

การพัฒนาปรับปรุงเว็บแอปพลิเคชันของระบบซีเอสดีในส่วนของ
ยูสเซอร์อินเทอร์เฟซ

DEVELOPING AND IMPROVING THE CSD WEB
APPLICATION IN THE USER INTERFACE



สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาตรีวิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEVELOPING AND IMPROVING THE CSD WEB
APPLICATION IN THE USER INTERFACE

Setthaporn Khamket

A COOPERATIVE EDUCATION SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENT FOR
THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE, SCHOOL OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2022

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	การพัฒนาปรับปรุงเว็บแอปพลิเคชันของระบบซีเอสดีในส่วนของยูสเซอร์อินเทอร์เฟซ
ชื่อนักศึกษา	นางสาวเศรษฐพร คำเกษร รหัสนักศึกษา 62050231
ปริญญา	วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2565
อาจารย์ที่ปรึกษา	ผศ.ดร.นวลสวาท หิรัญสกุลวงศ์

บทคัดย่อ

สหกิจศึกษานี้มีวัตถุประสงค์เพื่อพัฒนา และปรับปรุงเว็บแอปพลิเคชันของระบบ CSD ที่มีวัตถุประสงค์เพื่อรับฝากหลักทรัพย์แบบไร้ใบหลักทรัพย์ ในการพัฒนาได้ใช้เทคโนโลยีตามมาตรฐานในระดับองค์กร เพื่อปรับปรุงให้ระบบมีความทันสมัยมากขึ้น สามารถใช้งานได้ง่าย และเพิ่มประสิทธิภาพการทำงานของระบบให้ดียิ่งขึ้น พร้อมทั้งพัฒนาให้ง่ายต่อการส่งต่อ และบำรุงรักษาภายในอนาคต

โดยการพัฒนาครั้งนี้ผู้จัดทำได้รับผิดชอบในส่วนของการพัฒนาส่วนประสานงานกับผู้ใช้งานของเว็บแอปพลิเคชัน ซึ่งการพัฒนาของเว็บแอปพลิเคชันสร้างเพื่อตอบสนองความต้องการของผู้ใช้งาน และสามารถทำงานได้อย่างถูกต้อง ครบถ้วน

คำสำคัญ : เทคโนโลยี, ส่วนประสานงานกับผู้ใช้งาน, เว็บแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Developing and Improving the CSD Web Application
Students	Miss Setthaporn Khamket Student ID 62050231
Degree	Bachelor of Science (COMPUTER SCIENCE)
Department	COMPUTER SCIENCE
School	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2022
Advisor	Asst.Prof.Dr. Nualsawat Hiransakolwong

Abstract

The goal of this cooperative education is to create and maintain the CSD web application, which is in charge of securities depository. We apply enterprise-standard technology during the development stages to improve the system's modernity, usability, and performance. The development standards are preserved for transfer and maintenance in the future.

I oversee developing the user interface for the web application to make sure that it fits user demands and works effectively.

Keywords: Technology, User Interface, Web Application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การที่สหกิจศึกษาครั้งนี้สำเร็จ และผ่านไปได้ด้วยดี จะเกิดขึ้นไม่ได้หากไม่ได้รับการสนับสนุนจากบริษัทตลาดหลักทรัพย์แห่งประเทศไทยที่มีโอกาสผู้จัดทำได้เข้าไปศึกษาการทำงาน และเรียนรู้งานจากการสหกิจครั้งนี้ พร้อมทั้งยังสนับสนุนเครื่องมือที่ใช้ในการพัฒนา และช่วยแก้ปัญหาต่างๆ เรื่องอุปกรณ์เสมอมา

ขอขอบคุณพี่เลี้ยงที่เคารพ คุณสุภาพร คมสัน และที่ปรึกษาประจำทีมคุณพุดฉิมรัฐ เขวงชูดิรัตน์ ที่คอยสนับสนุน ช่วยเหลือ และให้คำแนะนำมาจนจบสหกิจศึกษา ขอขอบคุณพี่ๆ ในทีมพัฒนา UI, ทีม API และทีม Tester ที่คอยร่วมงานอย่างเข้าใจ ที่ช่วยสอน แก้ปัญหา และแนะนำการใช้เครื่องมือหรือโปรแกรมต่างๆ ที่จำเป็น พร้อมทั้งคอยให้กำลังใจเสมอมา

ขอขอบคุณ ดร.รุ่งรัตน์ เวียงศรีพนาวลัย และผศ.ดร.นวลสวาท หิรัญสกุลวงศ์ ที่คอยชี้แนะแนวทางการแก้ไขปัญหา พร้อมทั้งให้คำปรึกษา คอยให้ความร่วมมืออยู่เสมอ จนกระทั่งสหกิจศึกษาในครั้งนี้จบลงอย่างสมบูรณ์

ขอขอบคุณทุกท่านที่กล่าวไปอีกครั้งสหกิจศึกษาครั้งนี้จะผ่านไปไม่ได้เลยหากขาดส่วนใดส่วนหนึ่ง ขอขอบคุณสำหรับประสบการณ์ที่ดี การเรียนรู้ และความเชื่อมั่นที่มีให้เสมอมา

เศรษฐพร คำเกษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ช
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของสหกิจศึกษา.....	1
1.3 ขอบเขตของสหกิจศึกษา.....	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง.....	3
2.1 เทคโนโลยีที่เกี่ยวข้อง.....	3
2.1.1 เทคโนโลยี npm.....	3
2.2.2 ภาษา TypeScript.....	4
2.2.3 เทคโนโลยี React.....	6
2.2.4 เทคโนโลยี React Hook Form.....	7
2.2.5 เทคโนโลยี Redux-Saga.....	7
2.2.6 เทคโนโลยี Material User Interface.....	8
2.2.7 เทคโนโลยี Ant Design.....	8
2.2.8 เทคโนโลยี Swagger.....	8
2.2.9 เทคโนโลยี Lighthouse.....	8
บทที่ 3 วิธีการดำเนินงานสหกิจศึกษา.....	10
3.1 โครงสร้างของระบบ.....	10
3.1.1 Software Architecture.....	10
3.1.2 System Architecture.....	11
3.2 แผนภาพการใช้งานของผู้ใช้ระบบ.....	12
3.2.1 Structure Chart ของระบบ CSD.....	12
3.2.2 Use Case Diagram ของผู้ใช้ระบบ CSD.....	13

เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 แผนภาพแสดงการทำงานของระบบ (Use Case Diagram).....	15
3.3.1 หน้าประเภท Report	15
3.3.2 หน้าประเภท Create, Update และ View	19
3.3.3 หน้าประเภท Approve และ Reject.....	22
3.3.4 หน้าประเภท Generate Status.....	25
3.3.5 หน้าประเภท View	28
3.4 ขั้นตอนในการพัฒนา User Interface.....	30
3.4.1 Workflow ของเทคโนโลยีที่ใช้.....	30
3.4.2 โครงสร้างการเขียนโปรแกรม	35
3.4.3 เทคนิคการเขียนโปรแกรมที่ใช้.....	38
บทที่ 4 ผลการดำเนินงานและการอภิปรายผล.....	51
4.1 ภาพหน้าจอที่ได้รับผิดชอบในการพัฒนาของระบบ CSD.....	51
4.1.1 โมดูล Registrar.....	51
4.1.2 โมดูล Depository.....	55
4.2.3 โมดูล Inquiry/ Report.....	94
4.2 ผลการเปรียบเทียบประสิทธิภาพการทำงาน.....	96
4.2.1 โหมด Navigation	96
4.2.2 โหมด Timespan	102
4.2.3 โหมด Snapshot.....	105
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ.....	107
เอกสารอ้างอิง	108

