

ทรูมันนี่ เดเวลอปเปอร์ แซนด์บ็อกซ์

TRUEMONEY DEVELOPER SANDBOX



สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตาม  
หลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)  
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงแก้ไขเอกสารฉบับนี้  
ปีการศึกษา 2565

# TRUEMONEY DEVELOPER SANDBOX



JINJUTHA TEDCHAROEN  
PUNNARAT RATTANAPAWAN

A COOPERATIVE EDUCATION SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENT FOR

THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)

DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ **ACADEMIC YEAR 2022** ให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



หัวข้อสหกิจศึกษา	ทรูมันนี่ เดเวลอปเปอร์ แซนด์บ็อกซ์ TRUEMONEY DEVELOPER SANDBOX
ชื่อนักศึกษา	นางสาว จินต์จุฑา เทศเจริญ รหัสนักศึกษา 62050135 นาย ปุณณรัตน์ รัตนภาวรณ รหัสนักศึกษา 62050191
ปริญญา	วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2565
อาจารย์ที่ปรึกษา	ผศ.ดร. นवलสวาท หิรัญสกุลวงศ์

## บทคัดย่อ

โครงการสหกิจศึกษาหัวข้อ ทรูมันนี่ เดเวลอปเปอร์ แซนด์บ็อกซ์ ( TrueMoney Developer Sandbox ) มีวัตถุประสงค์เพื่อทำระบบทดสอบ API โดยการสร้างเว็บแอปพลิเคชันที่ผู้พัฒนาร้านค้าสามารถทดสอบการทำงานของ API ได้ และสร้างคู่มือแสดงรายละเอียดและวิธีการใช้เว็บแอปพลิเคชันดังกล่าว เพื่อให้ผู้พัฒนาร้านค้าใช้งานได้สะดวกที่สุด ระบบทดสอบ API เป็นการพัฒนาในส่วนที่ติดต่อกับผู้ใช้ ซึ่งได้รับโจทย์ปัญหาจากเจ้าของผลิตภัณฑ์ ( product owner ) โดยการพัฒนาครั้งนี้เป็นการพัฒนาระยะที่ 1 ที่เว็บแอปพลิเคชันจะสามารถประมวลผลได้บนเครื่องของผู้ใช้ ซึ่งผู้ใช้จำเป็นต้องลงทะเบียน เพื่อสร้าง username และ password สำหรับใช้ในการเข้าสู่เว็บแอปพลิเคชัน ซึ่งสามารถทดสอบ API ในกรณีศึกษาต่าง ๆ ได้ โดยการส่งค่าความต้องการผ่าน API ระบบจะแสดงผลลัพธ์ พร้อมมีการบันทึกประวัติผลลัพธ์ ซึ่งสามารถเรียกดูย้อนหลังได้ในเว็บแอปพลิเคชัน เมื่อพัฒนาระบบได้ตรงความต้องการของเจ้าของผลิตภัณฑ์แล้ว จึงได้ทำการทดสอบคุณภาพโดยการทำ automated test เพื่อตรวจสอบผลลัพธ์และยืนยันว่าการพัฒนาระบบดังกล่าวมีความถูกต้อง ก่อนนำไปปฏิบัติจริงกับผู้ใช้ทั่วไป

**คำสำคัญ :** mock API, automated test ;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>Title</b>	TRUEMONEY DEVELOPER SANDBOX
<b>Students</b>	Miss Jinjutha Tedcharoen Student ID 62050135 Mr. Punnarat Rattanaoawan Student ID 62050191
<b>Degree</b>	Bachelor of Science (Computer Science)
<b>Department</b>	Computer Science
<b>School</b>	Science
<b>University</b>	King Mongkut's Institute of Technology Ladkrabang (KMITL)
<b>Academic Year</b>	2022
<b>Advisor</b>	Asst. Prof. Dr. Nualsawat Hiransakolwong

## ABSTRACT

This cooperative education on TrueMoney Developer Sandbox aims to test the API by creating a web application that merchant developers can test the functionality of the API and create manuals detailing and how to use such web applications and to make it as convenient for merchant developers as possible. This API testing system is developed in the user interface which received the requirement from the product owner by this development is the development of the first phase where web applications can be processed on the user's machine. Users need to register to create a username and password for access to the web application. The API can be tested in various case studies by submitting requirements via the API. The results will be displayed with a record of the results history which can be viewed back in the web application. When the development of the system meets the needs of the product owner. Therefore, a quality test was conducted by automated testing to verify the results and confirm that the development of such systems is correct before putting it into practice with general users.

**Keywords :** mock API, automated test ;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

สหกิจศึกษาโครงการ เว็บแอปพลิเคชัน ทรมันนี่ เดเวลลอปเปอร์ แชนด์บ็อกซ์ สมบูรณ์ได้ ต้องขอขอบพระคุณ ผศ.ดร. นवलสวาท หิรัญสกุลวงศ์ และ ผศ.ดร. วิสันต์ ตั้งวงษ์เจริญ อาจารย์ประจำภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้คำแนะนำ เป็นที่ปรึกษาระหว่างสหกิจศึกษาตลอดมา ทำให้ทุกอย่างสำเร็จ ลุล่วงด้วยดี

ขอขอบคุณบริษัท ทรมันนี่ จำกัด ที่ได้ดูแลอย่างดีตลอดระยะเวลาการทำสหกิจศึกษา ให้ความรู้ คำแนะนำในการทำงานในชีวิตจริงและการทำงานร่วมกับผู้อื่น ทางผู้จัดทำหวังเป็นอย่างยิ่งว่าสหกิจศึกษาโครงการนี้จะเป็นประโยชน์แก่ผู้ที่สนใจ และสามารถนำไปพัฒนาต่อยอดได้ในอนาคต

นางสาว จินต์จุฑา เทศเจริญ  
นาย ปุณณรัตน์ รัตนภาวรณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

หน้า

บทคัดย่อ .....	ก
ABSTRACT .....	ข
กิตติกรรมประกาศ .....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ช
สารบัญรูป .....	ซ
บทที่ 1	
บทนำ .....	1
1.1 ความเป็นมาและความสำคัญของสหกิจศึกษา .....	1
1.2 วัตถุประสงค์ของระบบ .....	1
1.3 ขอบเขตของสหกิจศึกษา.....	2
1.4 แผนการดำเนินงาน.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ .....	4
1.6 อุปกรณ์ที่ใช้ในการพัฒนา .....	4
บทที่ 2	
ทฤษฎีและกรอบการดำเนินงาน .....	6
2.1 การทำงานของระบบต้นแบบ.....	6
2.2 ทฤษฎีที่เกี่ยวข้องกรอบงาน .....	9
2.2.1 REST API .....	9
2.2.2 การรับรองความถูกต้องพื้นฐาน.....	9
2.2.3 Mountebank.....	10
2.2.4. Robot Framework.....	15
2.3 เครื่องมือที่ใช้ในการพัฒนา .....	16
2.3.1 Bitbucket .....	16
2.3.2 Postman.....	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 Jenkins .....	16
2.3.4 Jira Software .....	17

### บทที่ 3

<b>วิธีการออกแบบและพัฒนาระบบ .....</b>	<b>18</b>
3.1 ขั้นตอนการศึกษาปัญหาและรวบรวมข้อมูล .....	18
3.2 ขั้นตอนการวิเคราะห์ระบบ .....	19
3.2.1 เว็บไซต์ ( Website ) .....	20
3.2.2 เว็บเซิร์ฟเวอร์ทรูมันนี่เดเวลอปเปอร์แซนด์บ็อกซ์ ( TrueMoney Developer Sandbox Web Server ) .....	20
3.2.3 API หรือเซิร์ฟเวอร์จำลอง ( Mock server ) .....	20
3.2.4 เซิร์ฟเวอร์การแจ้งเตือนร้านค้า ( Merchant Notification Server ) .....	20
3.3 ขั้นตอนการเตรียมข้อมูล .....	21
3.1.1 สร้างข้อมูลผู้ใช้ประเภทต่าง ๆ .....	22
3.1.2 สร้างข้อมูลจำลองกิจกรรมต่าง ๆ ที่เกิดขึ้นจริง .....	24
3.1.3 สร้างข้อมูลกิจกรรมการแจ้งเตือนร้านค้า .....	24
3.1.4 สร้างคู่มือวิธีใช้งานเว็บแอปพลิเคชัน .....	24
3.4 ขั้นตอนการออกแบบและการดำเนินงาน .....	24
3.3.1 Use Case Diagram .....	24
3.3.2 Use Case Description .....	25
3.3.3 ER Diagram .....	31
3.3.4 Data Dictionary .....	31
3.3.5 API Definition .....	34

### บทที่ 4

<b>ผลการวิจัยและการอภิปรายผล .....</b>	<b>39</b>
4.1 วิธีการใช้งานระบบ ทรูมันนี่ เดเวลอปเปอร์ แซนด์บ็อกซ์ ผ่านเว็บแอปพลิเคชัน .....	39
4.1.1 ลงทะเบียนร้านค้า .....	39
4.1.2 เข้าสู่ระบบ .....	40
4.1.3 ศึกษาวิธีใช้ระบบและ API .....	40
4.1.4 การยืนยันตัวตนด้วย Token .....	42
4.1.5 เลือก API ที่ต้องการทดสอบ .....	42
4.1.6 ทดสอบ API ผ่านเว็บแอปพลิเคชัน .....	44

4.1.7 ตัวอย่าง API อื่นๆ.....	46
4.2 ผลการทดสอบ API แบบ MANUAL TEST .....	47
4.3 ผลการทดสอบ API แบบ AUTOMATED TEST .....	49
<b>บทที่ 5</b>	
<b>สรุปผลการวิจัยและข้อเสนอแนะ .....</b>	<b>60</b>
5.1 สรุปผลการวิจัย.....	60
5.2 ปัญหาและอุปสรรค .....	60
5.3 ข้อเสนอแนะ .....	61
<b>เอกสารอ้างอิง.....</b>	<b>62</b>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่	หน้า
1.1 ตารางแผนการดำเนินงาน.....	4
3.1 รายละเอียดของ Use Case Register .....	26
3.2 รายละเอียดของ Use Case Login.....	27
3.3 รายละเอียดของ Use Case Access to documents.....	28
3.4 รายละเอียดของ Use Case Request API.....	29
3.5 รายละเอียดของ Use Case Get log .....	30
3.6 Data Dictionary : users จัดเก็บข้อมูลของผู้ใช้งาน.....	31
3.7 Data Dictionary : request_logs จัดเก็บบันทึกการ request ไปที่ mock open API.....	32
3.8 Data Dictionary : services จัดเก็บรายละเอียดของแต่ละ service .....	32
3.9 Data Dictionary : responses จัดเก็บ response ของแต่ละ API.....	33
3.10 Data Dictionary : request จัดเก็บ request ของแต่ละ API.....	33
3.11 Data Dictionary : request จัดเก็บรายละเอียดกรณีทดสอบของแต่ละ API .....	34
3.12 API Definition ของ Web Server.....	34
3.13 API Definition ของ Open API จำลอง .....	35
3.14 Common Response .....	36
3.15 Merchant Notification Response.....	37
4.1 รายละเอียดกรณีทดสอบ Get Balance.....	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

รูปที่	หน้า
2.1 Create Payment Links แบบไม่ใช้โค้ด.....	6
2.2 Retrieve a balance .....	7
2.3 Create Payment Links แบบไม่ใช้โค้ด.....	8
2.4 Project Structure.....	11
2.5 Mock ในส่วนของ fail .....	12
2.6 Mock ในส่วนของ success.....	13
2.7 Mountebank Imposter.....	14
2.8 Mountebank Proxy .....	15
2.9 Robot Framework format .....	15
2.10 ตัวอย่าง Task management ของ Jira Software.....	17
3.1 สถาปัตยกรรมระบบของทรูมันนี่ เดเวลอปเปอร์ แชนด์บ็อกซ์ .....	20
3.2 การทำงานการยืนยันลักษณะเฉพาะของข้อมูล .....	21
3.3 ข้อมูลในฐานข้อมูล web page.....	22
3.4 ข้อมูลสำหรับผู้พัฒนาฝั่งทรูมันนี่ .....	22
3.5 ข้อมูลเทคโนโลยีที่ใช้ในระบบ .....	23
3.6 Use Case Diagram ของเว็บแอปพลิเคชันทรูมันนี่ เดเวลอปเปอร์ แชนด์บ็อกซ์.....	25
3.7 ER Diagram.....	31
3.8 ตัวอย่างการสร้างและยืนยันลักษณะเฉพาะของข้อมูลในภาษา Python.....	38
4.1 ลงทะเบียนร้านค้า .....	39
4.2 เข้าสู่ระบบสำเร็จ.....	40
4.3 หน้า Document แสดงวิธีใช้ TrueMoney Developer Sandbox .....	41
4.4 Authorization แสดง Token .....	42
4.5 API Definitions และ API References.....	42
4.6 หน้าทดลอง API ในแต่ละกรณีทดสอบ .....	43
4.7 คำอธิบาย Response และ Error .....	43
4.8 รายละเอียดการทดลองส่ง Request Get Balance กรณีขอข้อมูลสำเร็จ.....	44
4.9 ทดลองส่ง request ขอข้อมูลเงินคงเหลือในบัญชีลูกค้า .....	45
4.10 แสดง log ผ่านหน้าเว็บแอปพลิเคชัน .....	45
4.11 คำอธิบายและวิธีการทดลอง Merchant Notification.....	46
4.12 วิธีการใช้และปุ่มดาวน์โหลด SDK.....	47

4.13 วิธีทดลอง Merchant Notification ในกรณีสำเร็จ ..... 47

4.14 Get Balance ..... 48

4.15 Create Order..... 48

4.16 Merchant Notification..... 49

4.17 Testcases..... 50

4.18 ผลการทดสอบ ( Test Report ) ..... 55



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## คำย่อ/สัญลักษณ์

คำย่อ/สัญลักษณ์	คำอธิบาย
mock API	การจำลองฟังก์ชันการทำงานหรือบริการของ API นั้น ๆ เพื่อใช้ในการทดสอบ
automated test	การทดสอบซอฟต์แวร์ เป็นกระบวนการอย่างหนึ่งในการพัฒนาซอฟต์แวร์ เพื่อหาข้อผิดพลาดและป้องกันข้อผิดพลาดที่อาจเกิดขึ้น หรือช่วยให้ซอฟต์แวร์ที่พัฒนานั้นมีประสิทธิภาพมากขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของสหกิจศึกษา

ร้านค้าที่ต้องการเพิ่มเว็บแอปพลิเคชันทรูมันนี่ วอลเล็ท ( TrueMoney wallet ) เป็นช่องทางชำระสินค้า นักพัฒนาของร้านค้าต้องพบกับข้อจำกัดต่าง ๆ เช่น ต้องมีการติดต่อกับบุคลากรของบริษัท ทรูมันนี่ จำกัด ในการเชื่อมต่อระบบ ต้องมีขั้นตอนการทำเอกสารต่าง ๆ ซึ่งต้องใช้เวลา ความต่างทางด้านภาษาและเขตเวลา และระบบสภาพแวดล้อมทดสอบของบริษัท ทรูมันนี่ จำกัด ที่ไม่สามารถให้บริการทดสอบได้ตลอดเวลา ทำให้เกิดความไม่สะดวกแก่ทั้งสองฝ่ายบริษัท ทรูมันนี่ จำกัด จึงต้องการพัฒนาระบบทดสอบ API สำหรับผู้พัฒนาของร้านค้าขึ้นมาในรูปแบบของเว็บแอปพลิเคชัน ชื่อ ทรูมันนี่ เดเวลอปเปอร์ แซนด์บ็อกซ์ ( TrueMoney Developer Sandbox ) เพื่อให้ นักพัฒนาของร้านค้าสามารถทดสอบการทำงานของ API ที่จะสามารถใช้ได้หลังการเชื่อมต่อเข้ากับระบบแอปพลิเคชัน ทรูมันนี่ วอลเล็ท ได้เอง เพื่อประกอบการตัดสินใจของร้านค้าก่อนที่จะเชื่อมต่อเข้ากับระบบจริง

ในระบบดังกล่าวเป็นการพัฒนาในระยะที่ 1 ที่สามารถประมวลผลได้บนเครื่องของผู้ใช้ โดยนักพัฒนาของร้านค้าจะต้องสร้างบัญชีผู้ใช้และลงทะเบียนเพื่อเข้าสู่ระบบทดสอบการทำรายการของร้านค้าในกรณีศึกษาต่าง ๆ โดยการส่งค่าความต้องการตามรูปแบบและตัวอย่างที่กำหนดไว้ในเว็บแอปพลิเคชันผ่าน API เช่น การขอข้อมูลของลูกค้า การขอช่องทางชำระสินค้า เป็นต้น โดยสามารถดูประวัติการทำรายการที่เกิดขึ้นทั้งหมด และแสดงคู่มือการใช้งานของระบบได้ผ่านเว็บแอปพลิเคชัน

ดังนั้น เว็บแอปพลิเคชันทรูมันนี่ เดเวลอปเปอร์ แซนด์บ็อกซ์ จึงเป็นระบบที่สร้าง API จำลอง และคู่มือการทดสอบ API ดังกล่าวในรูปแบบเว็บแอปพลิเคชันเพื่อทำให้ผู้พัฒนาของร้านค้าสามารถทดสอบการทำงานของบริการต่าง ๆ ผ่าน API ได้ด้วยตนเอง และสามารถทดสอบได้ตลอดเวลาก่อนที่จะเชื่อมต่อเข้ากับระบบจริง ทำให้ช่วยลดขั้นตอนของการติดต่อกับบุคลากรของบริษัท ขั้นตอนการทำเอกสาร และขจัดข้อจำกัดเรื่องเขตเวลาและและระบบสภาพแวดล้อมทดสอบของบริษัท นอกจากนี้ยังช่วยให้นักศึกษาได้รับการฝึกปฏิบัติงานร่วมกันเจ้าหน้าที่พัฒนาซอฟต์แวร์ของบริษัทตลอดทั้งกระบวนการพัฒนาตั้งแต่เริ่มโครงการ วิเคราะห์ออกแบบระบบ และพัฒนาโปรแกรมจนถึงสิ้นสุดโครงการด้วยการนำเสนอผลงาน

### 1.2 วัตถุประสงค์ของระบบ

1) เพื่อสร้าง API จำลอง ( mock API ) ที่ร้านค้าสามารถใช้ได้เมื่อเชื่อมต่อเข้ากับระบบ

เอกสารนี้เป็นเอกสารแอปพลิเคชัน ทรูมันนี่ วอลเล็ทเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ (2) เพื่อสร้างเว็บแอปพลิเคชันที่ร้านค้าสามารถทดสอบการทำงานของ API จำลองได้นำไปใช้

3) เพื่อสร้างคู่มืออธิบายรายละเอียดและวิธีการใช้เว็บแอปพลิเคชัน

### 1.3 ขอบเขตของสหกิจศึกษา

ศึกษาเกี่ยวกับการพัฒนาเว็บแอปพลิเคชัน ทูมนี้ เดเวลอปเปอร์ แชนด์บ็อกซ์ จากโจทย์ ปัญหาจากเจ้าของผลิตภัณฑ์ ( product owner ) โดยมีเป้าหมายเพื่อพัฒนาระบบให้ผู้ใช้สามารถ ทดสอบบริการต่าง ๆ ผ่าน API ได้สะดวกรวดเร็วที่สุด โดยแบ่งเป็นการพัฒนาดังนี้

