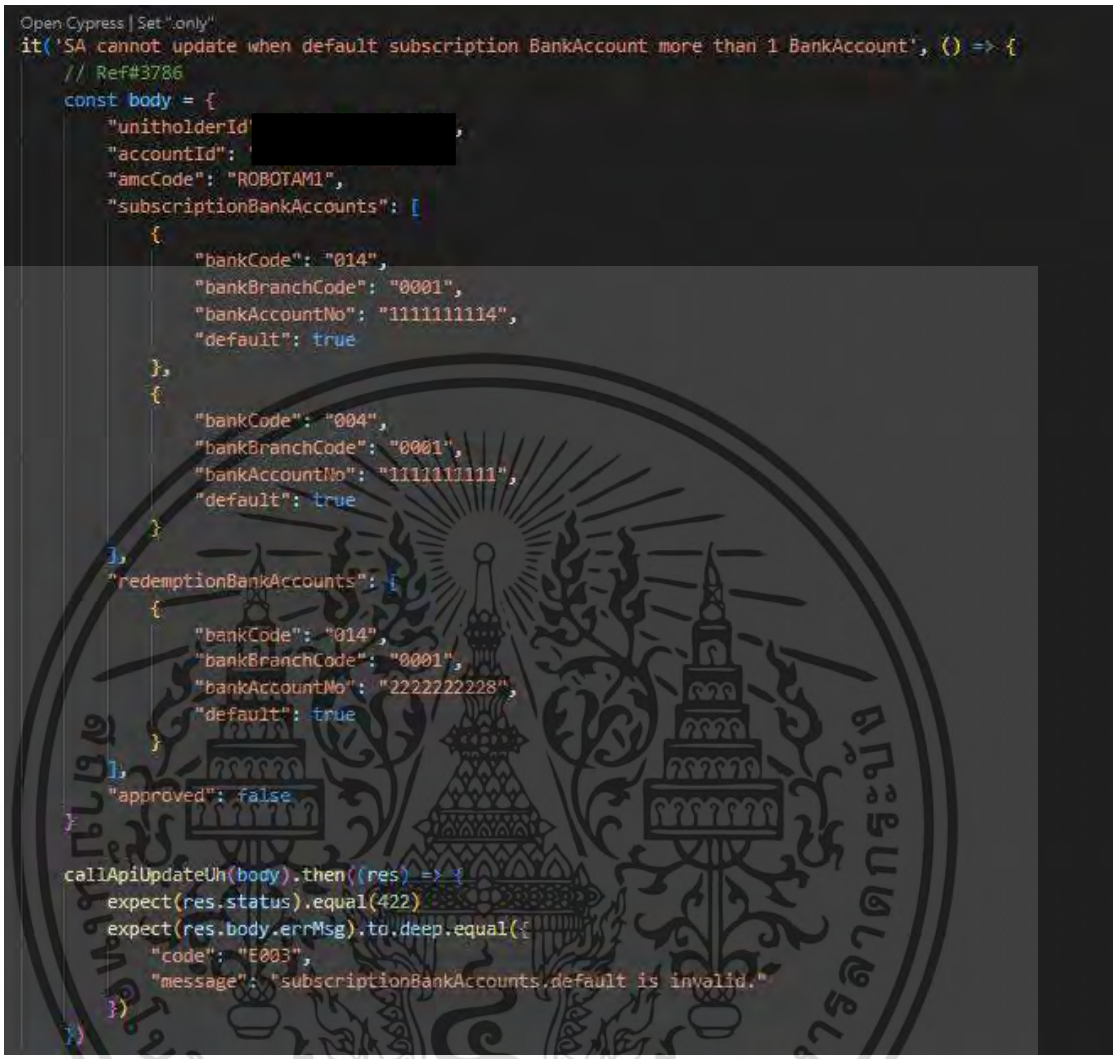
```
Open Cypress | Set ".only"
it('SA cannot update unitholder when invalid redemption BankAccount (wrong SCB bank account)', () => {
    // Ref#3785
    const body = {
        "unitholderId": [redacted]
        "accountId": [redacted]
        "amcCode": "ROBOTAM1",
        "subscriptionBankAccounts": [
            {
                "bankCode": "014",
                "bankBranchCode": "0001",
                "bankAccountNo": "1111111114",
                "default": false
            }
        ],
        "redemptionBankAccounts": [
            {
                "bankCode": "014",
                "bankBranchCode": "0001",
                "bankAccountNo": "2222222229",
                "default": true
            }
        ],
        "approved": false
    }

    callApiUpdateUh(body).then((res) => {
        expect(res.status).equal(422)
        expect(res.body.errMsg).to.deep.equal({
            "code": "E002",
            "message": "Invalid field format (bankAccountNo)."
        })
    })
})
```

รูปที่ 3.140 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 9 ไม่สามารถUpdate Unitholder Bankได้หากใส่ข้อมูล Sub Default Bank มากกว่า 1 ธนาคาร



```
Open Cypress | Set ".only"
it('SA cannot update when default subscription BankAccount more than 1 BankAccount', () => {
    // Ref#3786
    const body = {
        "unitholderId": [redacted],
        "accountId": [redacted],
        "amcCode": "ROBOTAM1",
        "subscriptionBankAccounts": [
            {
                "bankCode": "014",
                "bankBranchCode": "0001",
                "bankAccountNo": "1111111114",
                "default": true
            },
            {
                "bankCode": "004",
                "bankBranchCode": "0001",
                "bankAccountNo": "1111111111",
                "default": true
            }
        ],
        "redemptionBankAccounts": [
            {
                "bankCode": "014",
                "bankBranchCode": "0001",
                "bankAccountNo": "2222222228",
                "default": true
            }
        ],
        "approved": false
    }

    callApiUpdateUh(body).then((res) => {
        expect(res.status).equal(422)
        expect(res.body.errMsg).to.deep.equal({
            "code": "E003",
            "message": "subscriptionBankAccounts.default is invalid."
        })
    })
```

รูปที่ 3.141 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 10 ไม่สามารถUpdate Unitholder Bankได้หากใส่ข้อมูล Red Default Bank มากกว่า 1 ธนาคาร

```
it('SA cannot update when default redemption BankAccount more than 1 BankAccount', () => {
    // Ref#3787
    const body = {
        "unitholderId": [redacted]
        "accountId": [redacted]
        "amcCode": "ROBOTAM1",
        "subscriptionBankAccounts": [
            {
                "bankCode": "014",
                "bankBranchCode": "0001",
                "bankAccountNo": "1111111114",
                "default": true
            }
        ],
        "redemptionBankAccounts": [
            {
                "bankCode": "014",
                "bankBranchCode": "0001",
                "bankAccountNo": "2222222228",
                "default": true
            },
            {
                "bankCode": "004",
                "bankBranchCode": "0001",
                "bankAccountNo": "2222222222",
                "default": true
            }
        ],
        "approved": false
    }

    callApiUpdateUh(body).then((res) => {
        expect(res.status).equal(422)
        expect(res.body.errMsg).to.deep.equal({
            "code": "E003",
            "message": "redemptionBankAccounts.default is invalid."
        })
    })
})
```

รูปที่ 3.142 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 11 ไม่สามารถUpdate Unitholder Bankได้หากใส่ข้อมูล Sub Bank มากกว่า 5 ธนาคาร

```
Open Cypress | Set "only"
it('SA cannot update when update subscription BankAccount more than 5 BankAccount', () => {
    // Ref#3788
    const body = {
        "unitholderId": [redacted],
        "accountId": "[redacted]",
        "amcCode": "ROBOTAM1",
        "subscriptionBankAccounts": [
            {
                "bankCode": "014",
                "bankBranchCode": "0001",
                "bankAccountNo": "1111111114",
                "default": true
            },
            {
                "bankCode": "004",
                "bankBranchCode": "00000",
                "bankAccountNo": "1111111111",
                "default": false
            },
            {
                "bankCode": "006",
                "bankBranchCode": "00000",
                "bankAccountNo": "2222222222",
                "default": false
            },
            {
                "bankCode": "008",
                "bankBranchCode": "00000",
                "bankAccountNo": "3333333333",
                "default": false
            },
            {
                "bankCode": "010",
                "bankBranchCode": "00000",
                "bankAccountNo": "4444444444",
                "default": false
            }
        ]
    callApiUpdateUh(body).then((res) => {
        expect(res.status).equal(422)
        expect(res.body.errMsg).to.deep.equal({
            "code": "E003",
            "message": "subscriptionBankAccounts.limitBanks is invalid."
        })
    })
})
        v_pongsiris, 4 weeks ago • sp250 api update unitholder bank
```

รูปที่ 3.143 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 12 ไม่สามารถUpdate Unitholder Bankได้หากใส่ข้อมูล Red Bank มากกว่า 5 ธนาคาร

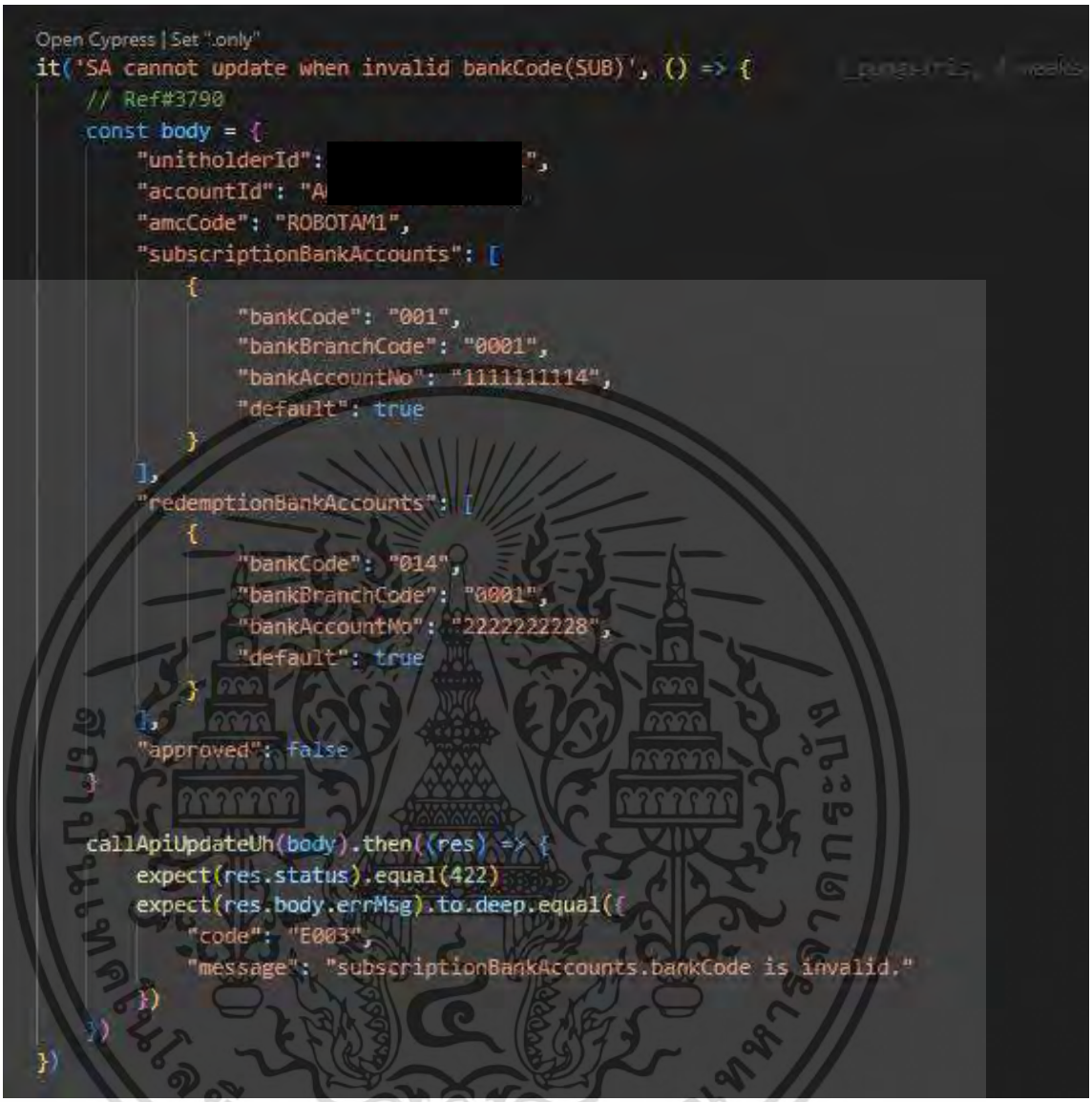
```
Open Cypress | Set ".only"
it('SA cannot update when update redemption BankAccount more than 5 BankAccount', () => {
    // Ref#3789
                "bankAccountNo": "7777777777",
                "default": false
            },
            {
                "bankCode": "008",
                "bankBranchCode": "00000",
                "bankAccountNo": "8888888888",
                "default": false
            },
            {
                "bankCode": "010",
                "bankBranchCode": "00000",
                "bankAccountNo": "9999999999",
                "default": false
            },
            {
                "bankCode": "032",
                "bankBranchCode": "00000",
                "bankAccountNo": "4444444444",
                "default": false
            }
    ],
    "approved": false
}

callApiUpdateUh(body).then((res) => {
    expect(res.status).equal(422)
    expect(res.body.errMsg).to.deep.equal({
        "code": "E003",
        "message": "redemptionBankAccounts.limitBanks is invalid."
    })
})
```

รูปที่ 3.144 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 13 ไม่สามารถUpdate Unitholder Bankได้หากใส่BankCode Sub ที่ไม่มีในระบบ



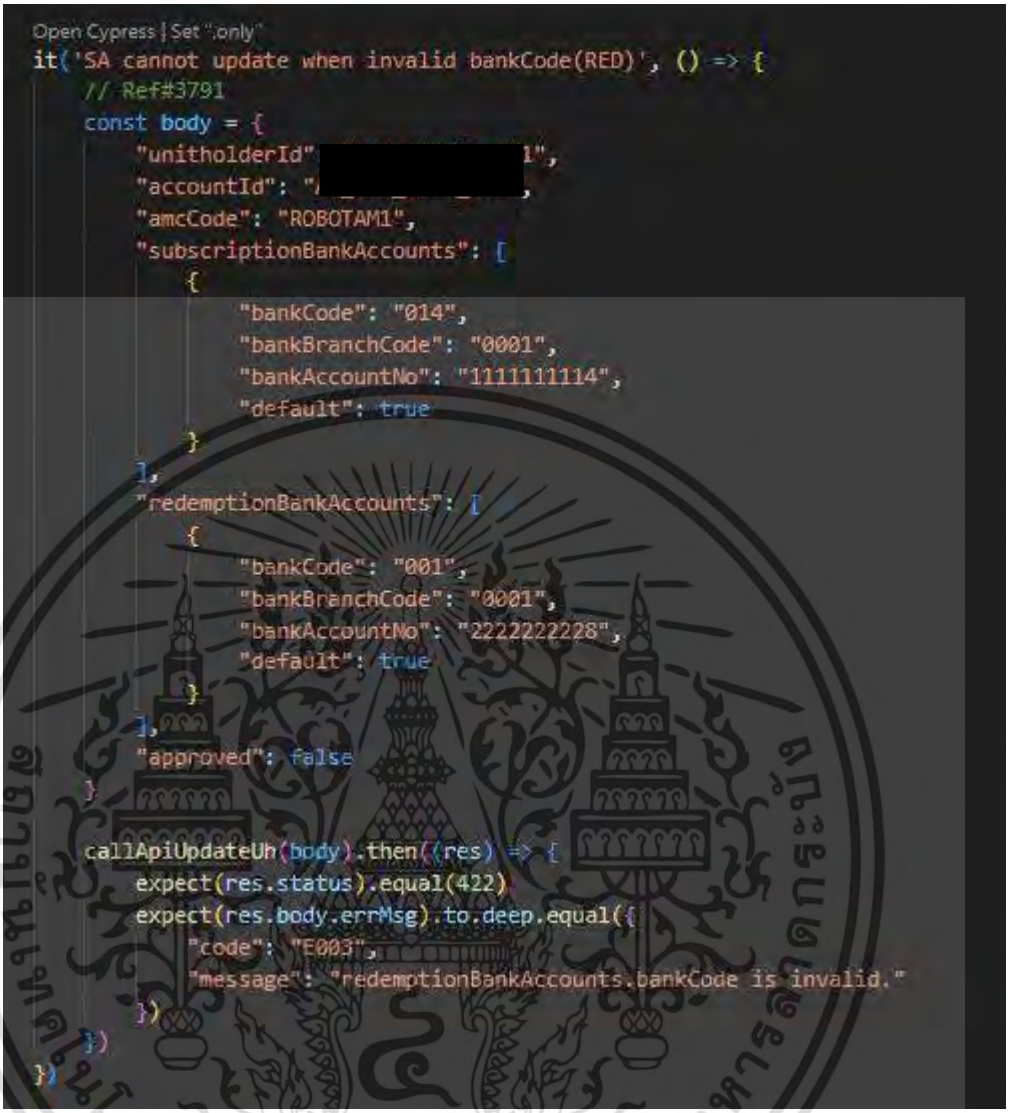
```
Open Cypress | Set "only"
it('SA cannot update when invalid bankCode(SUB)', () => {
    // Ref#3790
    const body = {
        "unitholderId": "[redacted]",
        "accountId": "A[redacted]",
        "amcCode": "ROBOTAM1",
        "subscriptionBankAccounts": [
            {
                "bankCode": "001",
                "bankBranchCode": "0001",
                "bankAccountNo": "1111111114",
                "default": true
            }
        ],
        "redemptionBankAccounts": [
            {
                "bankCode": "014",
                "bankBranchCode": "0001",
                "bankAccountNo": "2222222228",
                "default": true
            }
        ],
        "approved": false
    }

    callApiUpdateUh(body).then((res) => {
        expect(res.status).equal(422)
        expect(res.body.errMsg).to.deep.equal({
            "code": "E003",
            "message": "subscriptionBankAccounts.bankCode is invalid."
        })
    })
})
```

รูปที่ 3.145 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 14 ไม่สามารถUpdate Unitholder Bankได้หากใส่BankCode Red ที่ไม่มีในระบบ



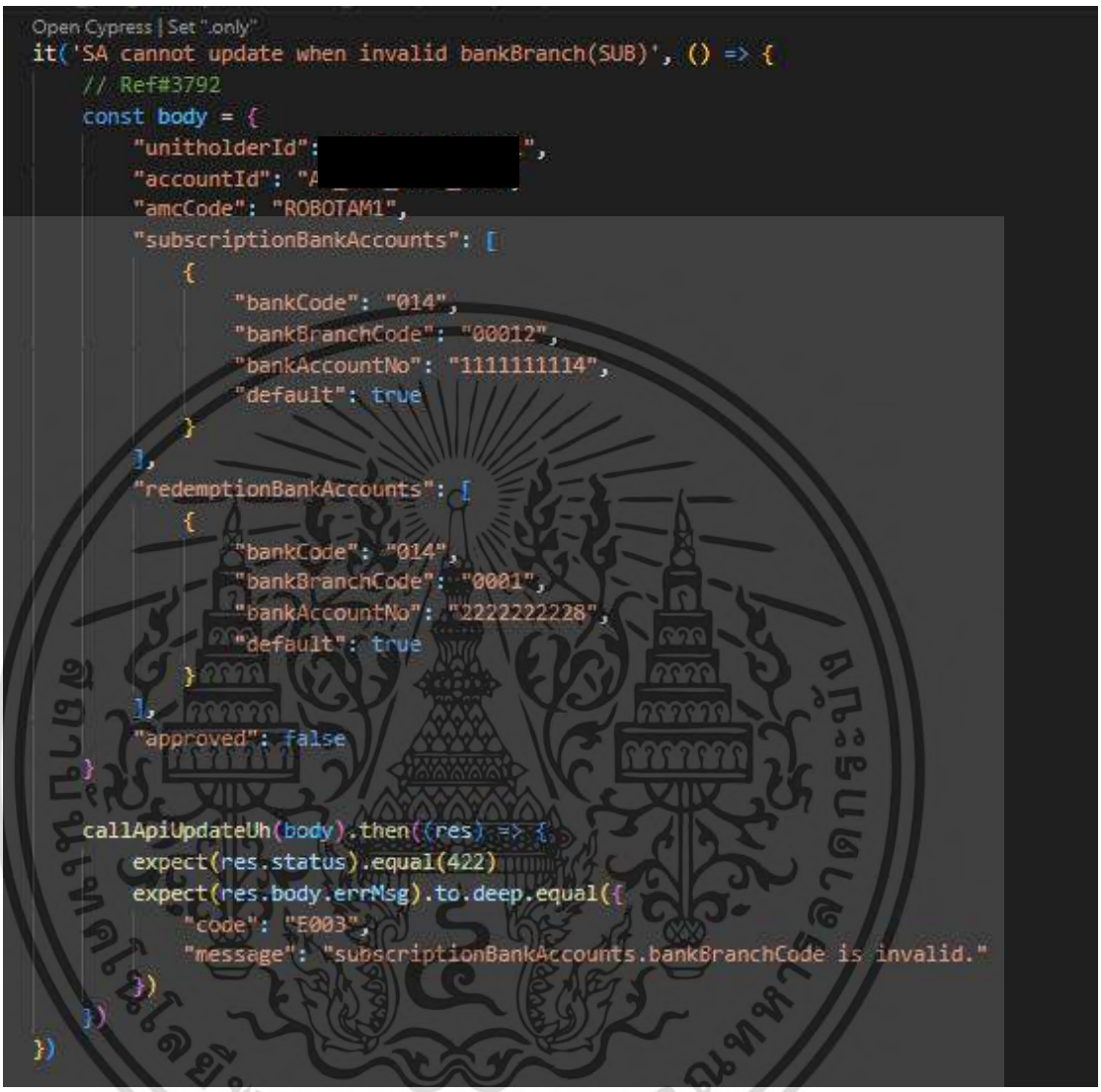
```
Open Cypress | Set ".only"
it('SA cannot update when invalid bankCode(RED)', () => {
    // Ref#3791
    const body = {
        "unitholderId": "[redacted]1",
        "accountId": "[redacted]",
        "amcCode": "ROBOTAM1",
        "subscriptionBankAccounts": [
            {
                "bankCode": "014",
                "bankBranchCode": "0001",
                "bankAccountNo": "1111111114",
                "default": true
            }
        ],
        "redemptionBankAccounts": [
            {
                "bankCode": "001",
                "bankBranchCode": "0001",
                "bankAccountNo": "2222222228",
                "default": true
            }
        ],
        "approved": false
    }

    callApiUpdateUh(body).then((res) => {
        expect(res.status).equal(422)
        expect(res.body.errMsg).to.deep.equal({
            "code": "E003",
            "message": "redemptionBankAccounts.bankCode is invalid."
        })
    })
})
```

รูปที่ 3.146 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 15 ไม่สามารถUpdate Unitholder Bankได้หากใส่BankBranch Sub ที่ไม่มีในระบบ

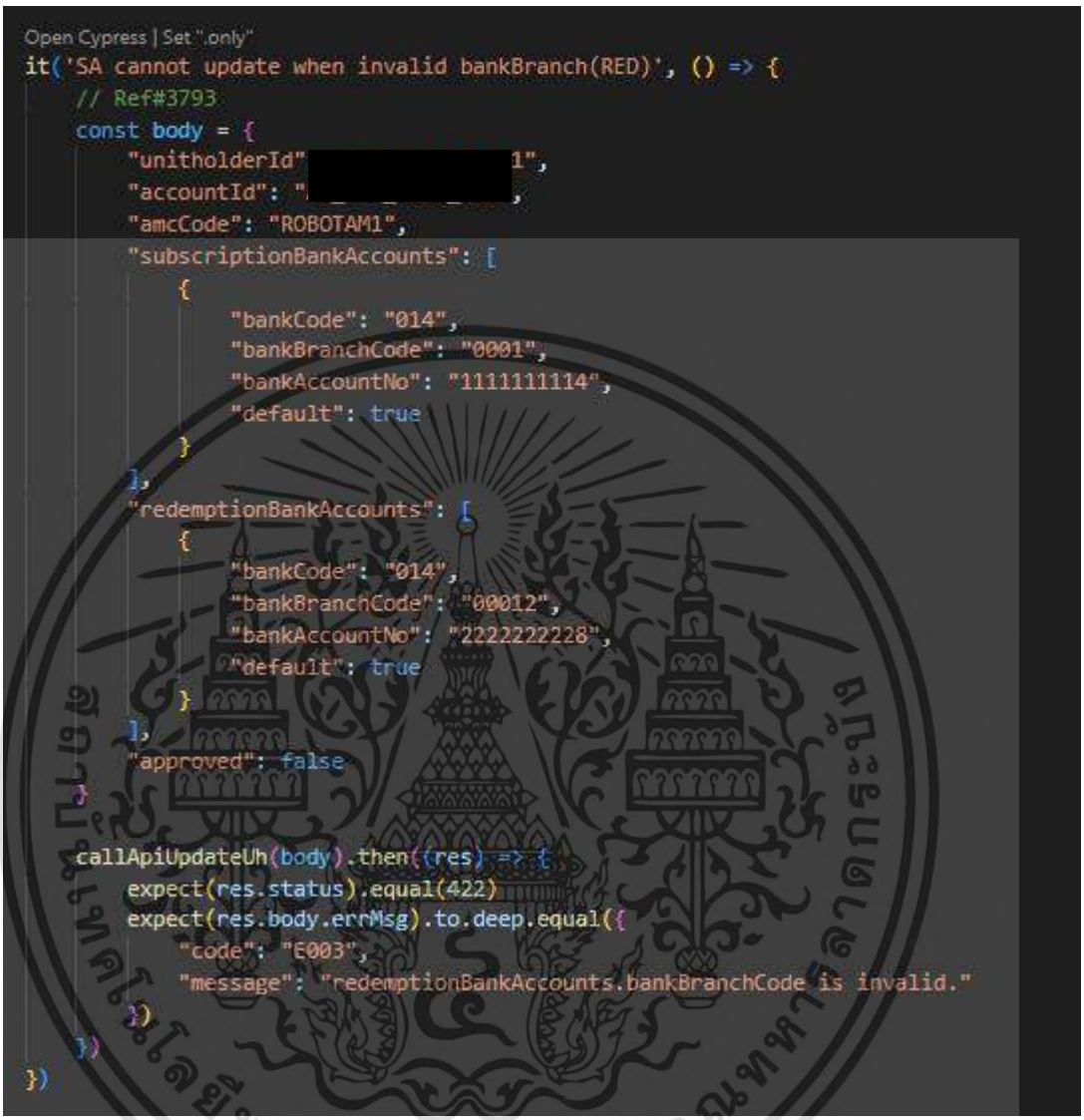


```
Open Cypress | Set ".only"
it('SA cannot update when invalid bankBranch(SUB)', () => {
    // Ref#3792
    const body = {
        "unitholderId": "[redacted]",
        "accountId": "[redacted]",
        "amcCode": "ROBOTAM1",
        "subscriptionBankAccounts": [
            {
                "bankCode": "014",
                "bankBranchCode": "00012",
                "bankAccountNo": "1111111114",
                "default": true
            }
        ],
        "redemptionBankAccounts": [
            {
                "bankCode": "014",
                "bankBranchCode": "0001",
                "bankAccountNo": "2222222228",
                "default": true
            }
        ],
        "approved": false
    }
    callApiUpdateUh(body).then((res) => {
        expect(res.status).equal(422)
        expect(res.body.errMsg).to.deep.equal({
            "code": "E003",
            "message": "subscriptionBankAccounts.bankBranchCode is invalid."
        })
    })
})
```

รูปที่ 3.147 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 16 ไม่สามารถUpdate Unitholder Bankได้หากใส่BankBranch Red ที่ไม่มีในระบบ



```
Open Cypress | Set ".only"
it('SA cannot update when invalid bankBranch(RED)', () => {
    // Ref#3793
    const body = {
        "unitholderId"                    1",
        "accountId": "                    ",
        "amcCode": "ROBOTAM1",
        "subscriptionBankAccounts": [
            {
                "bankCode": "014",
                "bankBranchCode": "0001",
                "bankAccountNo": "1111111114",
                "default": true
            }
        ],
        "redemptionBankAccounts": [
            {
                "bankCode": "014",
                "bankBranchCode": "00012",
                "bankAccountNo": "2222222228",
                "default": true
            }
        ],
        "approved": false
    }
    callApiUpdateUh(body).then((res) => {
        expect(res.status).equal(422)
        expect(res.body.errMsg).to.deep.equal({
            "code": "E003",
            "message": "redemptionBankAccounts.bankBranchCode is invalid."
        })
    })
})
```

รูปที่ 3.148 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีทดสอบที่ 17 ไม่สามารถUpdate Unitholder Bankได้หากใส่Unitholder ที่ไม่มีในระบบ

```
Open Cypress | Set ".only"
it('SA cannot update unitholder when unitholder not found', () => {
    // Ref#3794
    const body = {
        "unitholderId": "XXXX",
        "accountId": "[redacted]",
        "amcCode": "ROBOTAM1",
        "subscriptionBankAccounts": [
            {
                "bankCode": "014",
                "bankBranchCode": "0001",
                "bankAccountNo": "1111111114",
                "default": true
            }
        ],
        "redemptionBankAccounts": [
            {
                "bankCode": "014",
                "bankBranchCode": "0001",
                "bankAccountNo": "2222222228",
                "default": true
            }
        ],
        "approved": false
    }
    callApiUpdateUh(body).then((res) => {
        expect(res.status).equal(422)
        expect(res.body.errMsg).to.deep.equal({
            "code": "E103",
            "message": "Unitholder ID cannot be found."
        })
    })
})
```

รูปที่ 3.149 Script ในการทำ Automated Test API ตัวอย่างที่ 3 กรณีที่ 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 วิธีการบำรุงและดูแลรักษา

การจัดทำ Automated Test ของระบบฟันด์คอนเน็กต์จะมี Job ที่ใช้ในการรัน Automated Test ในทุกๆวัน ซึ่งจะ Dashboard ที่ใช้ในการแสดงผลลัพธ์ของ Automated Test ในแต่ละไฟล์ เพื่อสามารถรับรู้ถึงข้อผิดพลาดที่เกิดขึ้นหลังจากการพัฒนาซอฟต์แวร์ในแต่ละ Sprint แล้วนำข้อผิดพลาดนั้นแจ้งไปยังผู้พัฒนาซอฟต์แวร์ให้ทำการแก้ไข ด้วยเหตุนี้จึงต้องมีการบำรุงและดูแลรักษา Automated Test ที่ใช้ทดสอบระบบอยู่อย่างสม่ำเสมอ โดยปัญหาที่พบเจอจะมีทั้งหมดดังนี้

3.3.1 ข้อมูลสำหรับการทดสอบ

ปัญหาที่พบเจอได้บ่อยอันเกิดจาก Automated Test ในแต่ละไฟล์มีการใช้ Data ในการทดสอบชุดเดียวกัน แล้วไม่มีการอัพเดตข้อมูลหลังจากการทดสอบ หรือมีการปรับเปลี่ยนข้อมูลแล้วส่งผลกระทบกับการทดสอบอื่นๆ

ปัญหาจะเกิดขึ้นเมื่อ

- เตรียม Data Test ไม่ดี
- นำ Data เก่ามาใช้แล้วไม่คำนึงถึงผลกระทบที่เกิดขึ้น
- จัดเรียงลำดับ Test case ไม่ดีในไฟล์ Automated Test

วิธีการแก้ปัญหา

- จัดทำ Data ขึ้นใหม่
- ตรวจสอบ Data เก่าที่นำมาใช้ก่อนเสมอ
- หลีกเลี่ยงการลบและแก้ไขข้อมูลที่ใช้ร่วมกันกับไฟล์อื่นๆ

3.3.2 Flankly Test

การทดสอบแบบ Automated Test ในแต่ละไฟล์ที่บางครั้งการทดสอบผ่านบางครั้งการทดสอบไม่ผ่านทำให้การทดสอบนั้นไม่น่าเชื่อถือเท่าที่ควร

ปัญหาจะเกิดขึ้นเมื่อ

- จำนวน Case ที่ทดสอบเยอะเกินไป
- มี Element พวก Modal

วิธีการแก้ปัญหา

- ลดจำนวน Case ที่ทดสอบในแต่ละไฟล์
- เพิ่มคำสั่ง Wait() หรือ Then() เพื่อรอElementที่ต้องการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 กระทบจากการเพิ่มเติมหรือแก้ไขระบบ

เป็นเหตุผลหลักที่ต้องการจากการนำ Automated Test มาใช้ในการทดสอบระบบ เนื่องจากช่วยในการทดสอบแบบ Regression Test เพื่อหาข้อผิดพลาดที่เกิดขึ้นจากการพัฒนาซอฟต์แวร์ ซึ่ง Tester ต้องทำการตรวจสอบว่า Case ที่ไม่ผ่านนั้นเกิดข้อผิดพลาดจริงหรือไม่ก่อนนำไปแจ้งผู้พัฒนาเพื่อทำการแก้ไข

3.3.4 Localhost ผ่านแต่ Serverไม่ผ่าน

เป็นปัญหาที่พบได้แต่ไม่มากเนื่องจากใน Automated Test Script ที่จัดทำสามารถทดสอบบนเครื่องของแต่ละบุคคลได้แต่ไม่สามารถทดสอบบนเครื่องServerได้

3.4 การย้าย Automated Test Tools

ปัจจุบันระบบฟันด์คอนเน็กต์กำลังจะทำการย้าย Automated Test Tools จากการใช้ Robot Framework ไปยัง Cypress เนื่องจาก Cypress ในด้านของประสิทธิภาพในการทำงานดีกว่า Robot Feamework ทั้งความเร็วในการทดสอบ สภาพแวดล้อมที่ใช้งาน รวมถึงมีระบบต่างๆที่เข้ามาช่วยเหลือในการทดสอบให้ง่ายขึ้นมากกว่า Robot Framework โดยโครงสร้างและคำสั่งที่คล้ายกันมีดังนี้

ตารางที่ 3.14 ตารางเปรียบเทียบคำสั่งการเขียน Automated Test ของ Robot และ Cypress

คำสั่ง/โครงสร้าง	Robot	Cypresss
ส่วนที่ใส่คำสั่งที่ต้องการทำก่อนเริ่มทดสอบ	Test Set up	Before
ส่วนที่ใส่คำสั่งที่ต้องการทำหลังเริ่มทดสอบ	Test Teardown	After
กรอกข้อมูล	Input	.type()
เลือกDropdown	Select	.select()
คลิก	Click	.click()
ตรวจสอบ	Contain	.should(), .contain()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการดำเนินงานสหกิจศึกษาและการอภิปรายผล

จากที่ผู้จัดทำได้เข้าร่วมสหกิจศึกษาในหัวข้อการทดสอบอัตโนมัติสำหรับระบบฟันด์คอนเน็กต์ ซึ่งบทนี้จะเป็นการแสดงผลการทดสอบระบบฟันด์คอนเน็กต์ตามกรณีทดสอบที่ได้ออกแบบไว้ ซึ่งจะแสดงกรณีทดสอบ ตัวอย่างหน้าจอผลลัพธ์การทดสอบ ผลการทดสอบระบบในรูปแบบ Manual Test และผลลัพธ์การรันทดสอบ Automated Test Script ซึ่งประกอบไปด้วยผลการทดสอบดังนี้