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
3. 1 ตารางอธิบาย View Report Use Case	15
3. 2 ตารางอธิบาย Request Report Use Case	16
3. 3 ตารางอธิบาย Select Report Use Case	16
3. 4 ตารางอธิบาย Print Report Use Case.....	17
3. 5 ตารางอธิบาย Reprint Report Use Case.....	17
3. 6 ตารางอธิบาย Daily Report Use Case.....	17
3. 7 ตารางอธิบาย On Request Report Use Case	18
3. 8 ตารางอธิบาย Create Use Case Diagram.....	20
3. 9 ตารางอธิบาย View Data Use Case Diagram	20
3. 10 ตารางอธิบาย Update Data Use Case Diagram	20
3. 11 ตารางอธิบาย Inquiry Data Use Case Diagram	21
3. 12 ตารางอธิบาย Inquiry Data Use Case	23
3. 13 ตารางอธิบาย Select Record Use Case.....	23
3. 14 ตารางอธิบาย Accept Details Use Case.....	23
3. 15 ตารางอธิบาย Fill in Reject Reason Use Case.....	23
3. 16 ตารางอธิบาย Reject Record Use Case.....	24
3. 17 ตารางอธิบาย Approve Record Use Case.....	24
3. 18 ตารางอธิบาย Accept Generate Status Use Case	26
3. 19 ตารางอธิบาย Generate Status Use Case.....	26
3. 20 ตารางอธิบาย Fill in Reject Reason Use Case	26
3. 21 ตารางอธิบาย Select Record Use Case.....	27
3. 22 ตารางอธิบาย Inquiry Data Use Case	27
3. 23 ตารางอธิบาย View Data Use Case.....	29
3. 24 ตารางอธิบาย Inquiry Use Case.....	29
3. 25 ตารางอธิบาย Submit Use Case	29
3. 26 ตารางอธิบาย Fill in Search Form Use Case.....	29
4. 1 ตารางเปรียบเทียบค่า Metrics สำหรับค่า Performance.....	97
4. 2 ตารางแสดง Performance ของระบบเก่า	98
4. 3 ตารางเปรียบเทียบค่า Metrics สำหรับค่า Performance	100

เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตเห็นาไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. 4 ตารางแสดง Performance ของระบบใหม่ 100

4. 5 ตารางแสดงผลการเปรียบเทียบระยะเวลาของระบบเก่า และใหม่104



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2. 1 โลโก้ npm	4
2. 2 เว็บไซต์ npm.....	4
2. 3 ตัวอย่างโปรแกรมภาษา JavaScript.....	5
2. 4 ตัวอย่างโปรแกรมภาษา TypeScript	5
3. 1 Software Architecture ของระบบ.....	10
3. 2 System Architecture ของระบบ	11
3. 3 Structure Chart ของระบบ CSD	12
3. 4 Use Case Diagram ของ CSD System	13
3. 5 Use Case Diagram ของ Report	15
3. 6 Use Case Diagram ของ Create, Update และ View.....	19
3. 7 Use Case Diagram ของ Approve และ Reject	22
3. 8 Use Case Diagram ของ Generate Status	25
3. 9 Use Case Diagram ของ View.....	28
3. 10 Workflow ของเทคโนโลยีที่ใช้	30
3. 11 Component Code โดยใช้ React	31
3. 12 View เชื่อมต่อไป Action โดยใช้ React	31
3. 13 การสร้าง Action โดยใช้ React.....	32
3. 14 การสร้าง Types โดยใช้ React.....	32
3. 15 การสร้าง Saga โดยใช้ React	33
3. 16 ชนิดตัวแปร Interface ของ IHeader.....	33
3. 17 การสร้าง Reducer โดย React	34
3. 18 API Service Call โดย React.....	35
3. 19 โพลเดอร์ที่สำคัญ.....	35
3. 20 แผนภูมิโครงสร้างโพลเดอร์ src	37
3. 21 โปรแกรมหน้า App.tsx.....	38
3. 22 โปรแกรมหน้า Route.tsx	39
3. 23 โปรแกรมหน้า ModuleRoute.tsx	39
3. 24 โปรแกรมหน้า ModuleMapping.ts.....	40
3. 25 โปรแกรมหน้า ModuleMapping.ts.....	40

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตเห็นนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. 26 โปรแกรมหน้า LoaderAction.ts.....	41
3. 27 โปรแกรมหน้า LoaderSaga.ts.....	41
3. 28 โปรแกรมหน้า SearchSaga.ts.....	42
3. 29 ชนิดตัวแปร Interface ของ IHeader.....	42
3. 30 โปรแกรมหน้า ModuleService.ts.....	43
3. 31 โปรแกรมหน้า ModuleAPI.ts.....	43
3. 32 โปรแกรมหน้า HttpUtils.ts ฟังก์ชัน Get และ Post.....	44
3. 33 โปรแกรมหน้า HttpUtils.ts ฟังก์ชัน callApi.....	44
3. 34 ชนิดตัวแปร Interface ของ Axios Response.....	45
3. 35 ตัวอย่าง useState.....	45
3. 36 ตัวอย่าง useEffect.....	46
3. 37 ตัวอย่าง useContext.....	46
3. 38 ตัวอย่าง useCallback.....	47
3. 39 ตัวอย่าง useMemo.....	47
3. 40 ตัวอย่าง React Hook Form Control.....	48
3. 41 ตัวอย่าง React Hook Form Watch.....	48
3. 42 ตัวอย่าง React Hook Form Set Value.....	49
3. 43 ตัวอย่าง React Hook Form Handle Submit.....	49
3. 44 ตัวอย่าง React Hook Form - reset.....	49
3. 45 โปรแกรมหน้า ModuleName.tsx.....	50
4. 1 On Request หน้า Attached Release Letter Report ระบบเดิม.....	51
4. 2 On Request ของ Search Conditions หน้า Attached Release.....	52
4. 3 Daily Schedule หน้า Attached Release Letter Report.....	52
4. 4 Search Report List ของ Reprint หน้า Attached Release Letter Report.....	53
4. 5 Search Report List ของ Reprint Modal หน้า Attached Release Letter Report.....	53
4. 6 Search Report List ของ Print หน้า Attached Release Letter Report.....	54
4. 7 Search Report List Modal ของใบนำส่ง หน้า Attached Release Letter Report.....	54
4. 8 PDF Modal หน้า Attached Release Letter Report.....	55
4. 9 Search หน้า Unblock Shares Report ระบบเดิม.....	55
4. 10 Submit หน้า Unblock Shares Report ระบบเดิม.....	56
4. 11 Daily Schedule ของหน้า Unblock Shares.....	56
4. 12 Unblock Shares Summary Report ของ Daily Schedule หน้า Unblock Shares.....	57
4. 13 Search Report ของ Print หน้า Unblock Shares.....	57