- 1) สร้างข้อมูลผู้ใช้ประเภทต่าง ๆ สำหรับนักพัฒนาของร้านค้าเพื่อใช้ในระบบทดสอบ โดยประเภทผู้ใช้จะแบ่งเป็น
  - 1.1) ร้านค้า ( Merchant )
    - ร้านค้าที่มีสถานะพร้อมใช้งานครบถ้วน
    - ร้านค้าที่มีสถานะบางอย่างไม่พร้อมใช้งาน เช่น ร้านค้าไม่พร้อมใช้งาน สัญญาไม่พร้อมใช้งาน หรือ สัญญาหมดอายุ เป็นต้น
  - 1.2) ลูกค้า ( User )
    - ลูกค้าที่มีข้อมูลครบถ้วน
    - ลูกค้าที่มีข้อมูลไม่ครบถ้วน
- 2) สร้างข้อมูลจำลองกิจกรรม ( service ) ในกรณีทดสอบต่าง ๆ ที่เกิดขึ้นจริง คือ ตัวอย่างค่าความต้องการที่จะส่งผ่าน API และผลลัพธ์ของ API นั้น ๆ ที่จะแสดงผ่านเว็บแอปพลิเคชัน โดยแบ่งกิจกรรมออกเป็น 2 ส่วนหลักดังนี้
  - 2.1) ข้อมูลลูกค้า ทูมนี้ วอลเล็ต ( User Controller )
  - 2.2) การสั่งซื้อเพื่อขอช่องทางชำระเงิน ( Order Controller )
- 3) พัฒนาเซิร์ฟเวอร์ที่ใช้ในการทดสอบ API จำลอง ( mock server ) โดยใช้ Mountebank สร้าง API จำลอง ( mock API ) และใช้ Robot framework ในการทดสอบความถูกต้อง
- 4) สร้างการแจ้งเตือนร้านค้าผ่าน API ( Merchant Notification ) เพื่อให้ นักพัฒนาของร้านค้าทราบว่าการทำงานกิจกรรมต่าง ๆ บนระบบทดสอบ เช่น การสร้างรายการสั่งซื้อสินค้าสำเร็จหรือไม่
- 5) พัฒนาเว็บแอปพลิเคชันโดยใช้ React ให้ นักพัฒนาของร้านค้าลงทะเบียนสำหรับเข้าแอปพลิเคชันที่แสดงคู่มือการใช้งาน สามารถทดสอบ API กรณีทดสอบต่าง ๆ และแสดงประวัติผลลัพธ์การส่ง API ได้ผ่านเว็บแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.4 แผนการดำเนินงาน

ระยะเวลาในการดำเนินงานเริ่มตั้งแต่วันที่ 1 มิถุนายน พ.ศ.2565 และสิ้นสุดในวันที่ 29 ธันวาคม พ.ศ. 2565 โดยใช้กระบวนการพัฒนาระบบแบบ SDLC ( System Development Life Cycle ) โดยมีรายละเอียดดังนี้

### 1) Planning and requirement analysis

โดยเริ่มแรกจะได้รับโจทย์ปัญหาและความต้องการเบื้องต้นจากเจ้าของผลิตภัณฑ์ ( product owner ) ซึ่งจะมีการวิเคราะห์ความต้องการของระบบ และวางแผนแบ่งงานออกเป็นส่วนย่อยๆเพื่อทำใน 1 ระยะเวลา ( sprint ) ทุก ๆ 2 สัปดาห์ ในขั้นตอนการวิเคราะห์ความต้องการของระบบจะต้องมีการหาข้อมูลเพิ่มเติมจากทีมอื่นๆ ซึ่งลักษณะข้อมูลที่ได้นั้นมีทั้งข้อมูลจากเอกสาร จากการอบรมโครงสร้างระบบของบริษัทจากงานระบบเดิมในบริษัทและตัวบุคคลจากทีมที่เกี่ยวข้อง

### 2) Designing project architecture

ออกแบบภาษาโปรแกรมมิ่งที่จะใช้ในการพัฒนาแต่ละส่วนของงานให้เหมาะสม และการจัดการโครงสร้างของงานให้เป็นไปตามรูปแบบขององค์กรโดยเรียนรู้การใช้โครงร่างซอฟต์แวร์. ( framework ) จากงานอื่น ๆ ในบริษัทและบุคลากรภายในทีม

### 3) Development and programming

เรียนรู้การใช้เครื่องมือต่าง ๆ และพัฒนาระบบต่าง ๆ ตามที่ออกแบบไว้

### 4) Testing

ทดสอบระบบที่พัฒนาแล้วและแก้ไขหากพบข้อผิดพลาด

### 5) Deployment

นำบริการ ( service ) หรือแอปพลิเคชันที่ทดสอบเสร็จสิ้นแล้วมาปรับใช้เข้ากับระบบของบริษัท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- Jira Software
- Bitbucket
- Postman
- Hoppscotch
- Jenkins
- Visual Studio Code
- Tunnelblick

### 3) ภาษาโปรแกรมมิ่ง

- Python
- Node.js ( React )
- Javascript
- Typescript
- Ruby
- Golang



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎีและกรอบการดำเนินงาน

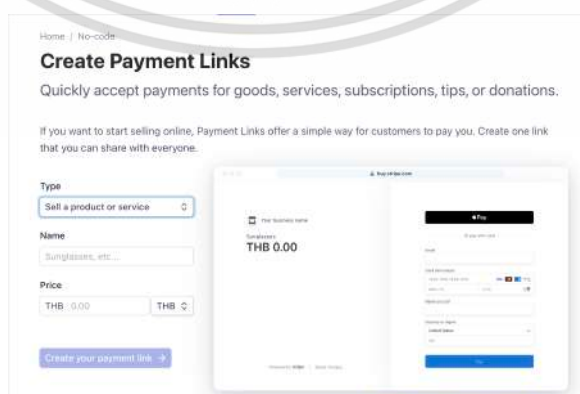
ศึกษาเรื่องการพัฒนาแอปพลิเคชันทรมันนี้ เดเวลอปเปอร์ แชนด์บ็อกซ์ จะต้องมีการศึกษาค้นคว้าทฤษฎีที่เกี่ยวข้องกับระบบ และมีองค์ความรู้ต่างๆที่ใช้ในการพัฒนาแบ่งออกเป็น 7 หัวข้อดังต่อไปนี้

- 2.1 การทำงานของระบบต้นแบบ
- 2.2 ทฤษฎีที่เกี่ยวข้องกับกรอบงาน
- 2.3 เครื่องมือที่ใช้ในการพัฒนา

#### 2.1 การทำงานของระบบต้นแบบ

ในขั้นตอนวิเคราะห์ความต้องการของระบบได้มีการสำรวจระบบทดสอบสำหรับผู้พัฒนา ( developer sandbox ) ซึ่งมีฟังก์ชันการทดสอบ API ที่เกี่ยวข้องกับการซื้อขายสินค้าและบริการด้วยสื่ออิเล็กทรอนิกส์ ( e-commerce ) ที่มีอยู่แล้วในท้องตลาดได้ที่ใกล้เคียงกับความต้องการของระบบเพื่อนำมาเป็นต้นแบบในการพัฒนา

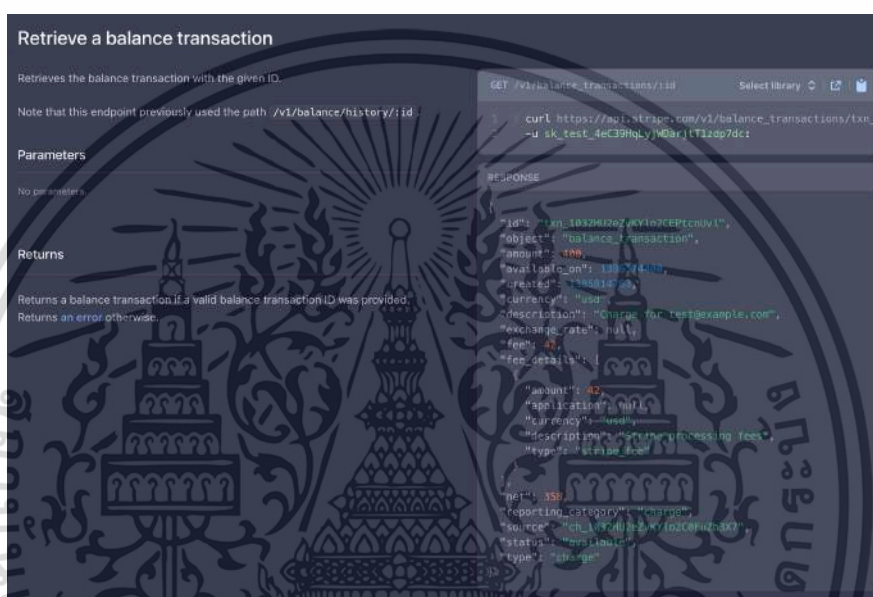
จากการสำรวจจึงได้นำระบบของสไตรป์ ( Stripe ) ซึ่งเป็นระบบให้บริการซอฟต์แวร์และ API เพื่อรับชำระเงินและจัดการธุรกิจทางออนไลน์มาเป็นต้นแบบ เพราะเป็นบริการที่ช่วยสร้างระบบด้วยฟังก์ชันธุรกรรมทางการเงินสำเร็จรูปผ่านการสมัครบัญชีของสไตรป์ ซึ่งมีตัวเลือกให้ผู้พัฒนาสามารถจำลองบริการธุรกรรมทางออนไลน์ได้โดยไม่ต้องเขียนโค้ดดังรูปที่ 2.1 โดยสามารถเลือกบริการที่มีตัวอย่างอยู่แล้วหรือปรับแต่งและใช้อินเทอร์เฟซการชำระเงินโดยตรงจากแชร์บอร์ด ( share board ) ของสไตรป์ จากนั้นสามารถทดสอบ API นั้นได้โดยส่งค่าผ่านระบบทดสอบในเว็บแอปพลิเคชัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามรูปที่ 2.1 Create Payment Links แบบไม่ใช้โค้ด ทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.1 แสดงการทดสอบการสร้างช่องทางชำระเงินให้กับลูกค้า ( Create Payment Links ) แบบไม่ใช้โค้ด ผู้พัฒนาสามารถปรับแต่งค่าต่างๆได้เองเพื่อสร้างฟังก์ชันช่องทางชำระเงิน

ในส่วนของการเลือกแบบต้องเขียนโค้ดสไตรปจะมีตัวอย่างการทำธุรกรรมทางการเงินออนไลน์ โดยการส่งค่าผ่าน REST API ซึ่งจะยอมรับค่าที่ส่งผ่านคำขอ ( request ) ที่เป็นไปตามการเข้ารหัสแบบฟอร์มที่กำหนด ( form-encoded ) และส่งการตอบกลับ ( response ) โดยเข้ารหัส JSON ( JSON-encoded ) ดังรูปที่ 2.2 และสามารถเพิ่มการส่งรหัสตอบกลับมาตรฐาน HTTP ( HTTP response codes ) ได้



รูปที่ 2.2 Retrieve a balance

จากรูปที่ 2.2 แสดงตัวอย่างการส่งคำขอเรียกยอดคงเหลือที่เรียกด้วย ID ที่กำหนด และแสดงผลการตอบกลับจากคำขอ API นั้น ซึ่งศึกษาคู่มือการใช้งานและคัดลอกไปใช้ในโหมดทดสอบของสไตรปได้ อีกทั้งยังมีไลบรารี ( library ) ต่าง ๆ ให้เลือกใช้อีกด้วย

การพิสูจน์ตัวตน ( authentication ) เพื่อรับคำคำขอสามารถพิสูจน์ตัวตนโดยใช้คีย์ API ( API key ) หรือ ใช้การรับรองความถูกต้องผ่าน HTTP ( HTTP Basic Auth ) โดยระบุคีย์ API ของผู้พัฒนาเป็นค่าชื่อผู้ใช้ตรวจสอบสิทธิ์พื้นฐาน ( basic auth username ) โดยไม่จำเป็นต้องระบุรหัสผ่าน ดังรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AUTHENTICATED REQUEST Select library
1  curl https://api.stripe.com/v1/charges \
2  -u sk_test_4eC39HqLyjWDarjtT1zdp7dc:
3  # The colon prevents curl from asking for a password.

If you need to authenticate via bearer auth (e.g., for a cross-origin request),
use -H "Authorization: Bearer sk_test_4eC39HqLyjWDarjtT1zdp7dc"
instead of -u sk_test_4eC39HqLyjWDarjtT1zdp7dc .

```

### รูปที่ 2.3 Create Payment Links แบบไม่ใช่โค้ด

จากรูปที่ 2.3 หากต้องการตรวจสอบสิทธิ์ผ่านผู้ถือสิทธิ์ (authenticate via bearer auth) ให้ใช้ `-H "Authorization: Bearer sk_test_4eC39HqLyjWDarjtT1zdp7dc"` แทน `-u sk_test_4eC39HqLyjWDarjtT1zdp7dc` โดยคำขอ API ทั้งหมดต้องทำผ่าน HTTPS การเรียกผ่าน HTTP ธรรมดาจะส่งคำขอไม่สำเร็จ คำขอ API ที่ไม่มีการรับรองความถูกต้องก็จะส่งคำขอไม่สำเร็จเช่นกัน

จุดเด่นของสไตรปคือผู้พัฒนาสามารถใช้ API ของสไตรปในโหมดทดสอบ (testing mode) ได้โดยไม่ส่งผลกระทบต่อข้อมูลจริงและสามารถทำการโต้ตอบกับเครือข่ายธนาคารได้ใกล้เคียงความเป็นจริง คีย์ API ที่ใช้ในการตรวจสอบความถูกต้องของคำขอจะสามารถกำหนดได้ว่าคำขอนั้นเป็นโหมดใช้งานจริงหรือโหมดทดสอบซึ่งผู้พัฒนาสามารถจัดการคีย์ API ได้ในหน้าจัดการข้อมูลของสไตรป (Stripe dashboard) และมีความเป็นมิตรกับผู้ใช้ (user friendly) มีการให้เอกสารคู่มือการใช้งานและวิธีการรับมือกับข้อผิดพลาดเบื้องต้น (handling errors) ไว้ให้ผู้พัฒนา

จากการศึกษาฟังก์ชันระบบทดสอบของสไตรป จึงได้นำฟังก์ชันดังกล่าวมาออกแบบตัดแปลงและกำหนดความต้องการให้เหมาะสมกับระบบ ทูร์มันนี่ เดเวลลอปเปอร์ แชนด์บ็อกซ์ โดยจะต้องสร้าง API จำลอง (mock API) ที่ร้านค้าสามารถใช้ได้เมื่อเชื่อมต่อเข้ากับระบบแอปพลิเคชัน ทูร์มันนี่ วอลเล็ต ซึ่งจะกำหนดตัวอย่างการส่งคำขอและผลการตอบกลับตามรูปแบบของบริษัทเท่าที่ API จำลองดังกล่าวจะสามารถทดสอบผ่านเว็บแอปพลิเคชันผ่านบัญชีผู้ใช้ (user account) ได้ มีการพิสูจน์ตัวตนเพื่อรับคำขอแบบการตรวจสอบสิทธิ์ผ่านผู้ถือสิทธิ์ (authenticate via bearer auth) โดยกำหนดโทเค็น (token) แทนผู้ใช้ เพื่อให้ผู้ใช้เห็นผลลัพธ์การส่งคำขอผ่าน API ได้รวดเร็วที่สุดตามเป้าหมายโดยไม่ต้องรอการยืนยันคีย์คู่ (key pair) และต้องมีการสร้างคู่มือการใช้งานสำหรับกรณีทดสอบต่าง ๆ รวมถึงสร้างไลบรารีหรือฟังก์ชันสำเร็จรูป (library หรือ SDK) ที่จำเป็นเพื่อความสะดวกแก่ผู้ใช้ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 ทฤษฎีที่เกี่ยวข้องกรอบงาน

ในการพัฒนาระบบทดสอบ API ตามแผนงานจำเป็นต้องมีการศึกษาเรียนรู้ทฤษฎีหลักต่าง ๆ ที่เกี่ยวข้อง ดังนี้

- 2.2.1) REST API
- 2.2.2) การรับรองความถูกต้องพื้นฐาน
- 2.2.3) Mountebank
- 2.2.4) Robot Framework

### 2.2.1 REST API

REST API เป็นอินเทอร์เฟซที่ระบบคอมพิวเตอร์สองระบบใช้เพื่อแลกเปลี่ยนข้อมูลผ่านอินเทอร์เน็ตได้อย่างปลอดภัย โดยจะรับรองการแลกเปลี่ยนข้อมูลนี้ให้เป็นไปตามมาตรฐาน

ฟังก์ชันพื้นฐานของ REST API คือ ไคลเอ็นต์ ( client ) จะติดต่อกับเซิร์ฟเวอร์ ( server ) โดยใช้ API เมื่อต้องใช้ทรัพยากร โดยไคลเอ็นต์ส่งคำขอไปยังเซิร์ฟเวอร์ตามรูปแบบคำขอในลักษณะที่เซิร์ฟเวอร์เข้าใจได้ เซิร์ฟเวอร์จะรับรองความถูกต้องของไคลเอ็นต์และประมวลผล จากนั้นจะส่งคืนผลการตอบรับกลับไปยังไคลเอ็นต์ โดยปกติแล้วเซิร์ฟเวอร์จะระบุเส้นทางไปยังทรัพยากร โดยใช้ตัวชี้แหล่งในอินเทอร์เน็ต ( Uniform Resource Locator หรือ URL )

นักพัฒนามักใช้ REST API โดยใช้เกณฑ์วิธีขนส่งข้อความหลายมิติ ( Hypertext Transfer Protocol หรือ HTTP ) วิธีการ HTTP จะบอกให้เซิร์ฟเวอร์ทราบถึงสิ่งที่ต้องทำกับทรัพยากร โดยวิธีการ HTTP ทั่วไปมี 4 วิธีดังต่อไปนี้

- 1) GET ไคลเอ็นต์ใช้เพื่อเข้าถึงทรัพยากรตามที่อยู่บน URL
- 2) POST ไคลเอ็นต์ใช้เพื่อส่งข้อมูลไปยังเซิร์ฟเวอร์
- 3) PUT ไคลเอ็นต์ใช้เพื่ออัปเดตทรัพยากรที่มีอยู่บนเซิร์ฟเวอร์
- 4) DELETE ไคลเอ็นต์ใช้คำขอเพื่อลบทรัพยากรออก

จากการศึกษา REST API ได้นำความรู้มาใช้ในการออกแบบและจำแนกประเภท API ต่าง ๆ ที่ต้องมีในระบบทดสอบทั้ง API เดิมของบริษัทและ API ใหม่ที่ต้องพัฒนาในระบบ

### 2.2.2 การรับรองความถูกต้องพื้นฐาน

ในการรับรองความถูกต้องพื้นฐาน ไคลเอ็นต์จะส่งชื่อผู้ใช้และรหัสผ่านในส่วนหัวของคำขอ ซึ่งจะเข้ารหัสด้วย base64 ซึ่งเป็นเทคนิคการเข้ารหัสที่แปลงทั้งคู่เป็นชุดอักขระ 64 ตัวเพื่อการรับส่งข้อมูลที่ปลอดภัย

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือมีเงื่อนไขเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1) การรับรองความถูกต้องของแบเรอร์ หรือ การตรวจสอบสิทธิ์ผ่านผู้ถือสิทธิ์

กระบวนการให้การควบคุมการเข้าถึงแก่ผู้ถือสิทธิ์ ( bearer ) ของโทเค็น ( token ) โดยโทเค็นแบบแบเรอร์มักจะเป็นสตริงอักขระที่เข้ารหัสซึ่งเซิร์ฟเวอร์สร้างขึ้นเพื่อตอบสนองต่อคำขอเข้าสู่ระบบ จากนั้นไคลเอ็นต์จะส่งโทเค็นในส่วนหัวของคำขอ ( request header ) เพื่อเข้าถึงทรัพยากร

### 2) คีย์ API ( API key )

การรับรองความถูกต้องที่เซิร์ฟเวอร์จะกำหนดค่าที่สร้างขึ้นเฉพาะให้กับไคลเอ็นต์ที่ใช้งานครั้งแรก เมื่อใดก็ตามที่ไคลเอ็นต์พยายามเข้าถึงทรัพยากร ก็จะใช้คีย์ API เฉพาะเพื่อยืนยันตัวเอง ทั้งนี้คีย์ API มีความปลอดภัยน้อยกว่าและใช้เวลามากกว่า เนื่องจากไคลเอ็นต์ต้องส่งคีย์ ซึ่งทำให้เสี่ยงต่อการถูกขโมยผ่านเครือข่าย

### 3) OAuth

การรวมรหัสผ่านและโทเค็นเข้าด้วยกันเพื่อการเข้าสู่ระบบที่มีความปลอดภัยสูง โดยเซิร์ฟเวอร์จะขอรหัสผ่านก่อนแล้วจึงขอโทเค็นเพิ่มเติมเพื่อให้กระบวนการรับรองความถูกต้องเสร็จสมบูรณ์ ซึ่งสามารถตรวจสอบโทเค็นได้ตลอดเวลาและเมื่อเวลาผ่านไปด้วยขอบเขตและระยะเวลาที่เฉพาะเจาะจง

ซึ่งใช้ความรู้ส่วนนี้ในการออกแบบการรับรองความถูกต้องในการรับคำขอให้เหมาะสมกับระบบที่พัฒนา โดยใช้การรับรองความถูกต้องแบบการตรวจสอบสิทธิ์ผ่านผู้ถือสิทธิ์ และกำหนดโทเค็นแทนบุคคลต่าง ๆ ให้ผู้พัฒนาของร้านค้านำไปใช้ทดสอบ ซึ่งเป็นการใช้การรับรองความถูกต้องที่ต่างจากบริการอื่น ๆ ในบริษัทเพื่อให้ตรงตามเป้าหมายที่เน้นความสะดวกและรวดเร็วในการทดสอบ

## 2.2.3 Mountebank

เมาท์แบงก์ ( Mountebank ) คือ เครื่องมือที่ใช้ทำระบบหลังบ้านจำลอง ( Legacy systems ) ที่เรียกว่าสตั๊ป ( stub ) เพื่อสร้างเงื่อนไขทดสอบฟังก์ชัน ประกอบด้วย 2 ส่วน ได้แก่

- 1) เงื่อนไข ( Predicate )
- 2) ผลการตอบกลับ ( Response )

ซึ่งสตั๊ปจะประกอบด้วยเงื่อนไขเท่าใดก็ได้ ความรู้ส่วนนี้ได้นำมาใช้ในการสร้างเซิร์ฟเวอร์จำลอง ( mock service ) เพื่อทดสอบผลการตอบกลับของ API จำลองและสร้าง

API จำลอง ( mock API ) จาก API เดิมของบริษัทโดยเลือกเฉพาะ API ที่ผู้พัฒนาของร้านค้าเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้ใช้จริงรวมถึง API ที่สร้างเพิ่มเติมเพื่อใช้ในการทดสอบ API บนเว็บแอปพลิเคชัน  
ทอร์มนี่ เดเวลอปเปอร์ แชนด์บ็อกซ์

เม้าท์แบงค์มีขั้นตอนการทำงาน คือ รับค่าที่ส่งมากับคำขอ ( request ) ผ่าน API มาตรวจสอบว่าตรงกับเงื่อนไข ( predicate ) ที่ตั้งไว้ก่อนหน้านี้หรือไม่ ถ้ามีรูปแบบตรงกัน จะส่งผลการตอบรับ ( response ) ของเงื่อนไขนั้นกลับมา ซึ่งเซิร์ฟเวอร์จำลองของเม้าท์แบงค์สามารถทดสอบได้ตลอด รองรับการ ทำงานผ่าน HTTPS และสามารถประมวลผลบน NodeJS ได้

ยกตัวอย่างเหตุการณ์ที่ต้องใช้เม้าท์แบงค์ สมมติว่ามี API สร้างผู้ใช้ ชื่อ create user จากเซอร์วิส ( service ) X ต้องการส่งคำขอไปยัง create user ที่เซอร์วิส Y แต่เนื่องจาก เซอร์วิส Y ยังทำงานไม่สำเร็จ หรือ การ create user มีข้อมูลที่อ่อนไหว เช่น เลขบัตรประชาชน เป็นต้น จึงจำเป็นต้องจำลองเซอร์วิส Y ขึ้นใหม่เพื่อให้สามารถทดสอบ API ได้ตลอด โดยมีรูปแบบโครงสร้างการพัฒนา ดังรูป 2.4 และ 2.5



รูปที่ 2.4 Project Structure

จากรูปที่ 2.4 แสดงรูปแบบโครงสร้างของเม้าท์แบงค์ประกอบด้วย

- 1) โฟลเดอร์ชื่อหัวข้อ API การสร้างผู้ใช้ ( profile )
  - a. กรณีทดสอบที่ส่งคำขอไม่สำเร็จ ( fail )
    - i. เงื่อนไข ( fail\_predicate )
    - ii. ผลการตอบกลับ ( fail\_response )
  - b. กรณีทดสอบที่ส่งคำขอสำเร็จ ( success )
    - i. เงื่อนไข ( fail\_predicate )
    - ii. ผลการตอบกลับ ( fail\_response )
- 2) ระบุเงื่อนไขทั้งหมดและปลายทางที่จะให้เม้าท์แบงค์จำลองเซอร์วิส

เอกสารนี้เป็นเอกสารที่สงวนไว้ ( imposter ) านเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีก 3) ปลายทางเพิ่มเติมในกรณีที่คำขอไม่ตรงตามเงื่อนไขทั้งหมดที่ตั้งไว้ ( proxy ) ้

## กรณีทดสอบที่ส่งคำขอไม่สำเร็จ ( fail )