4.1 ผลการทดสอบ Functional Test แบบ Manual Test

ตัวอย่างที่ 1 การทดสอบ Functional Test แบบ Manual Test

ตารางที่ 4.1 ผลการทดสอบ Functional Test แบบ Manual Test ตัวอย่างที่ 1

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result	Test Result
Code Management - SA							
1	Web	SA	Application Management	Code Management	- Upload Code(ก่อนเวลา 16.00) ตรวจสอบError	- Uploadไม่สำเร็จ - แสดงข้อความ “Must be uploaded after 16.00”	Pass
2	Web	SA	Application Management	Code Management	- Upload Code(หลังเวลา 16.00)	- Upload Code ได้สำเร็จ	Pass
3	Web	SA	Application Management	Code Management	- Upload Code ซ้ำที่มีในระบบแล้ว	- Uploadไม่สำเร็จ - แสดงข้อความ “Duplicate Code”	Pass
4	Web	SA	Application Management	Code Management	- Upload Code ที่ไม่ได้ Config Code Mapping ไว้	- Uploadไม่สำเร็จ - แสดงข้อความ “Incorrect Code Format”	Pass
5	Web	SA	Application Manage	Code Manage	- Upload Code ไม่ครบ 9	- Uploadไม่สำเร็จ - แสดงข้อความ	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			ment	ment	ตัวอักษร	“Incorrect Code Format”	
6	Web	SA	Application Manage ment	Code Manage ment	- Upload Code เกิน 9 ตัวอักษร	- Uploadไม่สำเร็จ - แสดงข้อความ “Incorrect Code Format”	Pass
7	Web	SA	Application Manage ment	Code Manage ment	- Upload Code อักษรตัวเล็ก	- Uploadไม่สำเร็จ - แสดง Point_code เป็นอักษรตัวใหญ่	Pass
8	Web	SA	Application Manage ment	Code Manage ment	- Upload Code ที่หมดอายุไปแล้ว	- Uploadไม่สำเร็จ - แสดงข้อความ “Incorrect Expiry Date”	Pass
9	Web	SA	Application Manage ment	Code Manage ment	- Upload Code ที่Expiry Date ตรงกับ วันที่Upload	- Upload Code ได้สำเร็จ	Pass
10	Web	SA	Application Manage ment	Code Manage ment	- Upload Code ที่มีคนละPrefix-Codeในไฟล์เดียวกัน	- Upload Code ได้สำเร็จ	Pass
11	Web	SA	Application Manage ment	Code Manage ment	- ตรวจสอบหัวข้อตารางที่แสดงบนเมนู Code Management	- แสดงหัวข้อตารางได้ถูกต้อง	Pass
12	Web	SA	Application Manage ment	Code Manage ment	- ตรวจสอบข้อมูล Status เมื่อนำPoint Codeไปใช้ทำรายการ	- StatusของPoint Code นั้นถูกปรับเป็น Redeemed ได้ถูกต้อง	Pass
13	Web	SA	Application Manage ment	Code Manage ment	- ตรวจสอบข้อมูล Redeemed Dateเมื่อนำ	-Redeemed Date ของPoint Code แสดงวันที่ Codeถูกใช้งานได้	Pass

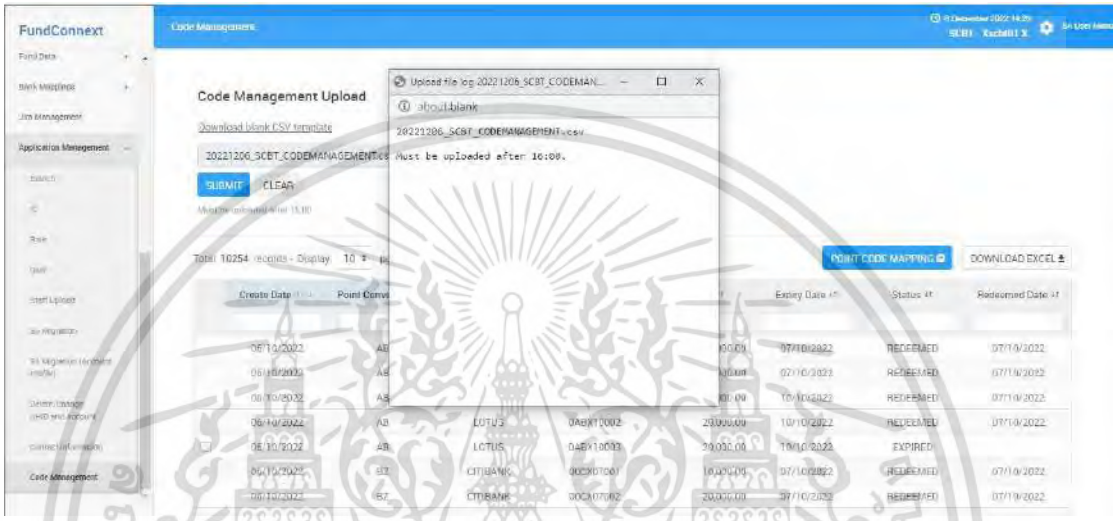
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือแสวงหาประโยชน์ทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

					Point Codeไปใช้ทำรายการ	ถูกต้อง	
14	Web	SA	Application Management	Code Management	- ตรวจสอบ Checkbox สำหรับลบข้อมูล Point Code	- แสดง Checkbox หน้าPoint Codeที่มี Status เป็น Available และ Expired - ไม่แสดง Checkbox หน้าPoint Codeที่มี Status เป็น Redeemed	Pass
15	Web	SA	Application Management	Code Management	- ตรวจสอบการลบข้อมูล Point Code	- สามารถลบ Point Codeที่ต้องการได้ถูกต้อง - แสดง Pop up เพื่อ Confirm การ Delete Point Code	Pass
16	Web	SA	Application Management	Code Management	- ตรวจสอบ Modal แสดงผล Point Code Mapping	- กดปุ่ม Point Code แล้วแสดง Modal ข้อมูล Point Code Mapping ได้ถูกต้อง	Pass
17	Web	SA	Application Management	Code Management	- ตรวจสอบFile Excel เมื่อ Filterข้อมูลในตารางแล้วทำการ Download File	- แสดงข้อมูลใน File Excel เฉพาะที่ Filter ได้ถูกต้อง	Pass
Code Management – TSD							
18	Web	TSD	File Management	Code Management	- ตรวจสอบหัวข้อตารางที่แสดงบนเมนู Code Management ของ TSD	- แสดงหัวข้อตารางได้ถูกต้อง	Pass
19	Web	TSD	File Management	Code Manage	- ตรวจสอบการแสดงผลข้อมูลใน	- แสดงข้อมูลในตารางได้ถูกต้อง	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

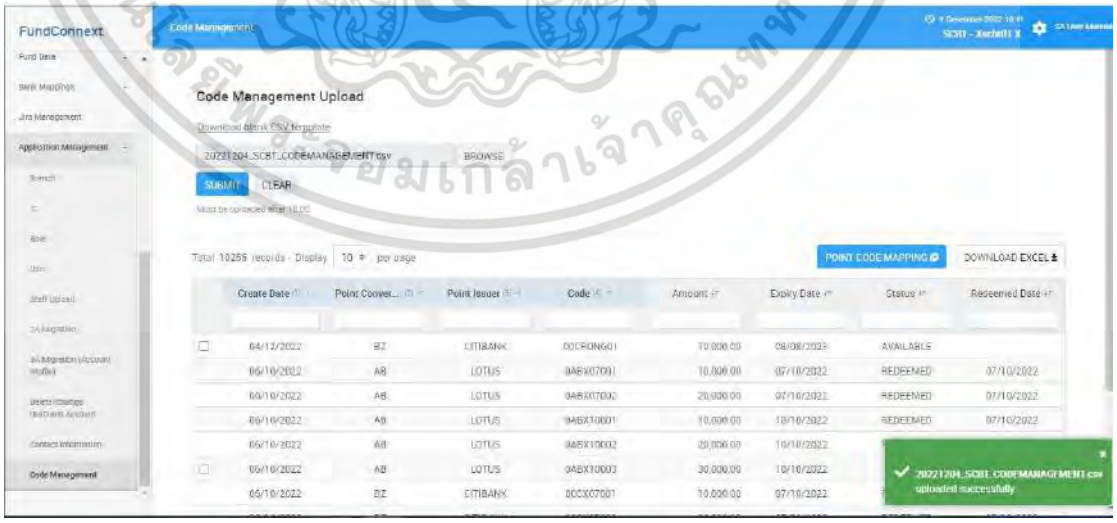
				ment	ตารางเมนู Code Management ของ TSD		

ผลการทดสอบกรณีที่ 1 Upload Point Code ก่อนเวลา 16.00 น. จะแสดง Error Modal “Must be uploaded after 16.00”



รูปที่ 4.1 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 1

ผลการทดสอบกรณีที่ 2 Upload Point Code หลังเวลา 16.00 น. ด้วย Code : 00CPONG01

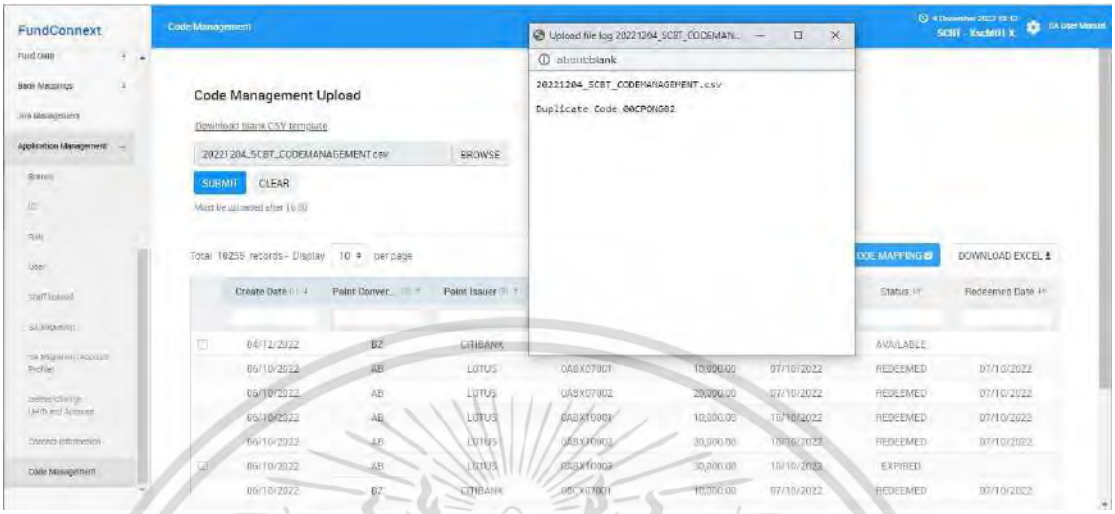


รูปที่ 4.2 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 3 Upload Point Code ที่ซ้ำกับในระบบ

Upload ด้วย Code : 00CPONG02 สองครั้ง



รูปที่ 4.3 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 3

ผลการทดสอบกรณีที่ 4 Upload Point Code ที่ไม่มีConfig BankMapping ไว้

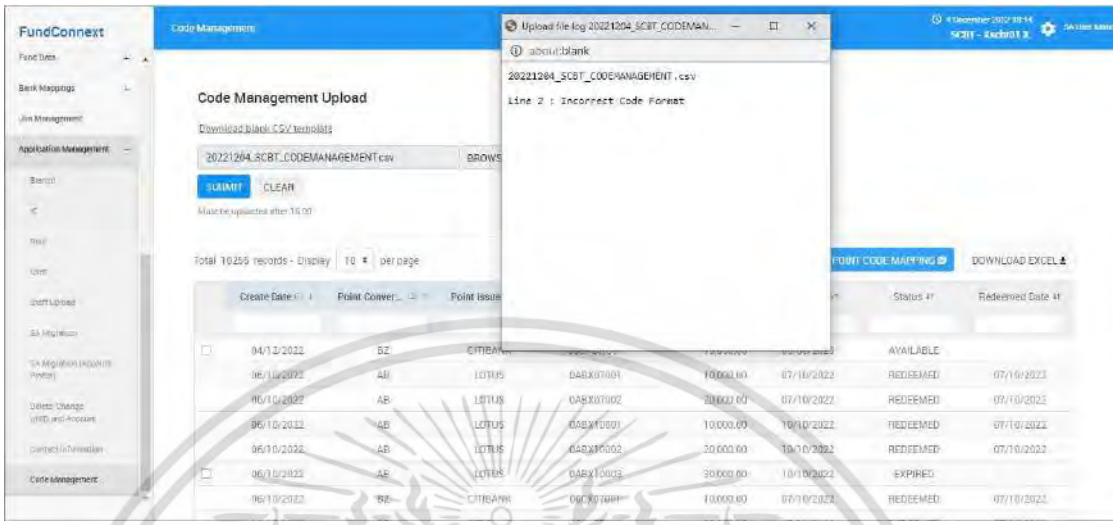
Upload ด้วย Code : 0CCPONG03

รูปที่ 4.4 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 5 Upload Point Code ไม่ครบ 9 ตัวอักษร

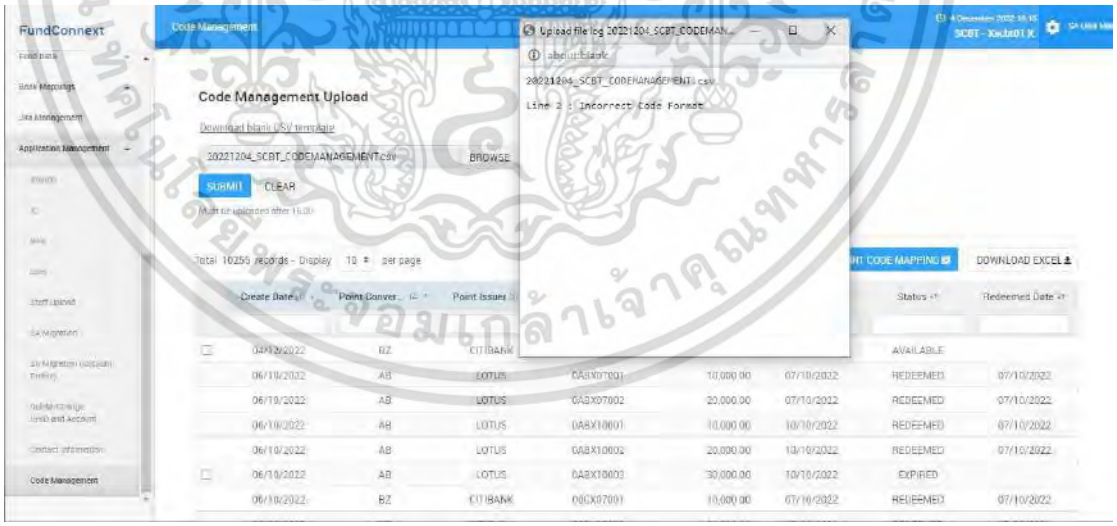
Upload ด้วย Code : 00CPONG



รูปที่ 4.5 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 5

ผลการทดสอบกรณีที่ 6 Upload Point Code เกิน 9 ตัวอักษร

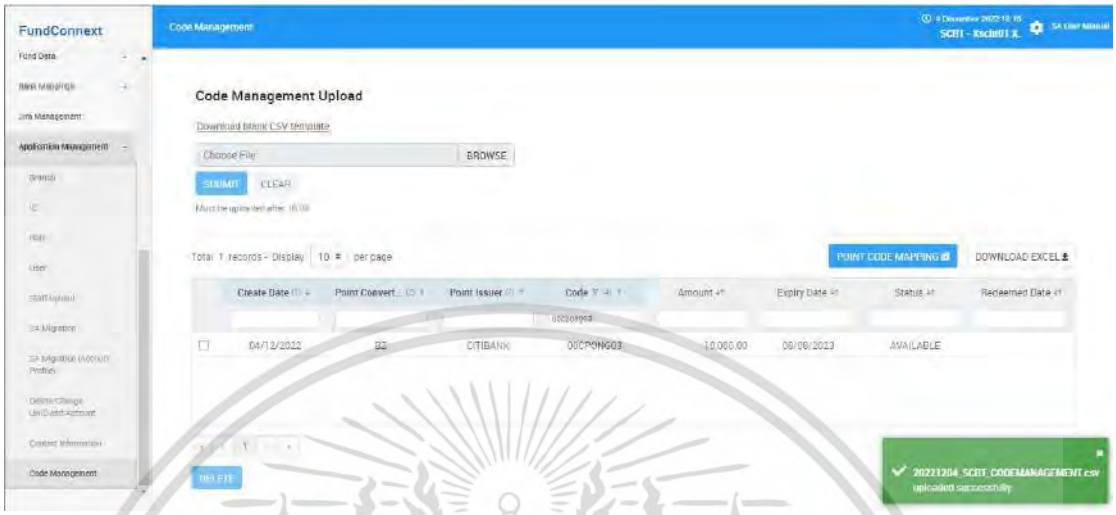
Upload ด้วย Code : 00CPONG001



รูปที่ 4.6 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 6

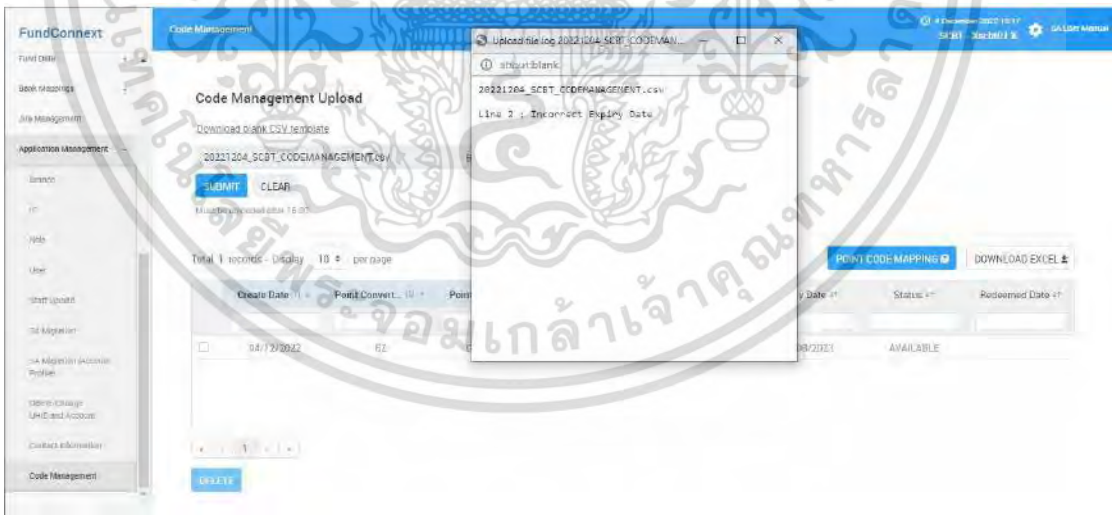
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 7 Upload Point Code ด้วยตัวอักษรตัวเล็กระบบจะทำการปรับให้เป็นตัวใหญ่ Upload ด้วย Code : 00Cpong03



รูปที่ 4.7 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 7

ผลการทดสอบกรณีที่ 8 Upload Point Code ด้วยExpiery Date ที่หมดอายุไปแล้ว Upload ด้วย Code : 00CPONG04 Expiry Date : 12/5/2021

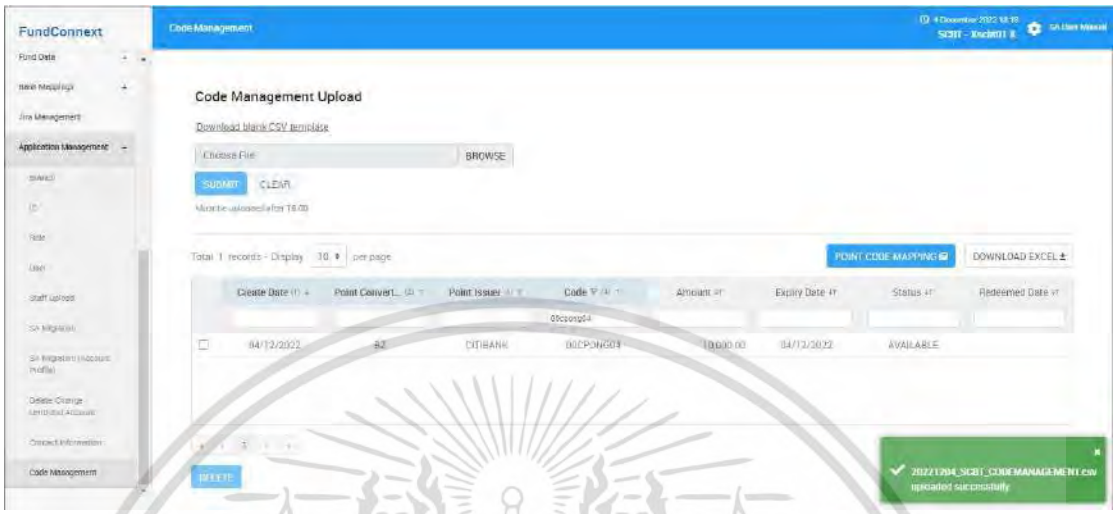


รูปที่ 4.8 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 9 Upload Point Code ด้วยExpiery Date ที่ตรงกับวันUpload

Upload ด้วย Code : 00CPONG05 Expiry Date : Upload Date



รูปที่ 4.9 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 9

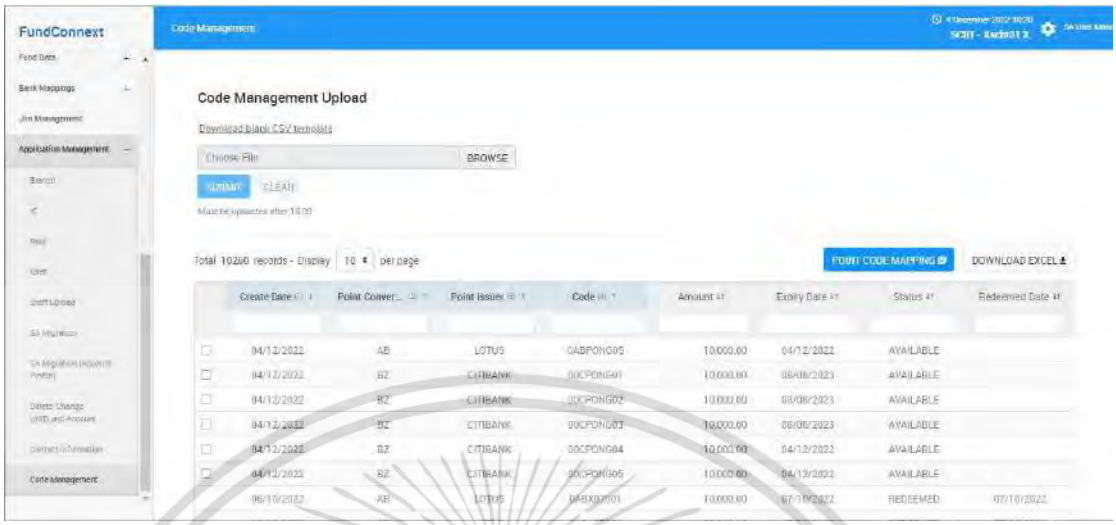
ผลการทดสอบกรณีที่ 10 Upload Point Code ด้วยคนละ Prefix-Codeในไฟล์เดียวกัน

Upload ด้วย Code : 00CPONG05 , 0ABPONG05

รูปที่ 4.10 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 11 ตรวจสอบหัวข้อตารางหน้าจอ Code Management ของ SA



รูปที่ 4.11 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 11

ผลการทดสอบกรณีที่ 12 ตรวจสอบStatusเมื่อ Point Code ถูกนำไปทำรายการ

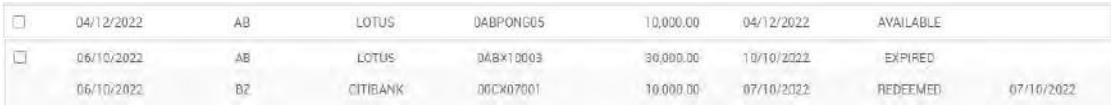
รูปที่ 4.12 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 12

ผลการทดสอบกรณีที่ 13 ตรวจสอบRedeemed Date เป็นวันที่ Code ถูกนำไปใช้งาน

รูปที่ 4.13 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

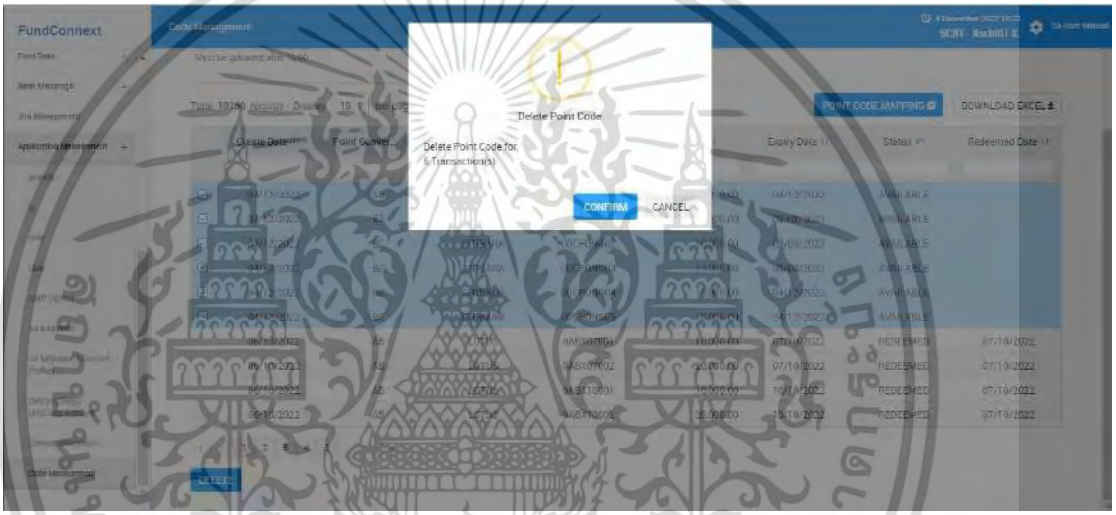
ผลการทดสอบกรณีที่ 14 แสดง Check box สำหรับDelete เมื่อ Status เป็น Available และ Expired



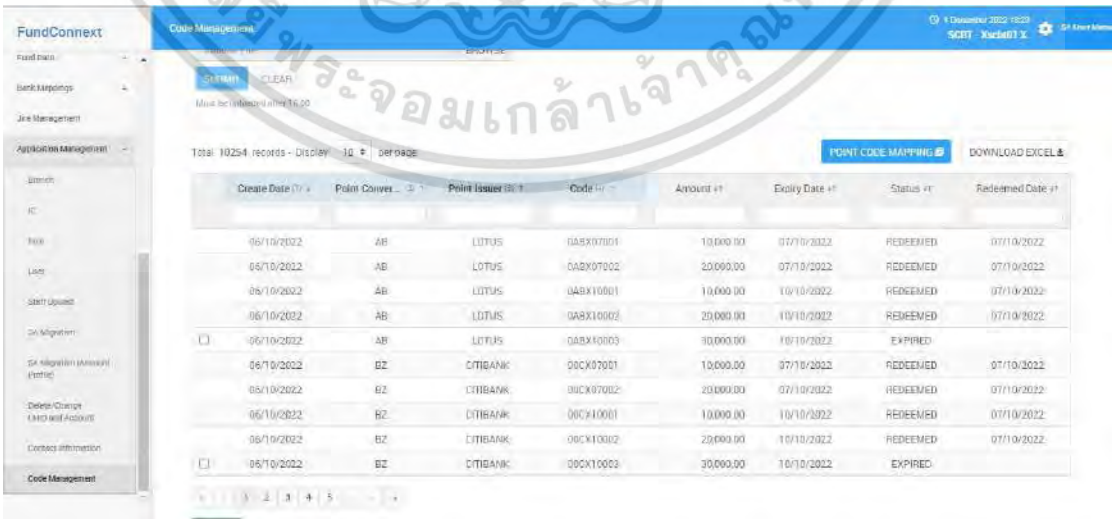
☐	04/12/2022	AB	LOTUS	0ABPONG05	10,000.00	04/12/2022	AVAILABLE	
☐	06/10/2022	AB	LOTUS	0ABX10003	30,000.00	10/10/2022	EXPIRED	
	06/10/2022	BZ	CITIBANK	00CX07001	10,000.00	07/10/2022	REDEEMED	07/10/2022

รูปที่ 4.14 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 14

ผลการทดสอบกรณีที่ 15 ตรวจสอบModal สำหรับลบ Point Code



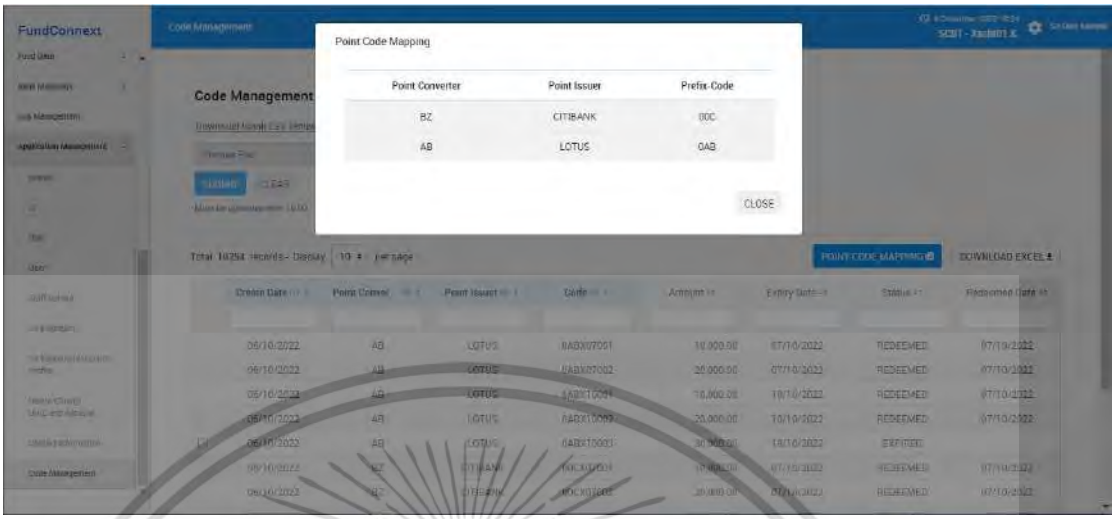
รูปที่ 4.15 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 15 (1)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.16 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 15 (2)

ผลการทดสอบกรณีที่ 16 ตรวจสอบModal ที่แสดงข้อมูลPoint Code Mapping



รูปที่ 4.17 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 16

ผลการทดสอบกรณีที่ 17 ตรวจสอบFile Excel โดยFilterด้วยTest

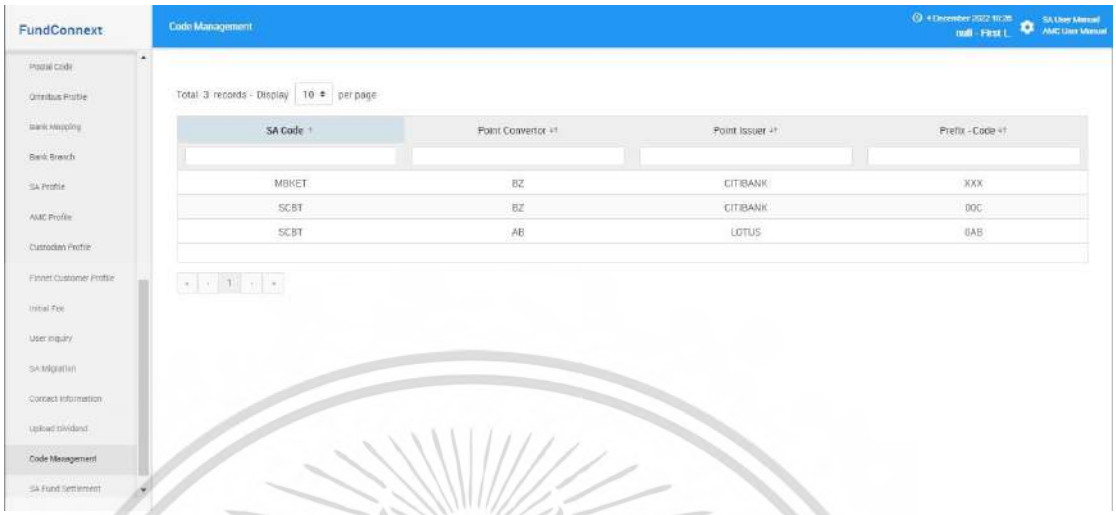
รูปที่ 4.18 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 17

	A	B	C	D	E	F	G	H	I
1	Create Dat	Point Conv	Point Issue	Code	Amount	Expiry Dat	Status	Redeemed Date	
2	08/08/202	BZ	CITIBANK	00CTEST0	10000	08/08/202	REDEEMEI	18/08/2022	
3	08/08/202	BZ	CITIBANK	00CTEST0	10000	08/08/202	AVAILABLE		
4	08/08/202	BZ	CITIBANK	00CTEST0	10000	08/08/202	AVAILABLE		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.19 ตัวอย่างไฟล์ผลลัพธ์การทดสอบตัวอย่างที่ 1 กรณีที่ 17

ผลการทดสอบกรณีที่ 18 ตรวจสอบหัวข้อตารางหน้าจอ Code Management ของ TSD



รูปที่ 4.20 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 18

ผลการทดสอบกรณีที่ 19 ตรวจสอบข้อมูลในตารางหน้าจอ Code Management ของ TSD

รูปที่ 4.21 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 1 กรณีที่ 19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2 การทดสอบ Functional Test แบบ Manual Test

ตารางที่ 4.2 ผลการทดสอบ Functional Test แบบ Manual Test ตัวอย่างที่ 2

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result	Test Result
SA Migrate UH							
1	Web	SA	Application Management	Change Unitholder to Another Account	- SA Migrate UH : SEG เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางไม่ติด Lock	- สามารถ Change Unitholder to Another Account ได้ถูกต้อง - มีการเก็บ Log - Orderทั้งหมดถูก Updateให้อยู่ภายใต้ Accountใหม่ถูกต้อง	Pass
2	Web	SA	Application Management	Change Unitholder to Another Account	- SA Migrate UH : OMN เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางไม่ติด Lock	- สามารถ Change Unitholder to Another Account ได้ถูกต้อง - มีการเก็บ Log - Orderทั้งหมดถูก Updateให้อยู่ภายใต้ Accountใหม่ถูกต้อง	Pass
3	Web	SA	Application Management	Change Unitholder to Another Account	- SA Migrate UH : SEG เข้าระบบโดยที่ Account ต้นทางติด Lock Account ปลายทางไม่ติด Lock	- Cannot change Account No. due to pending account Status.	Pass
4	Web	SA	Application Manage	Change Unitholder to	- SA Migrate UH : OMN เข้า	- Cannot change Account No. due to pending account	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือแสวงหาผลประโยชน์ทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			ment	Another Account	ระบบโดยที่ Account ต้นทางติด Lock Account ปลายทางไม่ติด Lock	Status.	
5	Web	SA	Application Management	Change Unitholder to Another Account	- SA Migrate UH : SEG เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางติด Lock	- Cannot change Account No. due to pending account Status.	Pass
6	Web	SA	Application Management	Change Unitholder to Another Account	- SA Migrate UH : OMN เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางติด Lock	- Cannot change Account No. due to pending account Status.	Pass
SA Send Order							
7	Web	SA	Application Management	Change Unitholder to Another Account	- SA Send Order : SEG เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางไม่ติด Lock	- สามารถ Change Unitholder to Another Account ได้ถูกต้อง - มีการเก็บ Log - Orderทั้งหมดถูก Updateให้อยู่ภายใต้ Accountใหม่ถูกต้อง	Pass
8	Web	SA	Application Manage	Change Unithold	- SA Send Order : OMN	- สามารถ Change Unitholder to	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			ment	er to Another Account	เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางไม่ติด Lock	Another Account ได้ถูกต้อง - มีการเก็บ Log - Orderทั้งหมดถูก Updateให้อยู่ภายใต้ Accountใหม่ถูกต้อง	
9	Web	SA	Application Management	Change Unitholder to Another Account	- SA Send Order : SEG เข้าระบบโดยที่ Account ต้นทางติด Lock Account ปลายทางติดไม่ Lock	- Cannot change Account No. due to pending account Status.	Pass
10	Web	SA	Application Management	Change Unitholder to Another Account	- SA Send Order : OMN เข้าระบบโดยที่ Account ต้นทางติด Lock Account ปลายทางติดไม่ Lock	- Cannot change Account No. due to pending account Status.	Pass
11	Web	SA	Application Management	Change Unitholder to Another Account	- SA Send Order : SEG เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางติด Lock	- Cannot change Account No. due to pending account Status.	Pass
12	Web	SA	Application Management	Change Unitholder to	- SA Send Order : OMN เข้าระบบโดยที่	- Cannot change Account No. due to pending account	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