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการดำเนินงาน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้วย
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. 14 Submit Report ของ Unblock Shares Summary หน้า Unblock Shares	58
4. 15 PDF Report Modal หน้า Unblock Shares	58
4. 16 Create หน้า Product Group ระบบเดิม.....	59
4. 17 Search หน้า Product Group ระบบเดิม	59
4. 18 ปุ่ม View หน้า Product Group ระบบเดิม.....	60
4. 19 View ของหน้า Product Group ระบบเดิม	60
4. 20 ปุ่ม Create หน้า Product Group.....	61
4. 21 Create Drawer หน้า Product Group	61
4. 22 Success Create Drawer หน้า Product Group.....	62
4. 23 ปุ่ม View และ Update หน้า Product Group	62
4. 24 View Drawer หน้า Product Group.....	63
4. 25 Update Drawer หน้า Product Group.....	63
4. 26 Success Update Drawer ของหน้า Product Group.....	64
4. 27 Create ของหน้า Fee Type ระบบเดิม.....	64
4. 28 Search หน้า Fee Type ระบบเดิม	65
4. 29 Update หน้า Fee Type ระบบเดิม.....	65
4. 30 ปุ่ม View หน้า Fee Type ระบบเดิม	66
4. 31 View หน้า Fee Type ระบบเดิม.....	66
4. 32 ปุ่ม Create หน้า Fee Type	67
4. 33 Create Drawer หน้า Fee Type.....	67
4. 34 Success Create Drawer หน้า Fee Type	68
4. 35 ปุ่ม View และ Update หน้า Fee Type.....	68
4. 36 View Drawer หน้า Fee Type	69
4. 37 Update Drawer ของหน้า Fee Type.....	69
4. 38 Success Update Drawer หน้า Fee Type.....	70
4. 39 Search Result หน้า Approve & Reject Fee Rate Custom ระบบเดิม.....	70
4. 40 Search ของ Participant หน้า Approve & Reject Fee Rate Custom.....	71
4. 41 Search ของ Other Registrar หน้า Approve & Reject Fee Rate Custom	71
4. 42 Approve หน้า Approve & Reject Fee Rate Custom.....	72
4. 43 Reject หน้า Approve & Reject Fee Rate Custom	72
4. 44 Confirm หน้า Approve & Reject Fee Rate Custom.....	73
4. 45 Success หน้า Approve & Reject Fee Rate Custom	73
4. 46 Search หน้า Adjust Fee Daily ระบบเดิม.....	74

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการดำเนินงานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านอื่นใด
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. 47 Search หน้า Adjust Fee Daily	74
4. 48 Reject หน้า Adjust Fee Daily	75
4. 49 Confirm หน้า Adjust Fee Daily	75
4. 50 Success หน้า Adjust Fee Daily	76
4. 51 Search ของหน้า Adjust Fee for Parti	76
4. 52 Search Result หน้า Adjust Fee for Parti	77
4. 53 Search หน้า Adjust Fee for Registrar.....	77
4. 54 Search Result หน้า Adjust Fee for Registrar	78
4. 55 Search Report หน้า Fees Report ระบบเดิม.....	78
4. 56 Search Report Submit ของหน้า Fees Report ระบบเดิม	79
4. 57 Search Report List ของหน้า Fees Report	79
4. 58 Submit Report List หน้า Fees Report	80
4. 59 รายงานสถิติ หน้า Fees Report ระบบเดิม.....	80
4. 60 รายงานสถิติ หน้า Fees Report.....	81
4. 61 รายงานสถิติ (Summary) หน้า Fees Report ระบบเดิม	81
4. 62 รายงานสถิติ (Summary) หน้า Fees Report.....	82
4. 63 รายงานสรุปค่าธรรมเนียม (รายเดือน) หน้า Fees Report ระบบเดิม.....	82
4. 64 รายงานสรุปค่าธรรมเนียม (รายเดือน) หน้า Fees Report	83
4. 65 Custody Fee Summary Government Bond หน้า Fees Report ระบบเดิม	83
4. 66 Custody Fee Summary Government Bond หน้า Fees Report.....	84
4. 67 Depository Fee Rate หน้า Fees Report ระบบเดิม	84
4. 68 Depository Fee Rate หน้า Fees Report.....	85
4. 69 Adjust Fee หน้า Fees Report ระบบเดิม.....	85
4. 70 Adjust Fee หน้า Fees Report.....	86
4. 71 Search หน้า Re-Auto Approve for Member ระบบเดิม	86
4. 72 Alert หน้า Re-Auto Approve for Member ระบบเดิม	87
4. 73 Search Result หน้า Re-Auto Approve for Member ระบบเดิม	87
4. 74 Search หน้า Re-Auto Approve for Member.....	88
4. 75 Alert Search หน้า Re-Auto Approve for Member.....	88
4. 76 Search Result หน้า Re-Auto Approve for Member.....	89
4. 77 Re-Auto Submit หน้า Re-Auto Approve for Member	89
4. 78 Alert Submit หน้า Re-Auto Approve for Member.....	90
4. 79 Status หน้า Re-Auto Approve for Member.....	90

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของสำนักงานกษาปณ์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้วย
 ใจว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. 80 Search หน้า Generate Entitlement ระบบเดิม.....	91
4. 81 Search Result หน้า Generate Entitlement ระบบเดิม.....	91
4. 82 Search หน้า Generate Entitlement	92
4. 83 Generate Submit หน้า Generate Entitlement.....	92
4. 84 Alert Submit หน้า Generate Entitlement.....	93
4. 85 Status ของหน้า Generate Entitlement.....	93
4. 86 Search หน้า DIF Message Inquiry ระบบเดิม	94
4. 87 View หน้า DIF Message Inquiry ระบบเดิม.....	94
4. 88 Search หน้า DIF Message Inquiry.....	95
4. 89 Search Result หน้า DIF Message Inquiry	95
4. 90 JSON Modal หน้า DIF Message Inquiry.....	96
4. 91 รายงานของโหมต Navigation ระบบเก่า.....	96
4. 92 ตัวอย่าง Network ที่ใช้ HTTP/1 ระบบเก่า.....	98
4. 93 ตัวอย่าง Cache Policy ระบบเก่า.....	98
4. 94 รายงานของโหมต Navigation ระบบใหม่.....	99
4. 95 ตัวอย่าง JavaScript ที่ไม่ได้ใช้งาน ระบบใหม่.....	101
4. 96 รายงานของโหมต Timespan ระบบเก่า.....	102
4. 97 ระยะเวลาการทำงาน ระบบเก่า	103
4. 98 รายงานของโหมต Timespan ระบบใหม่.....	103
4. 99 ระยะเวลาการทำงานของระบบใหม่	104
4. 100 รายงานของโหมต Snapshot ระบบเก่า	105
4. 101 รายงานของโหมต Snapshot ระบบใหม่	106

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันที่มีการพัฒนาเทคโนโลยีอย่างรวดเร็ว ทำให้องค์กรแต่ละองค์กรจำเป็นต้องมีการพัฒนาเพื่อตอบสนองต่อเทคโนโลยีที่เกิดขึ้น จึงเกิดระบบรับฝากหลักทรัพย์โดยไร้ใบหลักทรัพย์ หรือ CSD โดยพัฒนาในรูปแบบเว็บแอปพลิเคชัน เพื่อให้บริการด้านหลักทรัพย์แก่นายทะเบียนหลักทรัพย์ ทางการเงินชำระราคา ส่งมอบหลักทรัพย์ และงานศูนย์รับฝากหลักทรัพย์