```

fail_predicates.ejs                                fail_response.ejs
{
  "and": [{
    "equals": {
      "path": "/customer/v1/profiles/create",
      "method": "POST"
    }
  },
  {
    "matches": {
      "body": {
        "cardIdentification": "^FF.*",
        "requestTransactionId": "^.*"
      }
    }
  }
]}

{
  "is": {
    "statusCode": 200,
    "headers": {
      "Content-Type": "application/json"
    },
    "body": {
      "request_id": "12345678",
      "response_code": "1001",
      "response_desc": "mock error"
    }
  }
}
// การ return response แบบ fail

```

### รูปที่ 2.5 Mock ในส่วนของ fail

จากรูปที่ 2.5 การจำลอง ( mock ) กรณีทดสอบที่ส่งคำขอไม่สำเร็จ มีรายละเอียดดังนี้

#### 1) เงื่อนไข ( predicate )

ส่วน request

- มี path: /customer/v1/profiles/create
- มี method: POST

ส่วน body

- ส่ง cardIdentification มีค่านำหน้าคือ FF ตามด้วยอักขระอะไรก็ได้
- ส่ง requestTransactionId มีค่าเข้ามาเป็นอะไรก็ได้จึงจะเข้าเงื่อนไขทั้งหมด

#### 2) ผลการตอบกลับ ( response )

ส่วนตรวจสอบ

- ตรวจสอบเงื่อนไขว่าส่งคำขอสำเร็จ statusCode เท่ากับ 200
- ตรวจสอบ header ว่ามี Content-Type ตามที่กำหนด

ส่วน body

- ส่ง request\_id เท่ากับ 12345678
- ส่ง response\_code เท่ากับ 1001
- ส่ง response\_desc ส่งคำอธิบายเป็น mock error

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กรณีทดสอบที่ส่งคำขอสำเร็จ ( success )

```

success_predicates.ejs                                success_response.ejs
{
  "and": [{
    "equals": {
      "path": "/customer/v1/profiles/create",
      "method": "POST"
    }
  }, {
    "matches": {
      "body": {
        "cardIdentification": "^XY.*",
        "requestTransactionId": "^.*"
      }
    }
  }
]
}

{
  "is": {
    "statusCode": 200,
    "headers": {
      "Content-Type": "application/json"
    },
    "body": {
      "request_id": "12345678",
      "response_code": "0000",
      "response_desc": "SUCCESS"
    }
  }
}

// การ return response แบบ success

```

### รูปที่ 2.6 Mock ในส่วนของ success

จากรูปที่ 2.6 การจำลอง ( mock ) กรณีทดสอบที่ส่งคำขอสำเร็จ มีรายละเอียดดังนี้

#### 1) เงื่อนไข ( predicate )

ส่วน request

- มี path: /customer/v1/profiles/create
- มี method: POST

ส่วน body

- ส่ง cardIdentification มีค่านำหน้าคือ XY ตามด้วยอักขระอะไรก็ได้
- ส่ง requestTransactionId มีค่าเข้ามาเป็นอะไรก็ได้จึงจะเข้าเงื่อนไขทั้งหมด

#### 2) ผลการตอบกลับ ( response )

ส่วนตรวจสอบ

- ตรวจสอบเงื่อนไขว่าส่งคำขอสำเร็จ statusCode เท่ากับ 200
- ตรวจสอบ header ว่ามี Content-Type ตามที่กำหนด

ส่วน body

- ส่ง request\_id เท่ากับ 12345678
- ส่ง response\_code เท่ากับ 0000
- ส่ง response\_desc ส่งคำอธิบายเป็น SUCCESS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Mountebank demo

```

1. imposters.ejs
2. {
3.   "imposters": [{
4.     "port": "6969",
5.     "protocol": "http",
6.     "name": "profiles",
7.     "stubs": [{
8.       "predicates": [ <% include profiles/success/success_predicates.ejs %> ],
9.       "responses": [ <% include profiles/success/success_response.ejs %> ]
10.    },{
11.     "predicates": [ <% include profiles/fail/fail_predicates.ejs %> ],
12.     "responses": [ <% include profiles/fail/fail_response.ejs %> ]
13.    },{
14.     "responses": [ <% include proxy.ejs %> ]
15.    }
16.   ]
17. }

```

## รูปที่ 2.7 Mountebank Imposter

จากรูปที่ 2.7 แสดงตัวอย่าง การระบุเงื่อนไขทั้งหมดและปลายทางที่จะให้  
 เม้าท์แบงค์จำลองเซอร์วิส ( imposter ) บรรทัดที่ 4 คือช่องทาง (port) ของ  
 ปลายทางที่จะจำลองเม้าท์แบงค์ บรรทัดที่ 5 คือการกำหนดโปรโตคอล  
 ( protocol ) บรรทัดที่ 7-13 คือสตัป ( stub ) หรือกรณีทดสอบทั้งหมดที่ต้องการ  
 ทดสอบบนเม้าท์แบงค์ ประกอบด้วยกรณีที่ส่งคำขอสำเร็จและไม่สำเร็จ และบรรทัด  
 ที่ 13-15 คือปลายทางสำหรับคำขอที่ไม่เข้าเงื่อนไขข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1) proxy.ejs
2) {
3)   "proxy": {
4)     "to": "https://www.google.com",
5)     "mode": "proxyAlways"
6)   }

```

### รูปที่ 2.8 Mountebank Proxy

จากรูปที่ 2.8 แสดงกรณีที่มีการส่งคำขอ ( request ) ไม่เข้าเงื่อนไขที่กำหนด จะถูกส่งไปที่ปลายทางที่กำหนดในพร็อกซี ( proxy ) แทน คือ <https://www.google.com>

#### 2.2.4. Robot Framework

```

*** Settings ***
Documentation      Example using the space separated format.
Library           OperatingSystem

*** Variables ***
${MESSAGE}       Hello, world!

*** Test Cases ***
My Test
[Documentation]   Example test.
Log              ${MESSAGE}
My Keyword       ${CURDIR}

Another Test
Should Be Equal  ${MESSAGE}  Hello, world!

*** Keywords ***
My Keyword
[Arguments]      ${path}
Directory Should Exist  ${path}

```

### รูปที่ 2.9 Robot Framework format

จากรูปที่ 2.9 โรบอทเฟรมเวิร์ค ( Robot Framework ) เป็นซอฟต์แวร์ ที่ใช้ในการทดสอบการทำงานของ API จำลอง โดยภาษาที่ใช้คือ Python ซึ่งโครงสร้างหลักของโรบอทเฟรมเวิร์คมีดังนี้

#### 1) Settings

การกำหนดไลบรารี ( Library ) ที่ใช้ในการเรียกการทดสอบ และเรียก Keyword จากไลบรารีที่ต้องการมาใช้ เพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ 2) Variables มีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประกาศตัวแปรในการทำงานที่เรียกใช้บ่อย ๆ เมื่อต้องการเปลี่ยนค่าของตัวแปรสามารถเปลี่ยนทำได้โดยไม่ต้องเปลี่ยนในทุกส่วนที่เรียกใช้เอง

### 3) TestCases

การกำหนด ชื่อในกรณีทดสอบ ( Test Case ) ตามหัวข้อในการทำงาน และนำชื่อ Keyword มาไว้ใน Test cases เพื่อให้ Robot framework เรียกประมวลผลข้อมูลตามลำดับ

### 4) Keywords

ชุดคำสั่งที่สามารถเรียกใช้ได้จากไลบรารีสำเร็จรูป เช่น Selenium2Library หรือสร้าง keyword ขึ้นมาใช้งานเอง

## 2.2.5 CI/CD

กระบวนการในการทำงาน Pipeline ประกอบด้วย CI ( Continuous Integration ) คือ กระบวนการที่ใช้สำหรับการรวบรวมซอฟต์แวร์ที่มีการพัฒนาแยกส่วนกัน และ CD ( Continuous Delivery ) คือ กระบวนการต่าง ๆ ใน deployment pipeline ซึ่งจะเริ่มทำงานตั้งแต่การประมวลผล ( compile ) สร้าง ( build ) ไปจนถึงขั้นตอนการทดสอบต่างๆ แบบอัตโนมัติ ส่วนขั้นตอนการนำไปใช้ ( deployment ) ขึ้นระบบสภาพแวดล้อมจริง ( production ) นั้น จะต้องได้รับการอนุมัติจากเจ้าของผลิตภัณฑ์ ( product owner ) ก่อน

## 2.3 เครื่องมือที่ใช้ในการพัฒนา

### 2.3.1 Bitbucket

เครื่องมือในการไว้ใช้จัดการ Code collaboration และ Version control ซึ่งสามารถ test, deploy และสามารถสร้าง Private Git repositories ภายในทีมได้หรือกำหนดบุคคลในการ Comment การเปลี่ยนแปลงโค้ดก่อนการ deploy ได้ ซึ่งได้ใช้ในการสร้าง Private Git repositories ภายในทีม

### 2.3.2 Postman

เครื่องมือสำหรับช่วยในการพัฒนา API ทดสอบการทำงานของ Service รวมถึงการ Mock Service โดยความนิยมของ Postman หลัก ๆ มาจาก UI ที่สวยงามใช้งานง่ายกว่า Tools อื่น ๆ โดยผู้ใช้งานไม่จำเป็นต้องมีความรู้เรื่องภาษาโปรแกรมก็สามารถใช้งานได้

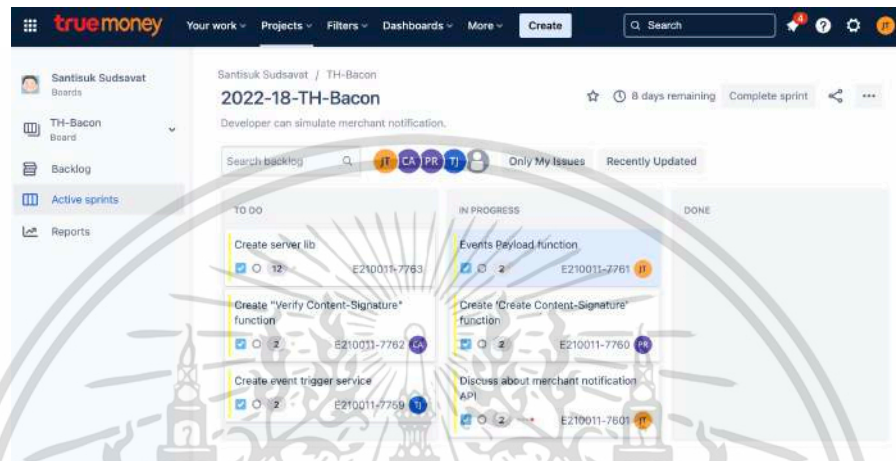
### 2.3.3 Jenkins

Jenkins เป็น Tool ถูกเขียนด้วยภาษา Java ถูกสร้างขึ้นมาเพื่อตอบสนอง CI/CD ที่จะรัน Test และเก็บ Test Report และคอยแจ้งปัญหาเวลา Test Fail นั้นเอง ทั้งยังมีชุดทดสอบเพื่อสามารถทดสอบย้อนหลังได้รวดเร็ว ซึ่งเราสามารถนำไปใช้งานได้ง่าย

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่ควรนำเอกสารนี้ไปใช้ในการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.4 Jira Software

แพลตฟอร์มที่มีระบบ Task Management ใช้วางแผนและมอบหมายงาน ซึ่งเป็นระบบที่ทุกคนภายในทีมสามารถติดตามความคืบหน้าของโครงการ และแก้ไขข้อมูลอัปเดตสถานะของงานส่วนที่ตนเองรับผิดชอบได้ ดังรูปที่ 2.10 แสดงตัวอย่างการวางแผนงานโดยใช้ Jira Software



รูปที่ 2.10 ตัวอย่าง Task management ของ Jira Software

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### วิธีการออกแบบและพัฒนาระบบ

ในการพัฒนาเว็บแอปพลิเคชัน ทรูมันนี่ เดเวลอปเปอร์ แซนด์บ็อกซ์ ( TrueMoney Developer Sandbox ) โดยจะเป็นการจำลองจากฟังก์ชันที่มีอยู่ก่อนแล้ว ในการพัฒนาเว็บแอปพลิเคชันได้มีการรวบรวมข้อมูลของฟังก์ชันจากทีมที่เป็นผู้พัฒนาฟังก์ชันเหล่านี้ เพื่อสร้างระบบจำลองที่สามารถให้นักพัฒนาจากร้านค้าต่าง ๆ สามารถนำไปใช้ได้ โดยสามารถแบ่งการดำเนินการออกเป็น 3 ขั้นตอน ดังนี้

- 3.1 ขั้นตอนการศึกษาและรวบรวมข้อมูล
- 3.2 ขั้นตอนการวิเคราะห์ระบบ
- 3.3 ขั้นตอนการเตรียมข้อมูล
- 3.4 ขั้นตอนการออกแบบและการดำเนินงาน

#### 3.1 ขั้นตอนการศึกษาปัญหาและรวบรวมข้อมูล

ศึกษาการทำงานของระบบที่มีอยู่เดิมเพื่อรวบรวมความต้องการและปัญหาต่าง ๆ ของระบบที่เกิดขึ้นเพื่อนำมาออกแบบระบบใหม่ ในขั้นต้นนั้นได้มีการรวบรวมข้อมูลที่ต้องใช้ในการทำเว็บแอปพลิเคชันก่อนซึ่งสิ่งที่ร้านค้าต้องใช้ในการรณำทรูมันนี่ วอลเล็ตไปเพิ่มเป็นช่องทางการชำระเงินต้องใช้ Open API

Open API คือ API ที่ทรูมันนี่มอบให้กับร้านค้าที่ต้องการเพิ่มทรูมันนี่ วอลเล็ตเป็นช่องทางการชำระเงิน API ส่วนใหญ่จะเป็น API ที่เกี่ยวข้องกับข้อมูลของลูกค้า เช่น ข้อมูลส่วนตัว ยอดเงินในบัญชีทรูมันนี่ วอลเล็ต เป็นต้น ข้อมูลต่าง ๆ สำหรับการใช้ API นี้เป็นการจำลองขึ้นมาใหม่ทั้งหมด เพื่อป้องกันข้อมูลที่มีความอ่อนไหวและเป็นการไม่กระทบกับสภาพแวดล้อมของระบบทรูมันนี่โดยตรง โดย Open API จำลองที่สร้างขึ้นมามีดังนี้

- Get TrueMoney ID คือ API ที่ใช้จำลองการส่งรหัสบัญชีทรูมันนี่ของผู้ใช้
- Get balance คือ API ที่ใช้จำลองการส่งยอดเงินในบัญชีทรูมันนี่ วอลเล็ตของผู้ใช้
- Get agent status คือ API ที่ใช้จำลองการส่งสถานะบัญชีของผู้ใช้
- Get NDID status คือ API ที่ใช้จำลองการส่งสถานะการยืนยันตัวตนดิจิทัล ( National Digital ID ) ของบัญชีผู้ใช้ ซึ่งผู้ใช้จะต้องได้รับการลงทะเบียน NDID ในระบบ TrueMoney