				Another Account	Account ต้นทางไม่ติด Lock Account ปลายทางติด Lock	Status.	
SA MoveSink							
13	Web	SA	Application Management	Change Unitholder to Another Account	- SA MoveSink : SEG เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางไม่ติด Lock	- สามารถ Change Unitholder to Another Account ได้ถูกต้อง - มีการเก็บ Log - Orderทั้งหมดถูก Updateให้อยู่ภายใต้ Accountใหม่ถูกต้อง	Pass
14	Web	SA	Application Management	Change Unitholder to Another Account	- SA MoveSink : OMN เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางไม่ติด Lock	- สามารถ Change Unitholder to Another Account ได้ถูกต้อง - มีการเก็บ Log - Orderทั้งหมดถูก Updateให้อยู่ภายใต้ Accountใหม่ถูกต้อง	Pass
15	Web	SA	Application Management	Change Unitholder to Another Account	- SA MoveSink : SEG เข้าระบบโดยที่ Account ต้นทางติด Lock Account ปลายทางติดไม่ Lock	- Cannot change Account No. due to pending account Status.	Pass
16	Web	SA	Application Management	Change Unitholder to	- SA MoveSink : OMN เข้าระบบโดยที่	- Cannot change Account No. due to pending account	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

				Another Account	Account ต้นทางติด Lock Account ปลายทางติดไม่ Lock	Status.	
17	Web	SA	Application Management	Change Unitholder to Another Account	- SA MoveSink : SEG เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางติด Lock	- Cannot change Account No. due to pending account Status.	Pass
18	Web	SA	Application Management	Change Unitholder to Another Account	- SA MoveSink : OMN เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางติด Lock	- Cannot change Account No. due to pending account Status.	Pass
SA Request API UH							
19	Web	SA	Application Management	Change Unitholder to Another Account	- SA Request API UH : SEG เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางไม่ติด Lock	- สามารถ Change Unitholder to Another Account ได้ถูกต้อง - มีการเก็บ Log - Orderทั้งหมดถูก Updateให้อยู่ภายใต้ Accountใหม่ถูกต้อง	Pass
20	Web	SA	Application Management	Change Unitholder to	- SA Request API UH : OMN เข้าระบบโดยที่	- สามารถ Change Unitholder to Another Account ได้	Pass

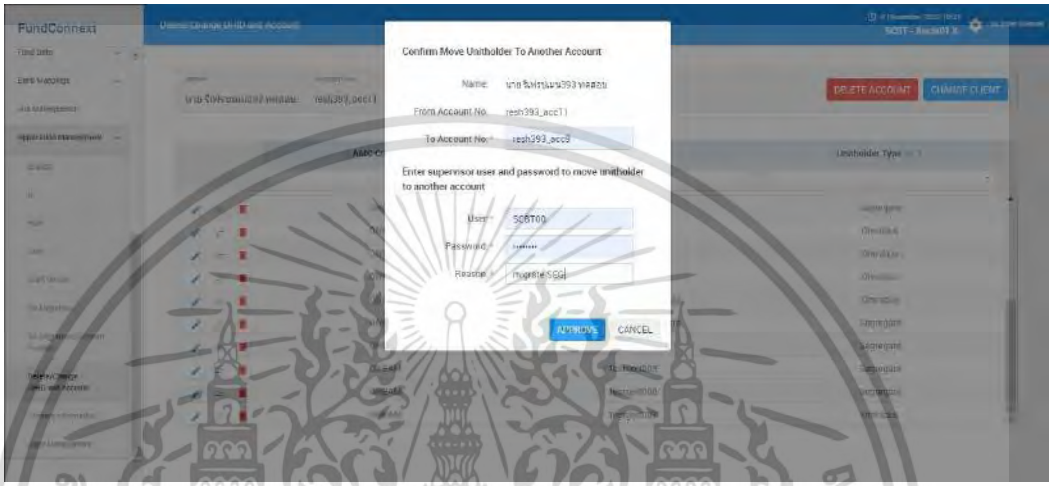
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

				Another Account	Account ต้นทางไม่ติด Lock Account ปลายทางไม่ติด Lock	ถูกต้อง - มีการเก็บ Log - Orderทั้งหมดถูก Updateให้อยู่ภายใต้ Accountใหม่ถูกต้อง	
21	Web	SA	Application Management	Change Unitholder to Another Account	- SA Request API UH : SEG เข้าระบบโดยที่ Account ต้นทางติด Lock Account ปลายทางติดไม่ Lock	- Cannot change Account No. due to pending account Status.	Pass
22	Web	SA	Application Management	Change Unitholder to Another Account	- SA Request API UH : OMN เข้าระบบโดยที่ Account ต้นทางติด Lock Account ปลายทางติดไม่ Lock	- Cannot change Account No. due to pending account Status.	Pass
23	Web	SA	Application Management	Change Unitholder to Another Account	- SA Request API UH : SEG เข้าระบบโดยที่ Account ต้นทางไม่ติด Lock Account ปลายทางติด Lock	- Cannot change Account No. due to pending account Status.	Pass
24	Web	SA	Application Management	Change Unitholder to Another	- SA Request API UH : OMN เข้าระบบโดยที่ Account ต้นทาง	- Cannot change Account No. due to pending account Status.	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

				Account	ไม่ติด Lock Account ปลายทางติด Lock		

Modal สำหรับกรอกข้อมูล Change Unitholder to Another Account

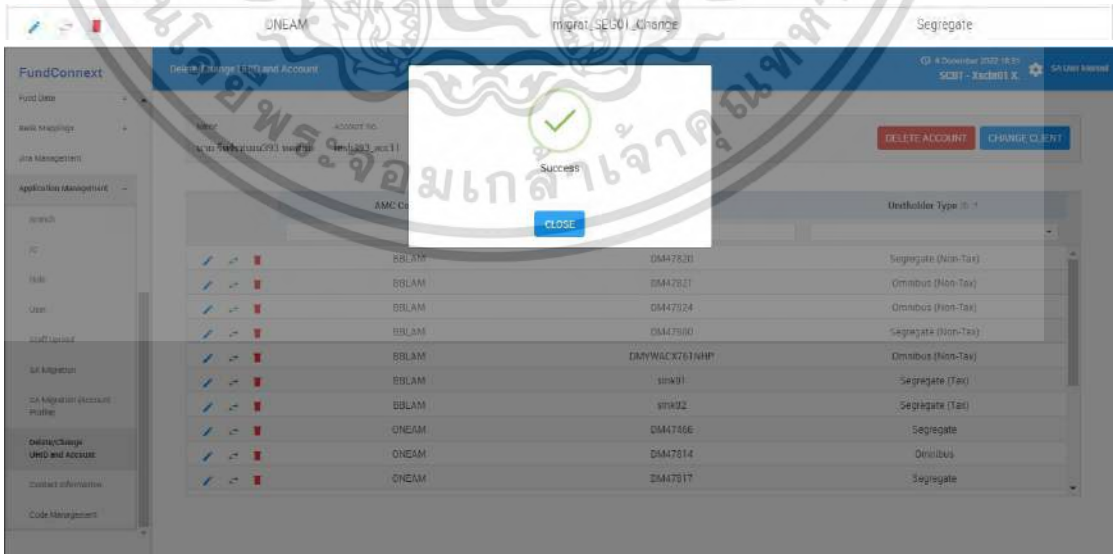


รูปที่ 4.22 หน้าจอสำหรับกรอก Change Unitholder to Another Account

ผลการทดสอบกรณีที่ 1 SA Migrate UH : SEG เข้าสู่ระบบ

Account ต้นทาง : ไม่ติดLock

Account ปลายทาง : ไม่ติดLock



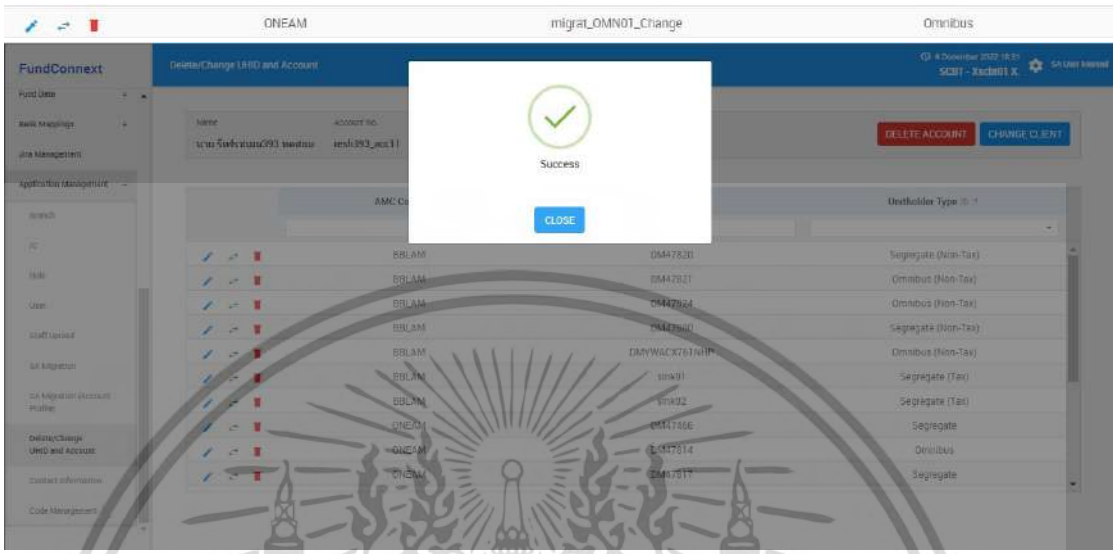
รูปที่ 4.23 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 2 SA Migrate UH : OMN เข้าสู่ระบบ

Account ต้นทาง : ไม่ติดLock

Account ปลายทาง : ไม่ติดLock

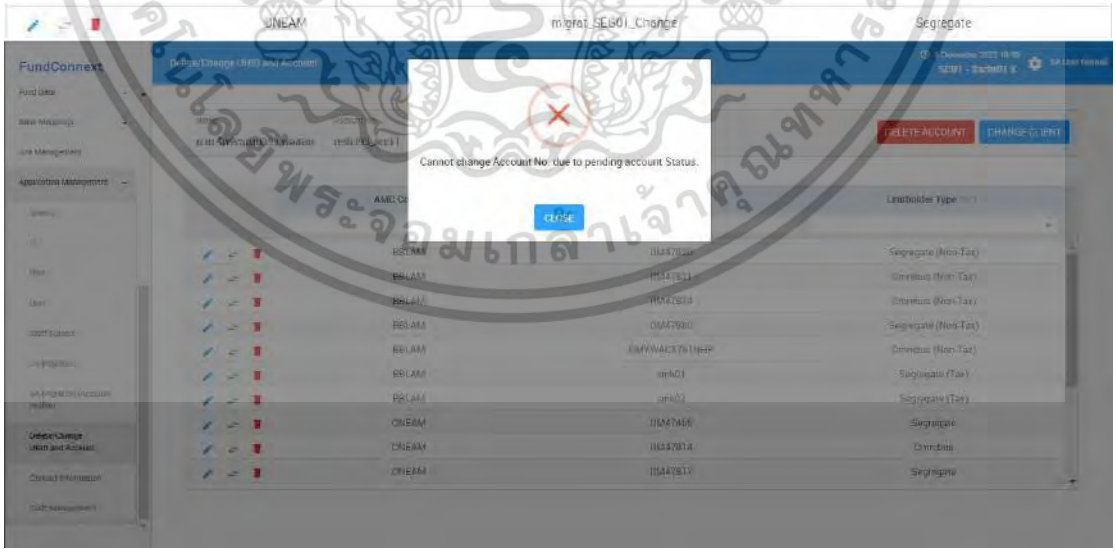


รูปที่ 4.24 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 2

ผลการทดสอบกรณีที่ 3 SA Migrate UH : SEG เข้าสู่ระบบ

Account ต้นทาง : ติดLock

Account ปลายทาง : ไม่ติดLock



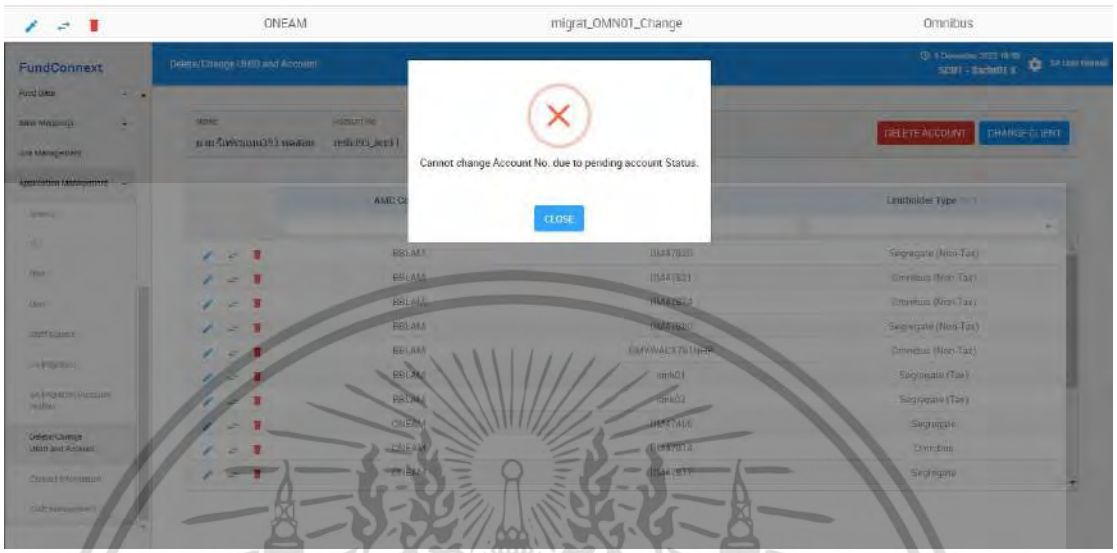
รูปที่ 4.25 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 4 SA Migrate UH : OMN เข้าสู่ระบบ

Account ต้นทาง : ติดLock

Account ปลายทาง : ไม่ติดLock

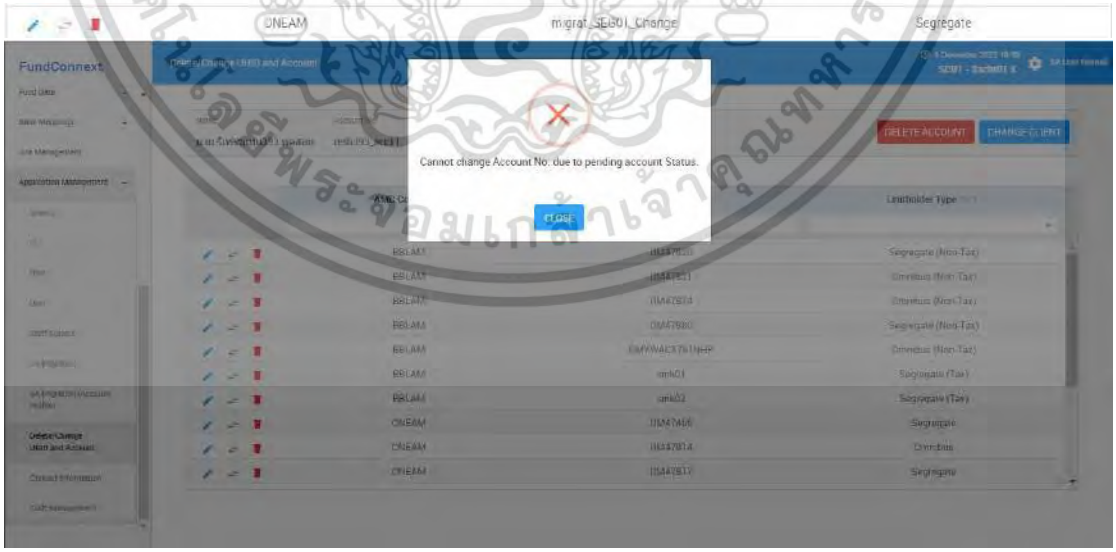


รูปที่ 4.26 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 4

ผลการทดสอบกรณีที่ 5 SA Migrate UH : SEG เข้าสู่ระบบ

Account ต้นทาง : ไม่ติดLock

Account ปลายทาง : ติดLock



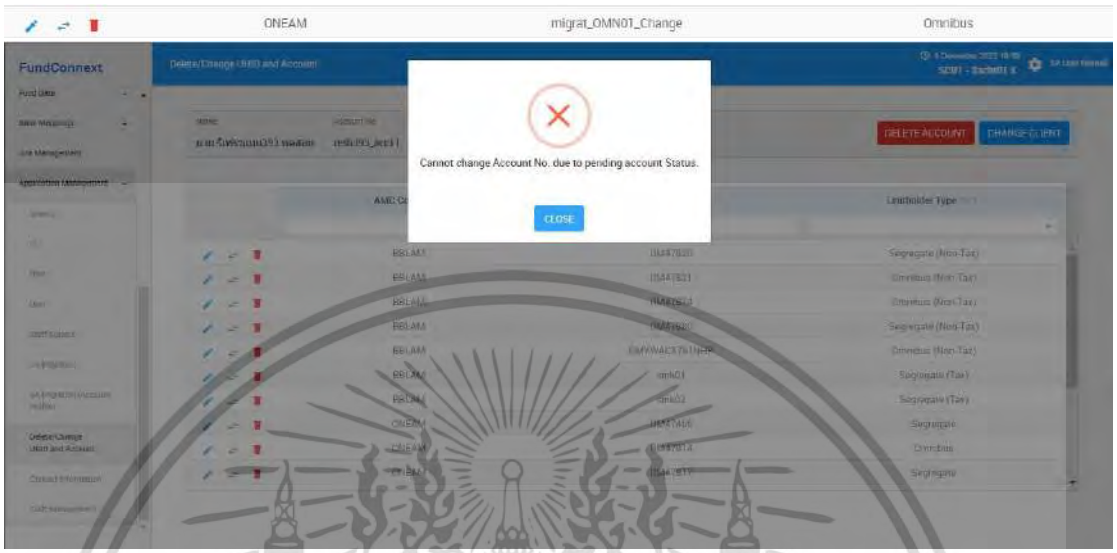
รูปที่ 4.27 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 6 SA Migrate UH : OMN เข้าสู่ระบบ

Account ต้นทาง : ไม่ติดLock

Account ปลายทาง : ติดLock

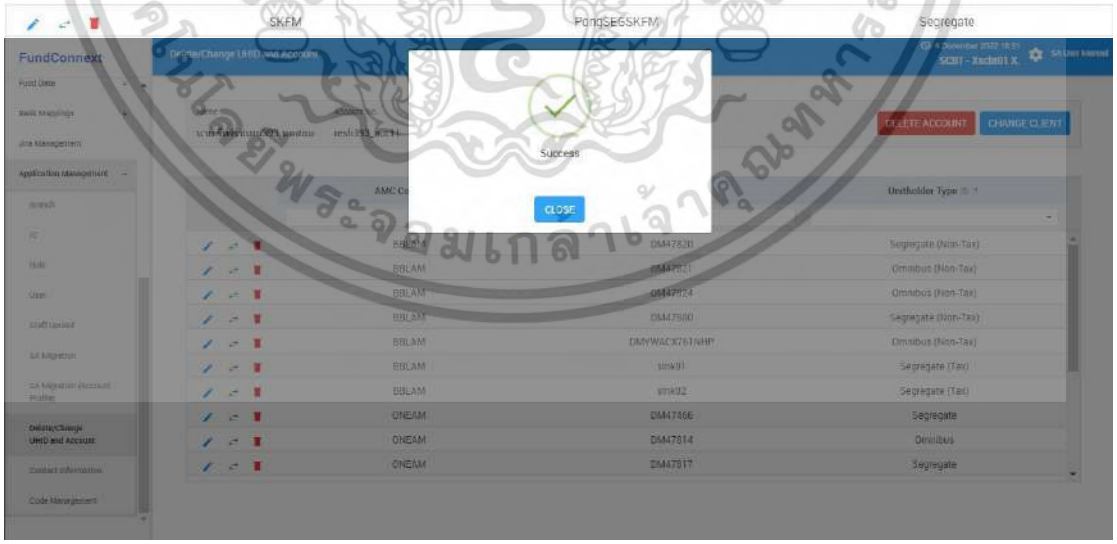


รูปที่ 4.28 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 6

ผลการทดสอบกรณีที่ 7 SA Send Order ด้วย UH : SEG

Account ต้นทาง : ไม่ติดLock

Account ปลายทาง : ไม่ติดLock



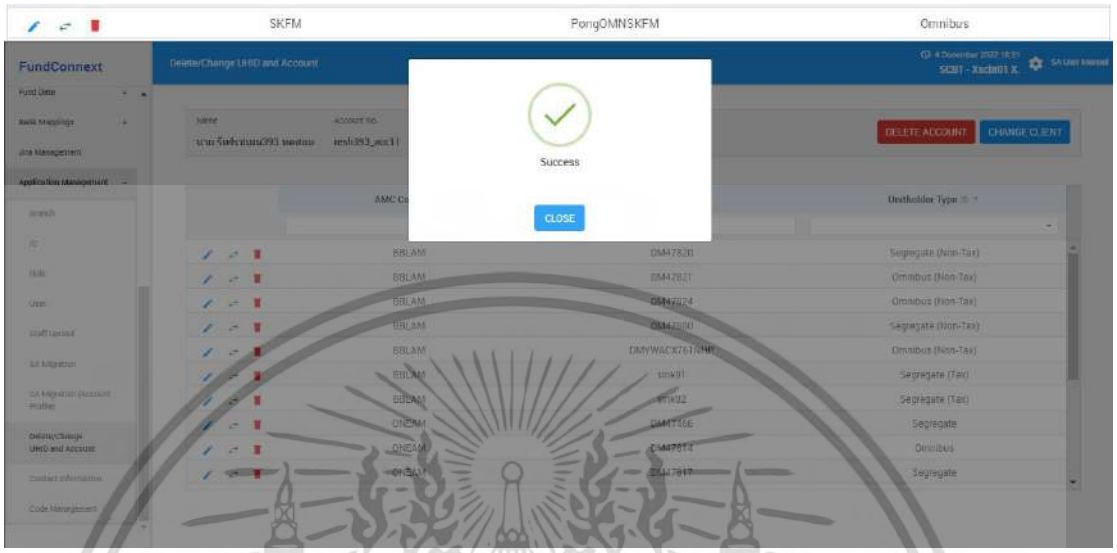
รูปที่ 4.29 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 8 SA Send Order ด้วย UH : OMN

Account ต้นทาง : ไม่ติดLock

Account ปลายทาง : ไม่ติดLock

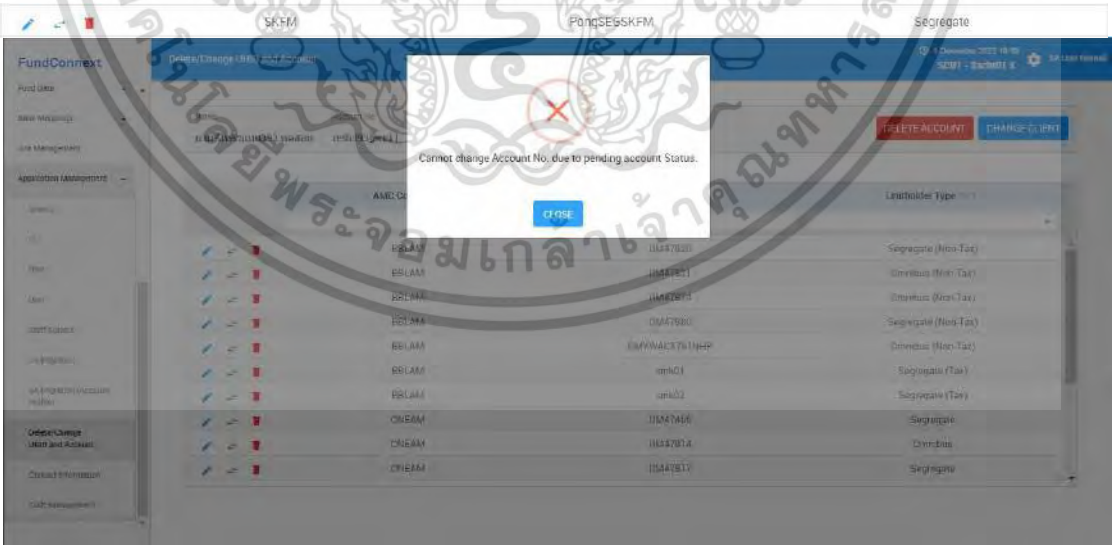


รูปที่ 4.30 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 8

ผลการทดสอบกรณีที่ 9 SA Send Order ด้วย UH : SEG

Account ต้นทาง : ติดLock

Account ปลายทาง : ไม่ติดLock



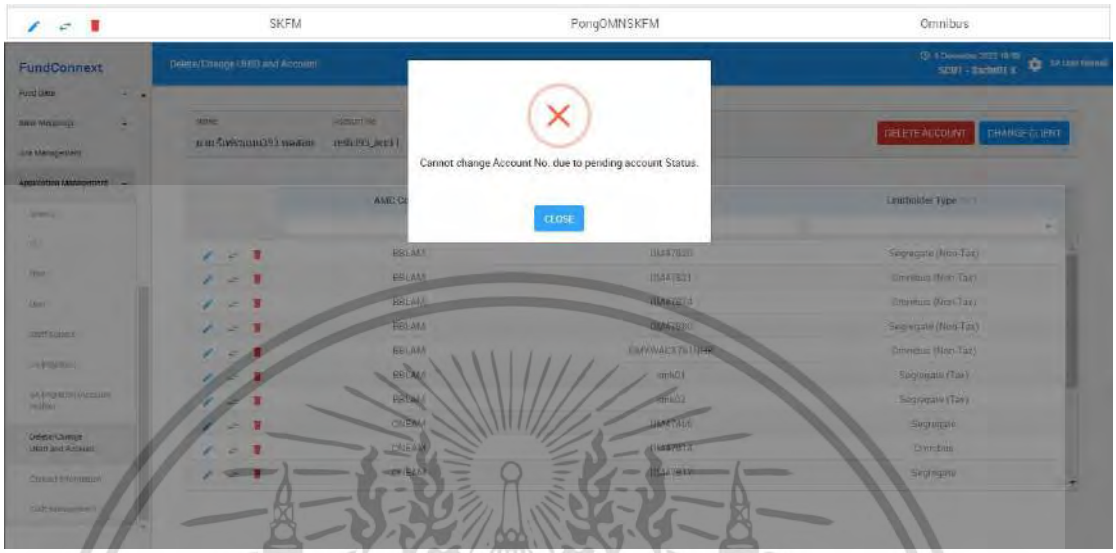
รูปที่ 4.31 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 10 SA Send Order ด้วย UH : OMN

Account ต้นทาง : ติดLock

Account ปลายทาง : ไม่ติดLock

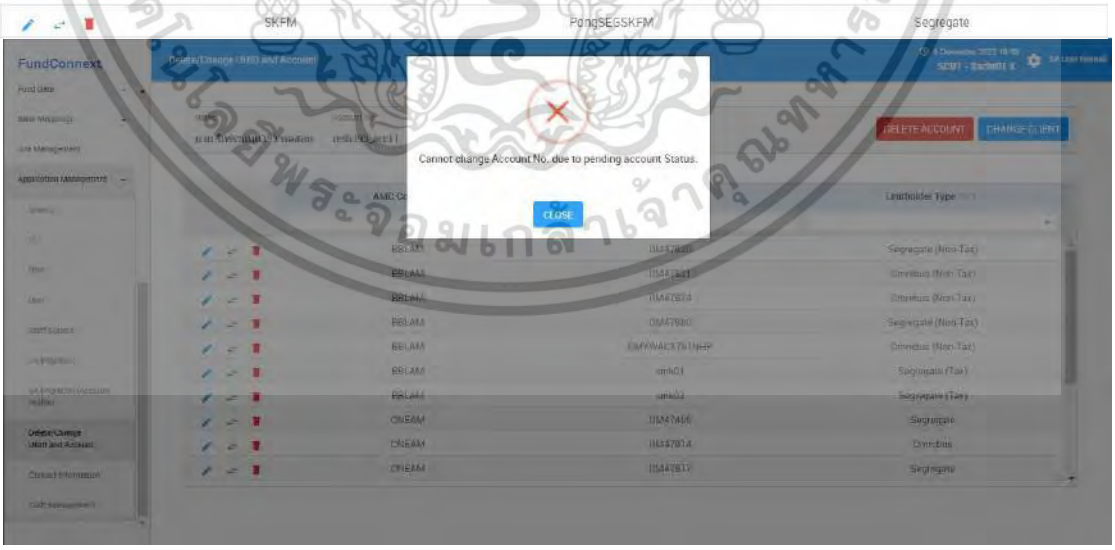


รูปที่ 4.32 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 10

ผลการทดสอบกรณีที่ 11 SA Send Order ด้วย UH : SEG

Account ต้นทาง : ไม่ติดLock

Account ปลายทาง : ติดLock



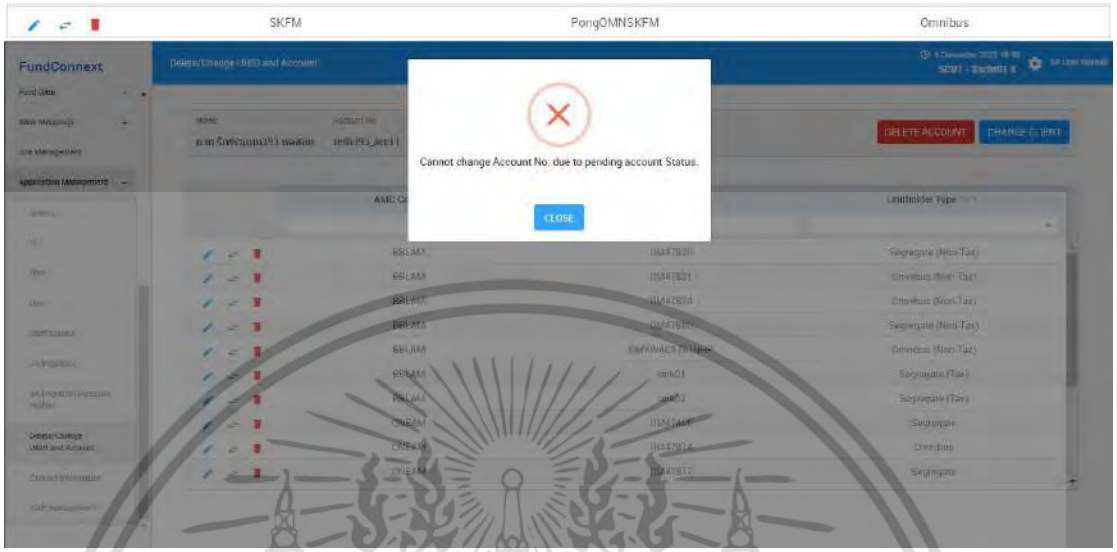
รูปที่ 4.33 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 12 SA Send Order ด้วย UH : OMN

Account ต้นทาง : ไม่ติดLock

Account ปลายทาง : ติดLock

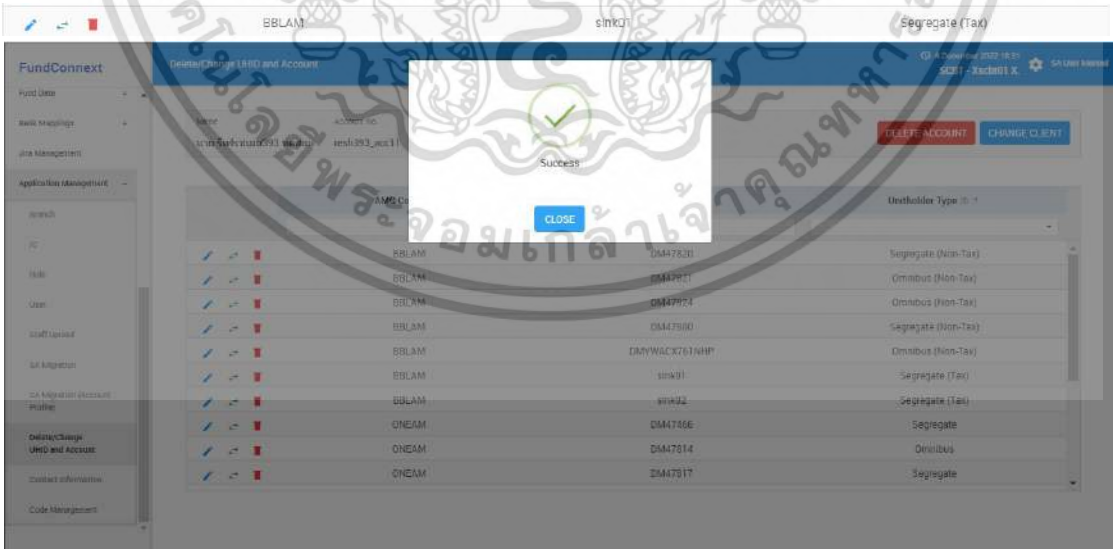


รูปที่ 4.34 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 12

ผลการทดสอบกรณีที่ 13 SA Move Sink UH : SEG

Account ต้นทาง : ไม่ติดLock

Account ปลายทาง : ไม่ติดLock



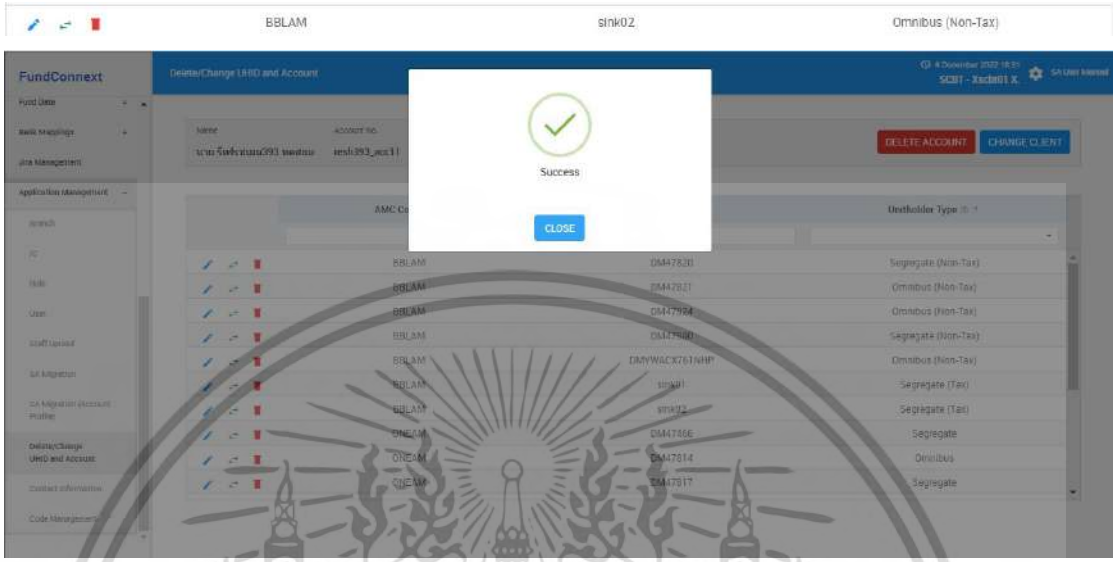
รูปที่ 4.35 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 14 SA Move Sink UH : OMN