ซึ่งในอดีตได้มีการพัฒนาระบบ CSD โดยใช้ Java Web Framework และใช้งานมาจนถึงปัจจุบัน เนื่องจากมีการประกาศมาจากผู้พัฒนา Java Web Framework ที่ใช้งานว่าจะยุติการพัฒนา ดังนั้นหลังจากการหยุดพัฒนาจึงเกิดปัญหาต่างๆ ตามมา รวมถึงปัญหาด้านการปรับปรุงระบบ และไม่เหมาะสมต่อการบำรุงรักษา จากเหตุผลดังกล่าวทางองค์กรจึงจัดทำนโยบาย CSD Refresh Technology เป็นนโยบายเพื่อการปรับปรุงเว็บแอปพลิเคชันเดิมให้มีความทันสมัย และมี User Interface ที่เข้าใจง่ายมากขึ้น

1.2 วัตถุประสงค์ของสหกิจศึกษา

ต้องการพัฒนาเว็บแอปพลิเคชันของระบบ CSD ให้มีความทันสมัยมากขึ้น มีการใช้งานที่ง่ายขึ้น และเพิ่มประสิทธิภาพของการทำงานให้ดีขึ้น

- 1) เพื่อนำ Web Framework ใหม่ ๆ ที่ตอบโจทย์งานของระบบ มาใช้ปรับปรุงระบบให้ดียิ่งขึ้น และเหมาะกับการแก้ไข บำรุงรักษา พร้อมทั้งง่ายต่อการส่งมอบต่อในอนาคต
- 2) เพื่อปรับเปลี่ยนหน้าจอของเว็บแอปพลิเคชันให้มีการใช้งานที่ทันสมัย เข้าใจง่าย และใช้งานมากขึ้น

1.3 ขอบเขตของสหกิจศึกษา

- 1) พัฒนาระบบรับฝากหลักทรัพย์แบบไร้ใบหลักทรัพย์ที่ใช้งานได้เฉพาะผู้ใช้งานภายในระบบ และสมาชิกศูนย์รับฝากหลักทรัพย์
- 2) พัฒนา User Interface ของเว็บแอปพลิเคชันจาก React JS และ Webpack
- 3) สามารถใช้งานระบบแบบออนไลน์เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ระบบที่มีความทันสมัยเหมาะต่อการใช้งาน ง่ายต่อผู้ใช้งาน และสามารถพัฒนาต่อได้ รวมถึงง่ายต่อการส่งต่อ หรือบำรุงรักษาภายในอนาคต
- 2) ได้เรียนรู้การทำงานร่วมกันในระดับองค์กร โดยใช้การทำงานแบบ Agile Methodology แบบ Scrum



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

การปรับปรุงโครงการ CSD Refresh Technology เพื่อให้มีความทันสมัย และใช้งานง่ายขึ้น จำเป็นต้องศึกษา และทำความเข้าใจเกี่ยวกับ ทฤษฎี และเทคโนโลยีที่เกี่ยวข้องกับการปรับปรุง ดังต่อไปนี้

2.1 เทคโนโลยีที่เกี่ยวข้อง

2.1.1 เทคโนโลยี npm

npm เป็น Software Registry ที่ใหญ่ที่สุดในโลก โดย npm มักจะถูกใช้เพื่อให้แบ่งปัน และ ยืม Package ที่ถูกสร้างโดยนักพัฒนา Open Source ทั้งหลายจากทั่วทุกทวีป อีกทั้งยังมีหลายองค์กรที่ใช้ npm เพื่อจัดการในการพัฒนาแบบส่วนตัว

npm ประกอบไปด้วย 3 สิ่งที่แตกต่างกัน ดังนี้

1) เว็บไซต์ (Website) มีไว้สำหรับใช้เพื่อค้นหา Package ที่ถูกสร้าง ใช้ตั้งค่าโปรไฟล์ของ Package ที่สร้างขึ้น อีกทั้งยังใช้ในการจัดการมุมมองจากการสร้าง Package โดยแค่เข้าร่วมใน Organization ของ npm ก็สามารถมีสิทธิ์ในการจัดการได้มุมมองของ Package ที่จะให้เป็นส่วนตัว หรือสาธารณะ

2) Command Line Interface หรือ CLI ใช้ดำเนินการผ่าน Terminal ซึ่งเป็นวิธีที่นักพัฒนา มักจะใช้กัน

3) Registry เป็นฐานข้อมูลขนาดใหญ่ และสามารถใช้ได้ในทุกบุคคลทั่วไป ของซอฟต์แวร์ JavaScript อีกทั้งยังมี Meta-Information ที่ไหลเวียนอยู่มากมาย

ประโยชน์ของการใช้ npm

1) ทำให้สามารถนำ Package ไปปรับใช้ในโปรแกรมที่กำลังเขียน หรือรวม Package ต่างๆ เข้าด้วยกันได้

2) สามารถดาวน์โหลดได้ในหลายระบบ โดยที่ไม่ต้องมีเครื่องมืออื่นๆ ในการช่วยในการโหลด

3) สามารถใช้ npx รัน Package ที่มีอยู่ได้เลย โดยไม่ต้องดาวน์โหลด

4) สะดวกในการแบ่งปันโปรแกรมให้กับผู้ใช้งาน npm ได้ในทุกๆ ที่

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้จัดทำเห็นว่าไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 5) สามารถจำกัดโค้ดให้นักพัฒนาที่เฉพาะเจาะจงได้
- 6) สร้างองค์กรที่สามารถควบคุมการบำรุงรักษา Package, การโค้ด Package และนักพัฒนาที่เข้าร่วมพัฒนา Package
- 7) สามารถจำลองทีมเสมือนได้ใน Organization
- 8) สามารถคั่นหานักพัฒนาที่สนใจเรื่องเดียวกัน และกำลังประสบปัญหาเดียวกัน



รูปภาพที่ 2. 1 โลโก้ npm

(ที่มา: <https://commons.wikimedia.org/wiki/File:Npm-logo.svg>)



รูปภาพที่ 2. 2 เว็บไซต์ npm

(ที่มา: <https://www.npmjs.com/>)

2.2.2 ภาษา TypeScript

TypeScript ถูกสร้างโดย Microsoft เป็น Superset ของ JavaScript ดังนั้นหมายความว่าอะไรก็ตามที่สามารถเขียนได้ในการเขียน JavaScript TypeScript ก็สามารทำได้เช่นกัน โดยเรียกกันง่ายๆ ก็คือ TypeScript นั้นได้สืบทอด JavaScript มาใส่ชนิดของข้อมูล (Type) และคุณสมบัติต่างๆ (Feature) โดยเป็นสิ่งที่สามารถใช้ประโยชน์ อำนวยความสะดวกได้ แต่ไม่มีใน JavaScript ในการ Compile สุดท้ายแล้ว TypeScript ก็ต้องเปลี่ยนเป็น JavaScript ก่อนที่จะถูกแสดงบนเบราว์เซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการใช้งาน JavaScript ในบางกรณีจะไม่สามารถดักจับ Error ที่ผิดพลาดได้ ถ้าหากสร้างตัวแปรตัวหนึ่งขึ้นมา แล้วดำเนินการต่างกับตัวแปรนั้นๆ ภายใต้ JavaScript จะไม่สามารถเช็คได้อย่างครบถ้วน ดังตัวอย่างต่อไปนี้

```
1  const sum = (a, b) => a + b;
2
```