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Send notify คือ API ที่ใช้จำลองการส่งผลลัพธ์ของการแจ้งเตือนไปยังผู้ใช้ ประเภทการส่งข้อความคือ SMS และข้อความที่ต้องการแจ้งเตือนผู้ใช้
- Get Thai ID card picture คือ API ที่ใช้จำลองการส่งภาพบัตรประชาชนของผู้ใช้ ซึ่งภาพบัตรประชาชนของผู้ใช้จะมีการเข้ารหัส base64
- Create order คือ API ที่ใช้จำลองการสร้างคำสั่งซื้อสินค้าของร้านค้า ซึ่งจะได้รับช่องทางการชำระเงิน (Checkout URL) ผ่านทรมันนี้ วอลเล็ตกลับมา
- Merchant notification คือ API ที่ใช้สามารถทดลองการแจ้งเตือนไปยังร้านค้าเมื่อมีการเมื่อมีการทำกิจกรรม (ธุรกรรม) ใด ๆ

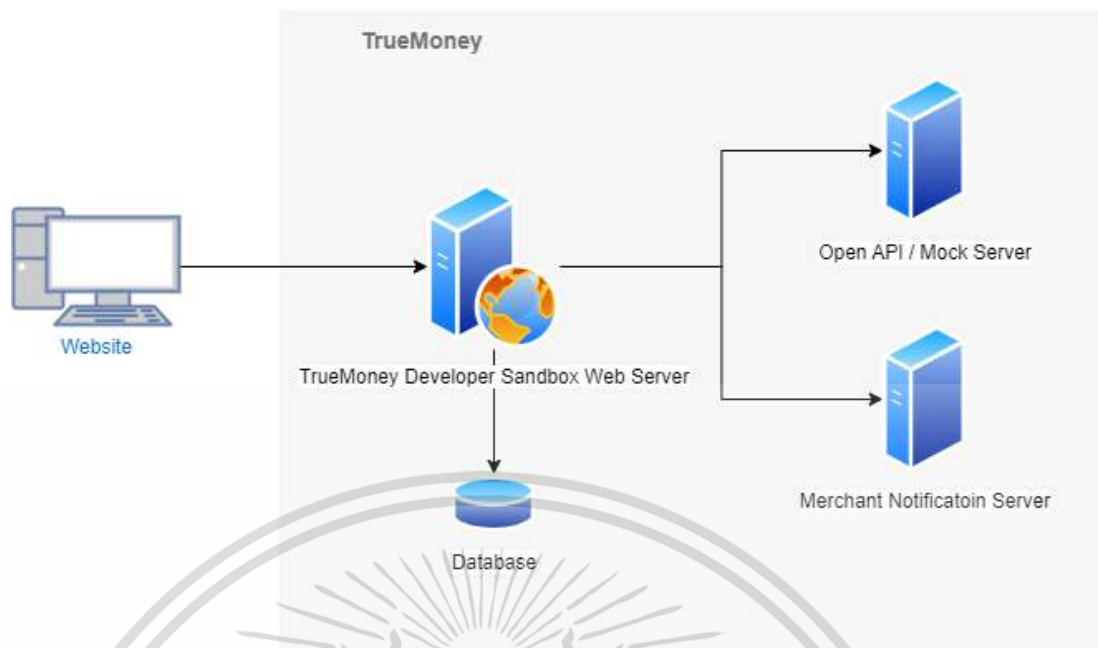
ขั้นตอนต่อไปจะเป็นการรวบรวมความต้องการของเว็บแอปพลิเคชันที่จะนำมาใช้สำหรับจำลองการใช้ Open API โดยจะการทำเป็นเว็บแอปพลิเคชันนี้จะช่วยให้ร้านค้ารัฐศึกษาและเข้าถึงได้ง่าย โดยฟังก์ชันของเว็บแอปพลิเคชันที่ต้องการมีดังนี้

- สามารถให้ผู้ใช้สร้างบัญชีผู้ใช้และเข้าสู่ระบบได้
- สามารถแสดงวิธีการใช้งาน Open API จำลองได้
- สามารถใช้งาน Open API จำลองได้
- สามารถเก็บและแสดงประวัติการเรียกใช้ Open API จำลองได้ ( Log )

### 3.2 ขั้นตอนการวิเคราะห์ระบบ

ศึกษาและวิเคราะห์การทำงานของแต่ละ API ในระบบ โดยสถาปัตยกรรมระบบของทรมันนี้ เดเวลอปเปอร์ แชนด์บ็อกซ์เป็น ดังรูป 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 สถาปัตยกรรมระบบของทรูมันนี่ เดเวลอปเปอร์ แซนด์บ็อกซ์

### 3.2.1 เว็บไซต์ ( Website )

เป็นส่วนต่อขยายที่เชื่อมกับฝั่งผู้ใช้ มีหน้าที่แสดงเอกสารการใช้งาน Open API จำลอง และทดสอบการใช้ Open API จำลอง

### 3.2.2 เว็บไซต์เวอร์ทรูมันนี่เดเวลอปเปอร์แซนด์บ็อกซ์ ( TrueMoney Developer Sandbox Web Server )

มีหน้าที่ในการส่งข้อมูลอย่างเอกสารการใช้งาน API จำลอง และมีหน้าที่รับคำสั่งการทดสอบ API จำลองแล้วส่งไปที่ API หรือเซิร์ฟเวอร์จำลอง ( mock server ) และ เซิร์ฟเวอร์การแจ้งเตือนร้านค้า ( Merchant notification server ) และมีหน้าที่บันทึกการใช้งาน Open API จำลอง

### 3.2.3 API หรือเซิร์ฟเวอร์จำลอง ( Mock server )

มีหน้าที่จำลองฟังก์ชันต่าง ๆ ที่จำลองมาจาก API จริง เพื่อให้ร้านค้าสามารถทดลองใช้ในระบบโดยใช้ Mountebank

### 3.2.4 เซิร์ฟเวอร์การแจ้งเตือนร้านค้า ( Merchant Notification Server )

มีหน้าที่ตรวจสอบและยืนยันกับฝั่งร้านค้าว่าทำกิจกรรมในทรูมันนี่ เดเวลอปเปอร์ แซนด์บ็อกซ์ได้ถูกต้องและสำเร็จหรือไม่ โดยใช้ การสร้างและตรวจสอบลักษณะเฉพาะของ

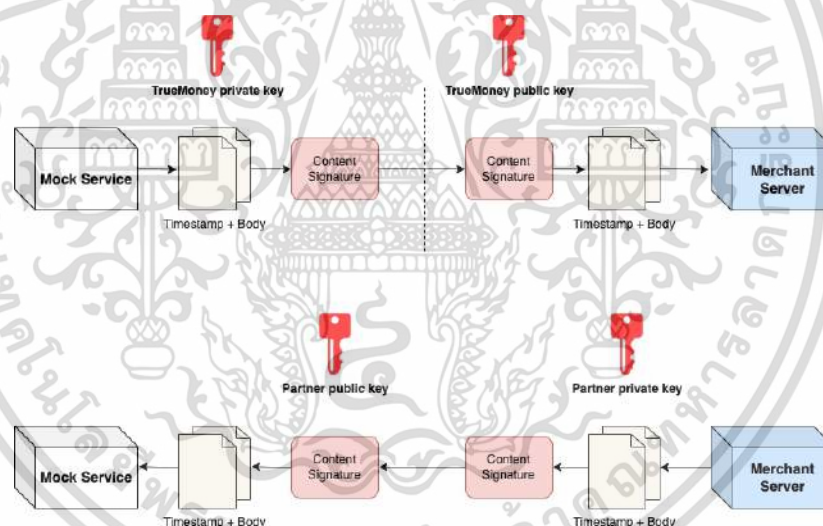
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้ใช้ภายในเท่านั้น ไม่สามารถเผยแพร่ไปภายนอกได้โดยไม่ได้รับอนุญาต  
ข้อมูลและภาษา Go ในการสร้างเซิร์ฟเวอร์จำลองรวมถึงมีการเตรียม SDK และตัวอย่างคำสั่ง  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ร้านค้าสามารถนำไปใช้กับภาษาโปรแกรมมิ่งได้ทั้งหมด 4 ภาษา ได้แก่ Java, Node.js, Python และ Ruby เพื่อลดความยุ่งยากในการใช้งานระบบของผู้พัฒนาฝั่งร้านค้า

### การสร้างและตรวจสอบลักษณะเฉพาะของข้อมูล

สร้างลักษณะเฉพาะของข้อมูลจากการนำข้อมูลหรือข้อความที่จะส่งให้กับร้านค้ามารวมกับประทับเวลาแล้วทำการเซ็นข้อมูลด้วยกุญแจส่วนตัวของทรูมันนี่หรือร้านค้า และการตรวจสอบลักษณะเฉพาะของข้อมูลจะเป็นการยืนยันลักษณะเฉพาะของข้อมูลด้วยกุญแจสาธารณะที่คู่กันกับกุญแจส่วนตัวที่ได้ทำการเซ็นไว้

ในการส่งคำขอที่มีการสร้างลักษณะเฉพาะของข้อมูลที่ฝั่งของเซิร์ฟเวอร์จำลองซึ่งสร้างด้วยกุญแจส่วนตัวของทรูมันนี่และตรวจสอบด้วยกุญแจสาธารณะของทรูมันนี่ที่ฝั่งเซิร์ฟเวอร์ของร้านค้า ในส่วนถัดไปที่มีการสร้างลักษณะเฉพาะของข้อมูลที่ฝั่งของร้านค้าจะถูกสร้างด้วยกุญแจส่วนตัวของร้านค้าและตรวจสอบด้วยกุญแจสาธารณะของร้านค้าในฝั่งเซิร์ฟเวอร์จำลองของทรูมันนี่ ดังภาพที่ 3.2



รูปที่ 3.2 การทำงานการยืนยันลักษณะเฉพาะของข้อมูล

### 3.3 ขั้นตอนการเตรียมข้อมูล

การพัฒนาระบบทดสอบ API จำลองจำเป็นต้องมีการเตรียมข้อมูลเสมือนจริงเพื่อใช้ในการทดสอบ API ต่าง ๆ จะมีการเตรียมข้อมูลไว้ 2 ส่วน ได้แก่ การเตรียมข้อมูลทดลองสำหรับเว็บแอปพลิเคชันดังรูปที่ 3.3 และรูปที่ 3.4 โดยมีแหล่งข้อมูลจากเอกสาร จากงานในเซอร์วิส ( service ) อื่น ๆ ที่คล้ายคลึงกันภายในบริษัท และจากตัวบุคคลของทีมอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

id	component_id	name	type	description	path	method	req_exp	resp_exp	created_at	updated_at
1	get-truemone...	Get T...	user	This API fo...	/use...	GET	# GET /...	('status'...	2022-1...	2022-1...
2	get-balance	Get B...	user	This API is...	/use...	GET	# GET /...	('status'...	2022-1...	2022-1...
3	get-ndid-stat...	Get...	user	This API fo...	/use...	GET	# GET /...	('status'...	2022-1...	2022-1...
4	get-agent-sta...	Get A...	user	This API is...	/use...	GET	# GET /...		2022-1...	2022-1...
5	get-thai-ID-c...	Get T...	user	This API fo...	/use...	GET	# GET /...		2022-1...	2022-1...
6	send-notify	Send...	user	This API fo...	/use...	POST	# POST...		2022-1...	2022-1...
7	save-consent	Save...	user	This API fo...	/use...	POST	# POST...		2022-1...	2022-1...
8	get-consent	Get...	user	This API fo...	/use...	GET	# GET /...		2022-1...	2022-1...
9	get-profile	Get P...	user	This API fo...	/use...	GET	# GET /...		2022-1...	2022-1...
10	create-order	Creat...	order	This API fo...	/pay...	POST	# POST...		2022-1...	2022-1...
11	get-payment...	Get P...	pay...	This API fo...	/pay...	GET	# GET /...		2022-1...	2022-1...
12	merchant-noti	Merc...	mer...	This API fo...	/trig...	POST	# POST...		2022-1...	2022-1...

รูปที่ 3.3 ข้อมูลในฐานข้อมูล web page

จากรูปที่ 3.3 แสดงภาพการเตรียมข้อมูลทดลองสำหรับเว็บแอปพลิเคชันดัง เพื่อทำการบันทึก (upload) เข้าไปในระบบโดยเก็บไว้ในฐานข้อมูล web page



รูปที่ 3.4 ข้อมูลสำหรับผู้พัฒนาฝั่งทรมันนี่

จากรูปที่ 3.4 แสดงภาพการเตรียมข้อมูลสำหรับอธิบายรายละเอียดงานเพื่อใช้ในการวางแผนงานในทีมและใช้เป็นเอกสารของบริษัท

โดยมีรายละเอียดข้อมูลที่ต้องเตรียมทั้งหมด 4 ส่วนดังนี้

### 3.1.1 สร้างข้อมูลผู้ใช้ประเภทต่าง ๆ

สร้างข้อมูลผู้ใช้ประเภทต่าง ๆ เพื่อให้ นักพัฒนาของร้านค้าใช้ระบบทดสอบ โดยประเภทผู้ใช้จะแบ่งเป็น

#### 1) ร้านค้า ( Merchant )

- ร้านค้าที่มีสถานะพร้อมใช้งานครบถ้วน
- ร้านค้าที่มีสถานะบางอย่างไม่พร้อมใช้งาน เช่น ร้านค้าไม่พร้อมใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับสัญญาไม่พร้อมใช้งาน หรือ สัญญาหมดอายุ เป็นต้น ซึ่งประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีก 2) ลูกค้า (User) ings เนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ลูกค้ำที่มีข้อมูลครบถ้วน
- ลูกค้ำที่มีข้อมูลไม่ครบถ้วน

### 3) ข้อมูลโทเค้น (Token) แสดงประเภทของข้อมูลในระบบของลูกค้ำดังรูปที่ 3.5

#### User Persona

Created by Jinjutha Tedcharoen, with a template  
Last updated: Sep 27, 2022 · 2 min read · 8 people viewed

For security access, client need to request with an authentication token which is a bearer token along with header. Each authorization tokens in header will be automatically generated various response .

##### Table of Tokens

- Trumoney's Customer
  - tok\_all\_scope\_complete\_info
  - tok\_all\_scope\_incomplete\_info
  - tok\_no\_scope
  - tok\_expired
- Merchant
  - Create Order Merchant
  - Product Code
  - Terminal Type

### รูปที่ 3.5 ข้อมูลโทเค้นที่ใช้ในระบบ

การทดลอง API ในเว็บแอปพลิเคชัน ทรูมันนี่ เดเวลอปเปอร์ แชนด์บ็อกซ์จะมีเฮดเดอร์ ( headers ) ที่ประกอบด้วยการพิสูจน์ตัวตน ( authorization ) และโทเค้น ( token ) จากรูปที่ 3.3 ที่กำหนดแทนประเภทผู้ใช้ระบบ โดยมีโทเค้นตามประเภทผู้ใช้นี้

#### 1) ลูกค้ำของ ทรูมันนี่ วอลเล็ต (User)

- ลูกค้ำที่มีสิทธิ์การเข้าถึงข้อมูลครบถ้วน ( tok\_all\_scope\_complete\_info )
- ลูกค้ำที่มีสิทธิ์การเข้าถึงข้อมูลไม่ครบถ้วน ( tok\_all\_scope\_incomplete\_info )
- ลูกค้ำที่ไม่มีสิทธิ์การเข้าถึงข้อมูลที่จำเป็น ( tok\_no\_scope )
- ลูกค้ำที่มีโทเค้น ( token ) ที่หมดอายุ ( tok\_expired )

#### 2) ร้านค้ำ ( Merchant )

- ร้านค้ำที่มีสถานะพร้อมใช้งานครบถ้วน
- ร้านค้ำที่มีสถานะบางอย่างไม่พร้อมใช้งาน เช่น ร้านค้ำไม่พร้อมใช้งาน, สัญญาไม่พร้อมใช้งาน หรือ สัญญาหมดอายุ เป็นต้น

โดยแต่ละโทเค้นต้องสร้างข้อมูลบุคคล ( user persona ) ทั้งของลูกค้ำ ทรูมันนี่ วอลเล็ต และของร้านค้ำประเภทต่าง ๆ ขึ้นมาเพื่อใช้เป็นข้อมูลในการส่งค่าคำขอผ่าน API ( request body )

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.2 สร้างข้อมูลจำลองกิจกรรมต่าง ๆ ที่เกิดขึ้นจริง

กำหนดค่าที่ส่งไปกับคำขอผ่าน API ในกรณีทดสอบต่างๆ และกำหนดผลการตอบกลับกรณีทดสอบนั้น ๆ สำหรับทุกกิจกรรม หรือ API ที่นักพัฒนาของร้านค้าจะสามารถใช้ได้ โดยแบ่งกิจกรรมต่างๆ ออกเป็น 3 ส่วนหลักดังนี้

- 1) ข้อมูลลูกค้า TrueMoney ( User Controller )
- 2) การสั่งซื้อเพื่อขอช่องทางชำระเงิน ( Order Controller )
- 3) การแจ้งเตือนร้านค้า ( Merchant Notification )

ข้อมูล API จำลอง ( mock API ) ข้างต้นจะมีรายละเอียดแตกต่างกันที่กรณีทดสอบ และจำนวนผลการตอบกลับ ( response ) โดยส่วนที่มีรายละเอียดเหมือนกันมีดังนี้

- อุปกรณ์ปลายทาง ( TrueMoney Endpoint )
- การยืนยันตัวตน ( Authentication )
- ตัวอย่างการร้องขอข้อมูลของร้านค้า ( API Request )
- ตัวอย่างการตอบกลับจากทรูมันนี่ ( API Response )

### 3.1.3 สร้างข้อมูลกิจกรรมการแจ้งเตือนร้านค้า

สร้างการแจ้งเตือนผู้พัฒนาของร้านค้าว่าการทำกิจกรรมต่าง ๆ บนระบบทดสอบนั้น ถูกต้องหรือไม่ เช่น การสร้างรายการสั่งซื้อสินค้า การเพิ่มสาขาของร้านค้า เป็นต้น

### 3.1.4 สร้างคู่มือวิธีใช้งานเว็บแอปพลิเคชัน

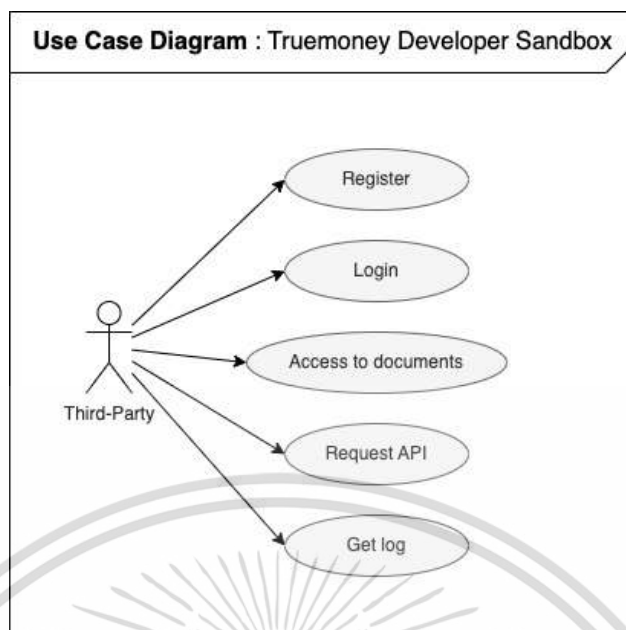
สร้างคู่มือการทดสอบ API จำลองบนเว็บแอปพลิเคชัน เอกสารการรับมือกับความผิดพลาดเบื้องต้น ( handling error ) วิธีติดตั้งและใช้งานฟังก์ชันเสริม ( SDK ) ในภาษา Java Python Node และ Ruby