Account ต้นทาง : ไม่ติดLock

Account ปลายทาง : ไม่ติดLock

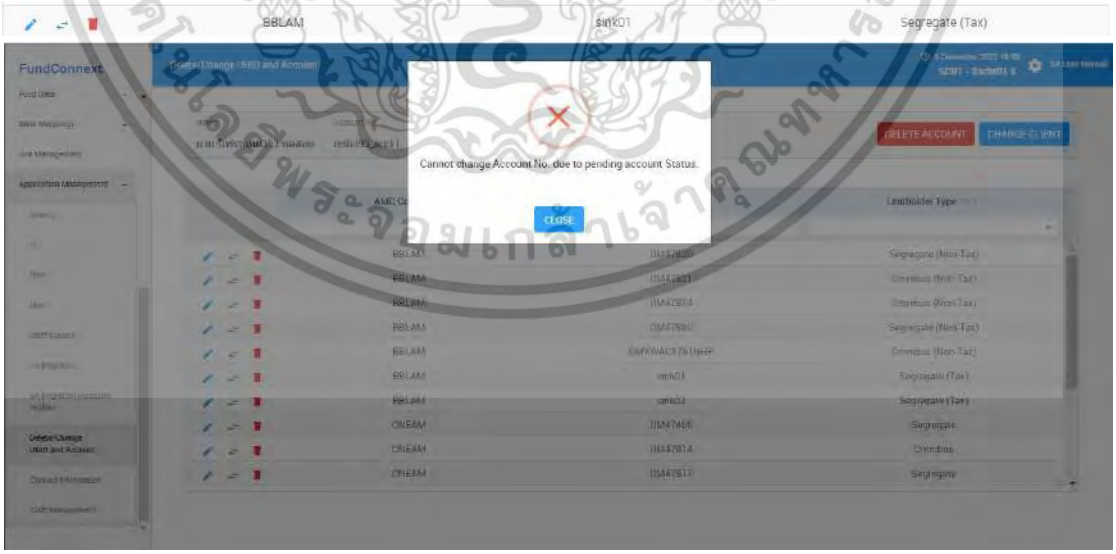


รูปที่ 4.36 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 14

ผลการทดสอบกรณีที่ 15 SA Move Sink UH : SEG

Account ต้นทาง : ติดLock

Account ปลายทาง : ไม่ติดLock



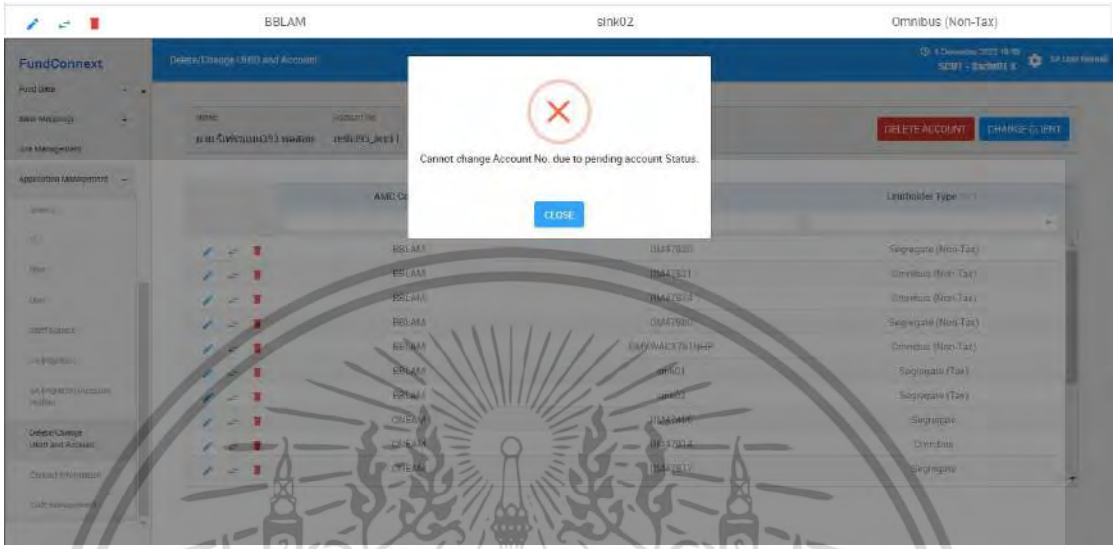
รูปที่ 4.37 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 16 SA Move Sink UH : OMN

Account ต้นทาง : ติดLock

Account ปลายทาง : ไม่ติดLock

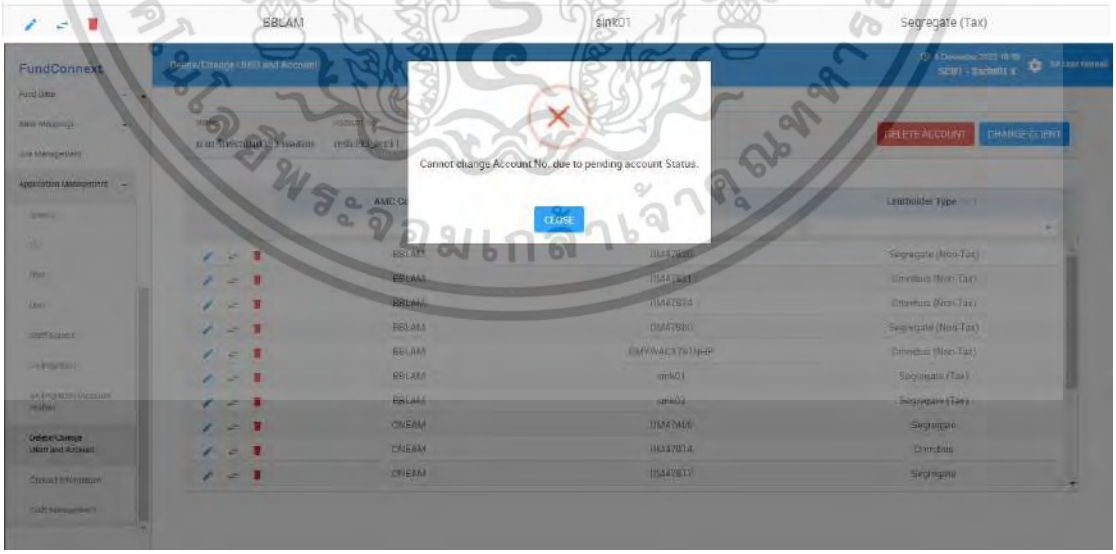


รูปที่ 4.38 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 16

ผลการทดสอบกรณีที่ 17 SA Move Sink UH : SEG

Account ต้นทาง : ไม่ติดLock

Account ปลายทาง : ติดLock



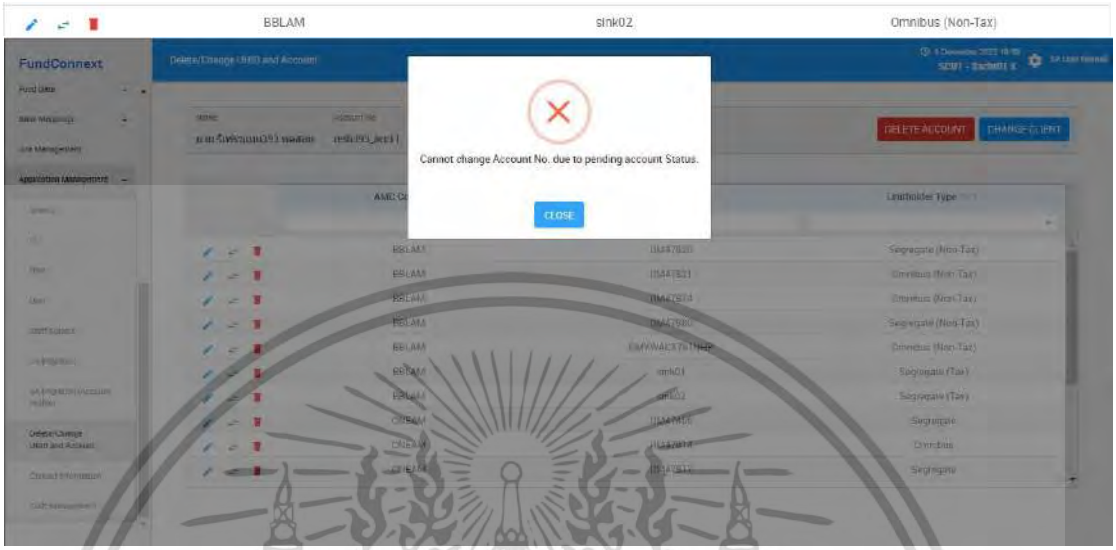
รูปที่ 4.39 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 18 SA Move Sink UH : OMN

Account ต้นทาง : ไม่ติดLock

Account ปลายทาง : ติดLock

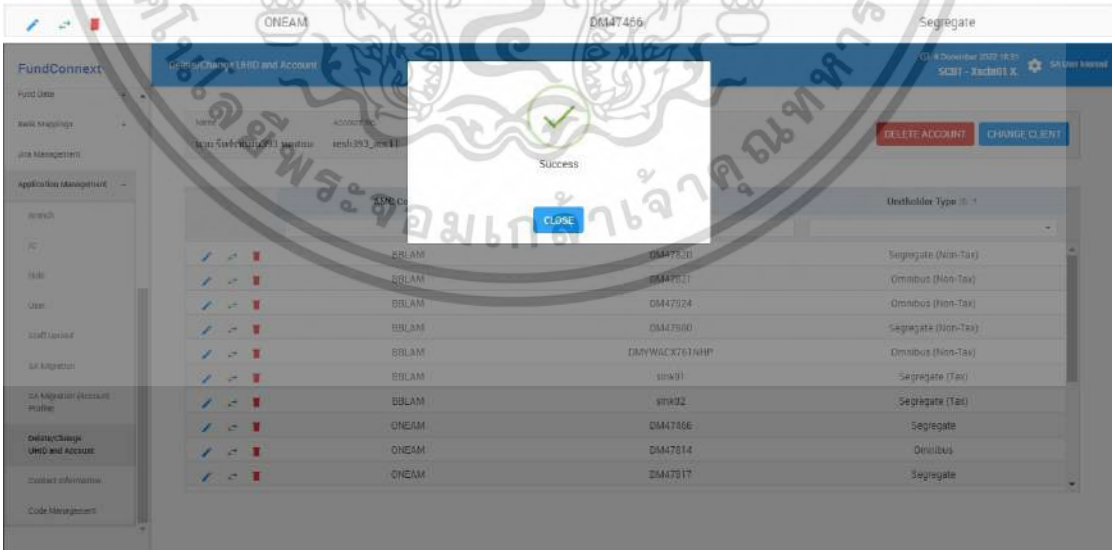


รูปที่ 4.40 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 18

ผลการทดสอบกรณีที่ 19 SA Request UH : SEG

Account ต้นทาง : ไม่ติดLock

Account ปลายทาง : ไม่ติดLock



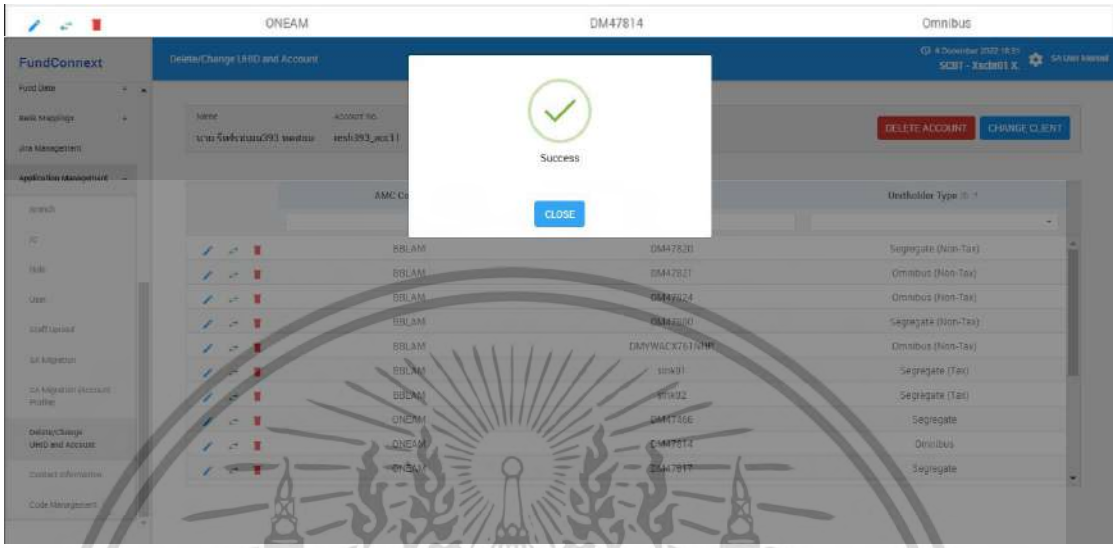
รูปที่ 4.41 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 20 SA Request UH : OMN

Account ต้นทาง : ไม่ติดLock

Account ปลายทาง : ไม่ติดLock

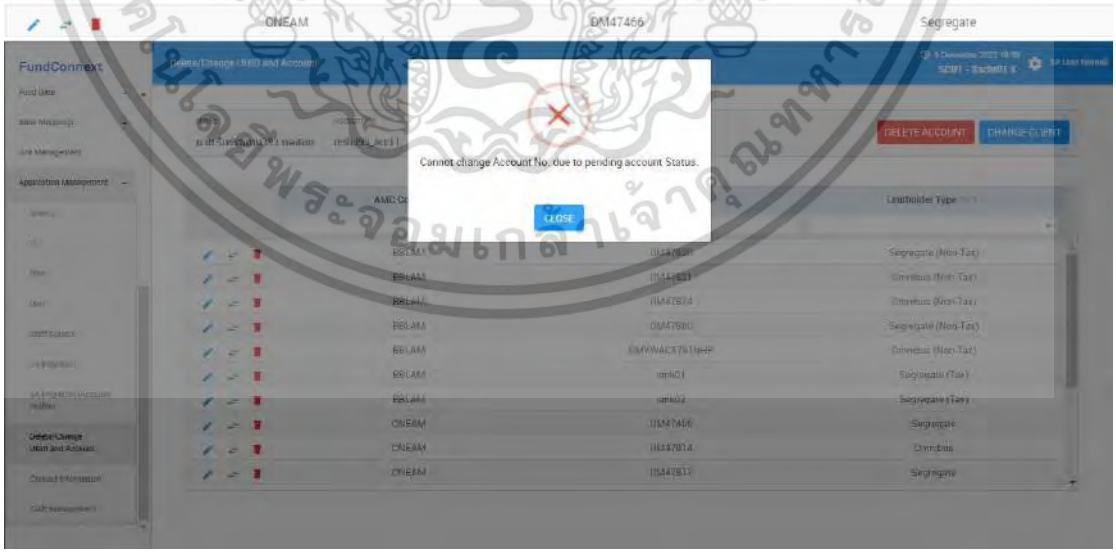


รูปที่ 4.42 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 20

ผลการทดสอบกรณีที่ 21 SA Request UH : SEG

Account ต้นทาง : ติดLock

Account ปลายทาง : ไม่ติดLock



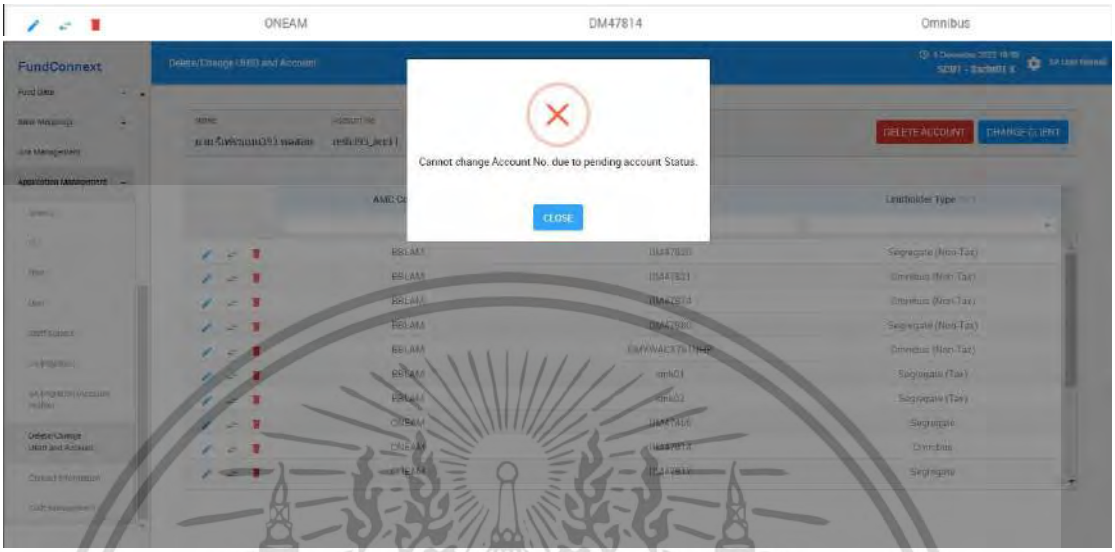
รูปที่ 4.43 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 22 SA Request UH : OMN

Account ต้นทาง : ติดLock

Account ปลายทาง : ไม่ติดLock

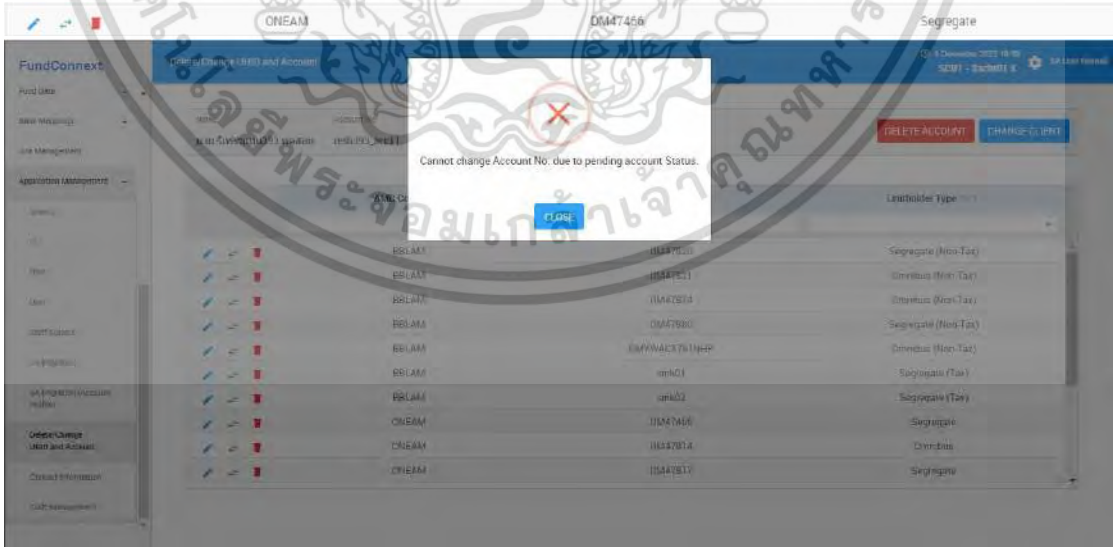


รูปที่ 4.44 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 22

ผลการทดสอบกรณีที่ 23 SA Request UH : SEG

Account ต้นทาง : ไม่ติดLock

Account ปลายทาง : ติดLock



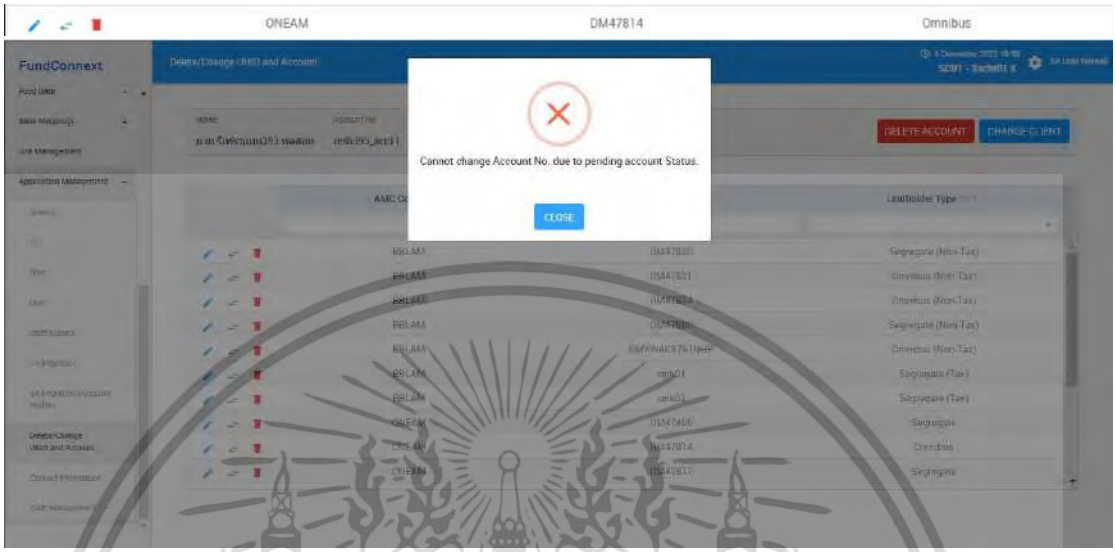
รูปที่ 4.45 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 24 SA Request UH : OMN

Account ต้นทาง : ไม่ติดLock

Account ปลายทาง : ติดLock



รูปที่ 4.46 หน้าจอผลลัพธ์การทดสอบแบบ Manual Test ตัวอย่างที่ 2 กรณีที่ 24

4.2 ผลการทดสอบ Automated Test แบบ Input Data

ตัวอย่างที่ 1 การทดสอบ Automated Test แบบ Input Data

ตารางที่ 4.3 ผลการทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result	Test Result
1	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึก Background เมื่อไม่กรอกข้อมูล Occupation	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง	Pass
2	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึก Background เมื่อไม่กรอกข้อมูล	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

					Occupation Other	ถูกต้อง	
3	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึก Background เมื่อไม่กรอกข้อมูล Business Type	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง	Pass
4	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึก Background เมื่อไม่กรอกข้อมูล Business Type Other	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง	Pass
5	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึก Background เมื่อไม่กรอกข้อมูล Workplace information	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง	Pass
6	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึก Background เมื่อไม่กรอกข้อมูล Source of income	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง	Pass
7	Web	SA	Account Management	Open Account	- ตรวจสอบการบันทึก Background เมื่อไม่กรอกข้อมูล monthly income	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง	Pass
8	Web	SA	Account	Open	- ตรวจสอบการ	- ไม่สามารถบันทึก	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			Manage ment	Account	บันทึก Background เมื่อไม่กรอก ข้อมูล fx risk	ข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง	
9	Web	SA	Account Manage ment	Open Account	- ตรวจสอบการบันทึก Background เมื่อไม่กรอก ข้อมูล derivatives risk	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง	Pass
10	Web	SA	Account Manage ment	Open Account	- ตรวจสอบการบันทึก Background เมื่อไม่กรอก ข้อมูล suitability form	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง	Pass
11	Web	SA	Account Manage ment	Open Account	- ตรวจสอบการบันทึก Background เมื่อไม่กรอก ข้อมูล vulnerable detail	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง	Pass
12	Web	SA	Account Manage ment	Open Account	- ตรวจสอบการบันทึก Background เมื่อไม่กรอก ข้อมูล fatca form	- ไม่สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง - แสดงข้อความ Required Field ได้ถูกต้อง	Pass
13	Web	SA	Account Manage ment	Open Account	- ตรวจสอบการบันทึก Background เมื่อกรอกข้อมูล	- สามารถบันทึกข้อมูล Backgroundได้ถูกต้อง	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์การรัน Automated Test Script แบบ Input Data ตัวอย่างที่ 1

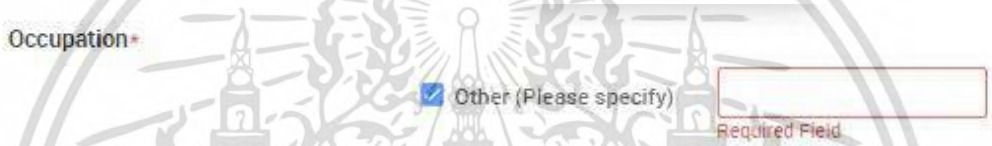
รูปที่ 4.47 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 1 แสดงข้อความ “Required Field” เมื่อไม่กรอกข้อมูล Occupation

รูปที่ 4.48 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 1

ผลการทดสอบกรณีที่ 2 แสดงข้อความ “Required Field” เมื่อไม่กรอกข้อมูล Occupation Other



รูปที่ 4.49 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 2

ผลการทดสอบกรณีที่ 3 แสดงข้อความ “Required Field” เมื่อไม่กรอกข้อมูล Business Type

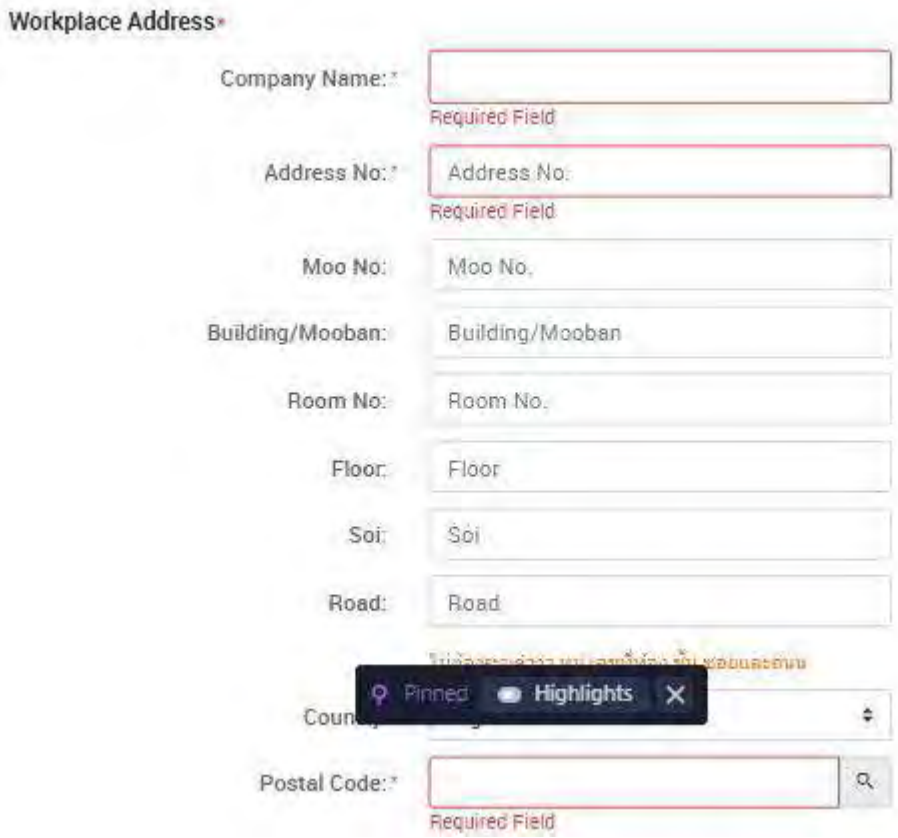
รูปที่ 4.50 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 4 แสดงข้อความ “Required Field” เมื่อไม่กรอกข้อมูล Business Type Other

รูปที่ 4.51 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 4

ผลการทดสอบกรณีที่ 5 แสดงข้อความ “Required Field” เมื่อไม่กรอกข้อมูล Workplace information



รูปที่ 4.52 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

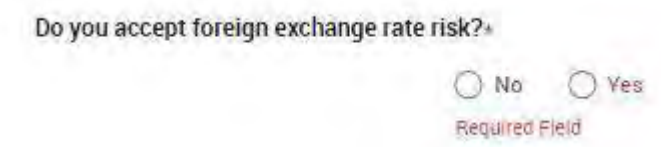
ผลการทดสอบกรณีที่ 6 แสดงข้อความ “Required Field” เมื่อไม่กรอกข้อมูล Source income

รูปที่ 4.53 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 6

ผลการทดสอบกรณีที่ 7 แสดงข้อความ “Required Field” เมื่อไม่กรอกข้อมูล Monthly income

รูปที่ 4.54 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 7

ผลการทดสอบกรณีที่ 8 แสดงข้อความ “Required Field” เมื่อไม่กรอกข้อมูล fx risk



รูปที่ 4.55 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

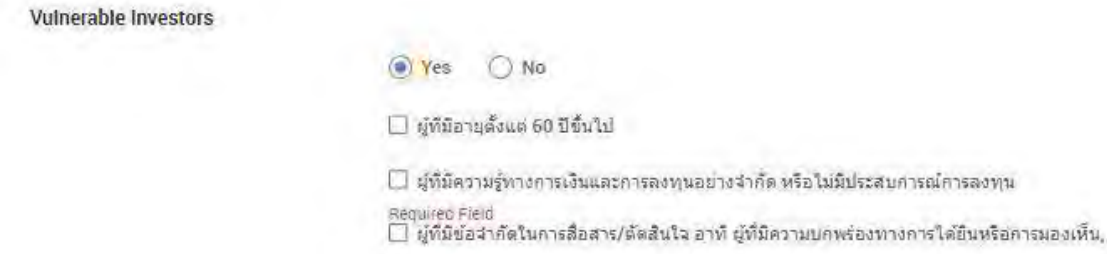
ผลการทดสอบกรณีที่ 9 แสดงข้อความ “Required Field” เมื่อไม่กรอกข้อมูล derivatives risk

รูปที่ 4.56 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 9

ผลการทดสอบกรณีที่ 10 แสดงข้อความ “Required Field” เมื่อไม่กรอกข้อมูล Suitability

รูปที่ 4.57 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 10

ผลการทดสอบกรณีที่ 11 แสดงข้อความ “Required Field” เมื่อไม่กรอกข้อมูล Vulnerable detail



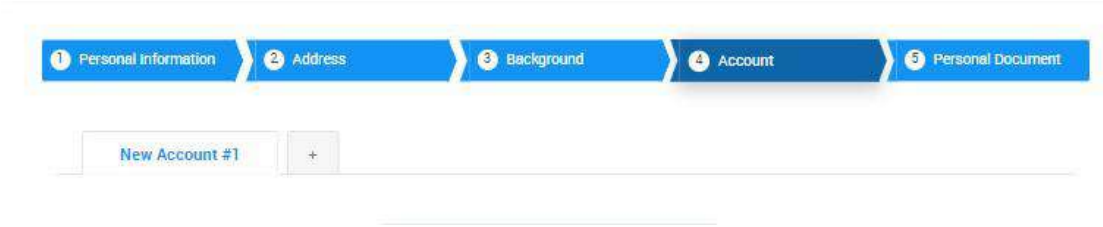
รูปที่ 4.58 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 11

ผลการทดสอบกรณีที่ 12 แสดงข้อความ “Required Field” เมื่อไม่กรอกข้อมูล fatca

รูปที่ 4.59 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 13 สามารถบันทึกข้อมูล Background Information



รูปที่ 4.60 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 1 กรณีที่ 13

ตัวอย่างที่ 2 การทดสอบ Automated Test แบบ Input Data

ตารางที่ 4.4 ผลการทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 2

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result	Test Result
1	Web	SA	Account Management	Open Account	- ตรวจสอบError Message เมื่อใส่ Card Noที่ไม่มีในระบบ	- แสดงError Message “Please create new individual/juristic profile”ได้ถูกต้อง	Pass
2	Web	SA	Account Management	Open Account	- ตรวจสอบError Message เมื่อใส่บัญชีJuristicที่ติดแม่กุญแจ	- แสดงError Message “ID inactive. Please approve individual/juristic profile”ได้ถูกต้อง	Pass
3	Web	SA	Account Management	Open Account	- ตรวจสอบการเปิดบัญชีBy Account เมื่อใส่Card No บัญชีJuristicที่ไม่ติดแม่กุญแจ	- สามารถเปิดบัญชี By Account ได้ถูกต้อง	Pass
4	Web	SA	Account Management	Open Account	- ตรวจสอบข้อมูลหลังจากเปิดบัญชี By	- สามารถเปิดบัญชี By Account ได้ถูกต้อง - แสดงข้อมูลที่เปิด	Pass

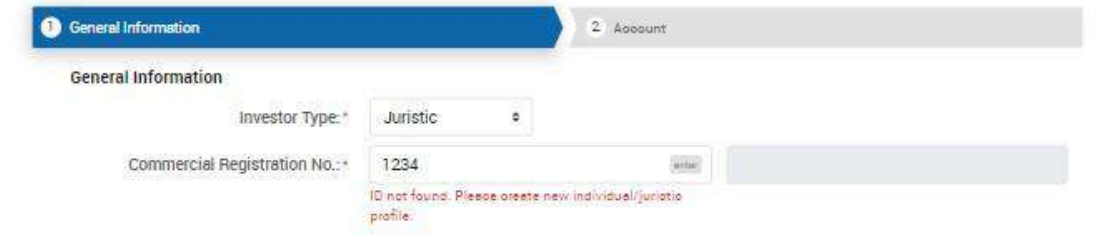
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

					Account	บัญชีได้ถูกต้อง	

ผลลัพธ์การรัน Automated Test Script แบบ Input Data ตัวอย่างที่ 2

รูปที่ 4.61 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 2

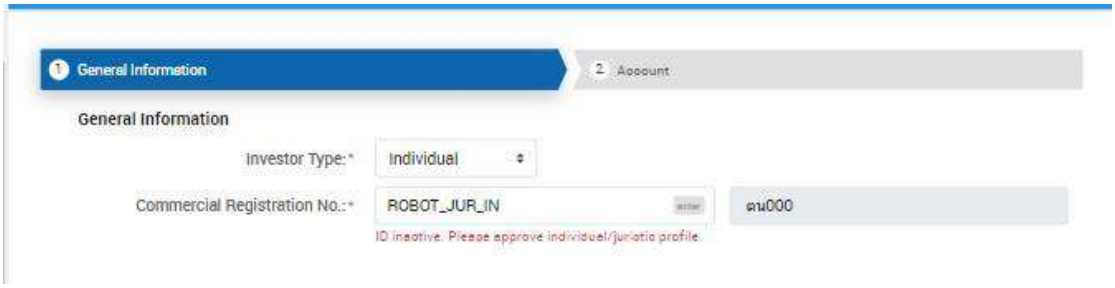
ผลการทดสอบกรณีที่ 1 แสดง Error Message” Please create new individual/juristic profile” เมื่อใส่ ID Card No.ที่ไม่ได้ทำการเปิดบัญชีในระบบ



รูปที่ 4.62 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 2 กรณีที่ 1

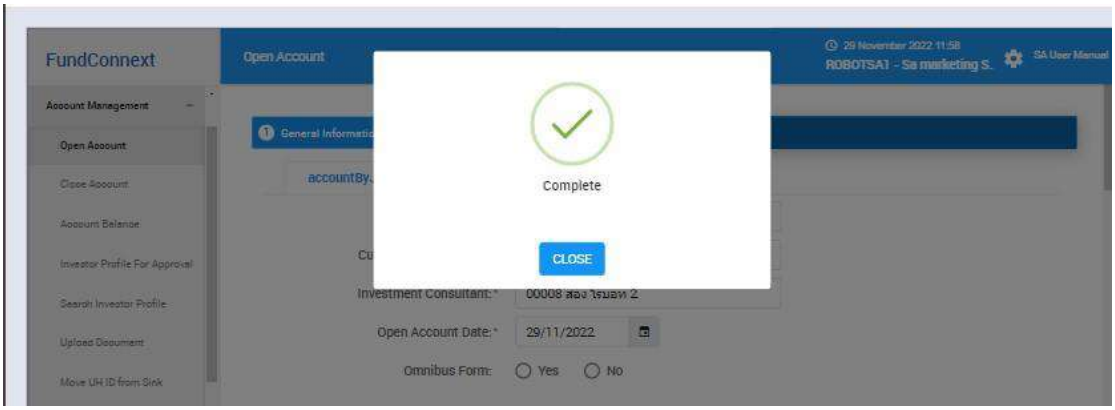
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 2 แสดง Error Message “Please approve individual/juristic profile”
เมื่อใส่ ID Card No.ที่ติด Lock



รูปที่ 4.63 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 2 กรณีที่ 2

ผลการทดสอบกรณีที่ 3 สามารถเปิดบัญชี By Account ได้สำเร็จ เมื่อใส่ ID Card No.ที่ไม่ติดLock



รูปที่ 4.64 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Input Data ตัวอย่างที่ 2 กรณีที่ 3

4.3 ผลการทดสอบ Automated Test แบบ Look and Field

ตัวอย่างที่ 1 การทดสอบ Automated Test แบบ Look and Field

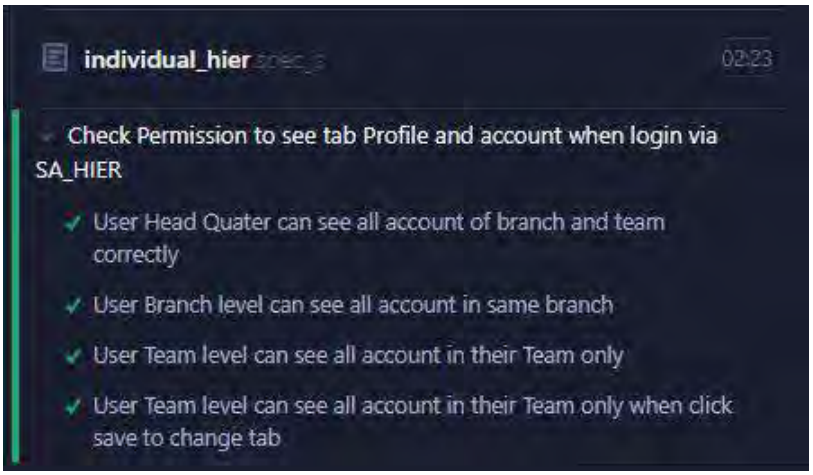
ตารางที่ 4.5 ผลการทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 1