รูปภาพที่ 2.3 ตัวอย่างโปรแกรมภาษา JavaScript

จากตัวอย่างด้านบนใน JavaScript สามารถเขียนและใช้งานได้อย่างปกติ แต่ไม่สามารถทราบได้เลยว่าอาร์กิวเมนต์ a และ b มีชนิดของข้อมูลเป็นอะไร ถ้าหากเกิดเหตุการณ์ที่ตัวแปร a เป็น String และตัวแปร b เป็น Number ขึ้นมา เหตุการณ์นี้จะไม่กระทบเลย ถ้าหากชิ้นงานที่คุณสร้างไม่ได้เป็นชิ้นงานที่ใหญ่ แต่หากงานชิ้นนั้นมีขนาดใหญ่ และเขียนเป็นหมื่นๆบรรทัด นี้อาจจะเป็นปัญหาของคุณ จากปัญหาที่กล่าวไปนี้ทำให้ TypeScript ถูกพัฒนาขึ้นมา โดยมีวิธีการประกาศแบบต่อไปนี้

```
1
2  const sum = (a: number, b: number) => a + b;
3
4  const s1 = sum(3, 5);
5  const s2 = sum(2, '33');
6
```

รูปภาพที่ 2.4 ตัวอย่างโปรแกรมภาษา TypeScript

จากภาพด้านบน คือ การประกาศตัวแปรชื่อ sum เป็นฟังก์ชันที่มี argument 2 ตัว คือ a เป็น number และ b เป็น number เช่นกัน แล้วให้ทำการนำ a + b จะเห็นได้ว่าเกิดการตรวจจับ error ที่เกิดขึ้นใน s2 เพราะว่ามี การประกาศ b เป็น String ทำให้มีการแจ้งเตือนว่า Argument of type 'string' is not assignable to parameter of type 'number'

ใน TypeScript มีการกำหนดชนิดของตัวแปร 2 ประเภท คือ Explicitly เป็นตัวแปรที่เกิดจากการกำหนดเองจากผู้เขียน และ Implicitly เป็นตัวแปรที่ถูกกำหนดโดย Compiler โดย ชนิดของข้อมูลที่พื้นฐานจะมี Boolean, Number, String และ Array โดยนอกจากตัวพื้นฐานแล้วยังมีตัวแปรอื่นๆ เช่น Any, Unknown, Null และ Undefined, Void, Never และ Enum อีกทั้งยังมีชนิดข้อมูลที่ซับซ้อน อย่าง Object และ Function อีกทั้งยังมีสำหรับตัวแปรที่มักถูกใช้ซ้ำ เช่น Interface, Union และ Intersect

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 เทคโนโลยี React

ในปี ค.ศ. 2013 Facebook ได้พยายามพัฒนาฟีเจอร์การพูดคุย (Chat Feature) โดยต้องการที่จะทำให้ฟีเจอร์การพูดคุยนี้สามารถไปอยู่ทุกหน้าที่ใช้งานอยู่ เลยเกิดความซับซ้อนเหมือนการที่ต้องนำแอปพลิเคชันที่กำลังจะเกิดขึ้นยัดเข้าไปในแอปพลิเคชันที่มีความซับซ้อนอยู่แล้ว เลยทำให้การควบคุมสิ่งที่เกิดขึ้นหลังจากที่ DOM (Document Object Model) นั้นมีความเป็นไปได้ยาก อีกทั้งยังมีปัญหาของเรื่องการจัดการทำงานแบบคู่ขนาน และการจัดจังหวะของผู้ใช้งานที่มากมาย หากนึกภาพไม่ออกให้ลองนึกถึงเรื่องการเรนเดอร์ของหน้าจอก็จะสามารถทราบได้อย่างรวดเร็วว่า DOM ที่จะถูกใช้งานได้ถูกเลือกมาใช้งาน และลำดับการทำงานของ DOM ที่ถูกนำมาใช้งานนั้นถูกต้องหรือไม่ โดยจากที่กล่าวไปจะไม่สามารถแน่ใจได้เลยก่อนหน้าที่จะมีการสร้าง React

ซึ่งการพัฒนา React อย่างแรกที่นักพัฒนาต้องคำนึงถึง คือ การแก้ไขปัญหาโดยการผูกมัดข้อมูลแบบทิศทางเดียว (Unidirectional data binding) หรือเรียกอีกชื่อว่า One-way Data Flow ของ Flux Architecture และการทำให้ Component State ที่ถูกออกแบบให้ไม่สามารถเปลี่ยนแปลงได้ (Immutable State) โดยหากมีการเปลี่ยนแปลงของ State จะทำการสร้าง State ใหม่ถูกแทนที่ State เก่า แต่หากสงสัยว่าทำไมต้องสร้าง State ใหม่แทน เพราะถ้าเป็นการ mutable เมื่อมีการเปลี่ยนแปลง State จะทำให้ State ที่ถูกกล่าวถึงถูกเปลี่ยนแปลงด้วย

โดย React เป็น JavaScript Library สำหรับการสร้าง UIs (User Interfaces) โดย React เหมาะกับการพัฒนาเว็บที่มี UIs ที่ซับซ้อน โดย React จะแบ่งส่วนของ UIs ต่างๆ ออกเป็นสิ่งเล็กๆ ที่เรียกว่าคอมโพเนนต์ (Component) โดย React จะใช้การเขียนโปรแกรมแบบ Declarative

React ได้มีการใช้ JSX ซึ่งเป็นส่วนขยายจาก JavaScript ที่เอาไว้ใช้ในการประกาศ Markup ในการเขียน JavaScript JSX เป็น Superset จาก JavaScript ก่อนหน้าที่จะมี JSX หากต้องการกำหนด UI จำเป็นต้องใช้ HTML Template เข้ามาช่วยในการแสดงผลลัพธ์ และไม่ได้มีมาตรฐานในการเขียน

ในการเขียนรู้ React เนื่องจากมีการแบ่งเป็น Component เลยจำเป็นต้องทำความเข้าใจกับ Component Lifecycle เพื่อให้เข้าใจ Component State ตีมากขึ้น และดึงประสิทธิภาพในการจัดการ State ได้ดีขึ้น และเพื่อไม่ให้เกิดการเขียนที่ทำให้ State นั้นเปลี่ยนแปลง อ้างอิงจากการที่นักพัฒนาตั้งใจจะสร้าง React ให้เป็น Immutable Component โดยใน React จะ Lifecycle ที่แตกต่างกันดังต่อไปนี้ คือ Mounting, Updating และ Unmounting ซึ่งใน Updating มี Render, Precommit และ Commit

ใน React เวอร์ชัน 16.0 ใน React Conference 2018 React ได้มีการเปิดตัว React Hook ครั้งแรก โดย React Hook ได้มีการอนุญาตให้ผู้เขียนได้เขียน Function ให้ตามหลัก Component

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Lifecycle ได้โดยไม่ต้องเขียนได้แค่ใน Class และยังสามารถแบ่งปันพฤติกรรมต่างๆ ระหว่าง Component ได้