## 3.4 ขั้นตอนการออกแบบและการดำเนินงาน

ออกแบบและดำเนินการสร้างระบบตาม requirement ที่ได้มาจากขั้นตอนที่ 3.4 และ 3.5 โดยแสดงขั้นตอนการทำงานของระบบในรูปของ diagram ได้ดังนี้

### 3.3.1 Use Case Diagram

แผนภาพแสดงบทบาทการทำงานของผู้ใช้ระบบทรูมันนี่ เดเวลอปเปอร์ แชนด์บ็อกซ์ผู้ที่เกี่ยวข้องกับระบบ รวมถึงความสัมพันธ์ของฟังก์ชันในระบบ ได้ดังรูป 3.6 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 Use Case Diagram ของเว็บแอปพลิเคชันทรูมันนี่ เดเวลอปเปอร์ แซนด์บ็อกซ์

จากรูปที่ 3.6 สามารถอธิบายกระบวนการทำงานได้ ดังนี้

- 1) Third-Party คือ ร้านค้าที่ต้องการเพิ่มทรูมันนี่ วอลเล็ตเป็นช่องทางการชำระเงิน
- 2) Register คือ การสร้างบัญชีผู้ใช้สำหรับผู้พัฒนาฝั่งร้านค้าที่ต้องการใช้เว็บแอปพลิเคชันทรูมันนี่ เดเวลอปเปอร์ แซนด์บ็อกซ์โดยลงทะเบียนด้วยอีเมล
- 3) Login คือ การเข้าสู่ระบบด้วยชื่อผู้ใช้และรหัสผ่านที่ตั้งไว้ตอนลงทะเบียน
- 4) Access to documents คือ การแสดงเอกสารการใช้งาน Open API
- 5) Request API คือ การเรียกใช้งาน Open API จำลอง
- 6) Get log คือ การแสดงประวัติการเรียกใช้ Open API จำลอง

### 3.3.2 Use Case Description

อธิบายการทำงานของเว็บแอปพลิเคชันเดเวลอปเปอร์ แซนด์บ็อกซ์ในแต่ละ Use Case โดยละเอียดจากรูปที่ 3.6 ข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 รายละเอียดของ Use Case Register

Use-Case Name :	Register	
Use-Case ID :	1	
Actor :	ผู้พัฒนาฝั่งร้านค้า	
Description :	การลงทะเบียนสร้างบัญชีผู้ใช้	
Precondition :	ผู้พัฒนาฝั่งร้านค้าต้องมี Email Account	
Postcondition :	ทำการลงทะเบียนเสร็จสมบูรณ์	
Typical Course of Events :	Actor Action	System Response
	Step 1: ผู้พัฒนาฝั่งร้านค้าทำการกรอกอีเมล ชื่อผู้ใช้ และรหัสผ่านเข้าสู่ระบบ	Step 2: ระบบทำการตรวจสอบความถูกต้อง และทำการบันทึกเข้าสู่ฐานข้อมูล user
Exception :	ถ้าตรวจสอบความถูกต้องของข้อมูลที่กรอกไม่ผ่าน จะไม่สามารถทำการลงทะเบียนได้ เช่น กรอกข้อมูลไม่ครบหรือไม่ถูกต้องตามเงื่อนไข	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 รายละเอียดของ Use Case Login

Use-Case Name :	Login	
Use-Case ID :	2	
Actor :	ผู้พัฒนาฝั่งร้านค้า	
Description :	ทำการเข้าสู่ระบบด้วย Username และ Password ที่ตั้งไว้ตอนลงทะเบียน	
Precondition :	ระบบจะต้องมีข้อมูลอยู่ในฐานข้อมูล user	
Postcondition :	ทำการเข้าสู่ระบบเสร็จสมบูรณ์	
Typical Course of Events :	Actor Action	System Response
	Step 1: ผู้พัฒนาฝั่งร้านค้าทำการกรอกชื่อผู้ใช้ และ รหัสผ่านเข้าสู่ระบบ	Step 2: ระบบทำการตรวจสอบความถูกต้อง และเข้าสู่หน้าแสดงเอกสารวิธีใช้งาน
Exception :	ถ้าตรวจสอบความถูกต้องของข้อมูลที่กรอกไม่ผ่าน จะไม่สามารถทำการเข้าสู่ระบบได้ เช่น กรอกข้อมูลไม่ครบ, ข้อมูลไม่ถูกต้องตามเงื่อนไข, ไม่มีข้อมูลผู้ใช้นั้นในฐานข้อมูล user จะไม่สามารถเข้าสู่หน้าแสดงเอกสารได้ เนื่องจากไม่มีข้อมูลอยู่ในระบบฐานข้อมูล	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 รายละเอียดของ Use Case Access to documents

Use-Case Name :	Access to documents	
Use-Case ID :	3	
Actor :	ผู้พัฒนาฝั่งร้านค้า	
Description :	แสดงเอกสารการทดลองใช้งาน Open API	
Precondition :	ผู้พัฒนาฝั่งร้านค้าจะต้องเข้าสู่ระบบก่อนเสมอ	
Postcondition :	แสดงเอกสารการทดลองใช้งาน Open API ที่เก็บในฐานข้อมูล web-page	
Typical Course of Events :	Actor Action	System Response
	Step 1: ผู้พัฒนาฝั่งร้านค้าเข้าสู่ระบบสำเร็จ Step 2: ทำการกดปุ่มข้อมูลที่ต้องการดูเพิ่มเติมบนหน้าเว็บ	Step 3: ระบบทำการแสดงเอกสารที่ถูกเก็บอยู่ในฐานข้อมูล web-page
Exception :	ถ้าไม่มีข้อมูลในระบบ จะแสดงรายละเอียดข้อมูลนั้นเป็นช่องว่าง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 รายละเอียดของ Use Case Request API

Use-Case Name :	Request API	
Use-Case ID :	4	
Actor :	ผู้พัฒนาฝั่งร้านค้า	
Description :	การเรียกใช้งาน Open API จำลอง	
Precondition :	มีการรันแอปอยู่ตลอดเวลา และต้องใส่ Endpoint, Authorization และ Request ตามที่กำหนด	
Postcondition :	แสดง Response และเก็บ Log เข้าฐานข้อมูล	
Typical Course of Events :	Actor Action	System Response
	Step 1: กดปุ่มเพื่อเข้าสู่หน้า Try Out API	Step 3: ระบบทำการรันแล้วส่งไปยัง Mock Server
	Step 2: ใส่ Endpoint, Authorization และ Request ตามที่กำหนดเพื่อทดลอง Open API Case ที่ต้องการ	Step 4: บันทึก Log เข้าฐานข้อมูล
Exception :	-	

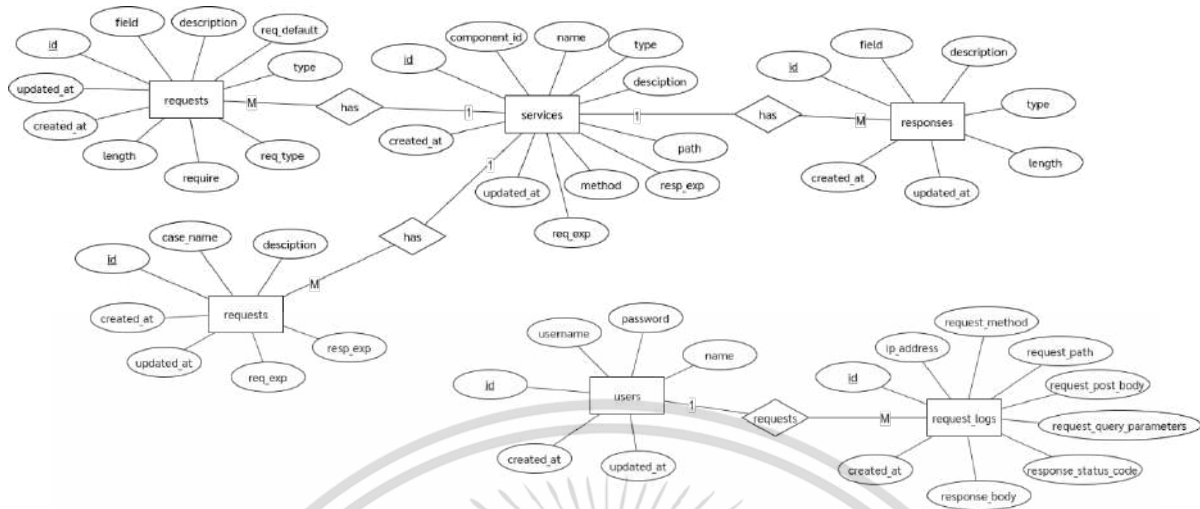
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 รายละเอียดของ Use Case Get log

Use-Case Name :	Get log	
Use-Case ID :	5	
Actor :	ผู้พัฒนาฝั่งร้านค้า	
Description :	แสดงประวัติการเรียกใช้ Open API จำลอง	
Precondition :	มีการรันแอปอยู่ตลอดเวลา	
Postcondition :	แสดงประวัติการเรียกใช้ Open API ในช่วงเวลานั้นๆ ได้ถูกต้อง	
Typical Course of Events :	Actor Action	System Response
	Step 1: แอปทำการเช็คใน ช่วงเวลาปัจจุบันว่ามีการเรียกใช้ Open API หรือไม่	Step 3: แอปทำการรันแล้ว ส่งไปยัง Web Server
	Step 2: แอปทำการรันเพื่อแสดง Log ที่อยู่ในฐานข้อมูลใน ช่วงเวลานั้นๆ	
Exception :	ถ้าแอปไม่ทำการรันไว้จะไม่มีประวัติ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.3 ER Diagram



รูปที่ 3.7 ER Diagram

### 3.3.4 Data Dictionary

พจนานุกรมข้อมูล แสดงรายละเอียดต่างๆ ในฐานข้อมูล หรือ Database จากรูปที่ 3.4 ข้างต้น

ตารางที่ 3.6 Data Dictionary : users จัดเก็บข้อมูลของผู้ใช้งาน

Attributes	Type	Description	Remark
id	integer	ID ของผู้ใช้งาน	Primary Key
username	varchar(50)	ชื่อเข้าใช้งาน	-
password	varchar(128)	รหัสผ่าน	-
name	varchar(50)	ชื่อร้านค้า	-
created_at	timestamp	วันและเวลาที่สร้าง	-
updated_at	timestamp	วันและเวลาที่แก้ไข	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.7 Data Dictionary : request\_logs จัดเก็บบันทึกการ request ไปที่ mock open API

Attributes	Type	Description	Remark
id	integer	ID ของการบันทึกแต่ละครั้ง	Primary Key
user_id	integer	ID ของผู้ใช้งาน	-
ip_address	varchar(25)	ที่อยู่ IP ของผู้ใช้งาน	-
request_method	http_method	http_method เป็น ENUM ได้แก่ GET, POST, PATCH, DELETE และ PUT	-
request_path	text	เป็นส่วนที่เราไว้ระบุการทำงาน เช่น /data, /data?r=2	-
request_post_body	json	Body ของ request	-
request_query_parameters	json	Query ของ request	-
response_status_code	integer	รหัสแสดงสถานะการตอบกลับ	-
response_body	json	Body ของการตอบกลับ	-
created_at	timestamp	วันและเวลาที่ส่งrequest	-

ตารางที่ 3.8 Data Dictionary : services จัดเก็บรายละเอียดของแต่ละ service

Attributes	Type	Description	Remark
id	integer	ID ของ service	Primary Key
component_id	varchar(50)	Id ของ component	-
name	varchar(50)	ชื่อของ service	-
type	varchar(10)	ชนิดของ service	-
description	Varchar	คำอธิบายของ service	-
path	varchar(255)	Path ของ service	-
method	varchar(10)	Method ของ service	-
req_exp	varchar	ตัวอย่างของการ request	-
resp_exp	varchar	ตัวอย่างของการ response	-
created_at	timestamp	วันและเวลาที่สร้าง	-
updated_at	timestamp	วันและเวลาที่แก้ไข	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเฉพาะการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.9 Data Dictionary : responses จัดเก็บรายละเอียดส่วนประกอบของการ response ของแต่ละ API

Attributes	Type	Description	Remark
id	integer	ID ของ response field	Primary Key
service_id	integer	ID ของ service	-
field	varchar(50)	ชื่อของ field	-
description	varchar	คำอธิบายของ field	-
type	varchar(10)	ชนิดของข้อมูล	-
length	integer	ขนาดและความยาวของข้อมูล	-
created_at	timestamp	วันและเวลาที่สร้าง	-
updated_at	timestamp	วันและเวลาที่แก้ไข	-

ตารางที่ 3.10 Data Dictionary : request จัดเก็บรายละเอียดส่วนประกอบของการ request ของแต่ละ API

Attributes	Type	Description	Remark
id	integer	ID ของ request field	Primary Key
service_id	integer	ID ของ service	-
field	varchar(50)	ชื่อของ field	-
description	Varchar	คำอธิบายของ field	-
req_default	varchar(50)	ค่าเริ่มต้น	-
type	varchar(10)	ชนิดของข้อมูล	-
req_type	varchar(10)	อยู่ในส่วนไหนของการ request Header หรือ body	-
require	boolean	จำเป็นต้องส่งในทุก request หรือไม่	-
length	integer	ขนาดและความยาวของข้อมูล	-
created_at	timestamp	วันและเวลาที่สร้าง	-
updated_at	timestamp	วันและเวลาที่แก้ไข	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.11 Data Dictionary : request จัดเก็บรายละเอียดกรณีทดสอบของแต่ละ API

Attributes	Type	Description	Remark
id	integer	ID ของแต่ละกรณีทดสอบ	Primary Key
service_id	integer	ID ของ service	-
case_name	varchar(50)	ชื่อของกรณีทดสอบ	-
description	varchar	คำอธิบายของกรณีทดสอบ	-
req_exp	varchar	ตัวอย่างการ request	-
resp_exp	varchar	ตัวอย่างการตอบกลับ	-
created_at	timestamp	วันและเวลาที่สร้าง	-
updated_at	timestamp	วันและเวลาที่แก้ไข	-

### 3.3.5 API Definition

ตารางที่ 3.12 API Definition ของ Web Server

API Name	Endpoint	Method	Description
Register	/auth/register	POST	สร้างบัญชีผู้ใช้
Login	/auth/login	POST	เข้าสู่ระบบ
Exchange token	/auth/token	POST	แลกเปลี่ยน Access Token และ Refresh Token
Get API document	/web/document/get/all	GET	ดึงข้อมูลเอกสารของ Open API จำลอง
Get testing document	/web/testing/get/all	GET	ดึงข้อมูลเอกสารการใช้งาน Open API จำลอง
Get request log	/web/log	GET	ดึงประวัติการร้องขอ Open API จำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.13 API Definition ของ Open API จำลอง

API Name	Endpoint	Method	Description
Get TrueMoney ID	/user/v2/tmn-id	GET	จำลองการส่ง TrueMoney ID ของผู้ใช้
Get balance	/user/v1/balance	GET	จำลองการส่งยอดเงินในบัญชี ของผู้ใช้
Get agent status	/user/v2/agent-status	GET	จำลองการส่งสถานะบัญชีของ ผู้ใช้
Get NDID status	/user/v2/ndid-status	GET	จำลองการส่งสถานะการยืนยัน ตัวตนดิจิทัล (National Digital ID) ของบัญชีผู้ใช้ ซึ่ง ผู้ใช้จะต้องได้รับการ ลงทะเบียน NDID ในระบบ TrueMoney ก่อน
Send notify	/user/v2/notify	POST	จำลองการส่งผลลัพธ์ของการ แจ้งเตือนไปยังผู้ใช้ ประเภท การส่งข้อความคือ SMS และ ข้อความที่ต้องการแจ้งเตือน ผู้ใช้
Get Thai ID card picture	/user/v2/thai-id-card-picture	GET	จำลองการส่งภาพบัตร ประชาชนของผู้ใช้ ซึ่งภาพบัตร ประชาชนของผู้ใช้จะมีการ เข้ารหัส base64
Create order	/payment/v1/order	POST	จำลองการสร้างคำสั่งซื้อ สินค้าของร้านค้า ซึ่งจะได้รับ ช่องทางการชำระเงิน ( Checkout URL ) ผ่าน TrueMoney Wallet กลับมา
Merchant notification	/trigger	POST	ทดลองการแจ้งเตือนไปยัง ร้านค้าเมื่อมีการเมื่อมีการทำ กิจกรรม ( ธุรกิจ ) ใดๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.14 Common Response

Tokens	HTTP Code	Error Code	Message	Description
tok_all_scope_complete_info	200	0	success	ส่ง request ถูกต้อง
ok_all_scope_incomplete_info	200	0	success	ส่ง request ถูกต้องส่ง request ถูกต้อง
tok_expired	401	OPA0001	Invalid Authorization	ไม่ใส่ Authorization ใน headers, ใส่ Authorization แต่ไม่ใส่ token, ใส่ Authorization แต่ token หมดอายุ
invalid_token	401	AUT0001	Invalid Token	Token ไม่ถูกต้อง
tok_no_scope	200	RES0001	Invalid Scope	Scope ไม่ถูกต้อง
tok_all_scope_complete_info	200	RES0002	Invalid Request	Request ไม่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.15 Merchant Notification Response

HTTP Code	Message	Description
200	"Signature verification Success"	ตรวจสอบ content-signature ได้สำเร็จ
	"Signature verification Fail"	ตรวจสอบ content-signature ไม่สำเร็จ เพราะ content-signature, timestamp, body หรือ private key ไม่ถูกต้อง
500	"Timestamp header is missing"	request ไม่ถูกต้องเพราะไม่พบ timestamp ใน headers
	"Unknown event type or invalid event type"	event type ใน request body ไม่ถูกต้องหรือ event type นั้นไม่มีอยู่ในระบบของ truemoney
	"Content-Signature header is missing",	request ไม่ถูกต้องเพราะไม่พบ content-signature ใน headers
	"Content-Signature and Timestamp header is missing",	request ไม่ถูกต้องเพราะไม่พบ timestamp และ content-signature ใน headers
	"Cannot map request body"	request ไม่ถูกต้องเพราะใส่ required fields ไม่ครบ
	"Timeout:Target not response"	ร้านค้าไม่ได้ตอบกลับมาในเวลาที่กำหนด
	"Target not found"	ไม่พบ target หรือ target ไม่มีอยู่จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## SDK

ชุดเครื่องมือพัฒนาซอฟต์แวร์ในแพ็คเกจเดียวที่สามารถติดตั้งได้ ช่วยอำนวยความสะดวกแก่ผู้พัฒนาร้านค้าในการใช้ระบบทราfficนี้เดเวลอปเปอร์แฮนด์บ็อกซ์โดย SDK ที่ทางคณะผู้จัดทำได้รับมอบหมายคือ การสร้างและยืนยันลักษณะเฉพาะของข้อมูลสำหรับฝั่งร้านค้าในภาษา Python และ Ruby แสดงตัวอย่างดังรูปที่ 3.8

```

1. from Crypto.PublicKey
2. import RSA from Crypto.Signature
3. import PKCS1_v1_5 from Crypto.Hash
4. import SHA256
5. import base64
6.
7. def createSignature(privateKey,message):
8.     keyBytes = RSA.importKey(open(privateKey, 'rb').read())
9.     h = SHA256.new(message.encode('utf-8'))
10.    signer = PKCS1_v1_5.new(keyBytes)
11.    signature = signer.sign(h)
12.    encoded = base64.b64encode(signature)
13.    return encoded
14.
15. def verifySignature(publicKey, message, signature):
16.    signature = signature[signature.find("data=")+len("data="):].strip()
17.    keyBytes = RSA.importKey(open(publicKey, 'rb').read())
18.    h = SHA256.new()
19.    signer = PKCS1_v1_5.new(keyBytes)
20.