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result	Test Result
1	Web	SA	Account Management	Open Account	- ตรวจสอบการแสดง Account ที่หน้าจอเมื่อ Login : HQ	- แสดง Account ทั้งหมดได้ถูกต้อง	Pass
2	Web	SA	Account	Open	- ตรวจสอบการ	- แสดง Account ใน	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			Manage ment	Account	แสดง Account ที่หน้าจอเมื่อ Login : B1	Branch1 เท่านั้น	
3	Web	SA	Account Manage ment	Open Account	- ตรวจสอบการแสดง Account ที่หน้าจอเมื่อ Login : B1T1 แล้วกด ที่Account Tab	- แสดง Account ใน Branch1 Team1 เท่านั้น	Pass
4	Web	SA	Account Manage ment	Open Account	- ตรวจสอบการแสดง Account ที่หน้าจอเมื่อ Login : B1T1 แล้วกดSaveเพื่อ เพื่อไปยัง Account Tab	- แสดง Account ใน Branch1 Team1 เท่านั้น	Pass

ผลลัพธ์การรัน Automated Test Script แบบ Look and Field ตัวอย่างที่ 1

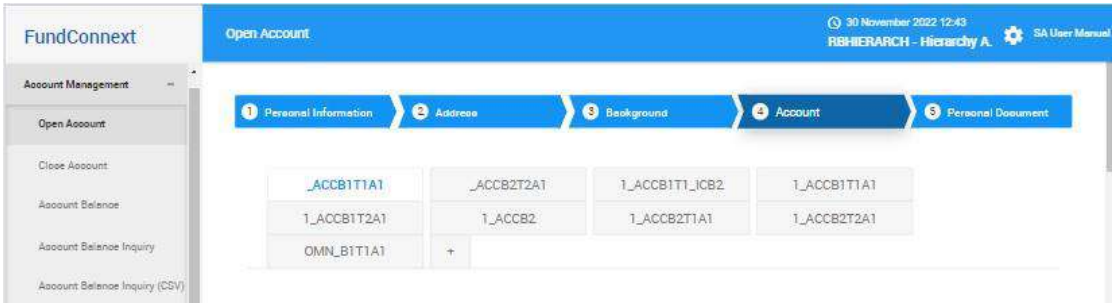


รูปที่ 4.65 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 1 แสดง Accountทั้งหมด

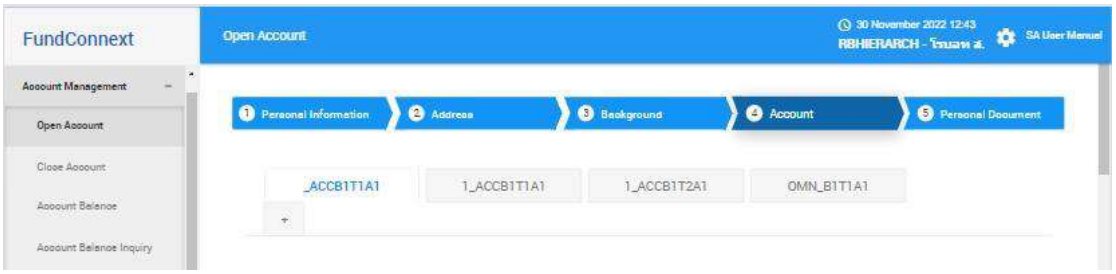
เมื่อ Login : HQ



รูปที่ 4.66 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 1 กรณีที่ 1

ผลการทดสอบกรณีที่ 2 แสดง Account ใน Branch1

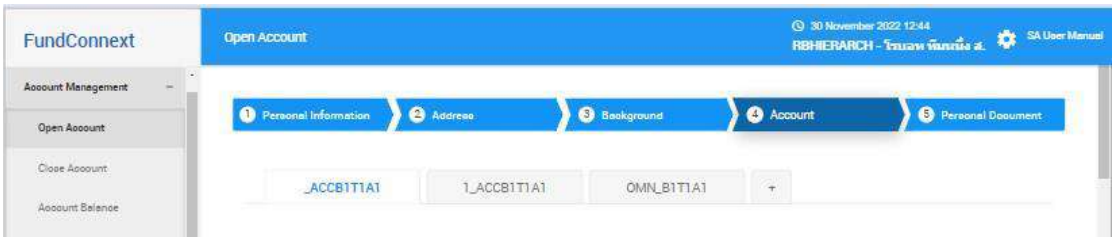
เมื่อ Login : Branch1



รูปที่ 4.67 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 1 กรณีที่ 2

ผลการทดสอบกรณีที่ 3 แสดง Account ใน Branch1 Team1

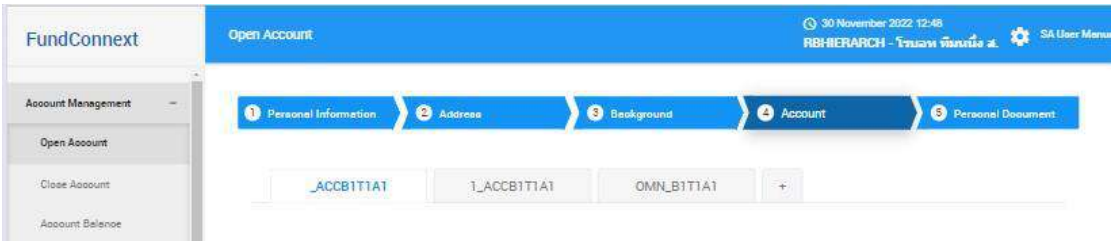
เมื่อ Login : Branch1Team1



รูปที่ 4.68 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 1 กรณีที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 4 แสดง Accountใน Branch1 Team1 โดยการกดSaveข้อมูล
เมื่อ Login : Branch1Team1



รูปที่ 4.69 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 1 กรณีที่ 4

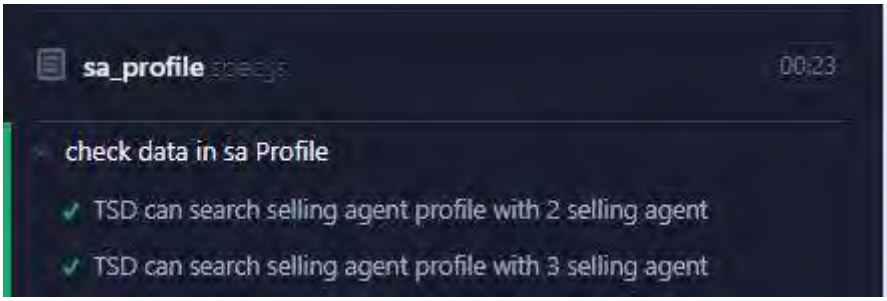
ตัวอย่างที่ 2 การทดสอบ Automated Test แบบ Look and Field

ตารางที่ 4.6 ผลการทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 2

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result	Test Result
1	Web	TSD	File Maintanance	SA Profile	- TSD สามารถค้นหาข้อมูลของ SA ได้ครั้งละ 2 SA	- แสดงข้อมูลของSA ทั้งสองได้ถูกต้อง	Pass
2	Web	TSD	File Maintanance	SA Profile	- TSD สามารถค้นหาข้อมูลของ SA ได้ครั้งละ 3 SA	- แสดงข้อมูลของSA ทั้งสามได้ถูกต้อง	Pass

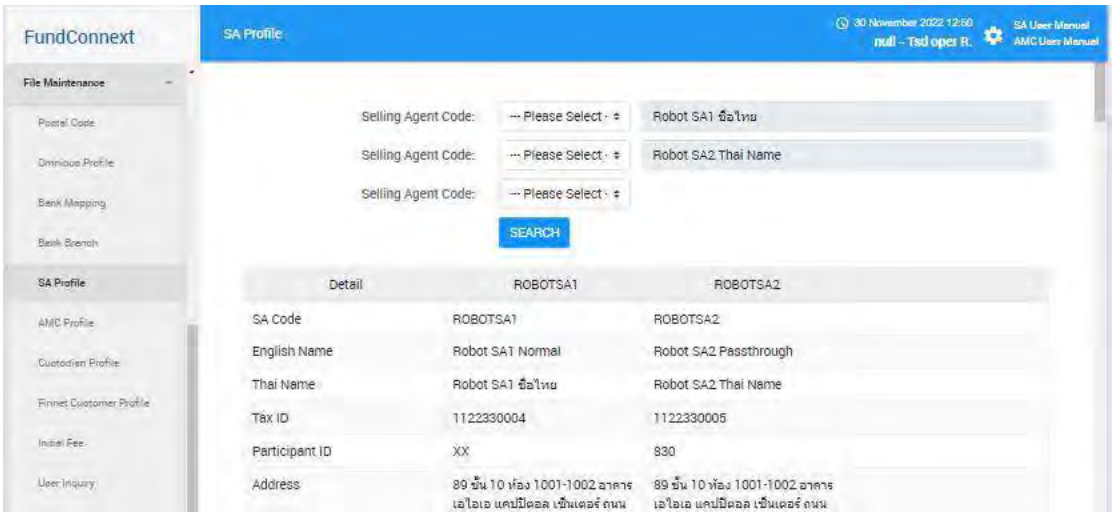
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์การรัน Automated Test Script แบบ Look and Field ตัวอย่างที่ 2



รูปที่ 4.70 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 2

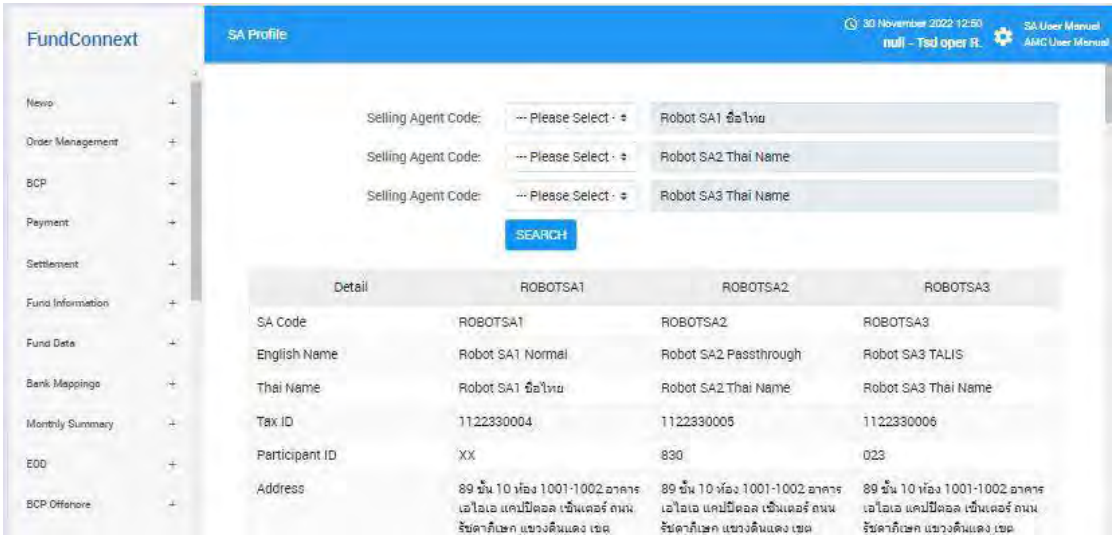
ผลการทดสอบกรณีที่ 1 แสดงข้อมูล SA Profile 2 SAพร้อมกันได้



รูปที่ 4.71 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 2 กรณีที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 2 แสดงข้อมูล SA Profile 3 SAพร้อมกันได้



รูปที่ 4.72 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field ตัวอย่างที่ 2 กรณีที่ 2

4.4 ผลการทดสอบ Automated Test แบบ Look and Field (ตาราง)

ตัวอย่างที่ 1 การทดสอบ Automated Test แบบ Look and Field (ตาราง)

ตารางที่ 4.7 ผลการทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 1

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result	Test Result
1	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบการแสดงผลหน้าจอ Failed Switching Order	- แสดงช่องรับข้อมูล Payment Date From/To ได้ถูกต้อง	Pass
2	Web	SA	Order Management	Failed Switching Order	- กรอกข้อมูลเฉพาะ Payment Date From	- ไม่สามารถค้นหาข้อมูลได้	Pass
3	Web	SA	Order Management	Failed Switching Order	- กรอกข้อมูลเฉพาะ Payment Date	- ไม่สามารถค้นหาข้อมูลได้	Pass

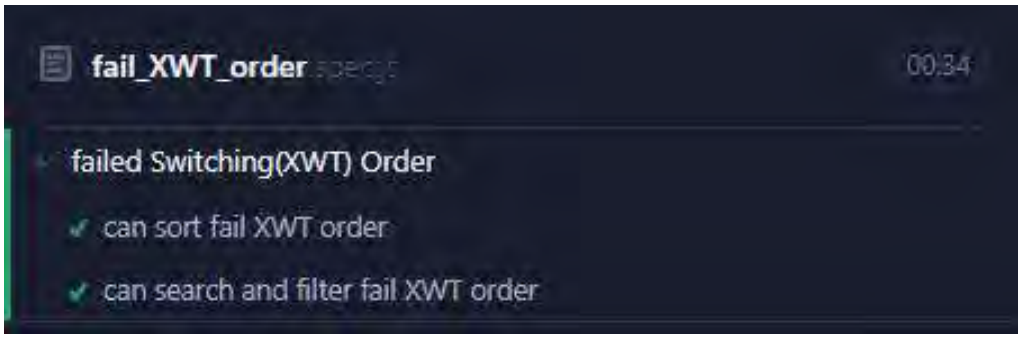
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

					To		
4	Web	SA	Order Management	Failed Switching Order	- แสดงหัวข้อตารางหน้าจอ Failed Switching Order	- แสดงหัวข้อตารางได้ถูกต้อง	Pass
5	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบข้อมูล Field : PaymentDate(RED) เมื่อ Gen รายการ Sub ไม่สำเร็จ	- แสดง Payment Date(RED) เป็น yyyymmdd ได้ถูกต้อง	Pass
6	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบข้อมูล Field : XWT Ref No. เมื่อ Gen รายการ Sub ไม่สำเร็จ	- แสดง XWT Ref No. เป็น xxxxxxxx ได้ถูกต้อง	Pass
7	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบข้อมูล Field : Account No เมื่อ Gen รายการ Sub ไม่สำเร็จ	- แสดง Account เป็น Accountที่ทำรายการได้ถูกต้อง	Pass
8	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบข้อมูล Field : Counter Unitholder ID เมื่อ Gen รายการ Sub ไม่สำเร็จ	- แสดง Counter Unitholder ID เป็น Unitholderที่ทำรายการขาซื้อได้ถูกต้อง	Pass
9	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบข้อมูล Field : Customer Name เมื่อ Gen	- แสดง Customer Name เป็นรายชื่อลูกค้าที่ทำรายการได้ถูกต้อง	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

					รายการ Sub ไม่สำเร็จ		
10	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบข้อมูล Field : Counter fund Code เมื่อ Gen รายการ Sub ไม่สำเร็จ	- แสดง Counter fund Code เป็น FundCodeที่ทำรายการขาซื้อได้ถูกต้อง	Pass
11	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบข้อมูล Field : Amount เมื่อ Gen รายการ Sub ไม่สำเร็จ	- แสดง Amount เป็นยอดเงินของรายการได้ถูกต้อง	Pass
12	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบข้อมูล Default Sort	- แสดง Default Sort ดังนี้ Payment Date น้อยไปมาก XWT Ref No น้อยไปมาก	Pass
13	Web	SA	Order Management	Failed Switching Order	- ตรวจสอบการ Filter และ Sort ข้อมูล	- สามารถ Filter และ Sort ข้อมูลได้	Pass

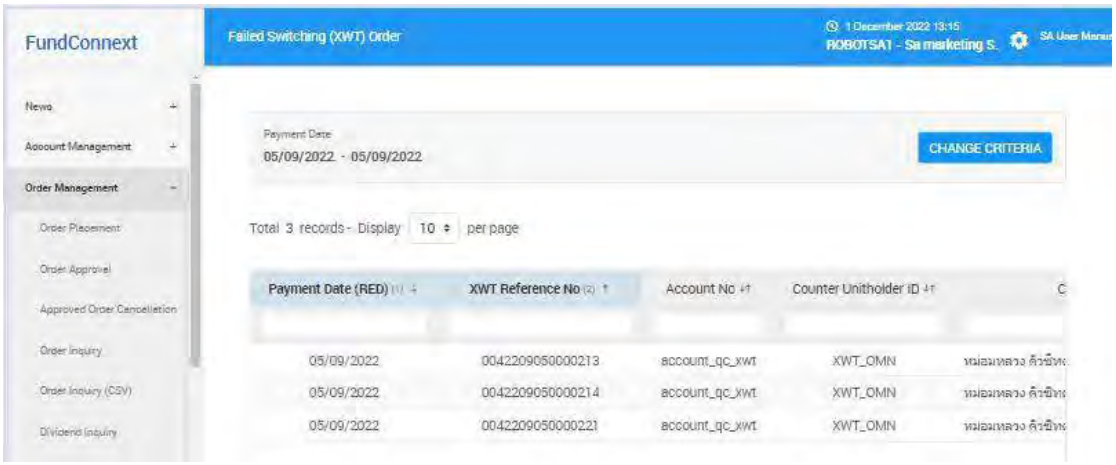
ผลลัพธ์การรัน Automated Test Script แบบ Look and Field (ตาราง) ตัวอย่างที่ 1



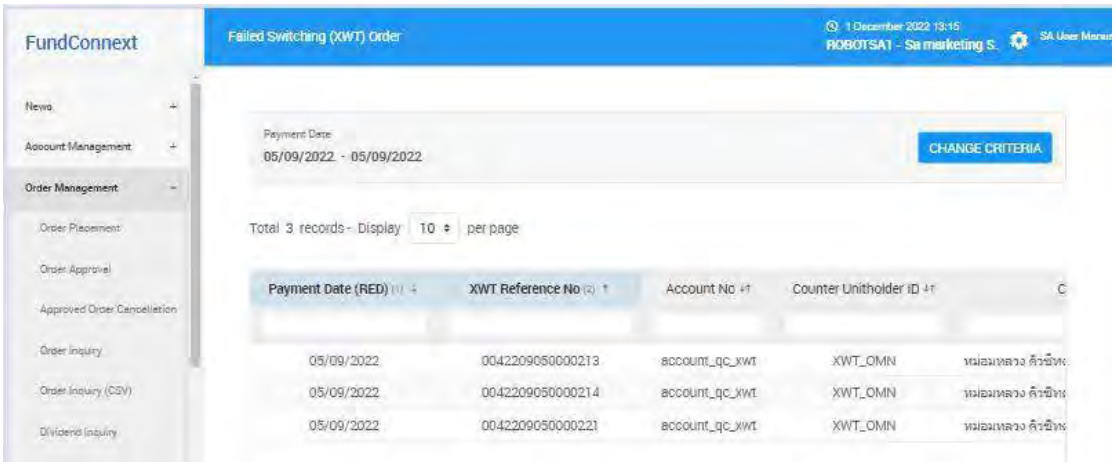
รูปที่ 4.73 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 1 แสดงDefault Sort หน้าจอ Failed XWT Order



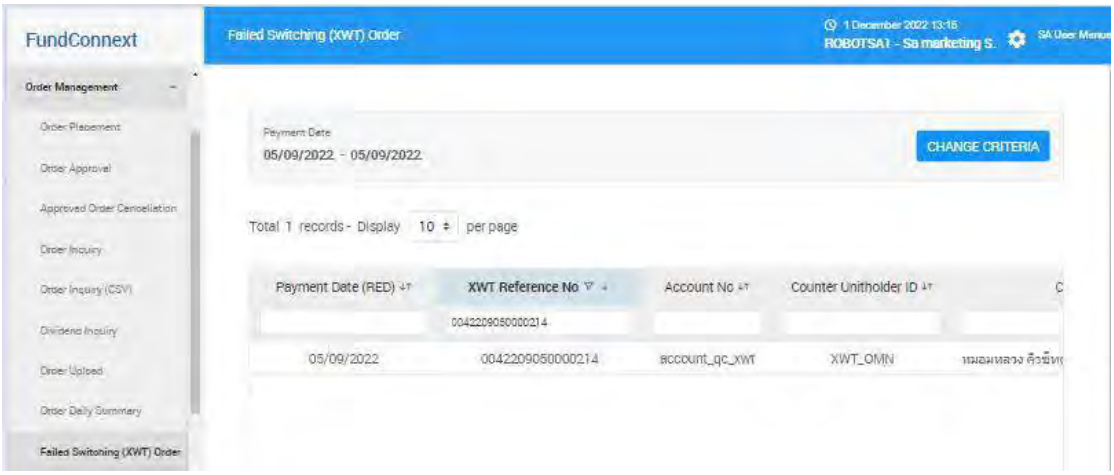
รูปที่ 4.74 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 1 กรณีที่ 1 (1)



รูปที่ 4.75 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 1 กรณีที่ 1 (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 2 แสดงข้อมูลหน้าจอ Failed XWT Order เมื่อ Filter



รูปที่ 4.76 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 1 กรณีที่ 2

ตัวอย่างที่ 2 การทดสอบ Automated Test แบบ Look and Field (ตาราง)

ตารางที่ 4.8 ผลการทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result	Test Result
1	Web	SA	Account Management	Account Balance Inquiry	- ตรวจสอบหัวข้อตารางเมนู Account Balance Inquiry	- แสดงหัวข้อตารางได้ถูกต้อง	Pass
2	Web	SA	Account Management	Account Balance Inquiry	- ตรวจสอบ Criteria เมื่อ Login : HQ	- แสดง Criteria Branch Input Team Input IC Input Investor Input	Pass
3	Web	SA	Account Manage	Account Balance	- ตรวจสอบข้อมูลในตาราง	- แสดงข้อมูล Account Balance	Pass

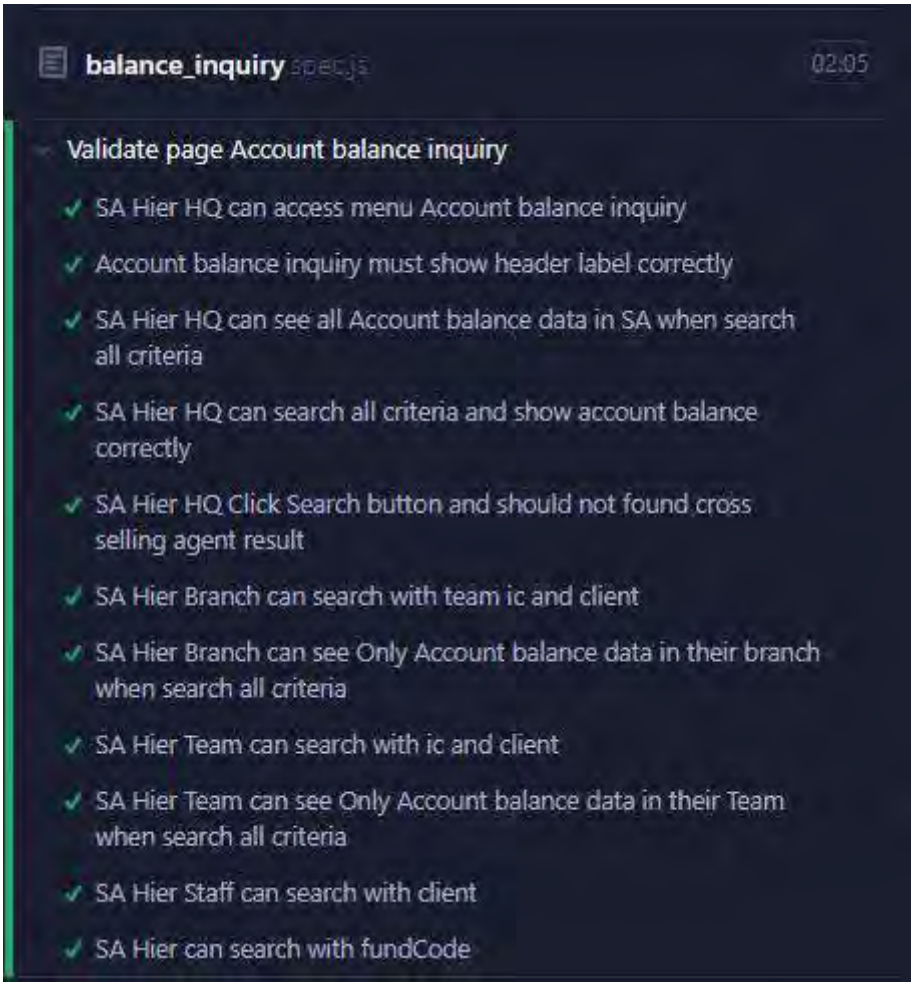
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือเพื่อการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			ment	Inquiry	เมื่อ Login : HQ ไม่ Criteria	ทั้งหมดในSA ได้ถูกต้อง	
4	Web	SA	Account Management	Account Balance Inquiry	- ตรวจสอบข้อมูลในตารางเมื่อ Login : HQ Criteria : Branch/Team/IC	- แสดงข้อมูล Account Balance ตามCriteria ได้ถูกต้อง	Pass
5	Web	SA	Account Management	Account Balance Inquiry	- ตรวจสอบ Criteria เมื่อ Login : Branch	- แสดง Criteria Team Input IC Input Investor Input	Pass
6	Web	SA	Account Management	Account Balance Inquiry	- ตรวจสอบข้อมูลในตารางเมื่อ Login : Branch ไม่ Criteria	- แสดงข้อมูล Account Balance ทั้งหมดในBranch เท่านั้น	Pass
7	Web	SA	Account Management	Account Balance Inquiry	- ตรวจสอบ Criteria เมื่อ Login : Team	- แสดง Criteria IC Input Investor Input	Pass
8	Web	SA	Account Management	Account Balance Inquiry	- ตรวจสอบข้อมูลในตารางเมื่อ Login : Branch ไม่ Criteria	- แสดงข้อมูล Account Balance ทั้งหมดในTeam เท่านั้น	Pass
9	Web	SA	Account Management	Account Balance Inquiry	- ตรวจสอบ Criteria เมื่อ Login : Staff	- แสดง Criteria Investor Input	Pass
10	Web	SA	Account	Account	- ตรวจสอบ	- แสดง Criteria	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			Manage ment	Balance Inquiry	Criteria Fund Code	FundCode ในทุกระดับของUser	

ผลลัพธ์การรัน Automated Test Script แบบ Look and Field (ตาราง) ตัวอย่างที่ 2



รูปที่ 4.77 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2

ผลการทดสอบกรณีที่ 1 แสดงหัวข้อตารางหน้าจอ Balance Inquiry

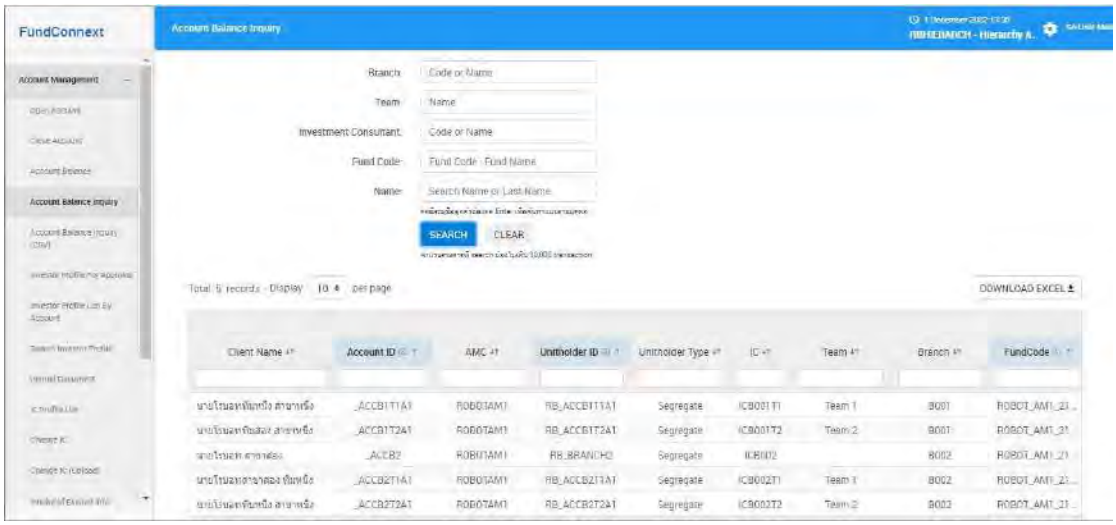
รูปที่ 4.78 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 2 แสดง Criteria เมื่อ Login : HQ ได้ถูกต้อง

รูปที่ 4.79 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 2

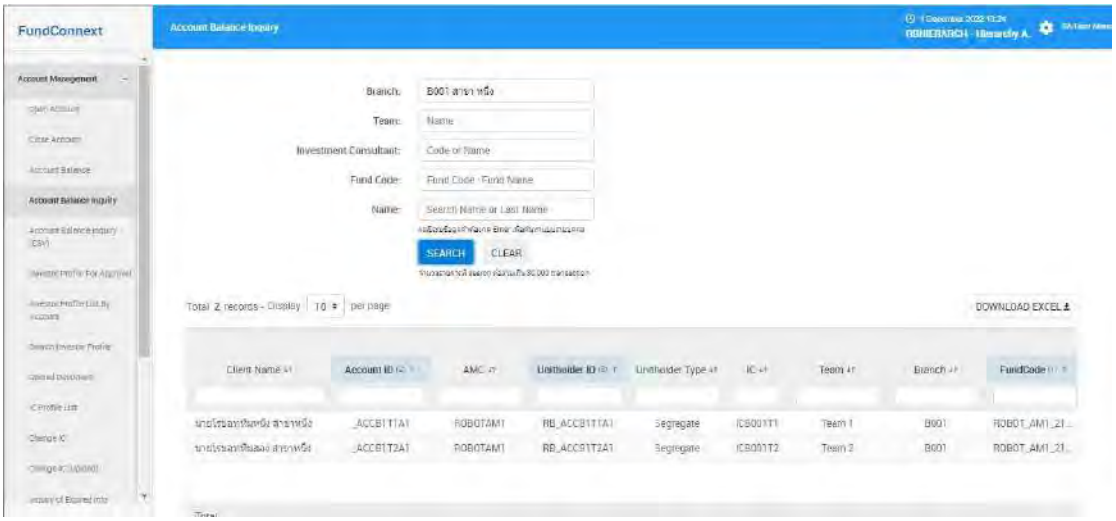
ผลการทดสอบกรณีที่ 3 แสดงข้อมูล Account Balance ทั้งหมด เมื่อ Login : HQ



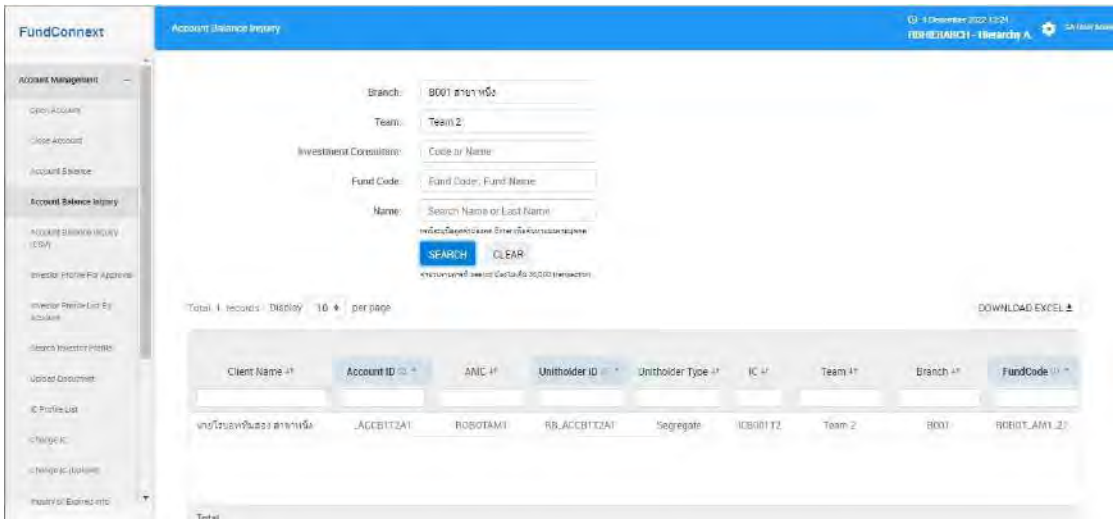
รูปที่ 4.80 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 4 แสดงข้อมูล Account Balance ตาม Criteria ถูกต้อง เมื่อ Login : HQ



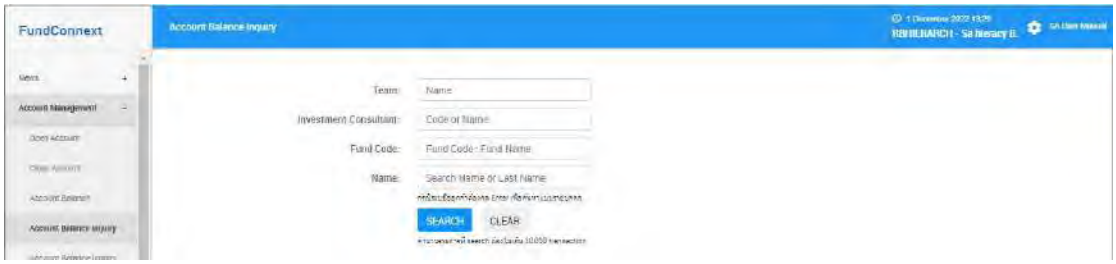
รูปที่ 4.81 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 4 (1)



รูปที่ 4.82 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 4 (2)

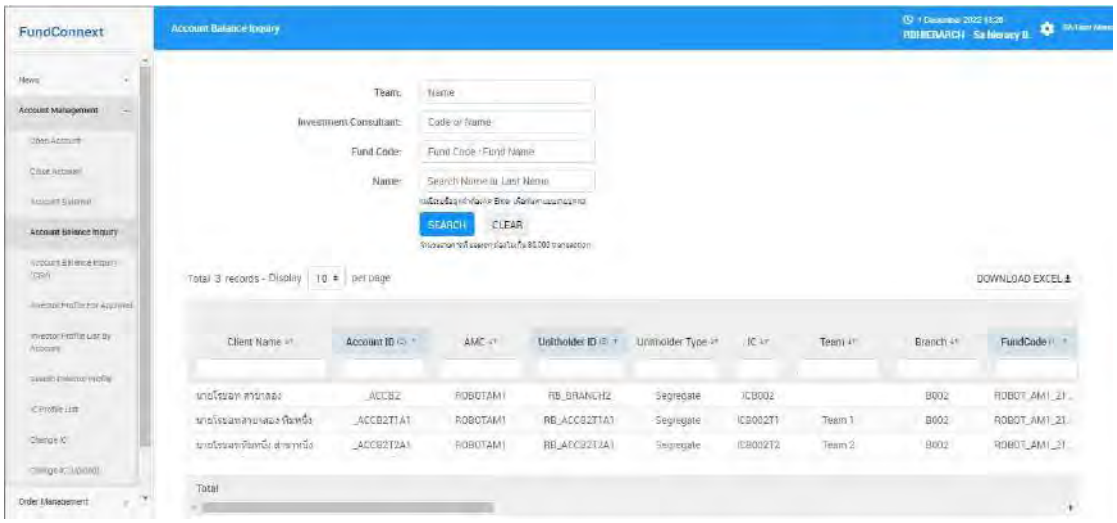
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 5 แสดง Criteria เมื่อ Login : Branch ได้ถูกต้อง