2.2.4 เทคโนโลยี React Hook Form

เป็น Library ที่มีไว้เพื่อช่วยให้การเขียนเว็บแอปพลิเคชันที่ต้องการใช้รูปแบบการแบบฟอร์ม สามารถจัดการเกี่ยวกับการตรวจสอบได้ง่ายขึ้น และสามารถลดการเขียนโปรแกรมให้ และในการติดตั้งใช้งาน เป็น Package ที่มีขนาดเล็ก และสามารถติดตั้งได้ง่ายโดยใช้ npm โดย Library นี้ได้ทำการเขียนตามมาตรฐาน HTML Standard ในด้านการใช้งานทำให้เขียนได้น้อยลง แต่ประสิทธิภาพที่ดีกว่าเดิม เนื่องจากการนำ React Hook Form สามารถลดการแสดงผลซ้ำที่ไม่จำเป็นของหน้าจอดี โดยตัวของ React Hook Form ในแต่ละ Input จะมีการทำงานที่เป็นอิสระต่อกัน ทำให้สามารถกำหนดให้บางตัวของ Input สามารถทำงานซ้ำได้ โดยไม่จำเป็นต้องทำงานซ้ำทั้งแบบฟอร์ม

2.2.5 เทคโนโลยี Redux-Saga

เป็น Library ที่มีวัตถุประสงค์หลักเพื่อจัดการผิดพลาดต่างๆ ที่เกิดขึ้นกับการสร้างแอปพลิเคชัน เช่น ปัญหาที่เกิดขึ้นจากการดึงข้อมูลแบบอะซิงโครนัส (Asynchronous) ปัญหาที่เกิดจากข้อมูลที่ไม่สมควรจะได้รับจากการส่งข้อมูล อีกทั้งยังเพิ่มประสิทธิภาพในการ Execute และยังสามารถทดสอบในภายหลัง และสามารถรับมือกับข้อผิดพลาดได้ดียิ่งขึ้น

แบบจำลอง Saga มีไว้เพื่อแบ่งแอปพลิเคชันที่ใช้ออกเป็นเธรด (Thread) เพื่อให้แต่ละเธรดสามารถรับมือกับข้อผิดพลาด หรือผลข้างเคียงที่เกิดขึ้น โดยตัว Redux-Saga จริง ๆ แล้วเป็น Redux Middleware ทำให้แต่ละเธรดสามารถ เริ่มต้น, หยุด และทำการยกเลิกจากแอปพลิเคชันหลักได้ในทุกเมื่อ อีกทั้ง Redux-Saga ยังยอมให้ใช้ State ของ Redux ตัวเต็ม ในแอปพลิเคชัน และยังสามารถส่ง Redux Action ได้ด้วย

Redux-Saga ใช้ ES6 Feature เพื่อเรียก Generator ในการจัดการการไหลที่ไม่ตรงกัน (Asynchronous Flow) เพื่อให้ง่ายต่อการอ่าน เขียน และทดสอบ โดยการจัดการกับ Asynchronous Flow ใช้วิธีการจัดการแบบเดียวกับการเขียนโปรแกรมใน JavaScript ในการจัดจังหวะ Async และ Await

ในการใช้ Redux ทุกคนอาจจะเคยใช้ Redux-Thunk ในการจัดการการดึงข้อมูล แต่ในการใช้ Redux-Thunk อาจทำให้เกิดสิ่งที่เรียกว่า Callback Hell ซึ่งทาง Redux-Saga ได้เกิดมาเพื่อให้ทดสอบการไหลของข้อมูลที่ไม่ตรงกันให้ง่ายขึ้น และการกระทำแต่ละอย่างของงานที่ใช้ Redux-Saga ก็จะดูดีมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 เทคโนโลยี Material User Interface

Material User Interface หรือที่เรียกกันว่า Material-UI โดย Material-UI เป็น Library ที่อนุญาตให้ Component เข้ามาใช้งานโครงเจกต์ และนำมาใช้งานเพื่อสร้าง User Interface ในแอปพลิเคชันที่ใช้ React โดย Material-UI จะช่วยประหยัดเวลาในการเขียน User Interface เป็นอย่างมาก และไม่จำเป็นต้องเขียนทุกอย่างตั้งแต่เริ่มต้น

โดย Material-UI ได้นำหลักการออกแบบ Material Design ของ Google มาใช้เป็นหลักในการสร้าง Component

2.2.7 เทคโนโลยี Ant Design

Ant Design เป็น Library ที่พัฒนา User Interface ถูกสร้างมาโดยนักพัฒนาชาวจีน โดยถูกนำมาใช้ใน React ซึ่ง Ant Design มีวิธีการนำ Component มาใช้ได้อย่างง่ายดาย และเป็นที่สามารถมีปฏิสัมพันธ์ได้ รวมถึงง่ายต่อการนำไปใช้ร่วมกับ Library อื่นๆ

2.2.8 เทคโนโลยี Swagger

Swagger Specification หรือที่เรียกกันว่า OpenAPI Specification เป็นที่เอาไว้สำหรับอธิบายรูปแบบ API สำหรับ REST APIs โดย OpenAPI อนุญาตให้คุณอธิบายรายละเอียดทุกอย่างของ API ทั้งหมด โดยข้อกำหนด API สามารถเขียนได้ในรูปแบบ JSON หรือ YAML และด้วยความที่รูปแบบการเขียนที่เรียนรู้ง่าย และสามารถอ่านได้ทั้งมนุษย์และเครื่อง

ดังนั้น Swagger เป็นเครื่องมือ Open-Source ที่สร้างตามข้อกำหนด OpenAPI ที่ช่วยในการออกแบบ สร้างเอกสารต่างๆ โดยเครื่องมือ Swagger ที่สำคัญ เช่น Swagger API Dom เป็นต้น

2.2.9 เทคโนโลยี Lighthouse

เป็น open-source และเป็นเครื่องมืออัตโนมัติที่มีไว้คอยพัฒนาคุณภาพของเว็บเพจ โดยสามารถรันได้ในหลายๆ เว็บเพจที่เป็นสาธารณะ หรือที่ต้องการยืนยันตัวตน (Authentication) มีเพื่อตรวจสอบประสิทธิภาพ (Performance), การเข้าถึง (Accessibility), Progressive Web Apps (PWAs) คือ ความก้าวหน้าของเว็บแอปพลิเคชันที่เทียบเท่ากับ Native Apps เช่น Twitter, SEO และอื่นๆ โดยสามารถรัน Lighthouse ใน Chrome DevTools, Command Line หรือจาก Node Module เพียงแค่ใช้ URL ก็สามารถใช้ในการตรวจสอบได้ หลังจากนั้นส่งออกมาเป็น JSON และ Github Gists นำไปเปิดบน Light Viewer หรือสามารถนำส่งออกเป็น PDF ได้ ค่าที่วัดได้ใน Lighthouse มีค่าที่ใช้ประเมิน ทั้งหมด 5 ตัวชี้วัด ดังนี้

1) Performance ใช้วัดประสิทธิภาพและค้นหาโอกาสในการเพิ่มความเร็วในการโหลดหน้าเว็บ มีค่าที่ประเมินดังต่อไปนี้ ค่าเริ่มแรกเมื่อถูกวาดขึ้นในหน้าเว็บ (First Contentful Paint), เวลาในการตอบสนองกลับ (Time to Interactive) หลังจากกระทำการต่างๆ, ความเร็วในการโหลดเนื้อหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของหน้าเว็บในทุกส่วน(Speed Index), ค่ารวมเวลาที่ปิดกั้น (Total of Block Time) รวมความเร็ว FCP และ Time to Interation เมื่อการทำงานมากกว่า 50 ms เป็นค่าที่โดนปิดกั้นไม่ให้ตอบสนอง ในการป้อนข้อมูล เช่น ตอนที่กดคีย์บอร์ด, เวลาในการวาดรูปภาพหรือข้อความที่ใหญ่ที่สุดขึ้นมา (Largest Contentful Paint) และค่าวัดการเคลื่อนไหวที่สามารถมองเห็น (Cumulative Layout Shift)