```

รูปที่ 3.8 ตัวอย่างการสร้างและยืนยันลักษณะเฉพาะของข้อมูลในภาษา Python

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ผลการวิจัยและการอภิปรายผล

ในขั้นตอนการพัฒนาจะมีการทดสอบระบบแบบไม่ใช่เครื่องคอมพิวเตอร์ ( manual test ) คือ การตรวจสอบการส่งการร้องขอโดยใช้ Postman และการทดสอบระบบแบบใช้เครื่องคอมพิวเตอร์ ( automated test ) ที่เป็นระบบหลังบ้าน ( back office ) เพื่อส่งมอบซอฟต์แวร์ภายในบริษัท ซึ่ง API ทุกตัวที่สร้างขึ้น ( mock API ) จะมีกรณีทดสอบ ( test cases ) เป็นของตัวเอง โดยในบทนี้จะอธิบายวิธีการใช้งานระบบ ทูร์มินี่ เดเวลอปเปอร์ แชนด์บ็อกซ์ TrueMoney Developer Sandbox ผ่านเว็บแอปพลิเคชัน และรายงานผลการทดสอบ API ( testing report ) ตามลำดับ ดังนี้

- 4.1 วิธีการใช้งานระบบ ทูร์มินี่ เดเวลอปเปอร์ แชนด์บ็อกซ์ ผ่านเว็บแอปพลิเคชัน
- 4.2 ผลการทดสอบ API แบบ Manual Test
- 4.3 ผลการทดสอบ API แบบ Automated Test

#### 4.1 วิธีการใช้งานระบบ ทูร์มินี่ เดเวลอปเปอร์ แชนด์บ็อกซ์ ผ่านเว็บแอปพลิเคชัน

##### 4.1.1 ลงทะเบียนร้านค้า

นักพัฒนาฝั่งร้านค้าจะต้องลงทะเบียนร้านค้าเพื่อเข้าสู่ระบบด้วยชื่อ ( username ) และรหัสผ่าน ( password ) ดังรูปที่ 4.1 โดยข้อมูลที่ลงทะเบียนจะถูกเก็บอยู่ในฐานข้อมูลชื่อ user

รูปที่ 4.1 ลงทะเบียนร้านค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.1.2 เข้าสู่ระบบ

นักพัฒนาฝั่งร้านค้าจะเข้าสู่ระบบด้วยชื่อและรหัสผ่านที่ลงทะเบียนไว้ หากลงทะเบียนถูกต้อง จะแสดงว่า Login Success ดังรูปที่ 4.2

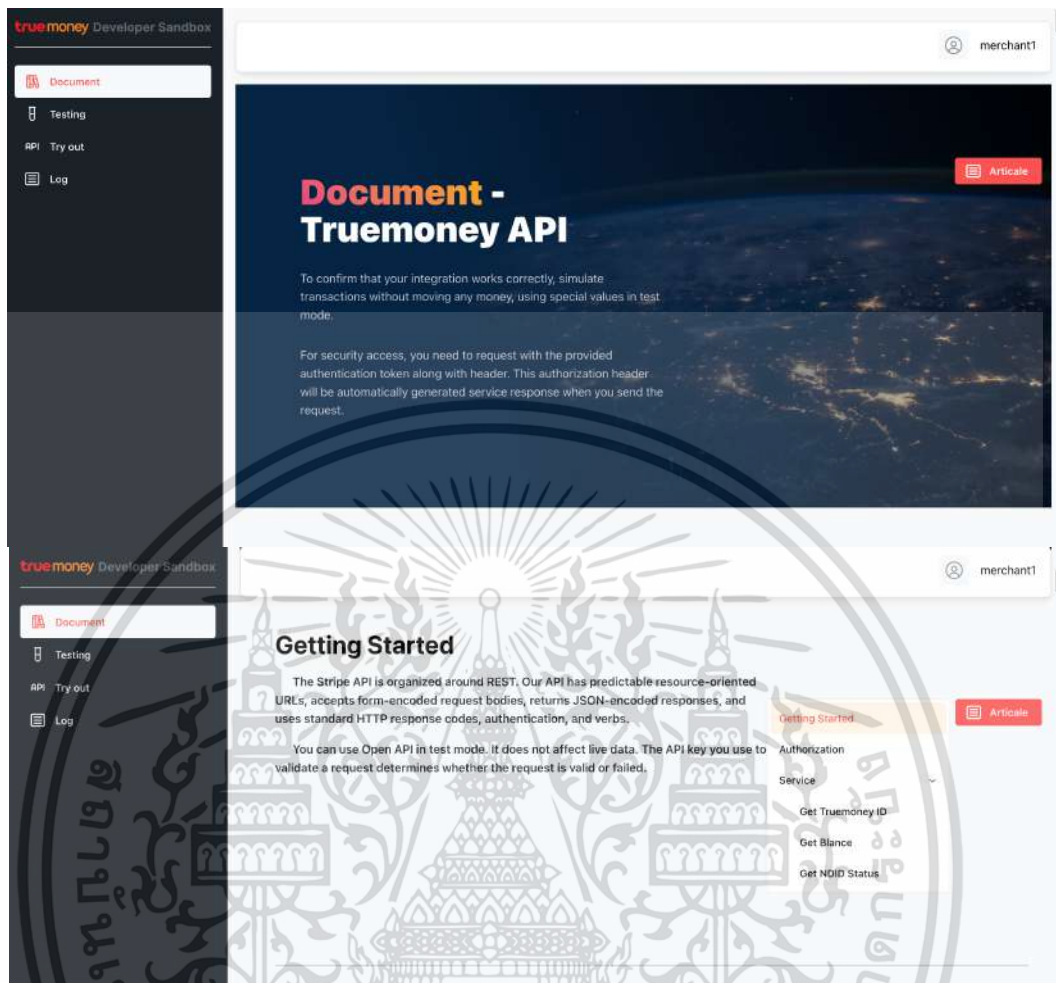


รูปที่ 4.2 เข้าสู่ระบบสำเร็จ

### 4.1.3 ศึกษาวิธีใช้ระบบและ API

เมื่อเข้าสู่ระบบแล้วจะเข้าสู่หน้าเว็บในส่วนอธิบายวิธีการใช้งานระบบทดสอบ ซึ่งจะแสดงข้อมูลสิทธิ์การเข้าถึงข้อมูลของผู้ใช้ ( authorization ) ผ่านตัวตนที่กำหนดไว้ ซึ่งเรียกว่า โทเคน ( token ) ที่แสดงถึงข้อมูลผู้ใช้ระบบประเภทต่าง ๆ ในหัวข้อ 3.3 และจะแสดงความหมายของแต่ละ API ( API description ) ที่ผู้พัฒนาฝั่งร้านค้าสามารถใช้ทดสอบได้บนเว็บแอปพลิเคชัน ทรูมันนี่ เดเวลอปเปอร์ แชนด์บ็อกซ์ ดังรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

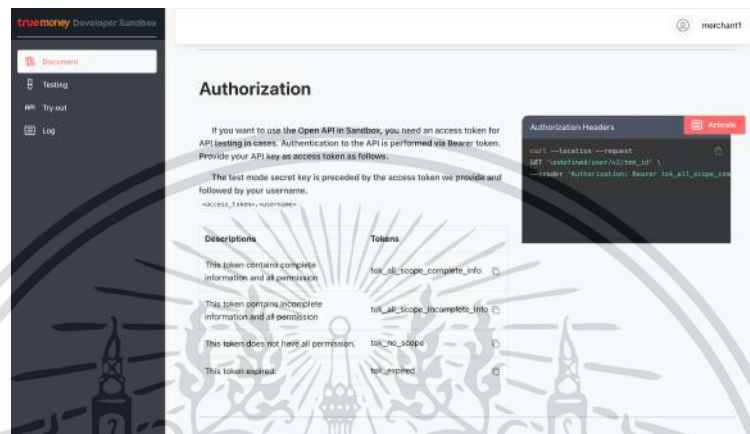


รูปที่ 4.3 หน้า Document แสดงวิธีใช้ TrueMoney Developer Sandbox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.4 การยืนยันตัวตนด้วย Token

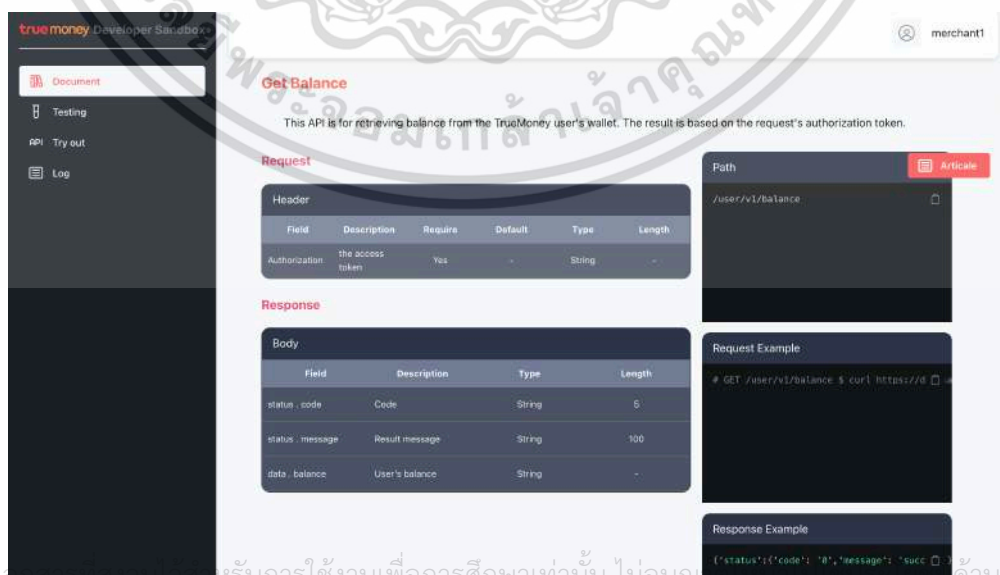
ในส่วนนี้จะอธิบายความหมายของแต่ละโทเคนที่สามารถใช้ได้พร้อมตัวอย่างที่สามารถคัดลอกจากปุ่มคัดลอกทางขวา ดังรูปที่ 4.4 และเนื่องจากข้อมูลส่วนใหญ่ถูกบันทึกในฐานข้อมูล หากมีการพัฒนาต่อจึงสามารถเปลี่ยนข้อมูลได้โดยไม่กระทบส่วนอื่น ๆ



รูปที่ 4.4 Authorization แสดง Token

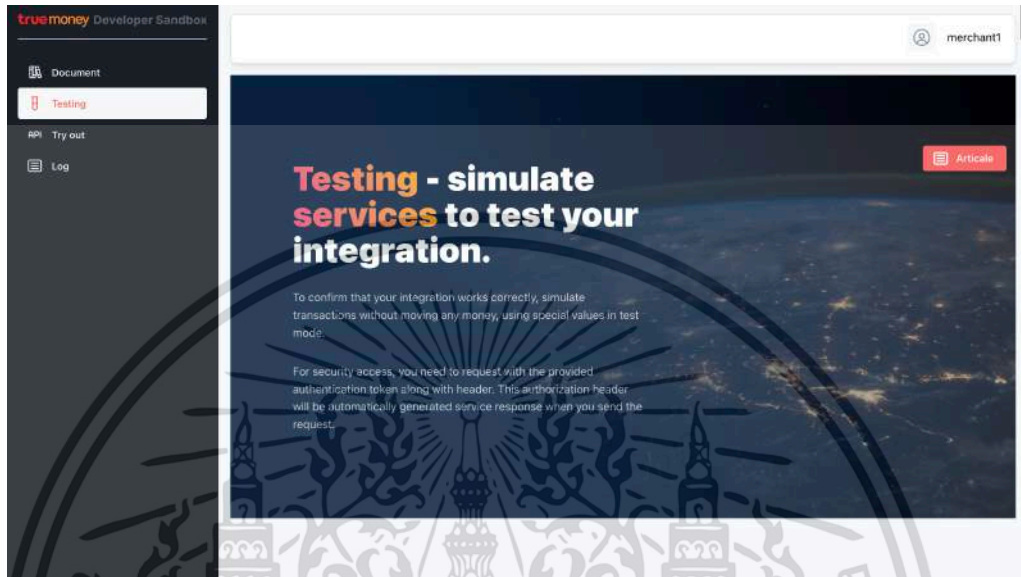
#### 4.1.5 เลือก API ที่ต้องการทดสอบ

ผู้พัฒนาฝั่งร้านค้าสามารถเลือก API ที่ต้องการทดสอบได้จากหน้า Document ดังรูปที่ 4.5 โดยเลือกกรณีที่จะทดสอบใน API นั้นๆ



เอกสารนี้เป็นเอกสารลิขสิทธิ์ของธนาคารเพื่อการศึกษานานาชาติ ไม่อนุญาตให้เผยแพร่หรือใช้ซ้ำโดยไม่ได้รับอนุญาต  
รูปที่ 4.5 API Definitions และ API References

จากนั้น ผู้พัฒนาฝั่งร้านค้าสามารถดูวิธีการทดสอบอย่างละเอียดได้ในหน้า Testing ดังรูปที่ 4.6 ซึ่งจะอธิบาย Error ที่สามารถเกิดขึ้นได้ ดังรูปที่ 4.7 และแสดงตัวอย่างผลลัพธ์จากการทดสอบ API ในกรณีทดสอบที่เลือก ดังรูปที่ 4.8 โดยสามารถคัดลอกจากตัวอย่าง เพื่อนำไปทดสอบได้

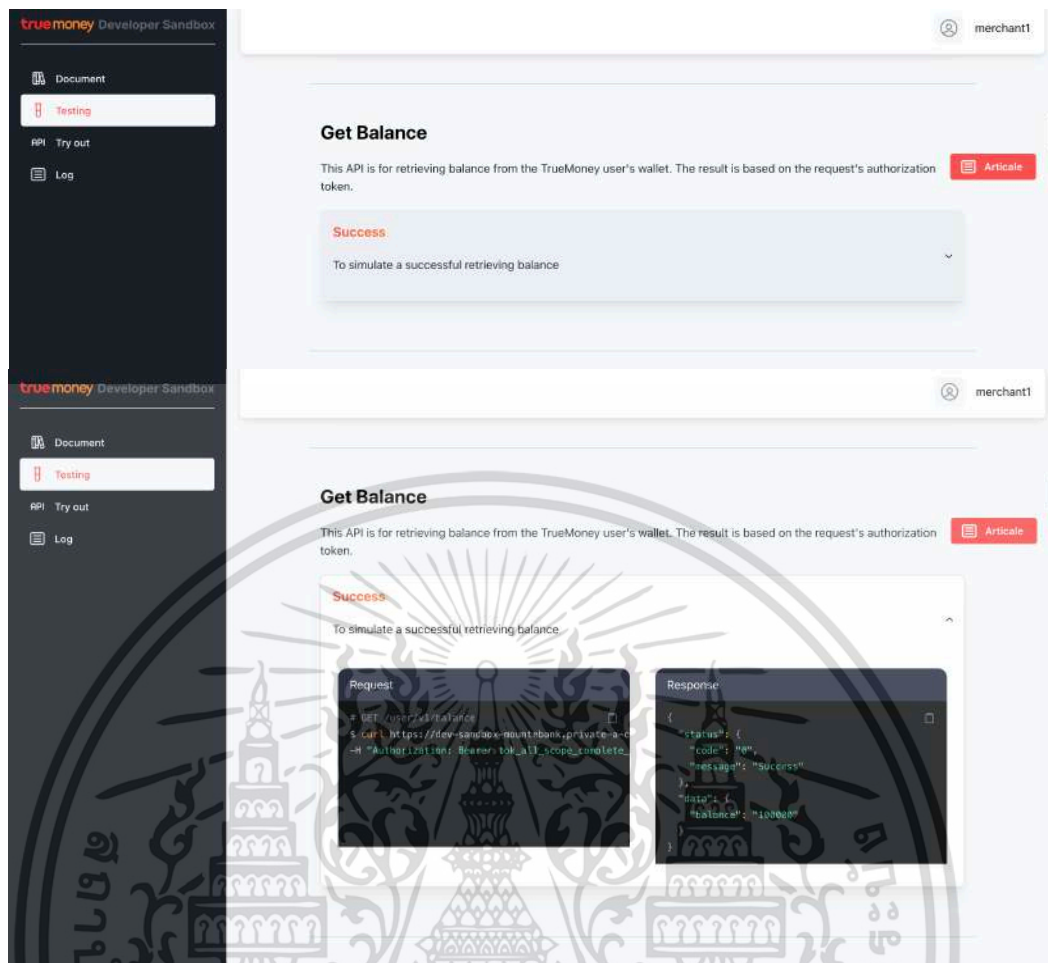


รูปที่ 4.6 หน้าทดลอง API ในแต่ละกรณีทดสอบ



รูปที่ 4.7 คำอธิบาย Response และ Error

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

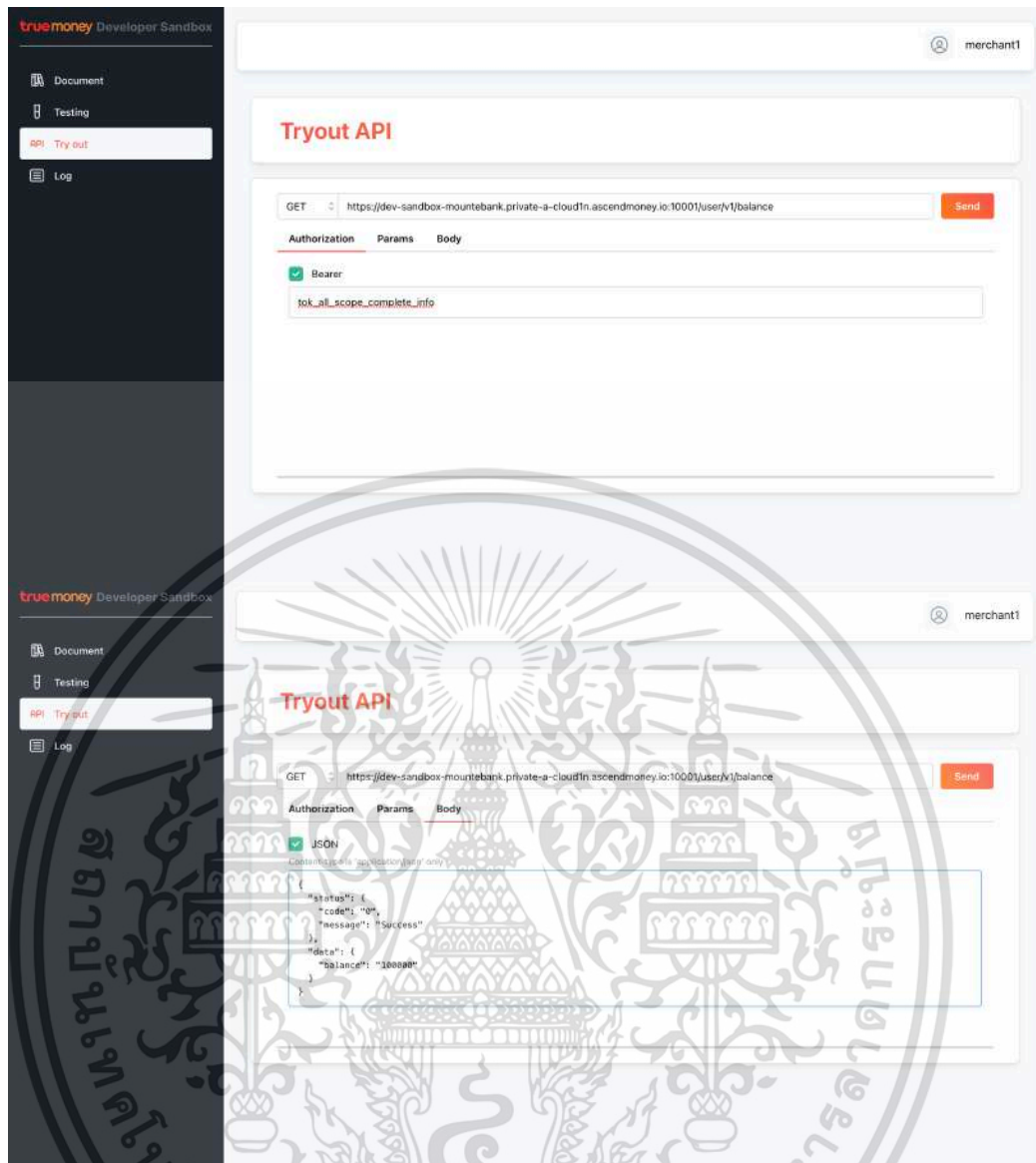


รูปที่ 4.8 รายละเอียดการทดลองส่ง Request Get Balance กรณีขอข้อมูลสำเร็จ

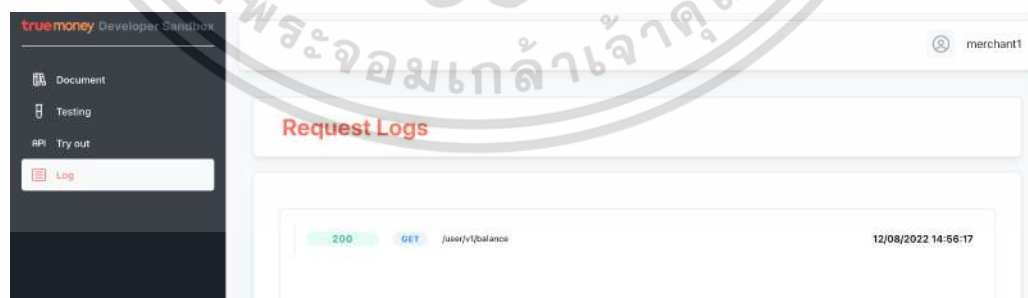
#### 4.1.6 ทดสอบ API ผ่านเว็บแอปพลิเคชัน