รูปที่ 4.83 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 5

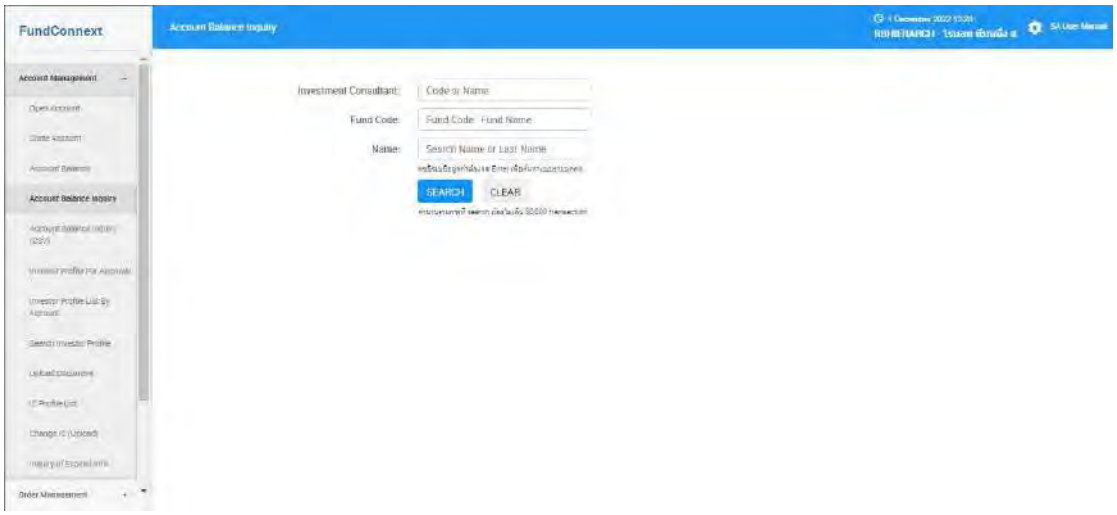
ผลการทดสอบกรณีที่ 6 แสดงข้อมูล Account Balance ของ Branch2 เมื่อ Login : Branch2



รูปที่ 4.84 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 6

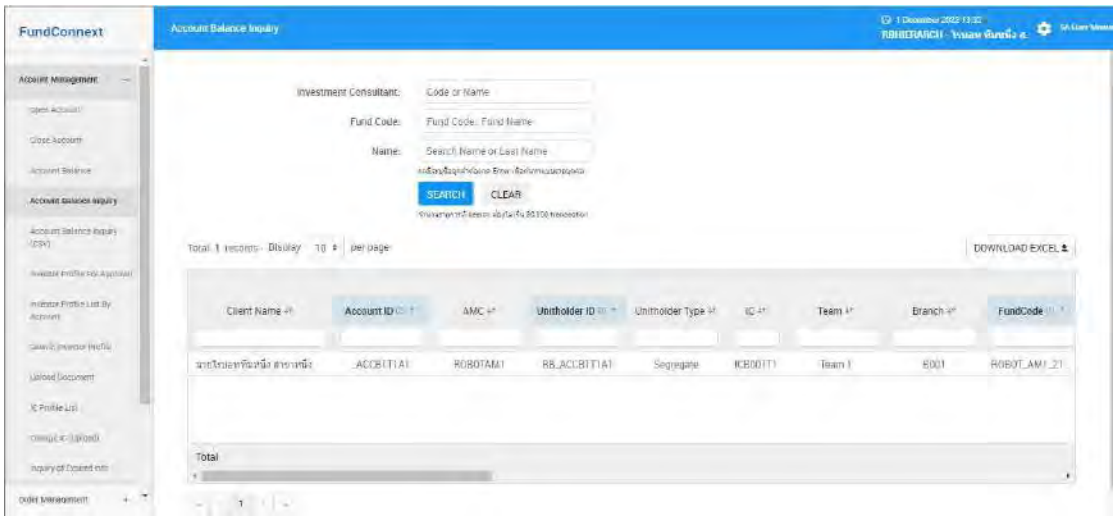
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 7 แสดง Criteria เมื่อ Login : Team ได้ถูกต้อง



รูปที่ 4.85 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 7

ผลการทดสอบกรณีที่ 8 แสดงข้อมูล Account Balance ของ Branch1Team1 เมื่อ Login : Branch1Team1



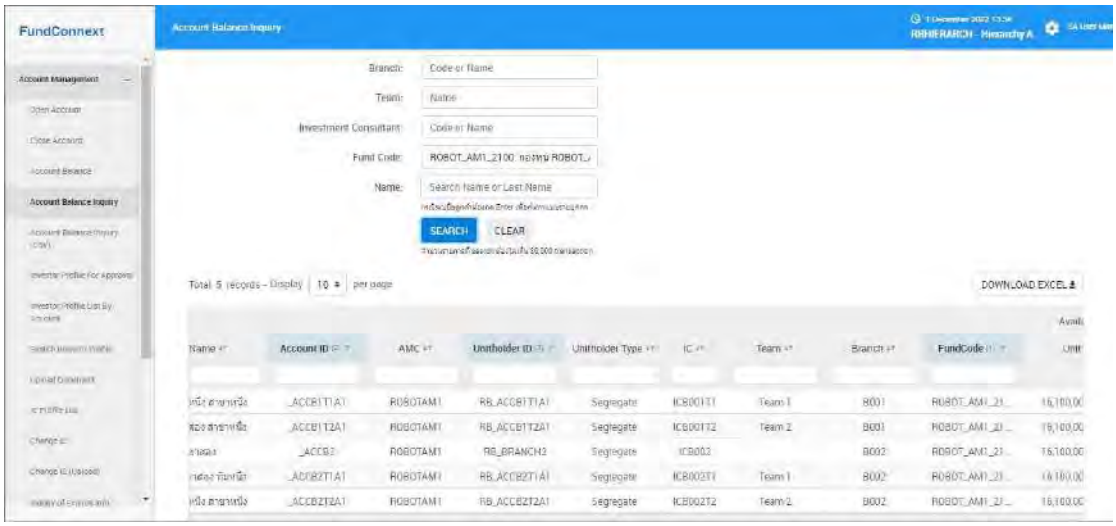
รูปที่ 4.86 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 9 แสดง Criteria เมื่อ Login : Staff ได้ถูกต้อง

รูปที่ 4.87 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 9

ผลการทดสอบกรณีที่ 10 แสดง Criteria FundCode



รูปที่ 4.88 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Look and Field (ตาราง) ตัวอย่างที่ 2 กรณีที่ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 ผลการทดสอบ Automated Test แบบ Validate File

ตัวอย่างที่ 1 การทดสอบ Automated Test แบบ Validate File

ตารางที่ 4.9 ผลการทดสอบ Automated Test แบบ Validate File ตัวอย่างที่ 1

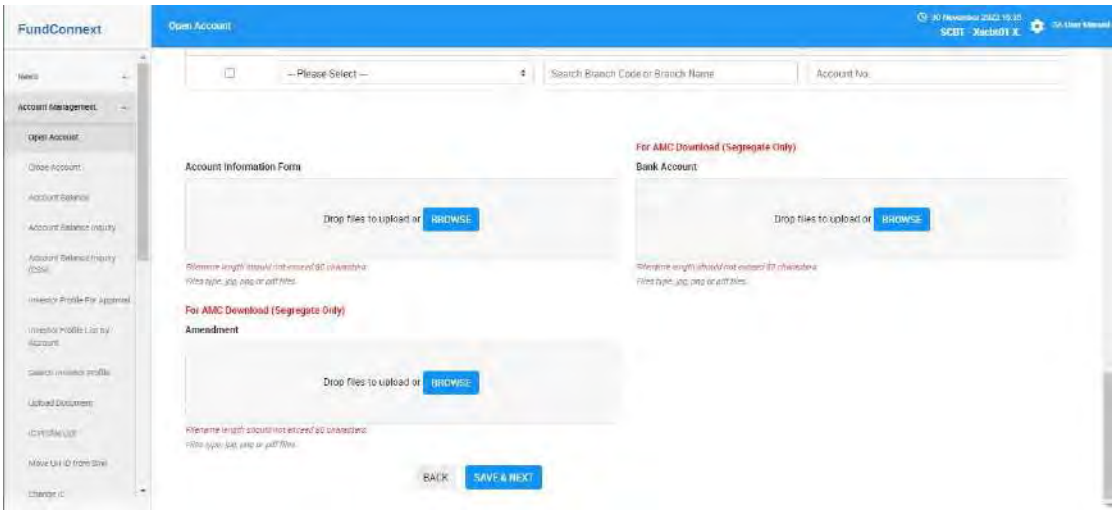
Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result	Test Result
1	Web	SA	Open Account	Individual	- ตรวจสอบข้อความใต้กล่อง Upload Attachment ใน Tab4(Account Tab) Tab5(Personal Document)	- แสดงข้อความสีแดง “Filename length should not exceed 80 Characters” ทุกกล่อง	Pass
2	Web	SA	Open Account	Individual	- Upload Attachment Filename > 80 ทุกกล่องใน Account Tab	- ไม่สามารถ Upload ไฟล์ได้ถูกต้อง - แสดง Popup “Filename length should not exceed 80 Characters”ได้ถูกต้อง	Pass
3	Web	SA	Open Account	Individual	- Upload Attachment Filename <= 80 ทุกกล่องใน Account Tab	- สามารถ Upload ไฟล์ได้ถูกต้อง	Pass
4	Web	SA	Open Account	Individual	- Upload Attachment Filename > 80 ทุกกล่องใน Personal Document Tab	- ไม่สามารถ Upload ไฟล์ได้ถูกต้อง - แสดง Popup “Filename length should not exceed 80 Characters”ได้ถูกต้อง	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์การรัน Automated Test Script แบบ Validate File ตัวอย่างที่ 1

รูปที่ 4.89 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Validate File ตัวอย่างที่ 1

ผลการทดสอบกรณีที่ 1 แสดงข้อความ “Filename length should not exceed 80 Characters”บริเวณใต้กล่องUpload



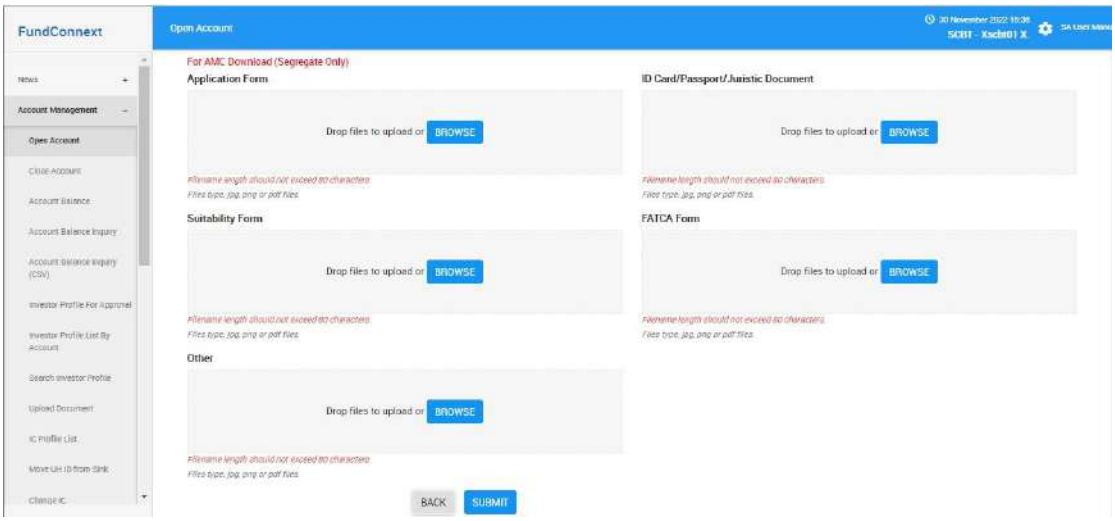
รูปที่ 4.90 หน้าจอผลลัพธ์การทดสอบAutomated Test แบบ Validate File ตัวอย่างที่ 1 กรณีที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 2 ไม่สามารถ Upload File ที่มีชื่อไฟล์มากกว่า 80 ตัวอักษร

รูปที่ 4.91 หน้าจอผลลัพธ์การทดสอบAutomated Test แบบ Validate File ตัวอย่างที่ 1 กรณีที่ 2

ผลการทดสอบกรณีที่ 3 แสดงข้อความ “Filename length should not exceed 80 Characters” บริเวณใต้กล่อง Upload ใน PersonalDoc information



รูปที่ 4.92 หน้าจอผลลัพธ์การทดสอบAutomated Test แบบ Validate File ตัวอย่างที่ 1 กรณีที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 4 ไม่สามารถ Upload File ที่มีชื่อไฟล์มากกว่า 80 ตัวอักษร ใน PersonalDoc information

รูปที่ 4.93 หน้าจอผลลัพธ์การทดสอบAutomated Test แบบ Validate File ตัวอย่างที่ 1 กรณีที่ 4

ตัวอย่างที่ 2 การทดสอบ Automated Test แบบ Validate File

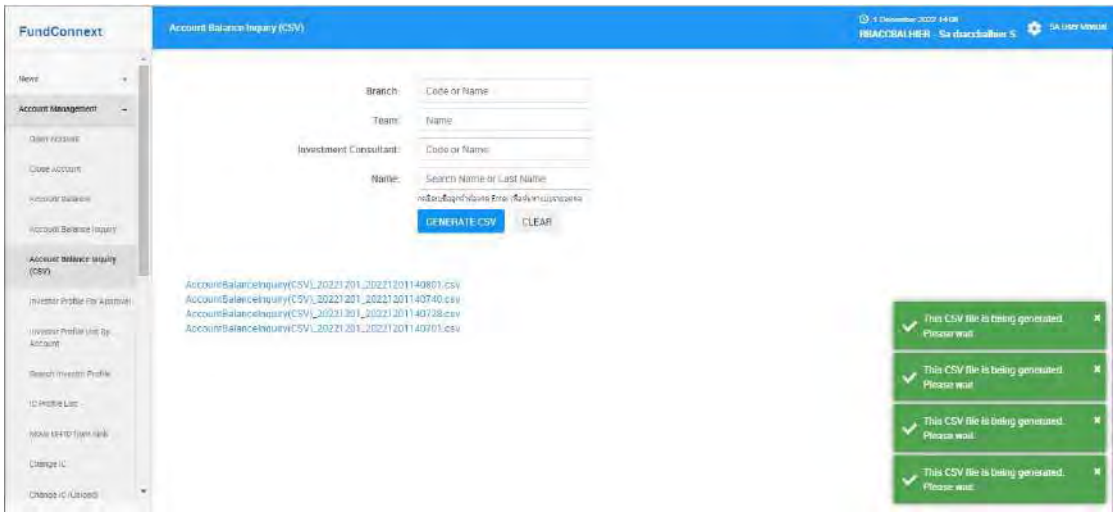
ตารางที่ 4.10 ผลการทดสอบ Automated Test แบบ Validate File ตัวอย่างที่ 2

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result	Test Result
1	Web	SA	Account Management	Balance Inquiry CSV	- ตรวจสอบไฟล์ที่ได้รับเมื่อ Generate CSV โดนไม่ใส่ Criteria	- แสดงไฟล์ที่ได้รับมาถูกต้อง	Pass
2	Web	SA	Account Management	Balance Inquiry CSV	- ตรวจสอบไฟล์ที่ได้รับเมื่อ Generate CSV โดน Criteria : Branch	- แสดงไฟล์ที่ได้รับมาถูกต้อง	Pass
3	Web	SA	Account Management	Balance Inquiry CSV	- ตรวจสอบไฟล์ที่ได้รับเมื่อ Generate CSV	- แสดงไฟล์ที่ได้รับมาถูกต้อง	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

					โดน Criteria : Investment Consultant		
4	Web	SA	Account Manage ment	Balance Inquiry CSV	- ตรวจสอบไฟล์ที่ได้รับเมื่อ Generate CSV โดน Criteria : Client Name	- แสดงไฟล์ที่ได้รับมาถูกต้อง	Pass

ไฟล์ CSV ที่ได้รับมีข้อมูลถูกต้อง เมื่อ Criteria ตามกรณีต่างๆ



รูปที่ 4.94 หน้าจอผลลัพธ์การทดสอบ Automated Test แบบ Validate File ตัวอย่างที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ผลการทดสอบ Automated Test API

ตัวอย่างที่ 1 การทดสอบ Automated Test API

ตารางที่ 4.11 ผลการทดสอบ Automated Test API ตัวอย่างที่ 1

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result	Test Result
1	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อไม่ใส่ token	- ไม่สามารถทำรายการได้	Pass
2	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อใส่ Account No	- แสดงรายการที่ Gen Sub ไม่สำเร็จของ account No นั้น	Pass
3	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อใส่ Payment Date from/to	- แสดงรายการที่ Gen Sub ไม่สำเร็จของ Payment Date นั้น	Pass
4	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อใส่ XWT Ref No.	- แสดงรายการที่ Gen Sub ไม่สำเร็จของ XWT Ref No. นั้น	Pass
5	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อใส่ Account No และ Payment Date from/to	- แสดงรายการที่ Gen Sub ไม่สำเร็จของ Account No และ Payment Dateที่ระบุนั้น	Pass

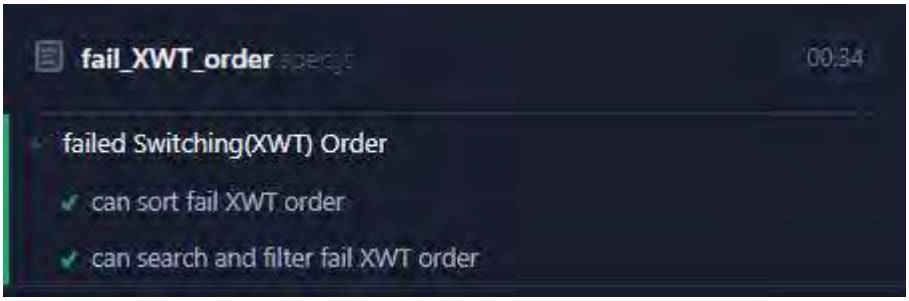
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อใส่ Account No และ XWT Ref No	- แสดงรายการที่ Gen Sub ไม่สำเร็จของ Account No และ XWT Ref Noที่ระบุนั้น	Pass
7	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อใส่ Payment Date to/from และ XWT Ref No	- แสดงรายการที่ Gen Sub ไม่สำเร็จของ Payment Date และ XWT Ref Noที่ระบุนั้น	Pass
8	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อใส่ Account No, Payment Date to/from และ XWT Ref No	- แสดงรายการที่ Gen Sub ไม่สำเร็จของ Account No, Payment Date และ XWT Ref Noที่ระบุนั้น	Pass
9	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อ Criteria รายการที่ไม่มีในระบบ	- ไม่แสดงรายการได้ถูกต้อง	Pass
10	API	SA	Failed Switching Order		- Send API Failed Switching Order เมื่อใส่ Payment Date to	- ไม่แสดงรายการได้ถูกต้อง - แสดง Error payment Date from is required field	Pass
11	API	SA	Failed Switching		- Send API Failed	- ไม่แสดงรายการได้	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

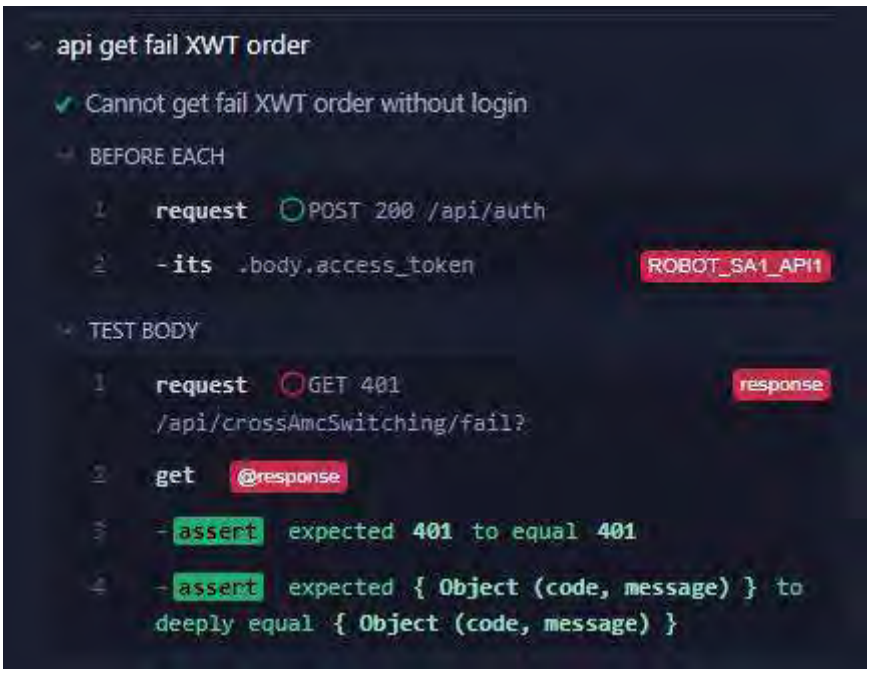
			Order		Switching Order เมื่อใส่ Payment Date from	ถูกต้อง - แสดง Error payment Date to is required field	

ผลลัพธ์การรัน Automated Test Script API ตัวอย่างที่ 1



รูปที่ 4.95 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1

ผลการทดสอบกรณีที่ 1 ไม่สามารถ Send API get Failed XWT Order ได้หากไม่ทำการ Login

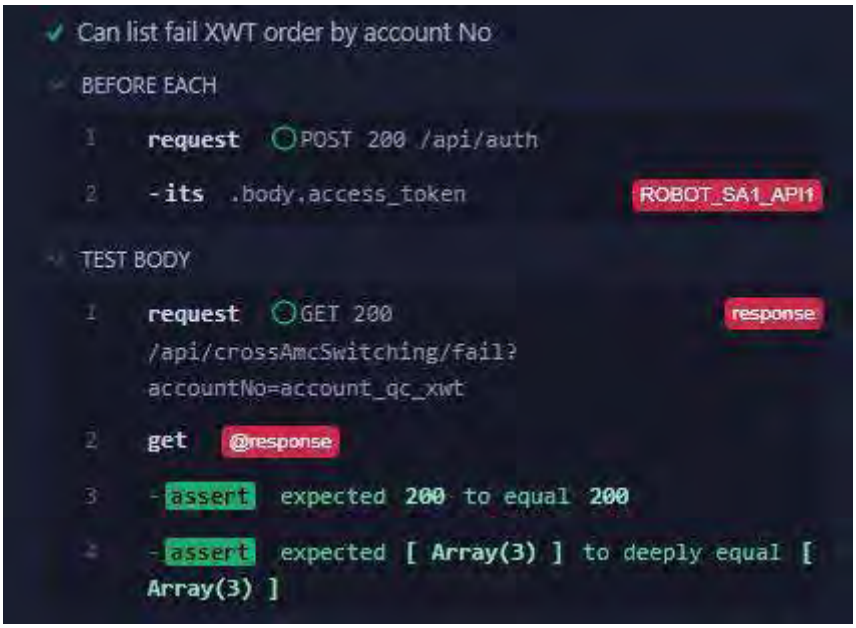


รูปที่ 4.96 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 2 Send API get Failed XWT Order

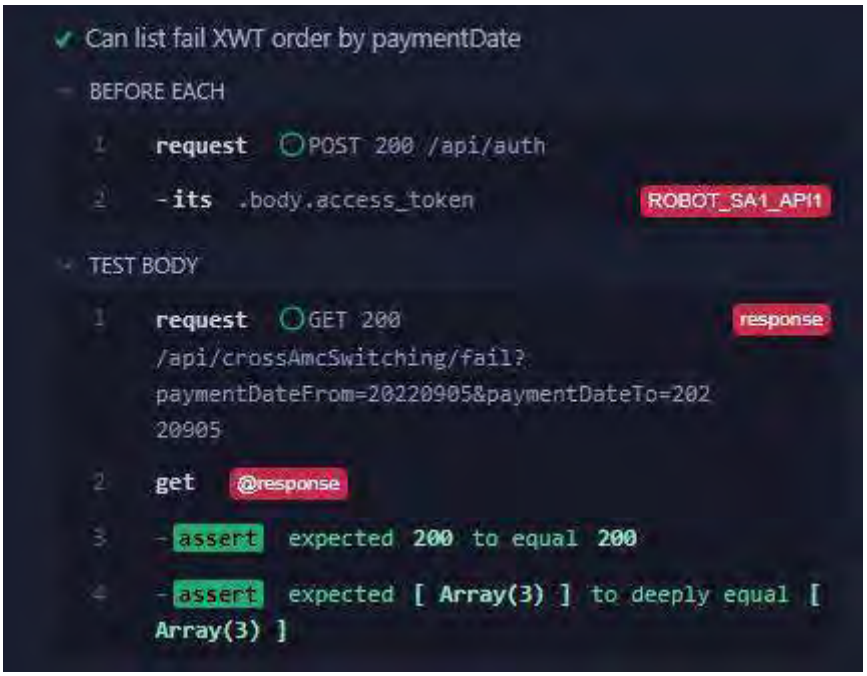
โดยค้นหาด้วย Account No : account_qc_xwt



รูปที่ 4.97 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 2

ผลการทดสอบกรณีที่ 3 Send API get Failed XWT Order

โดยค้นหาด้วย Payment Date From/To



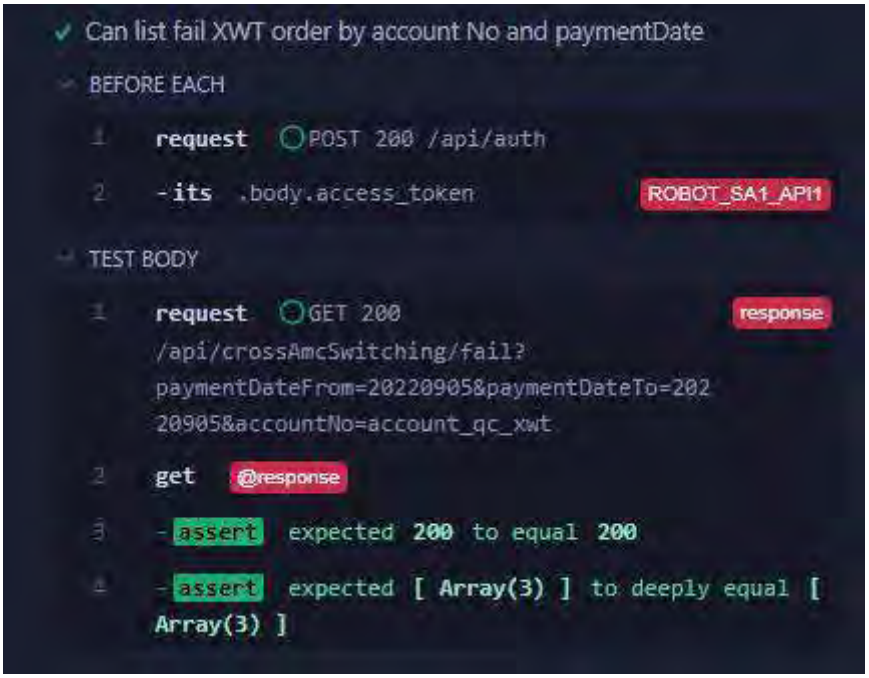
รูปที่ 4.98 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 4 Send API get Failed XWT Order
โดยค้นหาด้วย XWT Ref No.

รูปที่ 4.99 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 4

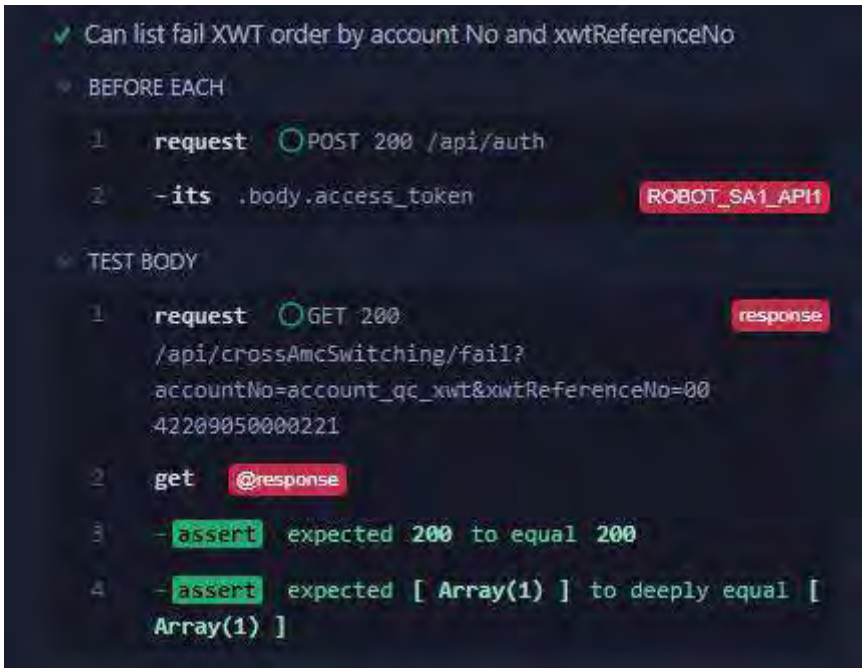
ผลการทดสอบกรณีที่ 5 Send API get Failed XWT Order
โดยค้นหาด้วย Account No. และ Payment Date From/To



รูปที่ 4.100 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 5

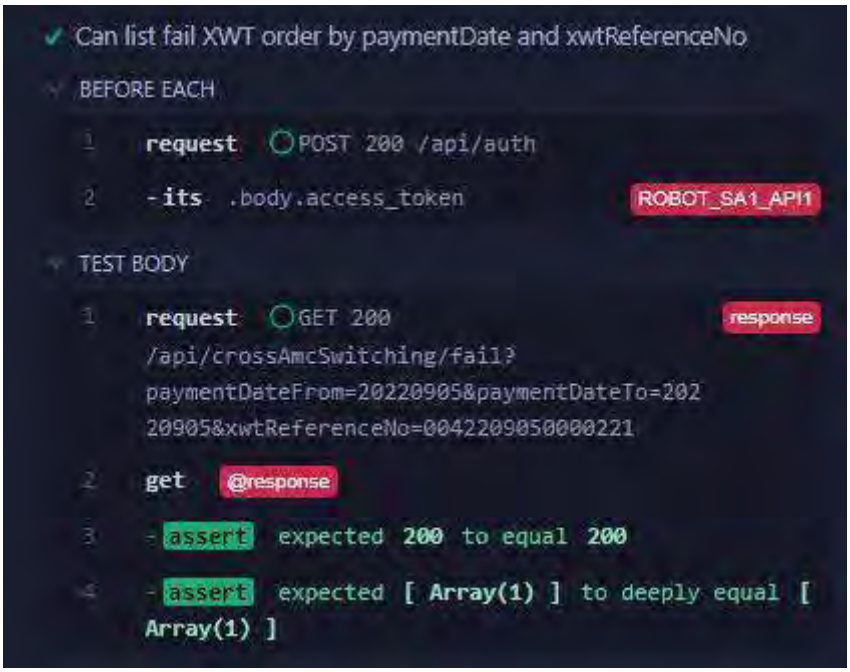
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 6 Send API get Failed XWT Order
โดยค้นหาด้วย Account No. และ XWT Ref No.



รูปที่ 4.101 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 6

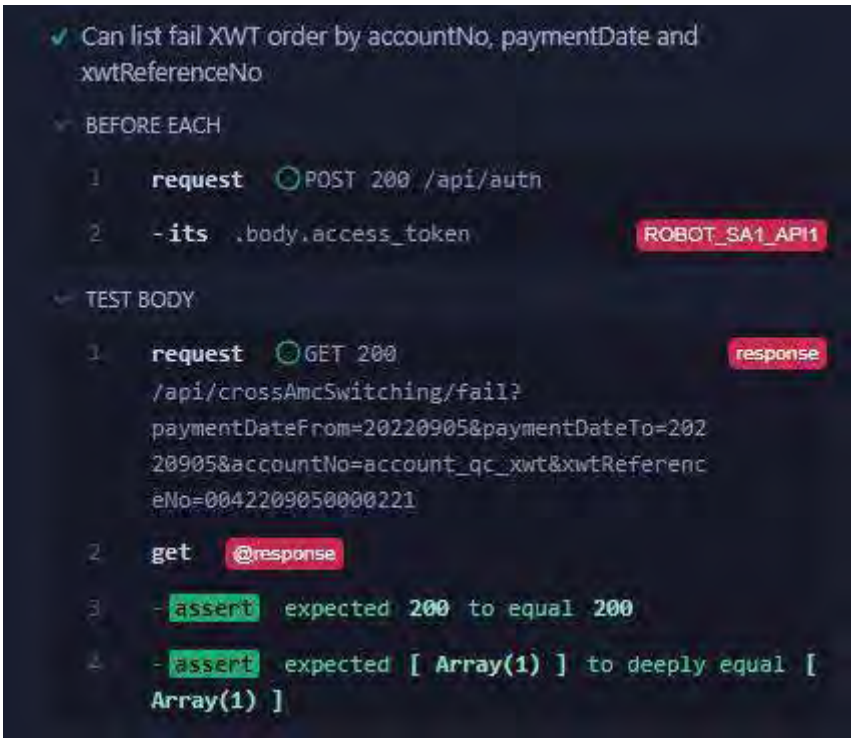
ผลการทดสอบกรณีที่ 7 Send API get Failed XWT Order
โดยค้นหาด้วย Payment Date From/To และ XWT Ref No.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

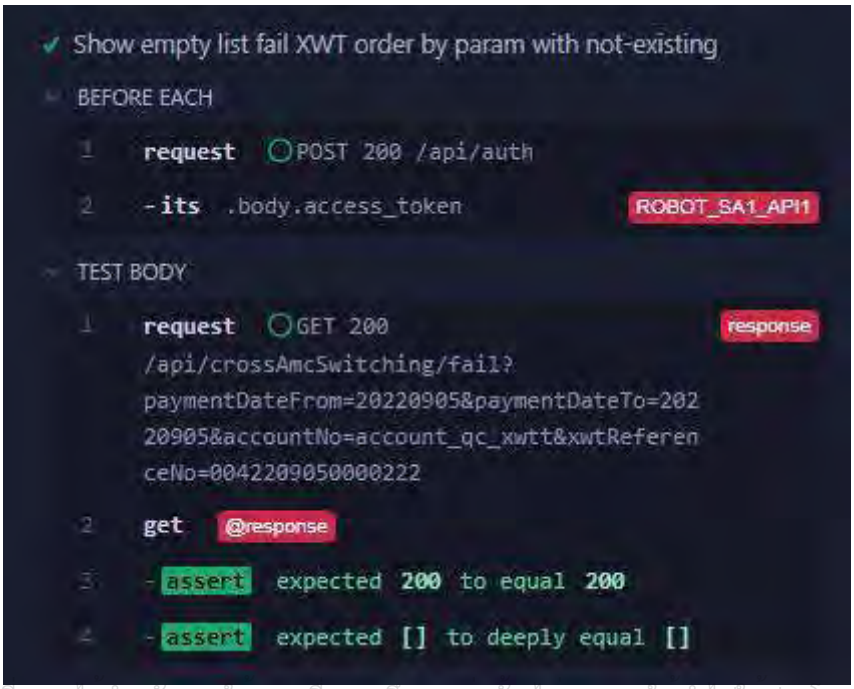
รูปที่ 4.102 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 7

ผลการทดสอบกรณีที่ 8 Send API get Failed XWT Order โดยค้นหาด้วย Account No, Payment Date From/To และ XWT Ref No.