2) Accessibility เป็นค่าที่วิเคราะห์ผู้ใช้ทั้งหมดสามารถเข้าถึงเนื้อหา และสำรวจไซต์ของคุณ ได้อย่างมีประสิทธิภาพหรือไม่

3) Best Practices เป็นค่าวิเคราะห์ที่ไว้ตรวจสอบว่าโปรแกรมที่เขียนขึ้นในหน้าเว็บ เหมาะสมหรือไม่ เช่น ใช้ HTTPS หรือไม่ มีการดึงข้อมูลจาก API ที่มีช่องโหว่หรือไม่

4) SEO ใช้ตรวจสอบว่าเมื่อมีการค้นหา หากค้นหาในเว็บจะสามารถค้นหาได้ดี หรือรวดเร็ว หรือไม่

5) Progressive Web Apps (PWA) ใช้ตรวจสอบเว็บว่าเป็นไปตามมาตรฐานใหม่ สามารถติดตั้งได้อย่างปลอดภัย และน่าเชื่อถือมากแค่ไหน โดยค่าด้านบนจะนำไปวิเคราะห์ใน 3 วิธีการ ดังนี้

- วิธีการ Navigation ใช้วิเคราะห์เว็บที่โหลดแบบหน้าเดียว ค่าที่วัดได้จะเรียกว่า Metric, ประเมินความสามารถของ PWAs ของเว็บ และวิเคราะห์การเข้าถึงหลังจากโหลดหน้าเว็บ โดยตัว Performance ไม่สามารถวิเคราะห์ Single Page Application ที่รู้จักกันว่าเป็นเว็บแบบ SPA สามารถใช้งานบน Browser เมื่อกด หรือมีการกระทำไปหน้าต่างๆ จะไม่มีการรีโหลด Page ใหม่ ตัวอย่าง Single Page Application เช่น Netflix มีวิเคราะห์ทั้งหมด ดังนี้
- วิธีการ Snapshot ใช้วิเคราะห์แต่ละหน้าใน State เฉพาะ ใช้หาปัญหาการเข้าถึงในเบื้องต้น เป็นวิธีการที่เหมาะสมกับการวิเคราะห์ User Exerience ยังสามารถประเมิน หรือวิเคราะห์แนวทางการปฏิบัติที่ดีที่สุดของเมนูต่างๆ ในเว็บ รวมถึงองค์ประกอบต่างๆ ที่อยู่หลังการโต้ตอบ
- วิธีการ Timespan ใช้วิเคราะห์เป็นช่วงเวลา ส่วนใหญ่ใช้ในการวิเคราะห์การโต้ตอบของผู้ใช้งาน (User Interaction) วัดการเปลี่ยนแปลงของ Layout ที่หน้าจอบริเวณ และวัดเวลาที่ใช้ดำเนินการของ JavaScript (JavaScript Execution) ค่าจาก Timespan จะไม่รวมค่า Performance จากวิธีการ Navigation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีการดำเนินงานสหกิจศึกษา

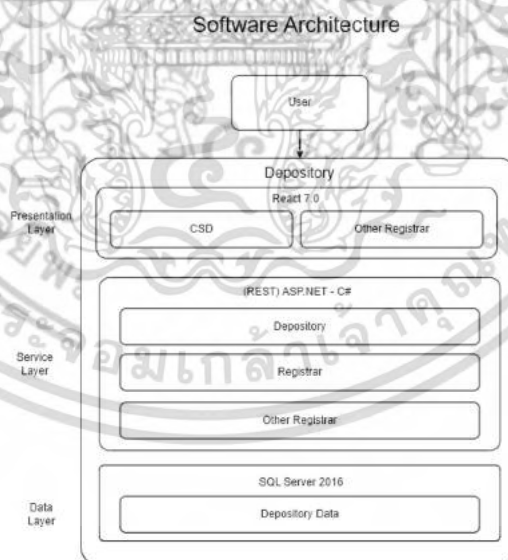
การพัฒนาเว็บแอปพลิเคชันได้พัฒนาจากระบบที่มีอยู่ขึ้นมาใหม่ของระบบ CSD และเปลี่ยนแปลงเทคโนโลยีที่ใช้ในการพัฒนา ซึ่งผู้จัดทำได้มีหน้าที่รับผิดชอบส่วนของการพัฒนา User Interface โดยมีโครงสร้าง แผนภาพการทำงาน และขั้นตอนการดำเนินการวิจัย ดังต่อไปนี้

3.1 โครงสร้างของระบบ

ระบบ CSD เป็นระบบหลักที่แบ่งการทำงานเป็นระบบย่อยๆ ตามรูปแบบการใช้งานของผู้ใช้งาน จึงวางแผนโครงสร้างก่อนเริ่มพัฒนา ตามสถาปัตยกรรมซอฟต์แวร์ และสถาปัตยกรรมระบบ ดังนี้

3.1.1 Software Architecture

การออกแบบ Software Architecture ของระบบ CSD ได้มีการวางแผนการพัฒนา ภาษาที่ใช้ในการพัฒนา และออกแบบโครงสร้างทางสถาปัตยกรรมซอฟต์แวร์ของระบบ แยกออกเป็น 3 ชั้น ได้แก่ ชั้นที่ติดต่อกับผู้ใช้งาน (Presentation Layer), ชั้นบริการที่มีให้ใช้งาน (Service Layer) และชั้นจัดเก็บข้อมูล (Data Layer) ดังนี้



รูปภาพที่ 3. 1 Software Architecture ของระบบ

จากรูปภาพที่ 3.1 เป็น Software Architecture หรือโครงสร้างสถาปัตยกรรมซอฟต์แวร์ของระบบรับฝากหลักทรัพย์ (Depository System) ในส่วน Presentation Layer จะเห็นว่า มีระบบ

ย่อยอีก 2 ระบบ คือ ระบบ CSD และระบบ Other Registrar ที่แยกออกจากกัน ชั้น Service Layer เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญตเห็นาไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นชั้นที่มีแค่บางระบบ ซึ่งในระบบนี้มีเพื่อแยกการเข้าถึงข้อมูล โดยแบ่งเป็น Depository, Registrar และ Other Registrar เป็นส่วนที่เก็บ Logic ของระบบ โดยระบบนี้ได้ใช้ REST API ในการพัฒนา ชั้น Data Layer เป็นชั้นที่เก็บข้อมูลของระบบ ในการพัฒนานี้ใช้ SQL Server 2016 มีฐานข้อมูล Depository

3.1.2 System Architecture

การออกแบบ System Architecture ของระบบ CSD ได้ออกแบบการใช้งานของ User และเทคโนโลยีที่นำมาพัฒนาระบบ ได้ดังนี้



รูปภาพที่ 3. 2 System Architecture ของระบบ

จากรูปภาพที่ 3.2 เป็น System Architecture โดยระบบที่ทำงานอยู่มีการเข้าถึงได้โดยใช้ https ผ่าน .Nginx และ Docker เชื่อมต่อ API ผ่าน .Net และสามารถเข้าไปสู่ฐานข้อมูล SQL Server กับ LDAP ที่ไว้เก็บข้อมูลต่างๆของผู้ใช้งาน