ตัวอย่างการส่ง API Get balance กรณีทดสอบการขอข้อมูลเงินคงเหลือในบัญชีทรูมันนี่ วอลเล็ต ( TrueMoney Wallet ) ของลูกค้าได้สำเร็จ ที่หน้า Tryout API ดังรูปที่ 4.9 ขั้นตอนแรก ระบุการร้องขอ ( method ) จุดหมายของ API ( path ) และโทเคน ( token ) ตามรายละเอียดของ API กรณีทดสอบ ( test case ) นั้นๆ หากเลือกการร้องขอเป็น POST จะต้องระบุข้อมูลที่ต้องการเพิ่มใน body ก่อนส่งการร้องขอ ซึ่งจะกำหนดให้ส่งแบบ JSON และ content-type เป็น application/json เท่านั้นตามระบบของบริษัท โดยผลลัพธ์ของแต่ละ API จะถูกกำหนดไว้ล่วงหน้าเพื่อให้ผู้พัฒนาฝั่งร้านค้าเห็นภาพการใช้ได้สะดวกเร็วที่สุด เช่น ถ้าจุดหมายเป็น user/v1/balance ก็จะได้ผลลัพธ์เป็นสถานะการขอข้อมูลในกรณีสำเร็จและแสดงข้อมูลเงินคงเหลือจำลองของลูกค้าที่กำหนดไว้ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 ทดลองส่ง request ขอข้อมูลเงินคงเหลือในบัญชีลูกค้า

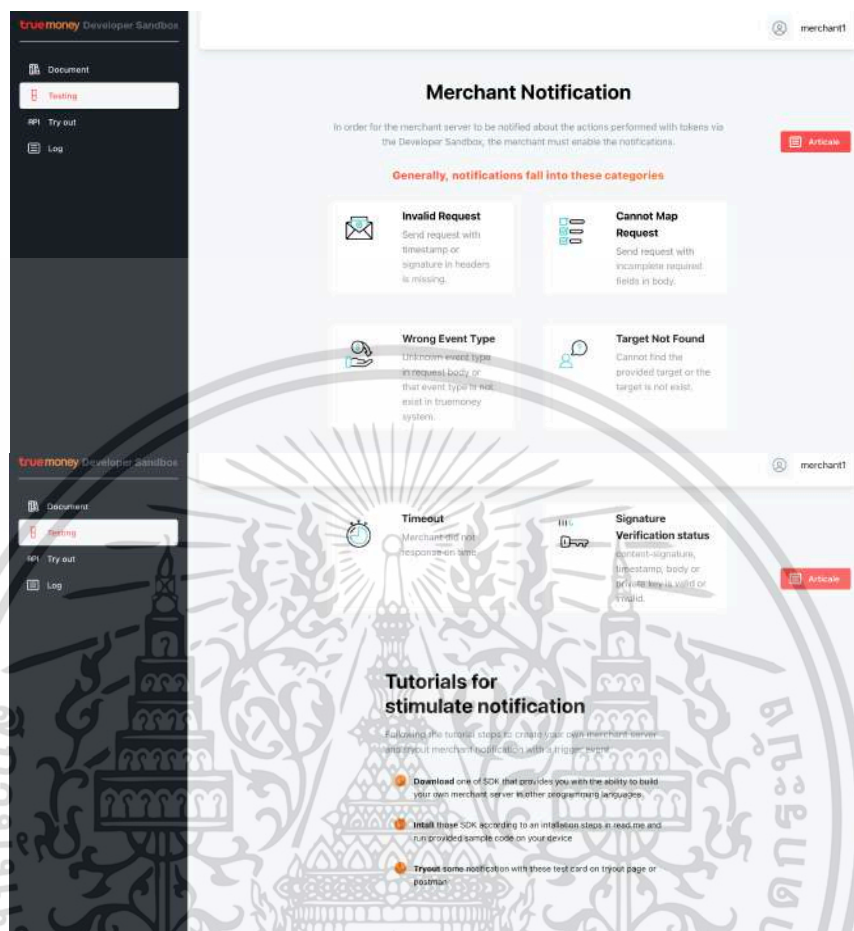


รูปที่ 4.10 แสดง log ผ่านหน้าเว็บแอปพลิเคชัน

จากรูปที่ 4.10 สามารถดูสถานะการขอข้อมูลได้ในหน้า Request Logs โดยจะแสดง HTTP code เป็น 200 แสดงประเภทการร้องขอ (method) เป็น GET แสดงจุดหมาย (path) เป็น user/v1/balance แสดงวัน และเวลาที่ส่งการร้องขอ ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่อนุญาตให้ท่านไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

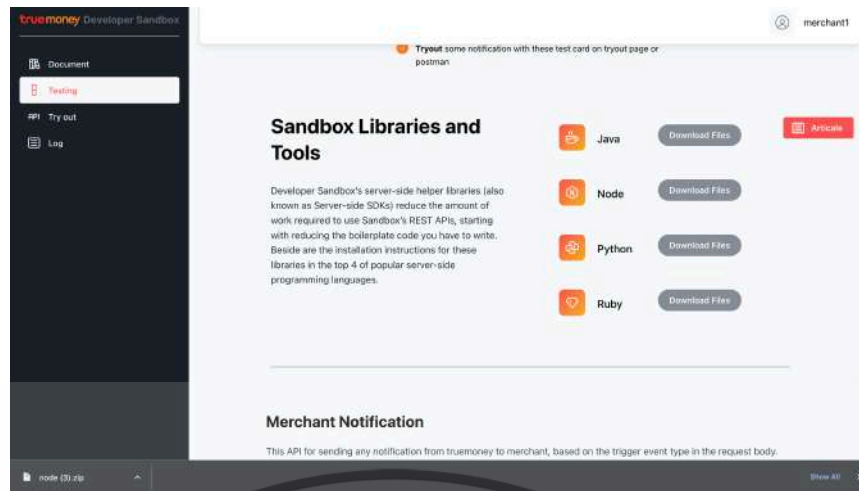
#### 4.1.7 ตัวอย่าง API อื่นๆ



รูปที่ 4.11 คำอธิบายและวิธีการทดลอง Merchant Notification

จากรูปที่ 4.11 แสดงวิธีการใช้ API การแจ้งเตือนไปยังร้านค้า ( merchant notification API ) เมื่อมีการอัปเดตข้อมูล หรือแจ้งเตือนการทำธุรกรรมต่าง ๆ รวมถึงการแจ้งเตือนเหตุขัดข้อง ( downtime ) โดยจะมีกรณีทดสอบที่ไม่สำเร็จ 6 กรณีดังรูป เนื่องจาก API นี้เป็น API ที่ทำการแจ้งเตือนไปยังร้านค้า ผู้พัฒนาฝั่งร้านค้าจึงจำเป็นต้องมีการสร้างเซิร์ฟเวอร์ของตนเอง โดยจะมีการเตรียม SDK และโค้ดต่าง ๆ ไว้ให้ใช้ทั้งหมด 4 ภาษา ซึ่งสามารถดาวน์โหลดผ่านเว็บแอปพลิเคชันได้ ดังรูปที่ 4.12 และสามารถทดสอบ API นี้ในกรณีทดสอบต่าง ๆ ตามรายละเอียด ดังรูปที่ 4.13 ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 วิธีการใช้และปุ่มดาวน์โหลด SDK



รูปที่ 4.13 วิธีทดลอง Merchant Notification ในกรณีสำเร็จ

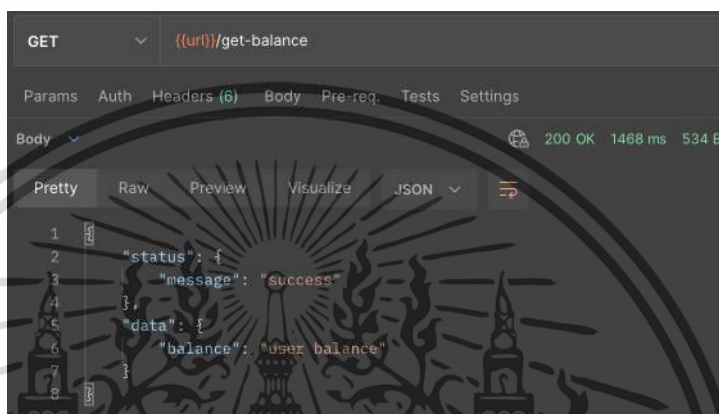
## 4.2 ผลการทดสอบ API แบบ Manual Test

การทดสอบระบบแบบไม่ใช้เครื่องคอมพิวเตอร์ ( manual test ) คือ การตรวจสอบการส่งการร้องขอโดยใช้ Postman โดยจะทำในระหว่างการพัฒนาเพื่อตรวจสอบผลลัพธ์เบื้องต้นก่อนเพื่อวางแผนและประเมินความคุ้มค่าในการทำ Automated Test ก่อนการทดสอบจึงต้องมีการศึกษาข้อมูลของระบบที่พัฒนาอยู่และศึกษาผลิตภัณฑ์ของบริษัทที่สามารถนำมาใช้เป็นตัวอย่างได้อย่างละเอียด เพื่อใช้ในการออกแบบกรณีทดสอบ ( test case ) เขียนขั้นตอนการทดสอบระบบ ( test script ) จากนั้นจึงเริ่มการทดสอบตามขั้นตอนที่วางแผนไว้ แม้ว่าผลจากการทำ Manual Test จะเอกสารนี้เป็นเอกสารที่ส่งงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นองถาดหนาไปใช้ประโยชน์ด้านการค้า ถูกต้อง แต่ไม่ได้รับรองว่าจะระบบจะทำงานถูกต้อง 100% เพราะเป็นการตรวจสอบด้วยการไม่ว่าการันใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุตบแต่งสเงนอเหตุ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คาดการณ์การทำงานของระบบโดยผู้พัฒนาเอง ซึ่งอาจไม่ครอบคลุมทุกกรณีที่จะเกิดขึ้น จึงต้องมีการทำ Automated Test ในลำดับต่อไป แสดงตัวอย่างการทดสอบ API ผ่าน Postman ดังนี้

### Get Balance

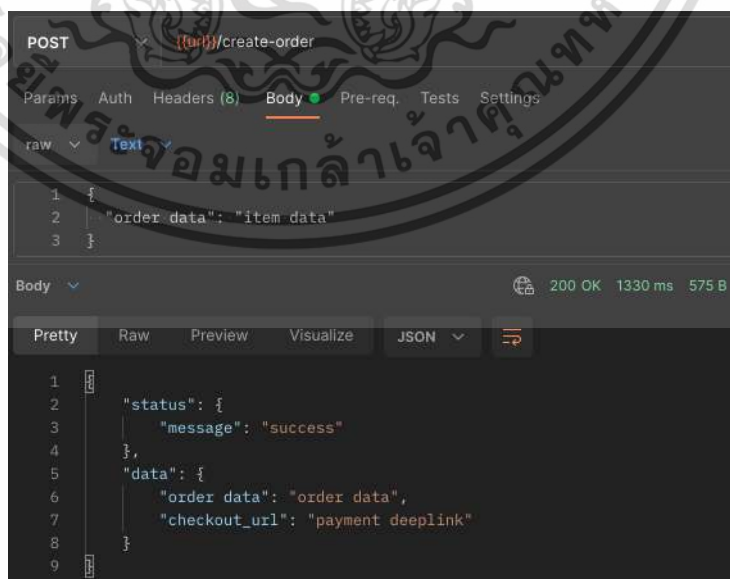
ทดสอบการส่งยอดเงินในบัญชีของผู้ใช้ ซึ่งได้ผลลัพธ์เป็นสถานะการส่งคำร้องขอที่สำเร็จและข้อมูลเงินในบัญชีของผู้ใช้ที่กำหนดไว้ ยกตัวอย่างเช่น “balance”: 1000000 เป็นต้น ดังรูป 4.14



รูปที่ 4.14 Get Balance

### Create Order

ทดสอบการสร้างคำสั่งซื้อสินค้าของร้านค้า ร้านค้าจะส่งรายละเอียดคำสั่งซื้อผ่านการร้องขอแบบ POST และจะได้ผลลัพธ์เป็นข้อมูลสินค้าและช่องทางการชำระเงิน ( Checkout URL ) ผ่านทวูมันนี่ วอลเล็ต ( TrueMoney Wallet ) กลับมา ดังรูป 4.15



รูปที่ 4.15 Create Order

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Merchant Notification

ทดสอบการแจ้งเตือนไปยังร้านค้าเมื่อมีการเมื่อมีการทำกิจกรรม หรือธุรกรรมใด ๆ โดยต้องส่งร้านค้าที่เป็นจุดหมาย ประเภทการแจ้งเตือน และกำหนดให้การยืนยันตัวตนร้านค้าต้องถูกต้องเท่านั้นจึงจะสำเร็จ และส่งผลลัพธ์เป็นข้อความกลับมา ดังรูป 4.16

```

POST {{url}}/merchant-noti

Params Auth Headers (8) Body Pre-req. Tests Settings
raw Text
1 {
2   "target": "https://test/merchant/success",
3   "eventType": "MERCHANT_UPDATE",
4   "sig_validate": true
5 }

Body 200 OK 1358 ms 491 B
Pretty Raw Preview Visualize HTML
1
2 "message": "Signature verification: success"
3

```

รูปที่ 4.16 Merchant Notification

## 4.3 ผลการทดสอบ API แบบ Automated Test

เขียนโปรแกรมทดสอบระบบอัตโนมัติ ( automated test ) ใช้คำสั่งประมวลผลผ่าน Terminal ที่สามารถดูผลการทดสอบได้โดยการเปิด Log.html บน Browser ซึ่งจะแสดงผลการทดสอบกรณีทดสอบ ( Testcases ) และแสดงรายละเอียดทุกขั้นตอนในการทดสอบ ดังรูปที่ 4.17 และ 4.18 ที่แสดงรายงานการทดสอบการทำงานของ API ที่ใช้ในระบบ ( Test Report ) ทั้งหมดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**SUITE Testcases**  
 Full Name: Testcases  
 Source: /home/jenkins/workspace/ROBOT-QA-acm-sandbox-service/robot/acm-dev-sandbox-robot/testcases  
 Start / End / Elapsed: 20220831 09:42:26.570 / 20220831 09:42:45.833 / 00:00:19.263  
 Status: 118 tests total, 118 passed, 0 failed, 0 skipped

**SUITE Payment**  
 Full Name: Testcases.Payment  
 Source: /home/jenkins/workspace/ROBOT-QA-acm-sandbox-service/robot/acm-dev-sandbox-robot/testcases/payment  
 Start / End / Elapsed: 20220831 09:42:26.648 / 20220831 09:42:36.156 / 00:00:09.508  
 Status: 43 tests total, 43 passed, 0 failed, 0 skipped

**SUITE Create Order**

**SUITE Get Payment Code**

**SUITE User**  
 Full Name: Testcases.User  
 Source: /home/jenkins/workspace/ROBOT-QA-acm-sandbox-service/robot/acm-dev-sandbox-robot/testcases/user  
 Start / End / Elapsed: 20220831 09:42:36.158 / 20220831 09:42:45.830 / 00:00:09.672  
 Status: 75 tests total, 75 passed, 0 failed, 0 skipped

**SUITE Get Agent Status**

**SUITE Get Balance**

**SUITE Get Consent**

**SUITE Get Ndid Status**

**SUITE Get Profile**

**SUITE Get Thai Id Card Picture**

**SUITE Get Tmn Id**

**SUITE Save Consent**

**SUITE Send Notify**

### รูปที่ 4.17 Testcases

จากรูปที่ 4.17 แบ่ง testcases เป็น 2 ประเภทคือ testcases ของ API ที่เกี่ยวกับการชำระเงิน ( Payment ) และ testcases ของ API ที่เกี่ยวกับข้อมูลผู้ใช้ ( User ) โดยจะแบ่งกรณีทดสอบเป็นกรณีที่ส่งคำร้องขอสำเร็จ ( success case ) และกรณีที่ส่งคำร้องไม่สำเร็จ ( fail case ) จากสาเหตุต่าง ๆ ก็จะมีข้อความตอบกลับที่แตกต่างกันตามการทำงานของแต่ละ API

API ที่เกี่ยวกับการชำระเงิน ( Payment ) มีชื่อ testcases ดังนี้

#### Create Order

- Success - Create Order - Single Item
- Success - Create Order - Multiple Item
- Success - Create Order - Only Require Field Mandatory
- Success - Create Order - Language Is Empty
- Success - Create Order - Partner Merchant ID, Partner Terminal ID, Partner Shop ID Null And Empty

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Success - Create Order - Language Is TH, EN, MY, KM
- Success - Create Order - Terminal Type is "ApplicationNonWeb"
- Success - Create Order - Terminal Type is "ApplicationWeb"
- Success - Create Order - Terminal Type is "Website"
- Success - Create Order - Terminal Type Is Null Or Empty
- Success - Create Order - Terminal Type Is "invalidTerminalType"
- Success - Create Order - Merchant Information have only merchant\_id
- Fail - Create Order - Order Invalid - Invalid Request
- Fail - Create Order - ISV Is Empty And No ISV Parameter - Invalid Request
- Fail - Create Order - Order Display Invalid - Invalid Request
- Fail - Create Order - Merchant Information least one (shop\_id, terminal\_id) - Invalid Request
- Fail - Create Order - Invalid Language - Invalid Request
- Fail - Create Order - Feild Mandatory Order Items - Invalid Request
- Fail - Create Order - Invalid Content Type Json - Invalid Request
- Fail - Create Order - Language Is Lowercase - Invalid Request
- Fail - Create Order - Product Code Is Invalid - Cannot Process Request & Invalid Request
- Fail - Create Order - Merchant Ineffective - Cannot Process Request
- Fail - Create Order - Contract Merchant Invalid - Cannot Process Request
- Fail - Create Order - Amount Invalid - Cannot Process Request
- Fail - Create Order - Expired Contract - Cannot Process Request
- Fail - Create Order - Value Missing - Cannot Process Request
- Fail - Create Order - Shop Not Active - Cannot Process Request
- Fail - Create Order - Shop Invalid - Cannot Process Request
- Fail - Create Order - Terminal Not Active - Cannot Process Request
- Fail - Create Order - Terminal Invalid - Cannot Process Request
- Fail - Create Order - Invalid Scope
- Fail - Create Order - Invalid Authorization
- Fail - Create Order - Invalid Token