รูปที่ 4.103 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 8

ผลการทดสอบกรณีที่ 9 Send API get Failed XWT Order โดยค้นหาด้วยรายการที่ไม่มีในระบบ



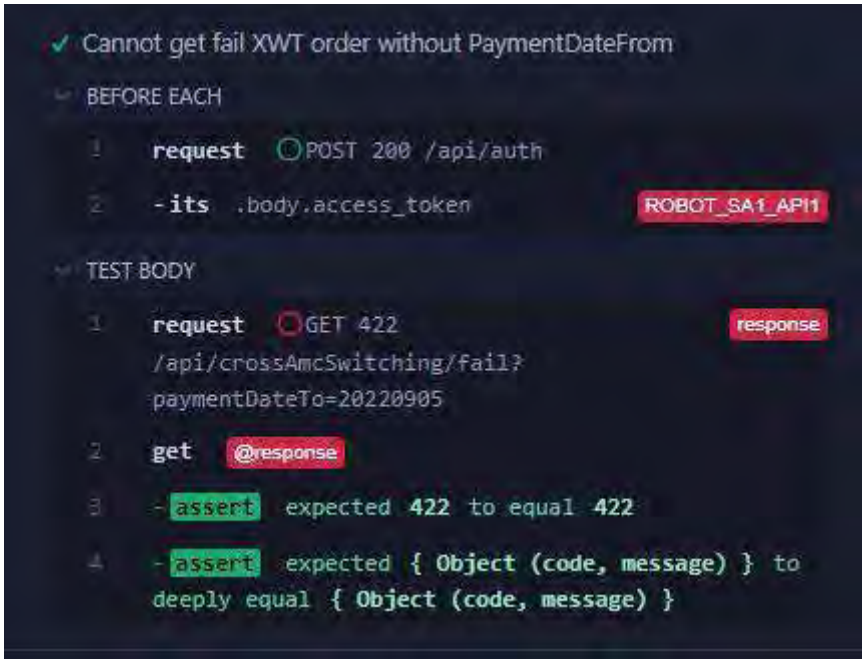
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.104 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 9

ผลการทดสอบกรณีที่ 10 ไม่สามารถ Send API get Failed XWT Order
โดยค้นหาด้วย Paymend Date To

รูปที่ 4.105 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 10

ผลการทดสอบกรณีที่ 11 ไม่สามารถ Send API get Failed XWT Order
โดยค้นหาด้วย Paymend Date From



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.106 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 1 กรณีที่ 11

ตัวอย่างที่ 2 การทดสอบ Automated Test API

ตารางที่ 4.12 ผลการทดสอบ Automated Test API ตัวอย่างที่ 2

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result	Test Result
1	API	SA	Approve cross amc switching		- Send API Approve โดยใส่ Transaction ID ที่ไม่มีในระบบ	- แสดงข้อความ “Transaction ID cannot be found”	Pass
2	API	SA	Approve cross amc switching		- Send API Approve โดยใส่ Transaction ID ที่มี Status Submit	- สามารถApprove รายการดังกล่าวได้	Pass
3	API	SA	Approve cross amc switching		- Send API Approve โดยไม่ใส่ Status	- แสดงข้อความ “Status is required field” - ไม่สามารถApprove รายการดังกล่าวได้	Pass
4	API	SA	Approve cross amc switching		- Send API Approve โดยใส่ Status ไม่ใช่ Approve, Cancel	- แสดงข้อความ “Invalid field format(Status)” - ไม่สามารถApprove รายการดังกล่าวได้	Pass
5	API	SA	Approve cross amc switching		- Send API Approve โดย Login : SA passthrough	- ไม่สามารถApprove รายการดังกล่าวได้	Pass
6	API	SA	Approve cross amc switching		- Send API Approve โดยไม่ Login	- ไม่สามารถApprove รายการดังกล่าวได้	Pass
7	API	SA	Approve cross amc		- Send API Approve โดยใส่	- แสดงข้อความ “Transaction is	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			switching		Transaction ID ที่มี Status Approved	already approved” - ไม่สามารถApprove รายการดังกล่าวได้	

ผลลัพธ์การรัน Automated Test Script API ตัวอย่างที่ 2

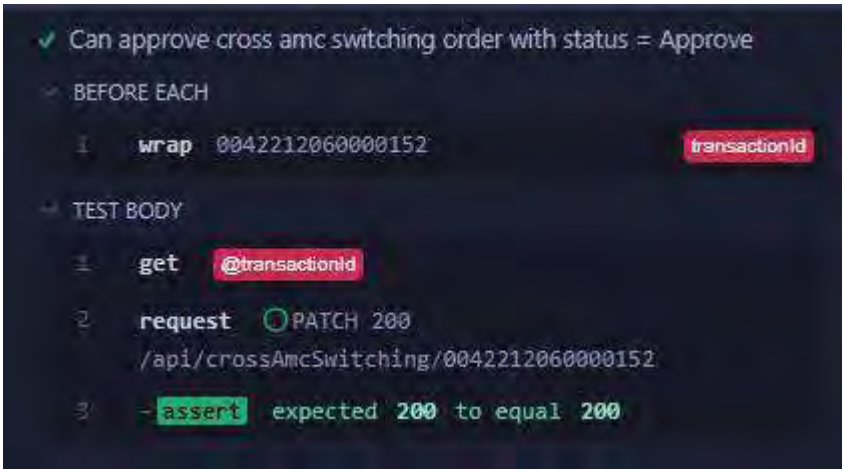
รูปที่ 4.107 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 1 ไม่สามารถ Send API Approve CrossAmcSwitching ได้
เมื่อ Transaction ไม่มีในระบบ

รูปที่ 4.108 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 2 กรณีที่ 1

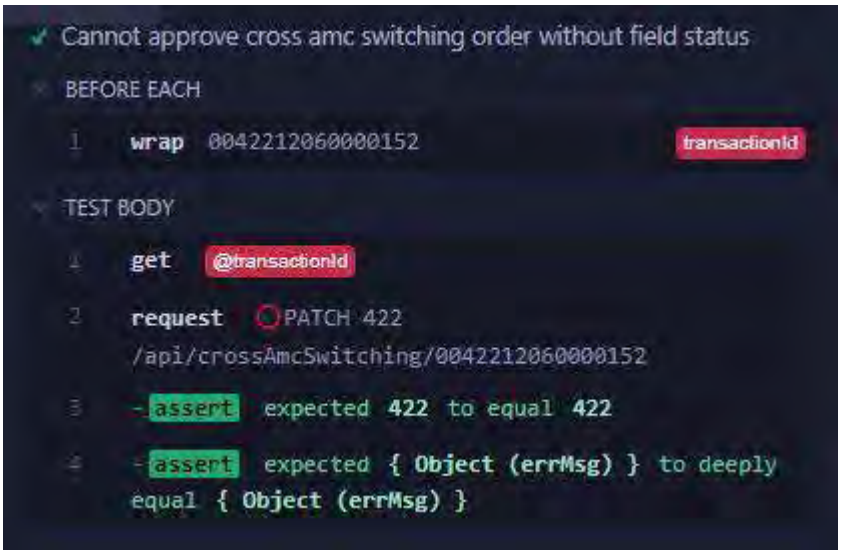
ผลการทดสอบกรณีที่ 2 Send API Approve CrossAmcSwitching ด้วย Transactionที่มีในระบบ
และ Status เป็น Submit



รูปที่ 4.109 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 2 กรณีที่ 2

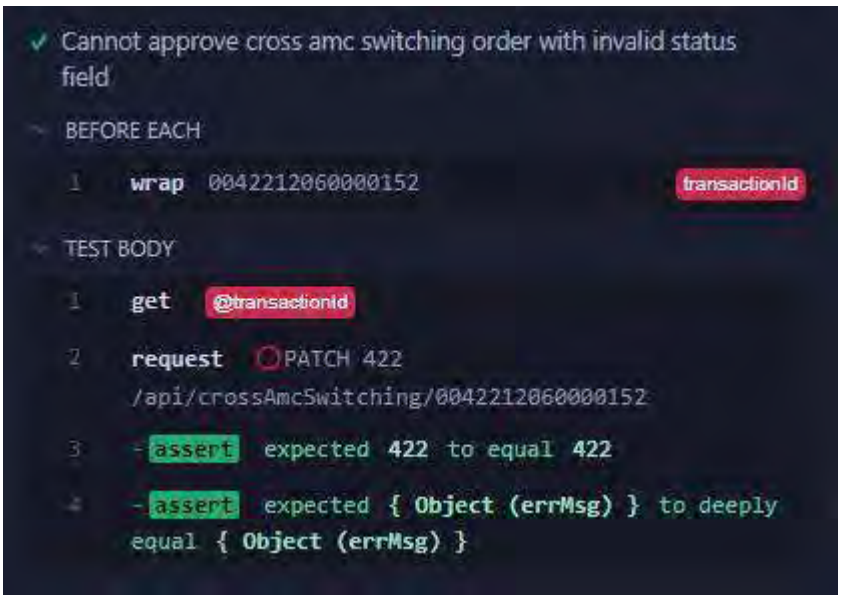
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 3 ไม่สามารถ Send API Approve CrossAmcSwitching ได้
โดยไม่ใส่ข้อมูลStatus



รูปที่ 4.110 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 2 กรณีที่ 3

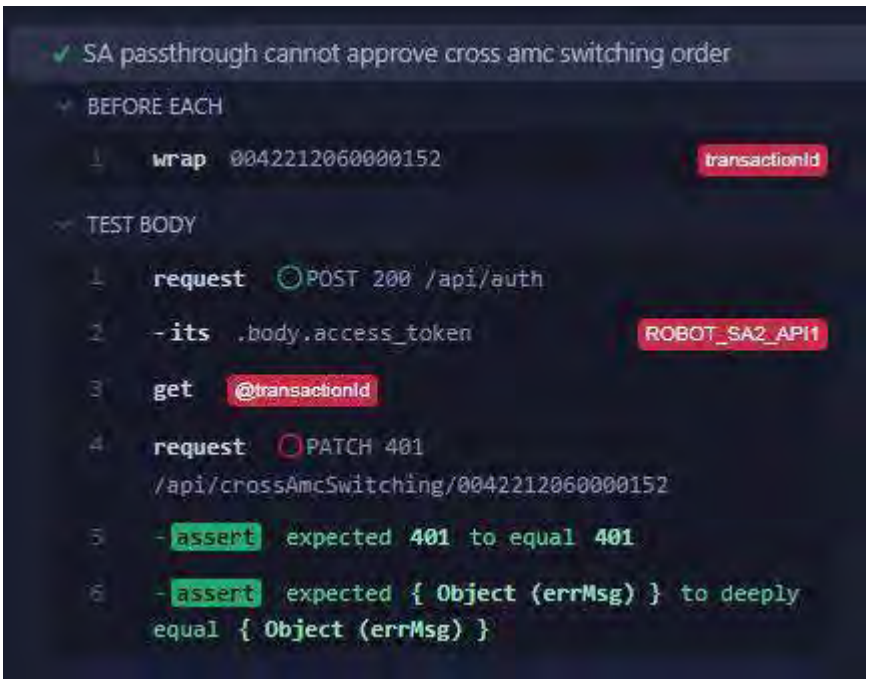
ผลการทดสอบกรณีที่ 4 ไม่สามารถ Send API Approve CrossAmcSwitching
โดยไม่ใส่ข้อมูล Status



รูปที่ 4.111 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 2 กรณีที่ 4

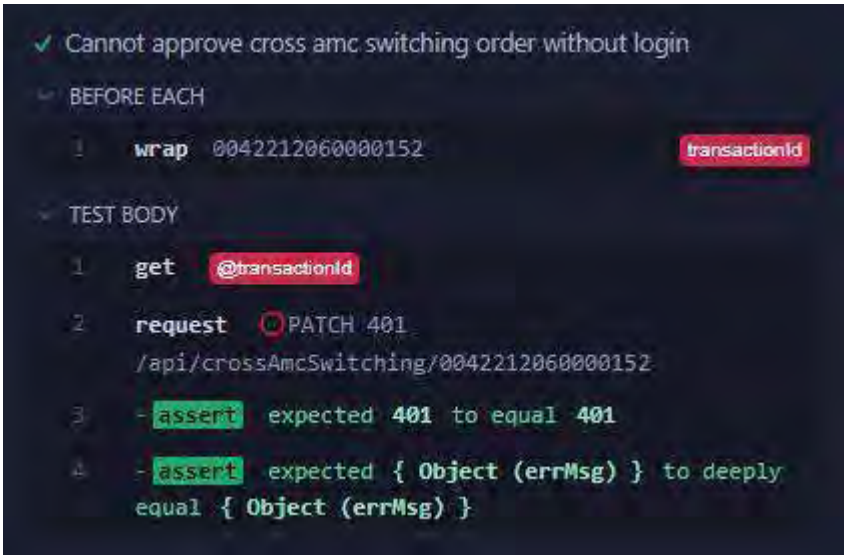
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 5 ไม่สามารถ Send API Approve CrossAmcSwitching
เมื่อ Login ด้วย SA Passthrough



รูปที่ 4.112 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 2 กรณีที่ 5

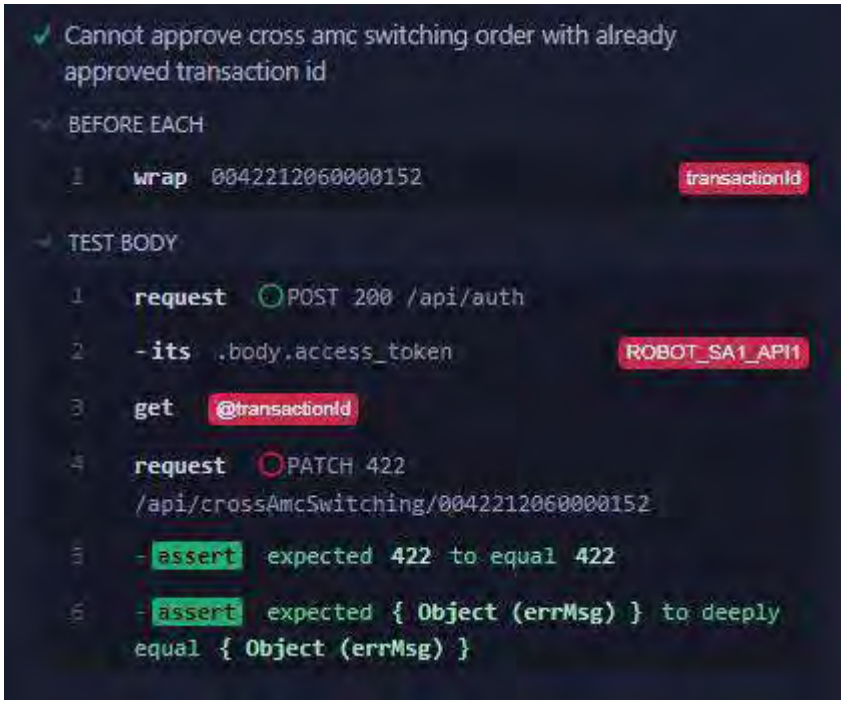
ผลการทดสอบกรณีที่ 6 ไม่สามารถ Send API Approve CrossAmcSwitching เมื่อไม่ Login



รูปที่ 4.113 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 2 กรณีที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 7 ไม่สามารถ Send API Approve CrossAmcSwitching
เมื่อ Transactionนั้นมี Status เป็น approve



รูปที่ 4.114 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 2 กรณีที่ 7

ตัวอย่างที่ 3 การทดสอบ Automated Test API

ตารางที่ 4.13 ผลการทดสอบ Automated Test API ตัวอย่างที่ 3

Test case ID	Channel	User	Function	Sub Function	Test Case	Expected Result	Test Result
1	API	SA	Update bank unitholder		- Send API โดย Unitholder Type : OMN	- ไม่สามารถ Update bank ได้ - แสดงข้อความ”Cannot update bank Account due to this unitholder type is Omnibus Account”	Pass
2	API	SA	Update bank		- Send API โดย Update Bank	-สามารถ Update bank ได้	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			unitholder		SUB/RED ครบทั้ง 5 ธนาคาร		
3	API	SA	Update bank unitholder		- Send API โดย Update Bank เฉพาะ SUB	- ไม่สามารถ Update bank ได้ - แสดงข้อความ"redemptionBank is required field"	Pass
4	API	SA	Update bank unitholder		- Send API โดย Update Bank เฉพาะ RED	- ไม่สามารถ Update bank ได้ - แสดงข้อความ"subscriptionBank is required field"	Pass
5	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูล BankAccountNo (SUB) น้อยกว่า 10 char	- ไม่สามารถ Update bank ได้ - แสดงข้อความ"Invalid field format(subscriptionBankAccount)"	Pass
6	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูล BankAccount No (RED) น้อยกว่า 10 char	- ไม่สามารถ Update bank ได้ - แสดงข้อความ"Invalid field format(redemptionBankAccount)"	Pass
7	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูลเลือก Bank SCB และ BankAccount No (SUB) ไม่ถูกต้องตาม	- ไม่สามารถ Update bank ได้ - แสดงข้อความ"Invalid field format(Bank Account No)"	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

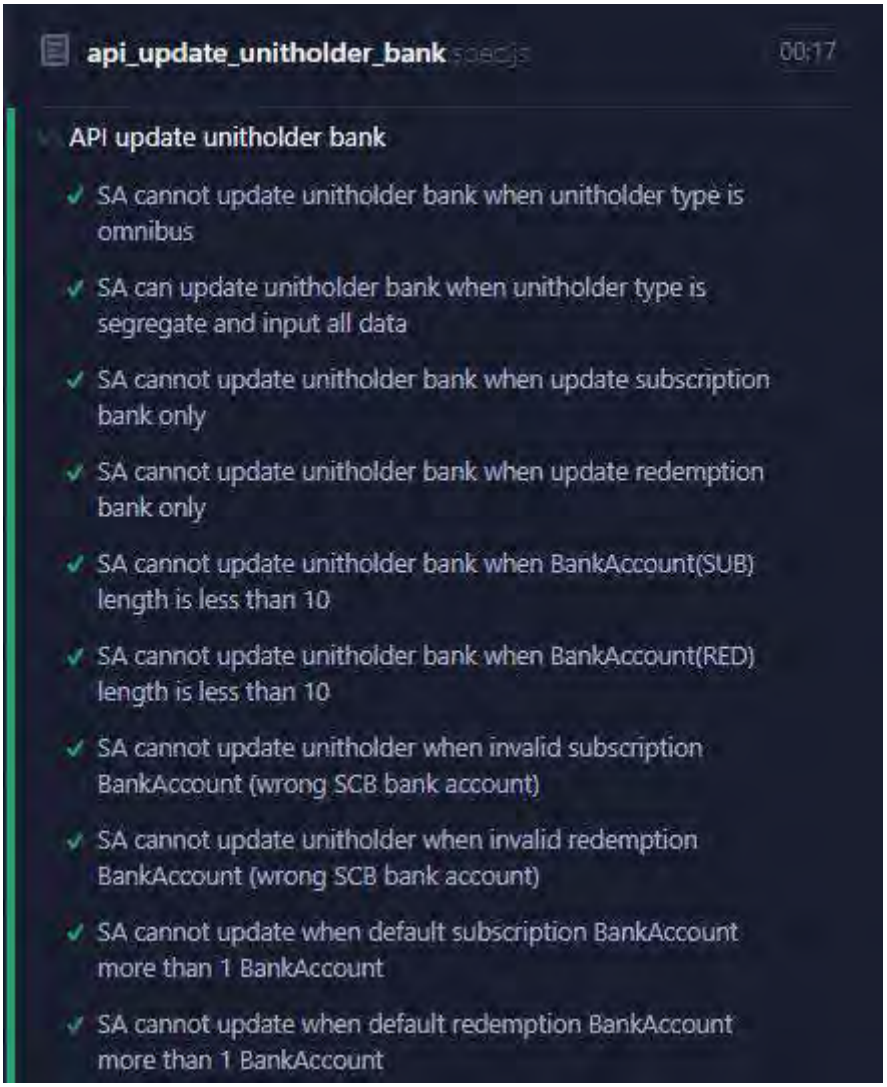
					Format SCB		
8	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูลเลือก Bank SCB และ BankAccount No (RED) ไม่ถูกต้องตาม Format SCB	- ไม่สามารถ Update bank ได้ - แสดงข้อความ"Invalid field format(Bank Account No)"	Pass
9	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูลเลือก Default Bank(SUB) มากกว่า 1 Bank	- ไม่สามารถ Update bank ได้ - แสดงข้อความ"subscriptionBankAccount.default is invalid"	Pass
10	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูลเลือก Default Bank(RED) มากกว่า 1 Bank	- ไม่สามารถ Update bank ได้ - แสดงข้อความ"redemptionBankAccount.default is invalid"	Pass
11	API	SA	Update bank unitholder		- Send API โดยข้อมูล Bank (SUB) มากกว่า 5 ธนาคาร	- ไม่สามารถ Update bank ได้ - แสดงข้อความ" subscriptionBankAccount.limitBanks is invalid"	Pass
12	API	SA	Update bank unitholder		- Send API โดยข้อมูล Bank (RED) มากกว่า 5 ธนาคาร	- ไม่สามารถ Update bank ได้ - แสดงข้อความ" redemptionBankAccount.limitBanks is invalid"	Pass
13	API	SA	Update bank		- Send API โดยใส่ข้อมูล Bank	- ไม่สามารถ Update bank ได้	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			unitholder		Code(SUB) ที่ไม่มีในระบบ	- แสดงข้อความ” subscriptionBankAccount.BankCode is invalid”	
14	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูล Bank Code(RED) ที่ไม่มีในระบบ	- ไม่สามารถ Update bank ได้ - แสดงข้อความ” redemptionBankAccount.BankCode is invalid”	Pass
15	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูล Bank Branch(SUB) ที่ไม่มีในระบบ	- ไม่สามารถ Update bank ได้ - แสดงข้อความ” subscriptionBankAccount.BankBranch is invalid”	Pass
16	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูล Bank Branch(RED) ที่ไม่มีในระบบ	- ไม่สามารถ Update bank ได้ - แสดงข้อความ” redemptionBankAccount.BankBranch is invalid”	Pass
17	API	SA	Update bank unitholder		- Send API โดยใส่ข้อมูล Unitholder ที่ไม่มีในระบบ	- ไม่สามารถ Update bank ได้ - แสดงข้อความ”Unitholder ID cannot be found”	Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

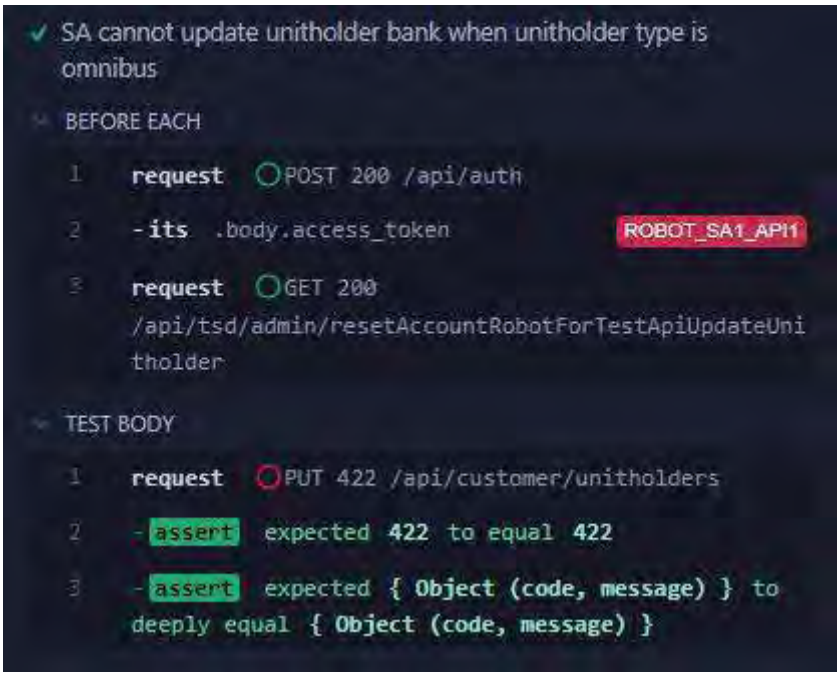
ผลลัพธ์การรัน Automated Test Script API ตัวอย่างที่ 3



รูปที่ 4.115 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3

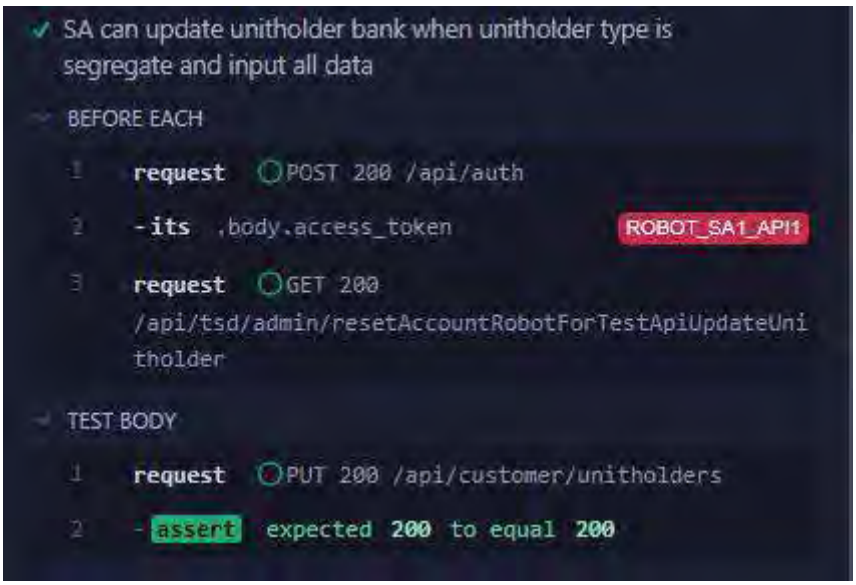
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 1 ไม่สามารถ Send API Update Unitholder Bank
เมื่อ UH Type เป็น Omnibus



รูปที่ 4.116 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 1

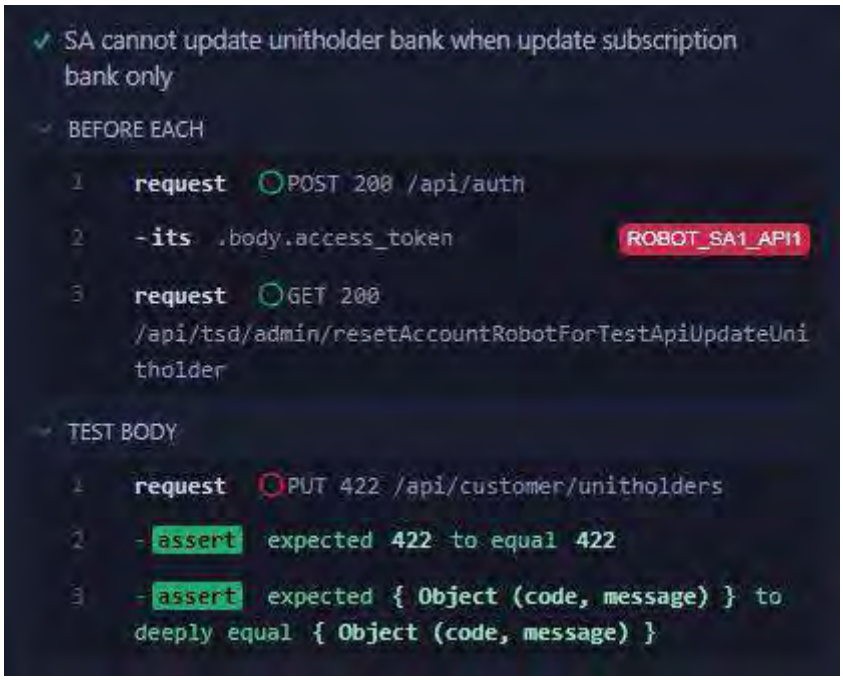
ผลการทดสอบกรณีที่ 2 สามารถ Send API Update Unitholder ได้
เมื่อใส่ข้อมูลBank ครบทั้ง 5 ธนาคาร



รูปที่ 4.117 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 2

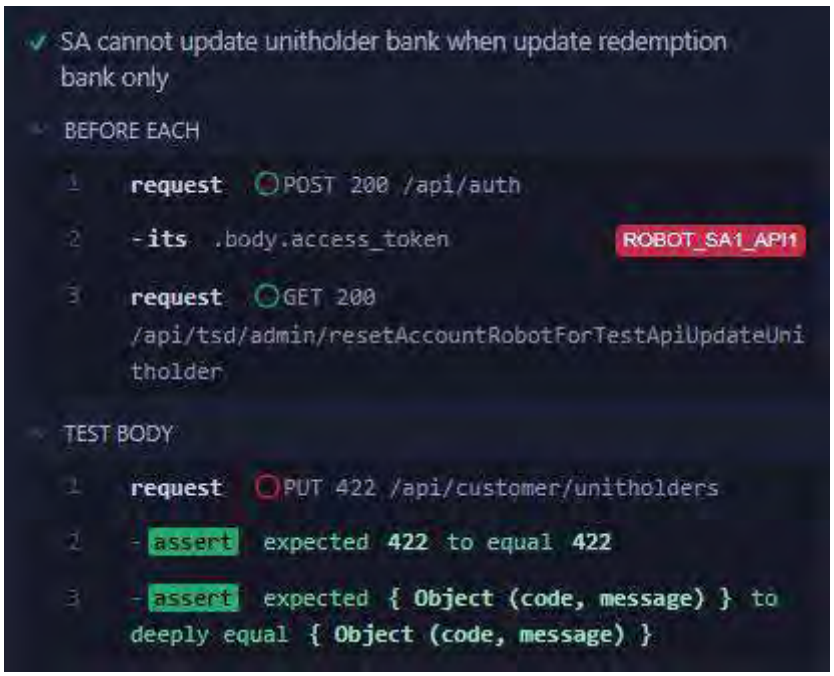
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 3 ไม่สามารถ Send API Update Unitholder Bank เฉพาะ Bank Sub



รูปที่ 4.118 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 3

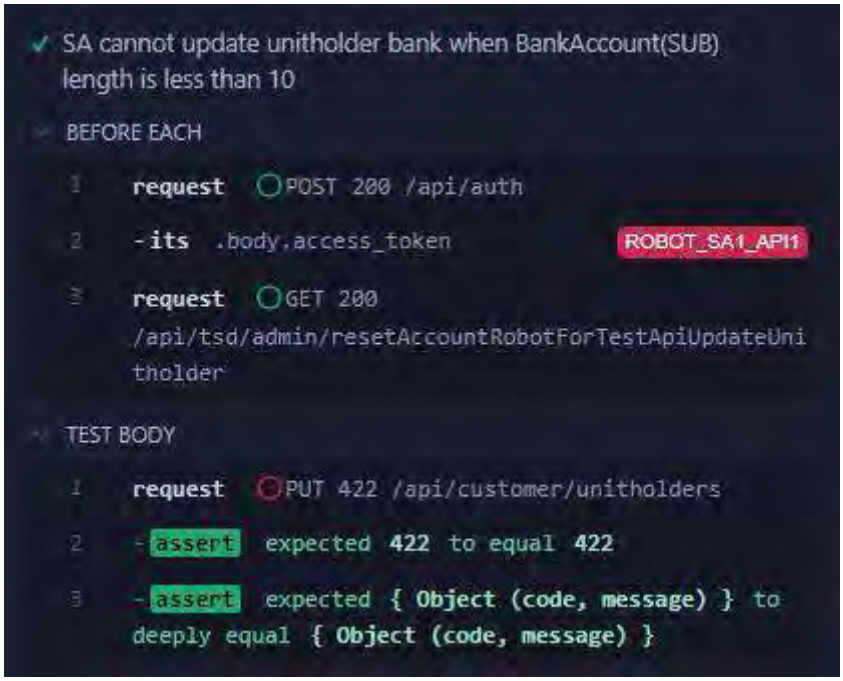
ผลการทดสอบกรณีที่ 4 ไม่สามารถ Send API Update Unitholder Bank เฉพาะ Bank Red



รูปที่ 4.119 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 4

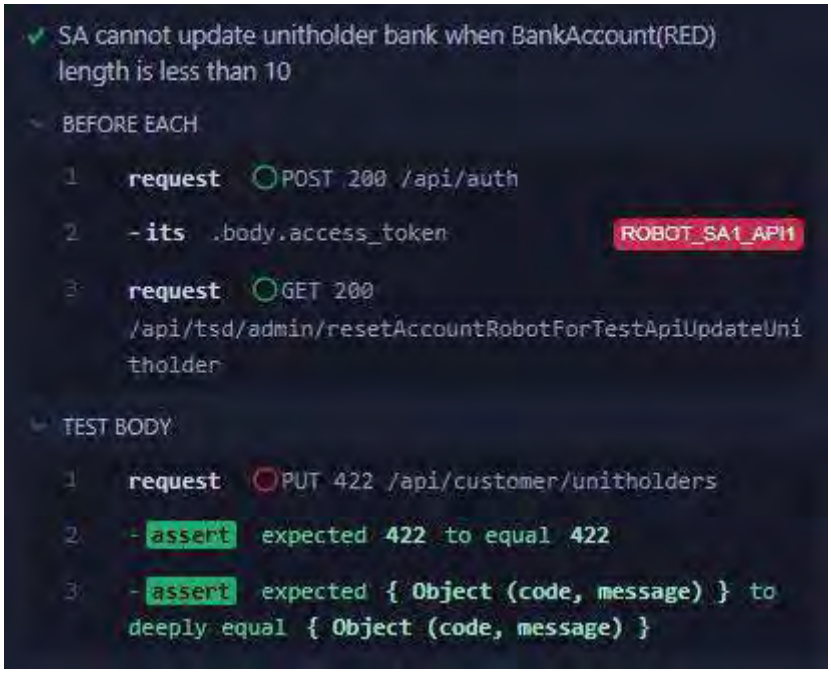
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 5 ไม่สามารถ Send API Update Unitholder Bank
โดยใส่ข้อมูล BankAccoutNo. Sub น้อยกว่า 10 Char



รูปที่ 4.120 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 5

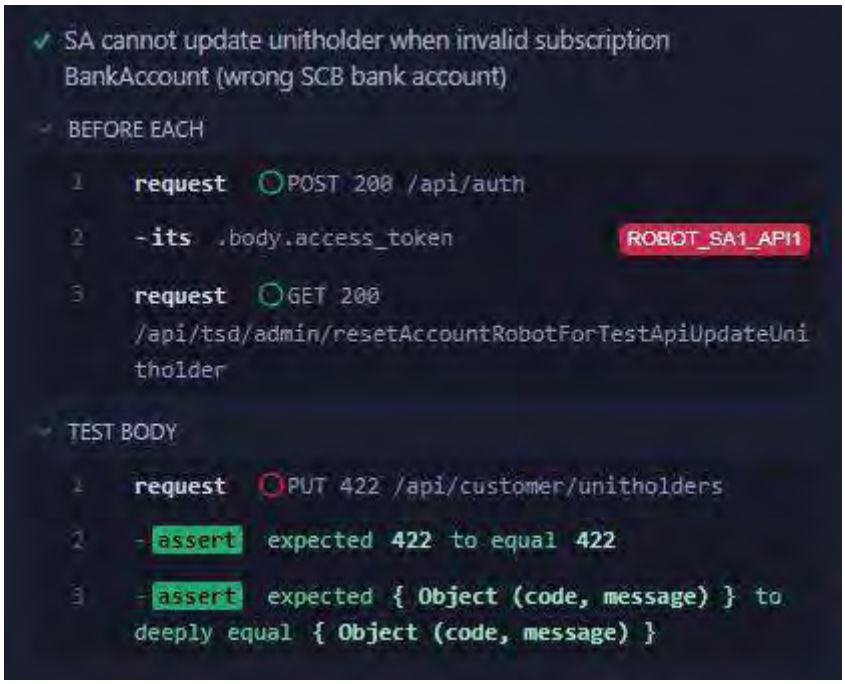
ผลการทดสอบกรณีที่ 6 ไม่สามารถ Send API Update Unitholder Bank
โดยใส่ข้อมูล BankAccoutNo. Red น้อยกว่า 10 Char



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

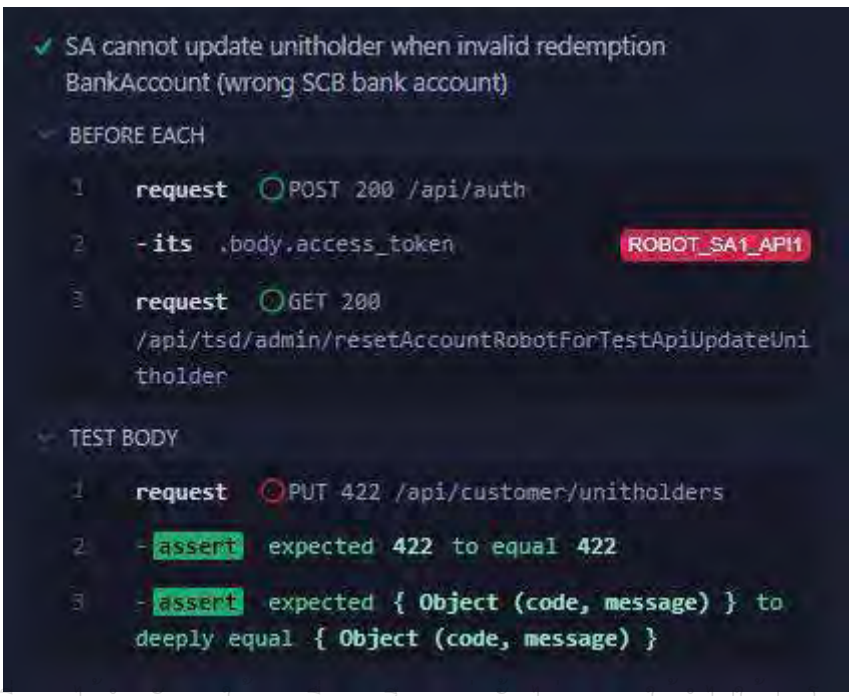
รูปที่ 4.121 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 6