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

API ที่เกี่ยวกับข้อมูลผู้ใช้ ( User ) มีชื่อ testcases ดังนี้

### Get Agent Status

- Success - Get Agent Status - All Statuses
- Fail - Get Agent Status - Invalid Authorization
- Fail - Get Agent Status - Invalid Token
- Fail - Get Agent Status - Invalid Scope

### Get Balance

- Success - Get Balance
- Fail - Get Balance - Invalid Authorization
- Fail - Get Balance - Invalid Token
- Fail - Get Balance - Invalid Scope

### Get Consent

- Success - Get Consent - With ACCEPT Status
- Success - Get Consent - With NOT\_ACCEPT Status
- Success - Get Consent - With REVOKE Status
- Success - Get Consent - With NOT\_FOUND Status
- Success - Get Consent - With Only Number
- Fail - Get Consent - Invalid Request With Number Is Empty Or Not Send Integer
- Fail - Get Consent - Invalid Request With Number Or Version Is Not Integer
- Fail - Get Consent - Agreement Not Found When Number Not Found
- Fail - Get Consent - Agreement Not Found When Version Not Found
- Fail - Get Consent - Invalid Scope
- Fail - Get Consent - Invalid Authorization
- Fail - Get Consent - Invalid token

### Get Ndid Status

- Success - Get NDID status API
- Fail - Get verify NDID enrollment status with ndid user not enrolled
- Fail - Get NDID status API with authorization invalid
- Fail - Get NDID status API with invalid Token

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Fail - Get NDID status API with invalid Scope

### Get Profile

- Get Profile Success - tok\_all\_scope\_complete\_info
- Get Profile Success - tok\_all\_scope\_incomplete\_info
- Get Profile Success - tok\_no\_scope
- Get Profile Success - tok\_all\_scope\_complete\_info With Valid Scope
- Get Profile Success - tok\_all\_scope\_complete\_info With Invalid Scope
- Get Profile Success - tok\_all\_scope\_incomplete\_info With Valid Scope
- Get Profile Success - tok\_no\_scope With Valid Scope
- Get Profile Success - tok\_all\_scope\_complete\_info With Double Valid Scope
- Get Profile Success - tok\_no\_scope With Double Invalid Scope
- Get Profile Fail - Invalid Authorization When Bearer Does Not Have Bearer token
- Get Profile Fail - Invalid Authorization When Without Authorization
- Get Profile Fail - Invalid Authorization When Empty Authorization
- Get Profile Fail - Invalid Authorization When Token Is Expired
- Get Profile Fail - Invalid Authorization When The Auth Header Is Not Bearer
- Get Profile Fail - Invalid Authorization When Bearer's Token Is Invalid
- Get Profile Fail - tok\_no\_scope With Invalid Scope
- Get Profile Fail - Invalid Scope With Double Scope

### Get Thai Id Card Picture

- Success - Get Thai ID Card Picture
- Fail - Get Thai ID Card Picture - Image Not Found
- Fail - Get Thai ID Card Picture - Invalid Scope
- Fail - Get Thai ID Card Picture - Invalid Authorization
- Fail - Get Thai ID Card Picture - Token Is Invalid

### Get Tmn Id

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของบริษัทฯ โปรดแจ้งเกี่ยวกับการใช้เอกสารนี้ ไม่เผยแพร่โดยไม่ได้รับอนุญาต  
 เอกสารนี้ - Get tmn\_id Success - Success With A tok\_all\_scope\_complete\_info  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Get tmn\_id Success - Success With A tok\_all\_scope\_incomplete\_info Token
- Get tmn\_id Fail - Invalid scope With A tok\_no\_scope Token
- Get tmn\_id Fail - Invalid Authorization When Bearer Does Not Have Bearer token
- Get tmn\_id Fail - Invalid Authorization When Token Is Expired
- Get tmn\_id Fail - Invalid Authorization When Without Authorization
- Get tmn\_id Fail - Invalid Authorization When The Auth Header Is Not Bearer
- Get tmn\_id Fail - Invalid Authorization When Bearer's Token is Invalid

### Save Consent

- Success - Save Consent - Consent Status Is ACCEPT
- Success - Save Consent - Consent Status Is NOT\_ACCEPT
- Success - Save Consent - Consent Status Is Lowercase
- Fail - Save Consent - Consent Status Is Not ACCEPT Or NOT\_ACCEPT
- Fail - Save Consent - Agreement ID Is EMPTY
- Fail - Save Consent - Consent Status Is EMPTY
- Fail - Save Consent - Agreement Not Found
- Fail - Save Consent - Invalid Scope
- Fail - Save Consent - Authorization Invalid
- Fail - Save Consent - Token Is Invalid

### Send Notify

- Success - Post Send Notify API
- Fail - Post send notify with invalid mobile number
- Fail - Post send notify with type is empty
- Fail - Post send notify with type isn't SMS
- Fail - Post send notify with userNotification is empty
- Fail - Post send notify with userNotification is over 200 digit
- Fail - Post send notify with userNotification is invalid format
- Fail - Post Send Notify API with authorization invalid

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการดำเนินงานด้านการศึกษา  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Fail - Post Send Notify API with invalid Scope

จากการทำ testcases ทั้งหมดที่กล่าวข้างต้น แสดงให้เห็นว่าแม้แต่ละ tesecases จะมีจำนวนครั้งในการทดสอบมากน้อยแตกต่างกัน แต่ผลการทดสอบจะมีสถานะผ่าน ( Pass ) คือทดสอบการทำงานตามเงื่อนไขได้สำเร็จ แสดงดังรูปที่ 4.18

Test Statistics						
Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	118	118	0	0	00:00:17	<div style="width:100%;"></div>
Statistics by Tag						
Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
fail	78	78	0	0	00:00:11	<div style="width:100%;"></div>
payment	43	43	0	0	00:00:09	<div style="width:100%;"></div>
ready	118	118	0	0	00:00:17	<div style="width:100%;"></div>
regression	118	118	0	0	00:00:17	<div style="width:100%;"></div>
success	40	40	0	0	00:00:05	<div style="width:100%;"></div>
user	75	75	0	0	00:00:08	<div style="width:100%;"></div>
Statistics by Suite						
Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Testcases	118	118	0	0	00:00:19	<div style="width:100%;"></div>
Testcases . Payment	43	43	0	0	00:00:10	<div style="width:100%;"></div>
Testcases . Payment . Create Order	33	33	0	0	00:00:08	<div style="width:100%;"></div>
Testcases . Payment . Get Payment Code	10	10	0	0	00:00:02	<div style="width:100%;"></div>
Testcases . User	75	75	0	0	00:00:10	<div style="width:100%;"></div>
Testcases . User . Get Agent Status	4	4	0	0	00:00:01	<div style="width:100%;"></div>
Testcases . User . Get Balance	4	4	0	0	00:00:01	<div style="width:100%;"></div>
Testcases . User . Get Consent	12	12	0	0	00:00:02	<div style="width:100%;"></div>
Testcases . User . Get Ndid Status	5	5	0	0	00:00:01	<div style="width:100%;"></div>
Testcases . User . Get Profile	17	17	0	0	00:00:02	<div style="width:100%;"></div>
Testcases . User . Get Thai Id Card Picture	5	5	0	0	00:00:01	<div style="width:100%;"></div>
Testcases . User . Get Tmn Id	8	8	0	0	00:00:01	<div style="width:100%;"></div>
Testcases . User . Save Consent	10	10	0	0	00:00:01	<div style="width:100%;"></div>
Testcases . User . Send Notify	10	10	0	0	00:00:01	<div style="width:100%;"></div>

รูปที่ 4.18 ผลการทดสอบ ( Test Report )

Testcases ทั้งหมดจะต้องผ่านการวางแผนขั้นตอนการทดสอบ ( test step ) ก่อนที่จะทำการทดสอบจริงเสมอ เพื่อช่วยในการพัฒนาและตรวจสอบการทำงานของฟังก์ชันนั้น ๆ ได้อย่างถูกต้องครบถ้วน จึงมีการทำตารางแจกแจงรายละเอียดขั้นตอนของแต่ละ testcases ซึ่งมีจำนวนมาก ในที่นี้จึงยกตัวอย่างรายละเอียดกรณีทดสอบการขอจำนวนเงินคงเหลือในบัญชีของลูกค้า ( Get Balance ) ดังตารางที่ 4.1 เพียง 1 ตัวอย่างเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 รายละเอียดกรณีทดสอบ Get Balance

Function	Get Balance		
Test case name	Success - Get Balance		
Test case	Test Step	Expect Result	Status
To get balance success	<ol style="list-style-type: none"> <li>1. Prepare request with Authorization and Content-Type  <code>{token_all_scope_complete_info }</code></li> <li>2. When Call Get Balance API headers = {'Authorization': 'Bearer tok_all_scope_complete_info', 'Content-Type': 'application/json'}</li> <li>3. HTTP Status Should Be 200</li> <li>4. Response Body Status Should Be 0, success</li> <li>5. Verify Response Get Balance API Success</li> </ol>	status: 200, code: 0, message: success	PASS
Test case name	Fail - Get Balance - Invalid Authorization		
Test case	Test Step	Expect Result	Status
To get balance fail with invalid authorization	<ol style="list-style-type: none"> <li>1. Prepare request without Authorization and Content-Type</li> <li>2. When Call Get Balance API</li> <li>3. HTTP Status Should Be 401</li> <li>4. Response Body Status Should Be C2BG4001, Error from open API</li> </ol>	status: 401, code: C2BG4001, message: Error from	PASS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่ควรเผยแพร่ให้บุคคลภายนอกโดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่เอกสารนี้ไปยังบุคคลอื่นโดยไม่ได้รับอนุญาตของเอกสารทุกครั้งที่มีการนำไปใช้

	<p>gateway Unauthorized: Invalid access token or access token expired</p> <p>5. Prepare request with Authorization and Content-Type  <code>#{token_invalid_auth.empty_auth}</code></p> <p>6. When Call Get Balance API</p> <p>7. HTTP Status Should Be 401</p> <p>8. Response Body Status Should Be C2BG4001, Error from open API gateway Unauthorized: Invalid access token or access token expired</p> <p>9. Prepare request with Authorization and Content-Type  <code>#{token_invalid_auth.empty_bearer}</code></p> <p>10. When Call Get Balance API</p> <p>11. HTTP Status Should Be 401</p> <p>12. Response Body Status Should Be C2BG4001, Error from open API gateway Unauthorized: Invalid access token or access token expired</p> <p>13. Prepare request with Authorization and Content-Type  <code>#{token_invalid_auth.invalid}</code></p> <p>14. When Call Get Balance API</p> <p>15. HTTP Status Should Be 401</p>	<p>open API gateway Unauthorized : Invalid access token or access token expired</p>	
--	---	---	--

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	<p>16. Response Body Status Should Be C2BG4001, Error from open API gateway Unauthorized: Invalid access token or access token expired</p> <p>17. Prepare request with Authorization and Content-Type <code>\${token_invalid_auth.expired}</code></p> <p>18. When Call Get Balance API</p> <p>19. HTTP Status Should Be 401</p> <p>20. Response Body Status Should Be C2BG4001, Error from open API gateway Unauthorized: Invalid access token or access token expired</p>		
<b>Test case name</b>	Fail - Get Balance - Invalid Token		
<b>Test case</b>	<b>Test Step</b>	<b>Expect Result</b>	<b>Status</b>
To get balance fail with invalid token	<ol style="list-style-type: none"> <li>1. Prepare request with Authorization and Content-Type <code>\${token_invalid_token}</code></li> <li>2. When Call Get Balance API</li> <li>3. HTTP Status Should Be 401</li> <li>4. Response Body Status Should Be AUT4049, token is invalid</li> </ol>	<p>status: 401, code: AUT4049, message: token is invalid</p>	PASS
<b>Test case name</b>	Fail - Get Balance - Invalid Scope		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test case	Test Step	Expect Result	Status
To get balance fail with invalid scope	<ol style="list-style-type: none"> <li>1. Prepare request with Authorization and Content-Type <code>{token_no_scope}</code></li> <li>2. When Call Get Balance API</li> <li>3. HTTP Status Should Be 200</li> <li>4. Response Body Status Should Be RES4006, invalid scope</li> </ol>	status: 401, code: RES4006, message: invalid scope	PASS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลการวิจัยและข้อเสนอแนะ

#### 5.1 สรุปผลการวิจัย

จากการเข้าร่วมโครงการสหกิจศึกษากับทางบริษัท ทูม่อนี่ จำกัด ทำให้ได้ศึกษาโครงสร้างของบริษัทรวมไปถึงขั้นตอนการดำเนินงานต่างๆไปจนถึงได้เข้าร่วมทีมผู้พัฒนาของบริษัท พัฒนาแอปพลิเคชันทูม่อนี่ เดเวลอปเปอร์ แซนด์บ็อกซ์ ( TrueMoney Developer Sandbox ) เพื่อสร้างระบบทดลองให้กับร้านค้าและสามารถดำเนินงานในส่วนที่ได้รับมอบหมายจนสำเร็จทันตามกำหนดการ และครบถ้วนตามวัตถุประสงค์ของระบบคือ ได้ API จำลอง ( mock API ) ที่ร้านค้าสามารถใช้ได้เมื่อเชื่อมต่อเข้ากับระบบแอปพลิเคชัน ทูม่อนี่ วอลเล็ต ได้เว็บแอปพลิเคชันที่ร้านค้าสามารถทดสอบการทำงานของ API จำลองได้ และได้คู่มืออธิบายรายละเอียดและวิธีการใช้เว็บแอปพลิเคชัน อีกทั้งยังได้รับการทดสอบว่าสามารถทำงานได้ตรงกับความต้องการหรือไม่ นอกจากนี้ยังได้ทำการปรับปรุงและแก้ไขระบบเพื่อให้สอดคล้องกับเป้าหมายของโครงการให้มากที่สุด ทั้งนี้การเข้าร่วมโครงการสหกิจศึกษากับทางบริษัท ทูม่อนี่ จำกัด ทำให้ได้รับประสบการณ์ที่ดีและหลากหลาย ซึ่งถือว่าเป็นโอกาสในการเพิ่มทักษะส่วนบุคคล และทักษะวิชาชีพ ได้เรียนรู้การแก้ไขปัญหาต่างๆ ไม่ว่าจะเป็นการแก้ปัญหาเฉพาะหน้าหรือปัญหาการทำงาน อีกทั้งยังได้ ทักษะในการทำงานร่วมกับผู้อื่น ความรับผิดชอบงานที่ได้รับมอบหมาย การวางแผนการทำงานให้เป็นระบบและรอบคอบ

#### 5.2 ปัญหาและอุปสรรค

5.2.1 ใช้เวลาในรวบรวม Requirement ของระบบมากโดยใช้เวลา 1 สัปดาห์ในทุก ๆ sprint เนื่องจากเป็นระบบทดลองที่สร้างใหม่แยกจากระบบจริง และมีการเปลี่ยนแปลงเพื่อตอบสนองความต้องการด้านประสบการณ์การใช้ระบบของลูกค้าให้รู้สึกว่าย่างยไ้ได้มากที่สุด

5.2.2 ใช้เวลาในการทดลอง และเปลี่ยนเครื่องมือในการพัฒนาทั้งหมด 3 ครั้งทำให้การทำงานล่าช้ากว่ากำหนดการเดิม เนื่องจากผู้จัดทำไม่มีความรู้เกี่ยวกับโครงสร้าง ภาษาและเครื่องมือที่ใช้ที่มากเพียงพอ

5.2.3 การปรับตัวให้เข้ากับระบบทำงานแบบ Agile ในช่วงแรก เนื่องจากจะต้องแบ่งเวลาประชุมแผนงานและเขียนโค้ด และรับมือกับงานที่เปลี่ยนแปลงอยู่แทบจะตลอดเวลาหากมีการปรับเปลี่ยนงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3 ข้อเสนอแนะ

ศึกษาเครื่องมือและภาษาที่ใช้ก่อนที่จะมาเริ่มทำงานจริง และให้ความสำคัญกับการประชุมวางแผนงานให้ละเอียดและสม่ำเสมอ จะทำให้สามารถดำเนินงานได้เร็วยิ่งขึ้นในการพัฒนาระบบทดลอง และปรับปรุงระบบขั้นต่อไปให้ผู้พัฒนาฝั่งร้านค้าสามารถปรับแต่งข้อมูลใน API ต่าง ๆ ได้ตามจริง เพื่อเพิ่มความน่าสนใจ และทำให้ลูกค้าจะได้เห็นผลลัพธ์ที่ใกล้เคียงความเป็นจริงมากที่สุด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง

- [1] blog.cloudhm. สืบค้นเมื่อ 22 พฤศจิกายน 2565. **CI/CD คืออะไร? ช่วยให้ Developer ทำงานง่ายขึ้นได้มากขนาดไหน?**. [online]. สืบค้นได้จาก. <https://blog.cloudhm.co.th/ci-cd/>
- [2] medium. สืบค้นเมื่อ 22 พฤศจิกายน 2565. **Agile คืออะไร เริ่มใช้งานอย่างไร.** [online]. สืบค้นได้จาก. <https://medium.com/fastwork-engineering/agile-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-%E0%B9%80%E0%B8%A3%E0%B8%B4%E0%B9%88%E0%B8%A1%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99%E0%B8%AD%E0%B8%A2%E0%B9%88%E0%B8%B2%E0%B8%87%E0%B9%84%E0%B8%A3-ab749306d96e>
- [3] medium. สืบค้นเมื่อ 25 พฤศจิกายน 2565. **Scrum คืออะไร เริ่มใช้งานอย่างไร.** [online]. สืบค้นได้จาก. <https://medium.com/fastwork-engineering/scrum%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3%E0%B9%80%E0%B8%A3%E0%B8%B4%E0%B9%88%E0%B8%A1%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99%E0%B8%AD%E0%B8%A2%E0%B9%88%E0%B8%B2%E0%B8%87%E0%B9%84%E0%B8%A3-2483e761a47e>
- [4] blog.nextzy. สืบค้นเมื่อ 25 พฤศจิกายน 2565. **Test Case คืออะไร?**. [online]. สืบค้นได้จาก. <https://blog.nextzy.me/what-is-test-case-1c368f0c3ffb>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## คำรับรองเล่มสหกิจศึกษาโดยสถานประกอบการ

วันที่ 21 เดือน เมษายน พ.ศ. 2566

ข้าพเจ้า นายสันติสุข สุตสวาท ตำแหน่ง Development Manager เป็นตัวแทนของสถานประกอบการ บริษัททรูมันนี่ จำกัด ขอรับรองว่า ทางสถานประกอบการได้ตรวจสอบเล่มสหกิจศึกษา เรื่อง ทรูมันนี่ เดเวลอปเปอร์ แชนด์บ็อกซ์ ของนักศึกษาชื่อ นางสาว จินต์จุฑา เทศเจริญ และ นาย ปุณณรัตน์ รัตนภาวรณ ซึ่งเป็นนักศึกษาภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เรียบร้อยแล้ว และไม่มีส่วนหนึ่งส่วนใดในเล่มสหกิจศึกษาที่มีข้อมูลอ่อนไหว และ/หรือ ข้อมูลอันเป็นความลับอันจะก่อให้เกิดความเสียหายต่อสถานประกอบการ รวมทั้งอนุญาตให้สามารถเผยแพร่ต่อสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังได้ จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ ..... *สันติสุข* .....

(.....นาย สันติสุข สุตสวาท.....)

ตัวแทนสถานประกอบการ

ข้าพเจ้า ผศ.ดร. นवलสวาท หิรัญสกุลวงศ์ อาจารย์ที่ปรึกษาสหกิจศึกษา ได้ตรวจสอบเล่มสหกิจศึกษาแล้วและรับทราบว่าสถานประกอบการดำเนินการตรวจสอบเล่มสหกิจศึกษาแล้ว จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ ..... *นवलสวาท หิรัญสกุลวงศ์* .....

(.....ผศ.ดร. นवलสวาท หิรัญสกุลวงศ์.....)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้