ผลการทดสอบกรณีที่ 7 ไม่สามารถ Send API Update Unitholder Bank โดยใส่ข้อมูล BankAccoutNo. Sub ของ SCB ไม่ตรงตาม Format



รูปที่ 4.122 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 7

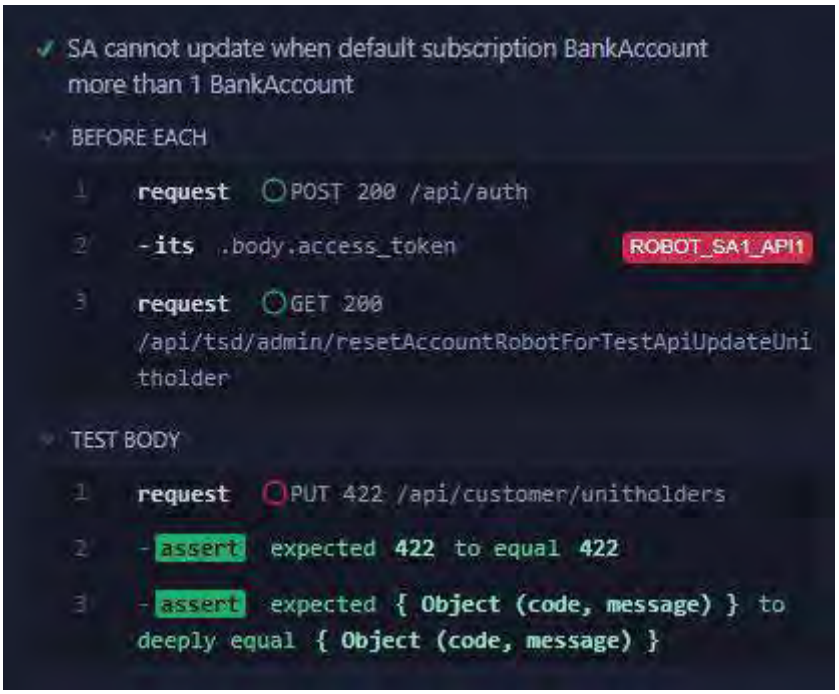
ผลการทดสอบกรณีที่ 8 ไม่สามารถ Send API Update Unitholder Bank โดยใส่ข้อมูล BankAccoutNo. Red ของ SCB ไม่ตรงตาม Format



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

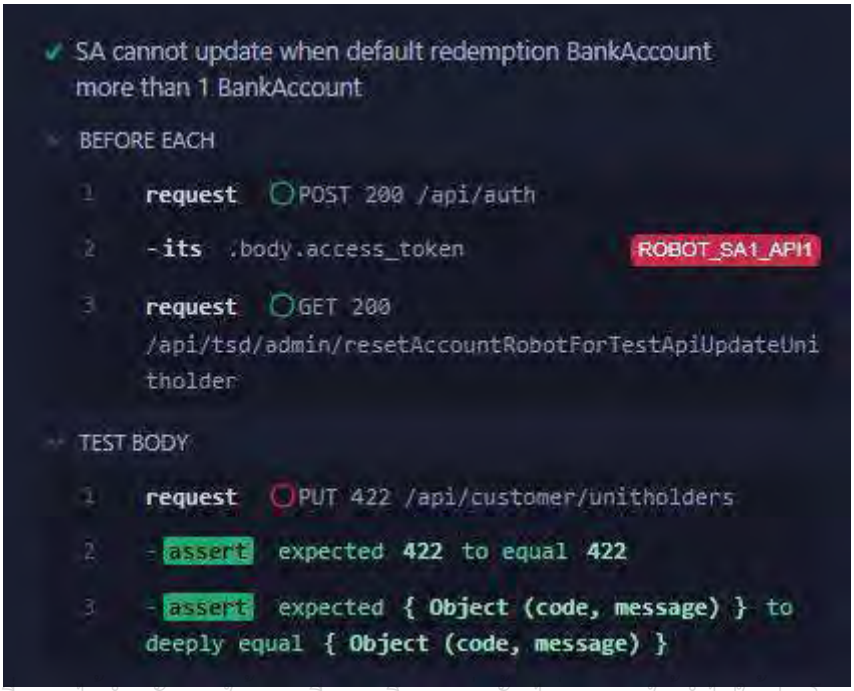
รูปที่ 4.123 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 8

ผลการทดสอบกรณีที่ 9 ไม่สามารถ Send API Update Unitholder Bank
โดยใส่ข้อมูล Default Bank Account Sub มากกว่า 1 ธนาคาร



รูปที่ 4.124 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 9

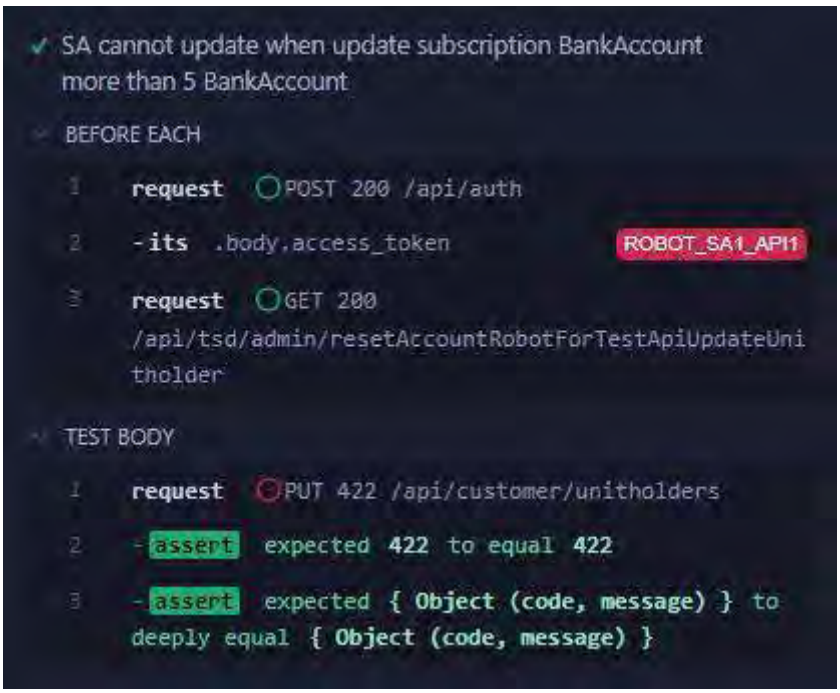
ผลการทดสอบกรณีที่ 10 ไม่สามารถ Send API Update Unitholder Bank
โดยใส่ข้อมูล Default Bank Account Red มากกว่า 1 ธนาคาร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

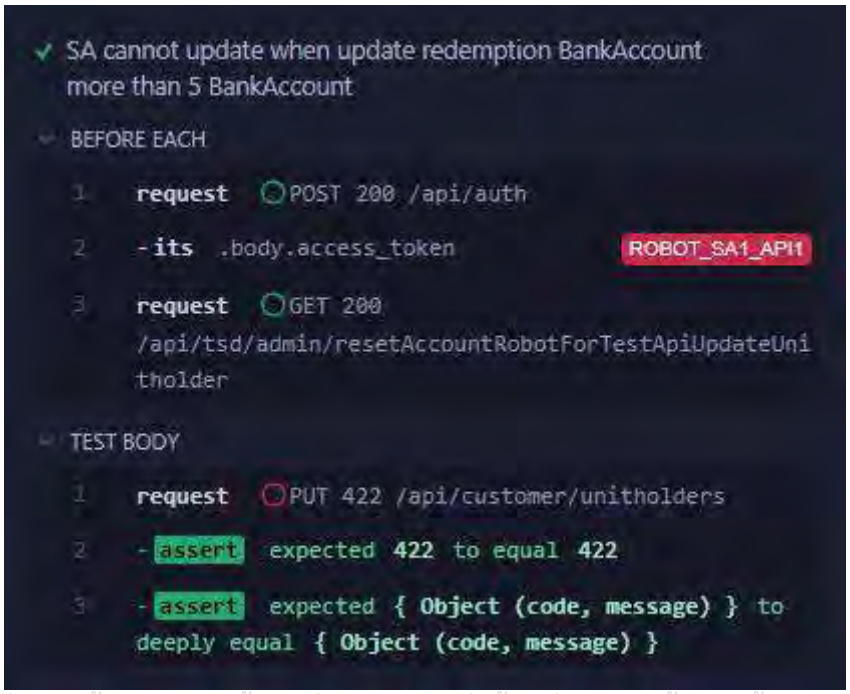
รูปที่ 4.125 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 10

ผลการทดสอบกรณีที่ 11 ไม่สามารถ Send API Update Unitholder Bank
โดยใส่ข้อมูล Bank Sub มากกว่า 5 ธนาคาร



รูปที่ 4.126 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 11

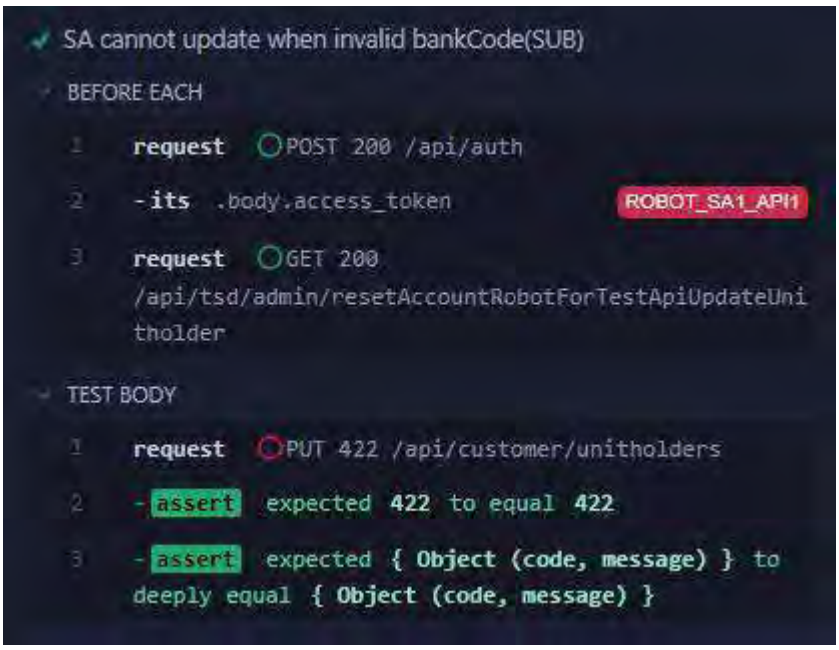
ผลการทดสอบกรณีที่ 12 ไม่สามารถ Send API Update Unitholder Bank
โดยใส่ข้อมูล Bank Sub มากกว่า 5 ธนาคาร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

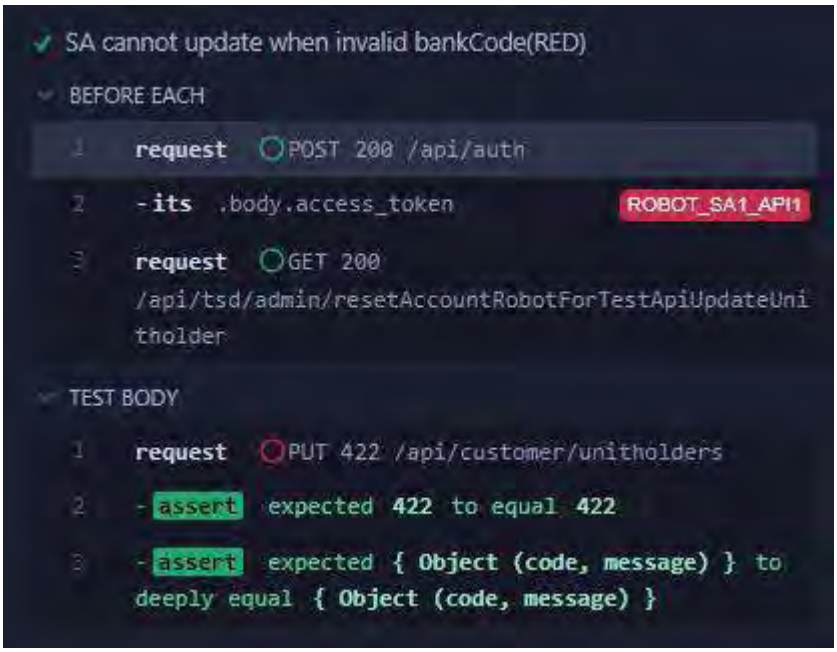
รูปที่ 4.127 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 12

ผลการทดสอบกรณีที่ 13 ไม่สามารถ Send API Update Unitholder Bank
โดยใส่ข้อมูล BankCode Sub ที่ไม่มีในระบบ



รูปที่ 4.128 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 13

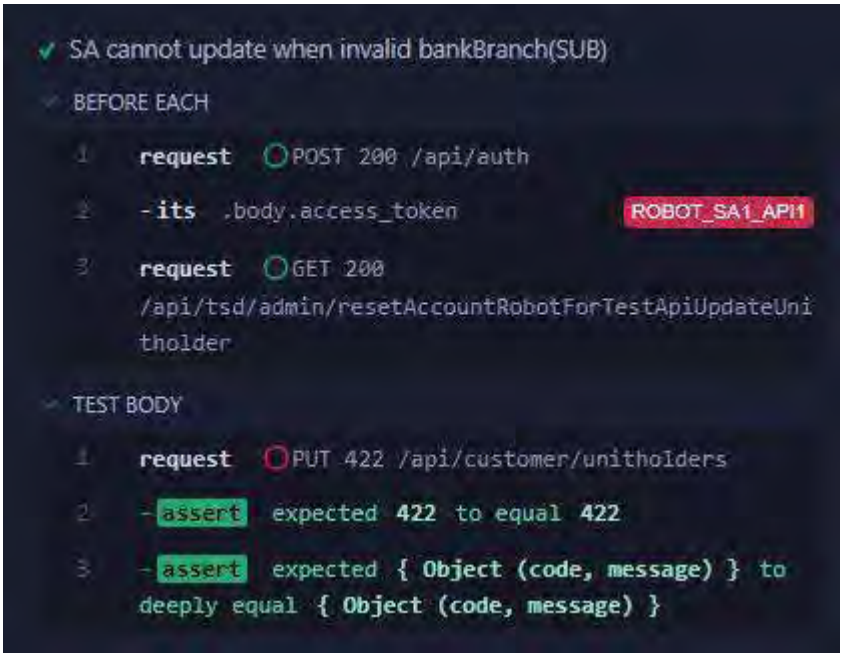
ผลการทดสอบกรณีที่ 14 ไม่สามารถ Send API Update Unitholder Bank
โดยใส่ข้อมูล BankCode Red ที่ไม่มีในระบบ



รูปที่ 4.129 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 14

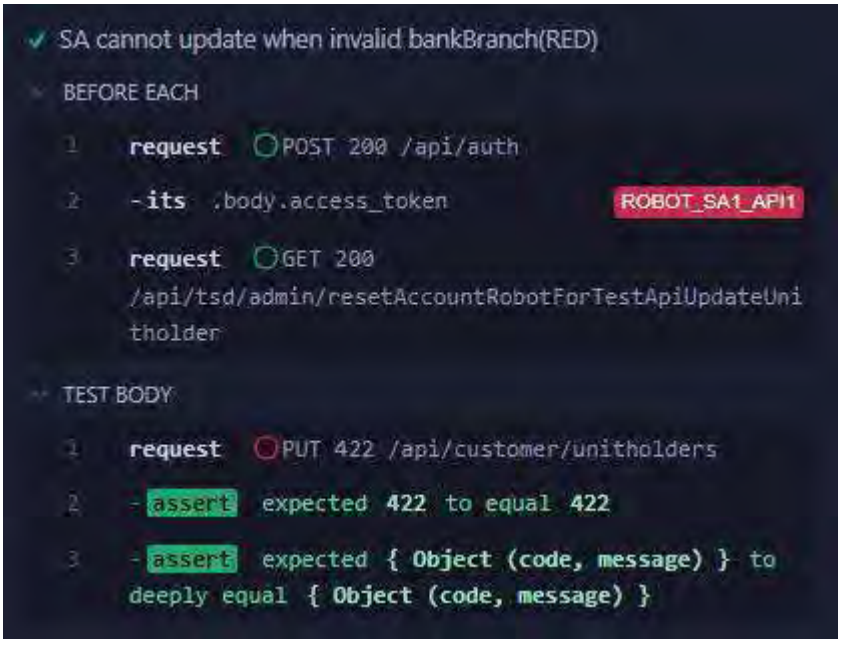
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 15 ไม่สามารถ Send API Update Unitholder Bank
โดยใส่ข้อมูล BankBranch Sub ที่ไม่มีในระบบ



รูปที่ 4.130 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 15

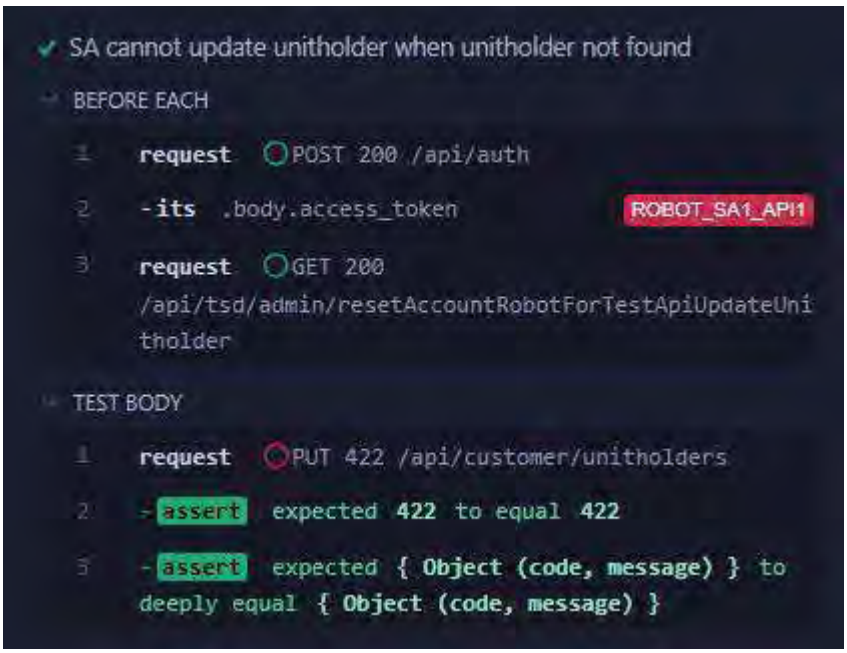
ผลการทดสอบกรณีที่ 16 ไม่สามารถ Send API Update Unitholder Bank
โดยใส่ข้อมูล BankBranch Red ที่ไม่มีในระบบ



รูปที่ 4.131 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบกรณีที่ 17 ไม่สามารถ Send API Update Unitholder Bank
โดยใส่ข้อมูล Unitholder ที่ไม่มีในระบบ



รูปที่ 4.132 หน้าจอผลลัพธ์การทดสอบ Automated Test API ตัวอย่างที่ 3 กรณีที่ 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการดำเนินงานสหกิจศึกษาและข้อเสนอแนะ

5.1 สรุปผลสหกิจศึกษา

จากการเข้าร่วมโครงการสหกิจศึกษาในหัวข้อการทดสอบอัตโนมัติสำหรับระบบฟันด์คอนเน็กต์ที่บริษัทตลาดหลักทรัพย์แห่งประเทศไทย ผู้จัดทำได้รับมอบหมายหน้าที่ Tester ภายในทีมฟันด์คอนเน็กต์ โดยได้รับมอบหมายตั้งแต่วิเคราะห์ความต้องการที่ได้รับมอบหมายจาก Project Owner ในแต่ละ Sprint การออกแบบกรณีทดสอบ การทดสอบแบบ Manual Test และการทดสอบแบบ Automated Test รวมถึงการบำรุงรักษา Automated Test Script ซึ่งจะต้องมีการศึกษาเกี่ยวกับโครงสร้างการทำงานของระบบฟันด์คอนเน็กต์ การทำความเข้าใจความต้องการได้รับมอบหมายก่อนการออกแบบกรณีทดสอบเพื่อทดสอบระบบแบบ Manual Test และการทดสอบแบบ Automated Test โดยผู้จัดทำได้ทำการทดสอบในระดับ Functional Test ซึ่งได้นำฟังก์ชันที่ได้ทำการทดสอบในช่วงสหกิจศึกษามายกตัวอย่างภายในเล่มสหกิจศึกษา เช่น การทดสอบ Functional Test ในรูปแบบต่างๆ การทดสอบ API เป็นต้น โดยเครื่องมือที่ใช้ในการจัดทำ Automated Test Script คือ Cypress รวมไปถึงการบำรุงรักษา Automated Test ซึ่งจะทำการแก้ไข Automated Test Script ที่ทดสอบไม่ผ่านในแต่ละ Sprint โดยการทดสอบฟังก์ชันที่ได้รับมอบหมายมาเป็นไปได้อย่างถูกต้อง แม้จะมีบางกรณีที่คาดไม่ถึงหรือตกหล่น แต่สามารถพบเจอและแก้ไขได้ทันก่อนที่ระบบจะให้ลูกค้าใช้งานจริง

5.2 ปัญหาและข้อเสนอแนะ

1. การเขียน Automated Test เมนูใหม่มักจะใช้เวลาในการเขียนและการเตรียมที่มากกว่าการเพิ่มเติม Test Case จากที่มีอยู่
2. รูปแบบในการเขียน Automated Test ของ Cypress ยังไม่มีแนวทางหรือวิธีการเขียนที่ชัดเจนทำให้ Automated Test มีความแตกต่างขึ้นอยู่กับผู้ที่จัดทำ ควรมีการประชุมทีมเพื่อหาแนวทางการเขียนให้อยู่ในรูปแบบเดียวกัน
3. บางคำสั่งที่ไม่สามารถใช้งานเมื่อรัน Automated Test บน Server ได้
4. Notebook ที่ใช้ในการทำงานที่บริษัทจัดหาให้มีประสิทธิภาพที่ไม่เพียงพอต่อการ Run Automated Test ในบางครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

Codebee. 2021. **วิธีใช้งาน Postman สำหรับการทดสอบ API.** [Online]. Available: https://www.codebee.co.th/labs/%E0%B8%A7%E0%B8%B4%E0%B8%98%E0%B8%B5%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99-postman/

ICONEXT WRITER. 2021 **Automated Testing คืออะไร** [Online]. Available : https://iconext.co.th/th/2021/07/19/automated-testing-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/

KhomKrit. 2020. **แนะนำการใช้งาน Cypress** [Online]. Available : https://www.khomkrit.com/%E0%B9%81%E0%B8%99%E0%B8%B0%E0%B8%99%E0%B8%B3%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99-cypress/

MintMint. 2018. **Cypress คืออะไร | ลองใช้ Cypress ฉบับ Beginner** [Online]. Available : https://medium.com/@mintmintttttt/qa-cypress-for-beginner-4c6a5a623292

Patipat. 2019. **ข้อดี / ข้อเสียของ Manual Testing และ Automate Testing.** [Online]. Available :

Traitanit Huangsri. 2018 **รู้จัก Cypress: Web Test Framework ที่จะทำให้คุณลืม Seleniumไปได้เลย** [Online]. Available : https://medium.com/cypress-io-thailand https://www.mindphp.com/forums/viewtopic.php?f=85&t=57739

Thomas Hamilton. 2021. **What is Software Testing? Definition, Basics & Types in Software Engineering.** [Online]. Available : https://www.guru99.com/software-testing-introduction-importance.html#1

Waerohanee Waehayee. 2019. **Postman คืออะไร.** [Online]. Available : https://medium.com/@waerohaneewaehayee/postman-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-c6461461cc55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

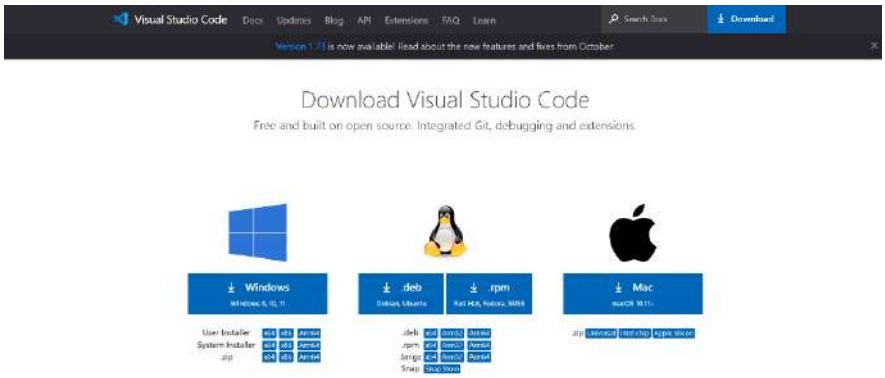
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

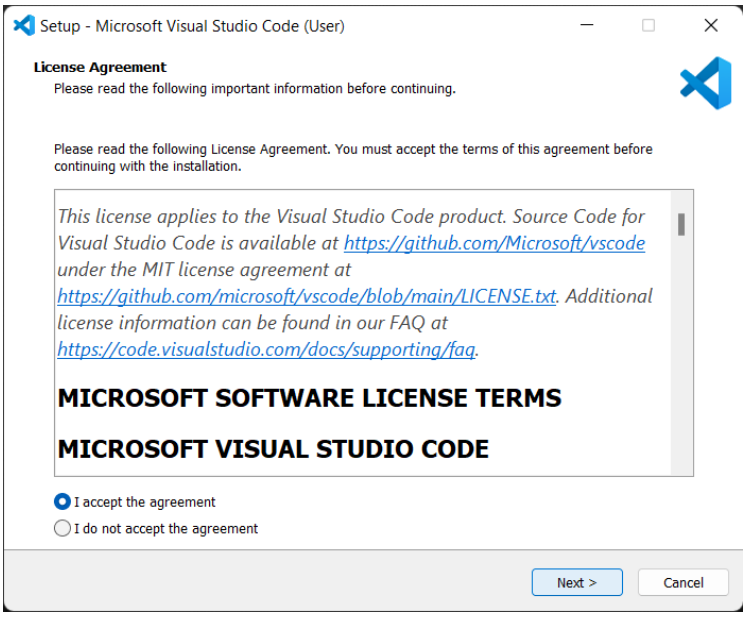
การติดตั้ง Visual Studio Code

1. ทำการติดตั้ง Visual Studio Code โดยการกด Download บนเว็บไซต์ https://code.visualstudio.com/download



รูปที่ ก.1 หน้าเว็บไซต์ในการ Download

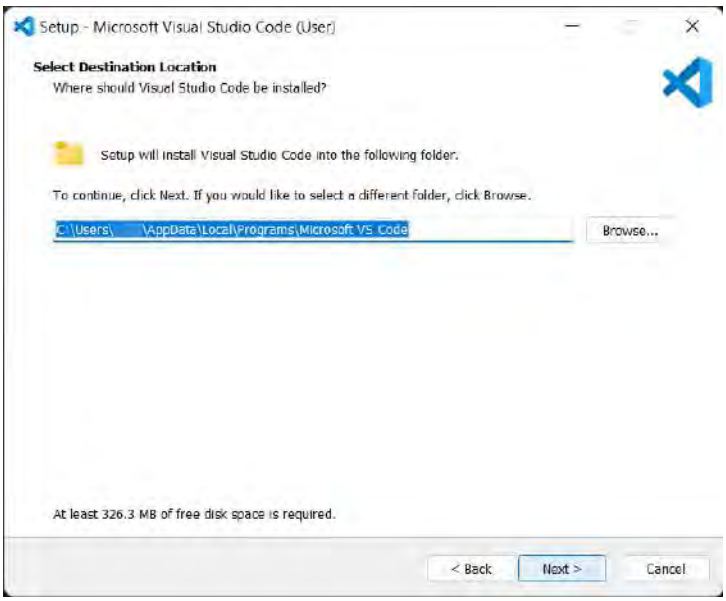
2. ทำการกดยอมรับข้อตกลง



รูปที่ ก.2 หน้าจอเงื่อนไขและข้อตกลงในการติดตั้ง

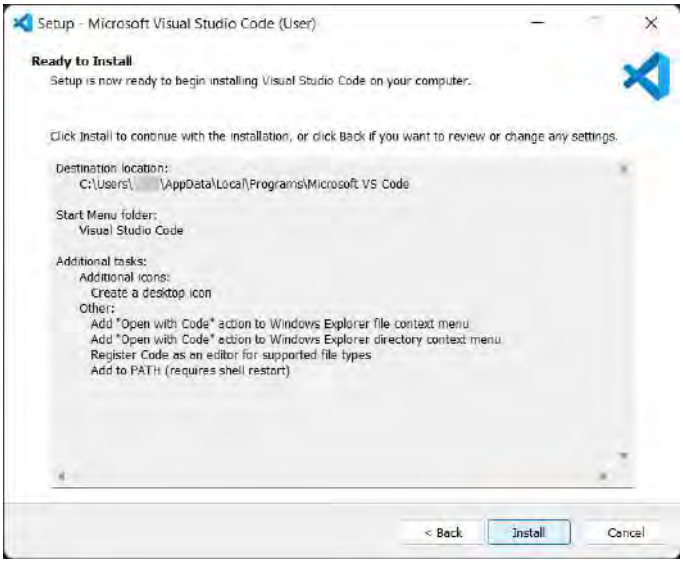
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลือกพื้นที่ในการติดตั้งโปรแกรม



รูปที่ ก.3 หน้าจอเลือกพื้นที่ในการติดตั้งโปรแกรม

4. ทำการกด Install



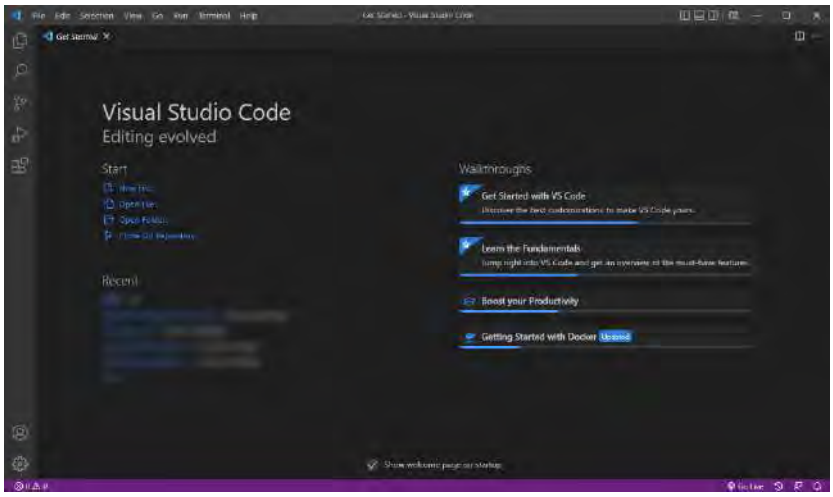
รูปที่ ก.4 หน้าจอกด Install

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เมื่อทำการติดตั้งเสร็จแล้วกด Finish

รูปที่ ก.5 หน้าจอเมื่อทำการติดตั้งเสร็จสิ้น

6. เมื่อแสดงหน้าจอแบบนี้ แสดงว่าการติดตั้งโปรแกรม Visual Studio Code เสร็จสมบูรณ์



รูปที่ ก.6 หน้าจอภายในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

การติดตั้ง Node.js

1. ทำการติดตั้ง Node.js โดยการกด Download บนเว็บไซต์ https://nodejs.org/en/

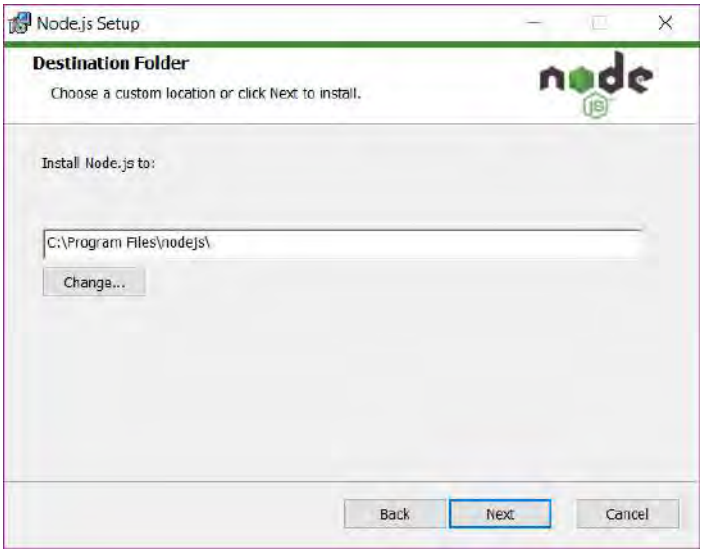
รูปที่ ข.1 หน้าเว็บไซต์ในการ Download

2. กดยอมรับข้อตกลงและเงื่อนไข

รูปที่ ข.2 หน้าจอเงื่อนไขและข้อตกลงในการติดตั้ง

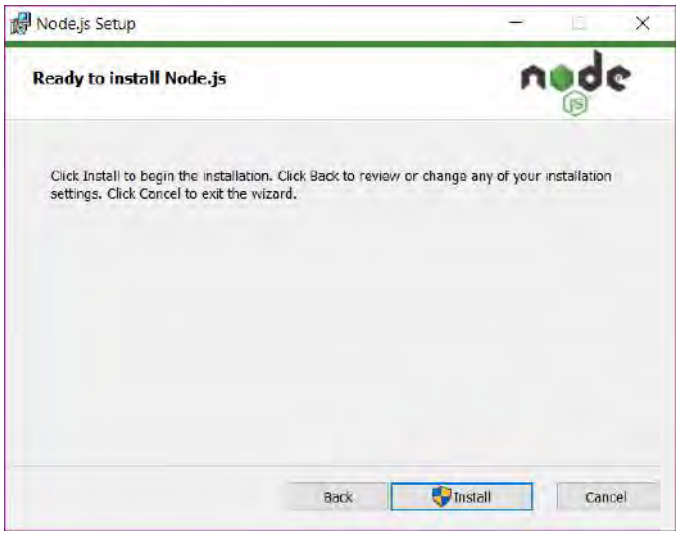
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลือก Location ในการติดตั้ง



รูปที่ ข.3 หน้าจอเลือกพื้นที่ในการติดตั้ง

4. ทำการกด Install



รูปที่ ข.4 หน้าจอกด Install

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

การติดตั้ง Cypress.io

1. ทำการติดตั้ง Node.js โดยการกด Download บนเว็บไซต์ https://nodejs.org/en/

รูปที่ ค.1 หน้าเว็บไซต์ในการ Download

2. สร้าง Folder ของProject แล้ว CD เข้าไป

```
cd /your/project/path
```

รูปที่ ค.2 CD ไปยัง Path Project

3. ติดตั้ง Cypress ด้วยคำสั่ง ดังรูป

```
npm install cypress --save-dev
```

รูปที่ ค.3 คำสั่งการติดตั้ง Cypress

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานทะเบียนคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

คำรับรองเล่มสหกิจศึกษา

วันที่ 28 เดือน พฤษภาคม พ.ศ 2566

ข้าพเจ้า นายพงษ์ศิริ ศิริพงษ์ รหัสประจำตัว 62050194 นักศึกษาหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาวิทยาการคอมพิวเตอร์ ขอรับรองว่าสหกิจศึกษา เรื่อง ชื่อภาษาไทย การทดสอบอัตโนมัติสำหรับระบบฟันด์คอนเน็กต์ ชื่อภาษาอังกฤษ Automated Test for FundConnext System ปีการศึกษา 2565 เป็นผลงานสหกิจศึกษาที่มิได้คัดลอกหรือละเมิดลิขสิทธิ์ของผู้อื่นและได้ผ่านการตรวจสอบความซ้ำซ้อนเรียบร้อยแล้ว และได้แนบเอกสารการตรวจสอบการลอกเลียนงานวรรณกรรมที่ตรวจสอบจากเล่มสหกิจศึกษาฉบับสมบูรณ์แล้ว โปรแกรมอักขราวิสุทธิ์ 0.14 %

ลงชื่อ.....

(นายพงษ์ศิริ ศิริพงษ์)

นักศึกษา

ข้าพเจ้า ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์ อาจารย์ที่ปรึกษาสหกิจศึกษาได้ตรวจสอบสหกิจศึกษาของนักศึกษาข้างต้นแล้วขอรับรองว่าเป็นผลงานสหกิจศึกษาของนักศึกษาจริงและมีเนื้อหาสมบูรณ์ จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ.....

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้