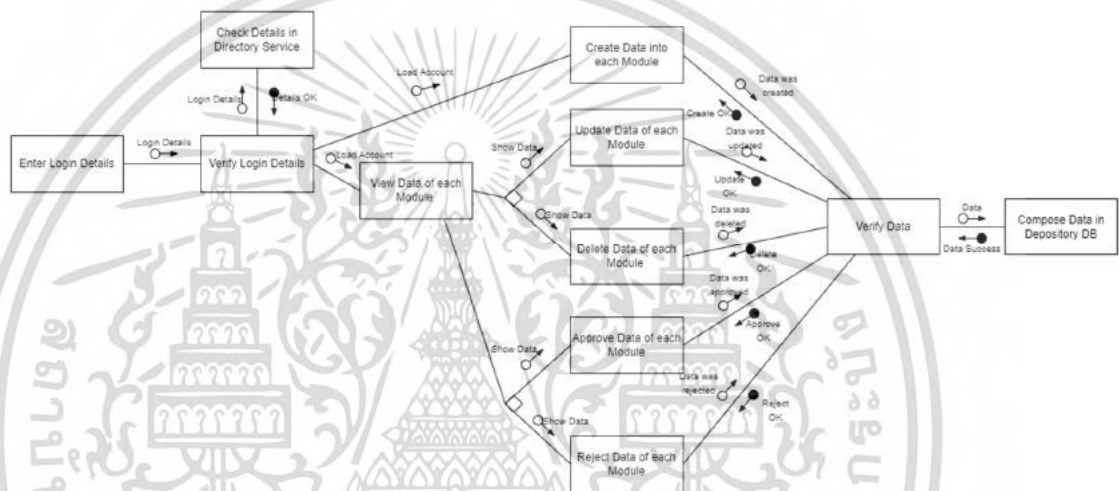
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 แผนภาพการใช้งานของผู้ใช้ระบบ

การพัฒนา ระบบ CSD ได้วางแผนการเข้าใช้งาน ออกแบบการเข้าใช้งาน และขั้นตอนที่การประมวลผลของส่วนต่าง ๆ เพื่อแสดงให้เห็นการทำงานของระบบ สามารถอธิบายได้ดังนี้

3.2.1 Structure Chart ของระบบ CSD

การออกแบบ Structure Chart ของระบบ เพื่ออธิบายฟังก์ชันหลักที่ผู้ใช้งานเข้าใช้งาน โดยมีการใช้งานคล้ายกันในทุกระบบย่อย ดังนี้ สอบถาม (Inquiry), สร้าง (Create), ดู (View), อัปเดต (Update) และลบ (Delete) ดังนี้



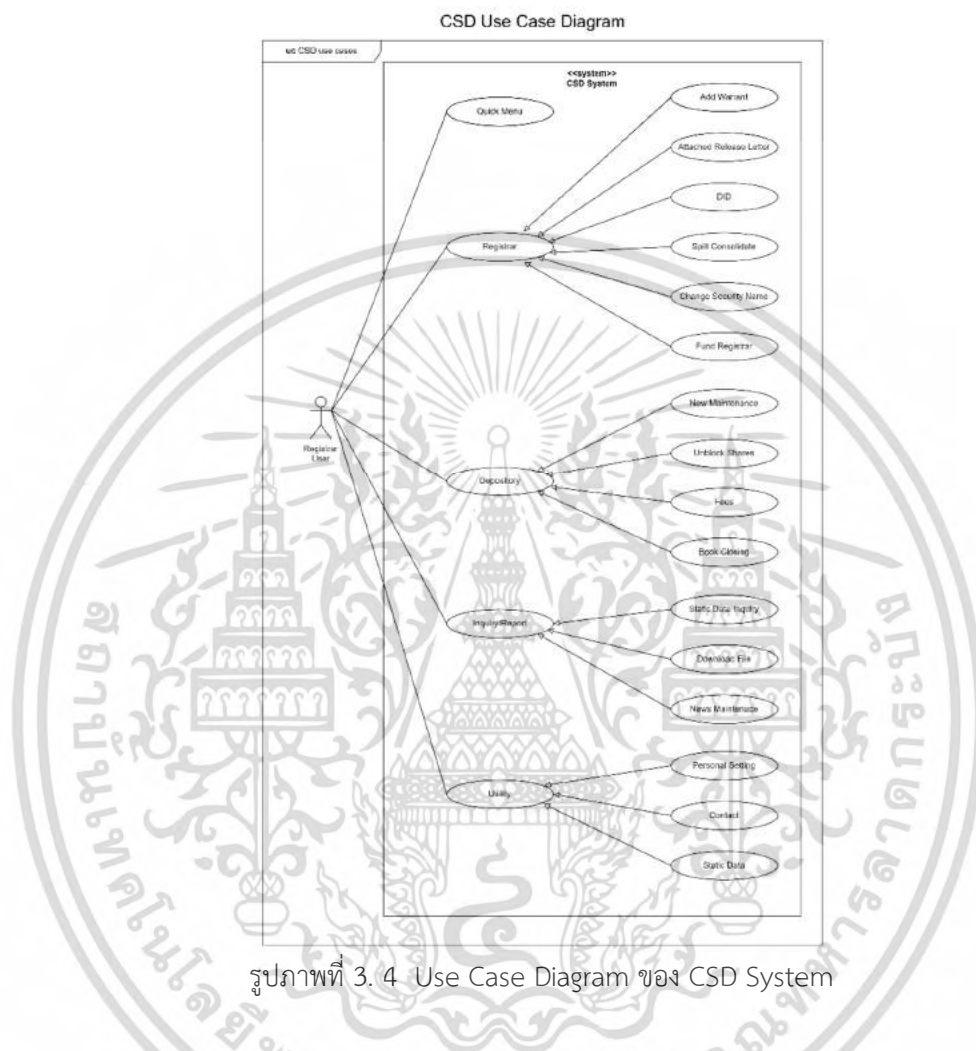
รูปภาพที่ 3. 3 Structure Chart ของระบบ CSD

จากรูปที่ 3.3 Structure Chart ของระบบ CSD มีการทำงานเริ่มต้นจากการ Login เข้าสู่ระบบโดยกรอกรายละเอียด Login หลังจากมีข้อมูลแล้วจะทำการตรวจสอบรายละเอียด (Verify Details) และนำรายละเอียดไปเทียบกับฐานข้อมูล Depository (Depository Database) หากรายละเอียดที่ตรวจสอบถูกต้องก็จะถูกโหลดข้อมูลบัญชีก่อนเข้าสู่ระบบ เมื่อเข้าสู่ระบบแล้ว สามารถเลือกฟังก์ชันที่ต้องการ ดังนี้ สร้างข้อมูลเข้าสู่ระบบ (Create) และดูข้อมูลที่มีอยู่ภายในระบบ (View) ซึ่งในการดูข้อมูล (View) จะสามารถใช้เงื่อนไขในการเลือกฟังก์ชันที่ต้องการ โดยมีทั้งหมด 4 แบบ คือ แก้ไข หรืออัปเดตข้อมูล (Update), ลบข้อมูล (Delete), ยืนยันรายการข้อมูล (Approve) และยกเลิกรายการข้อมูล (Reject)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 Use Case Diagram ของผู้ใช้ระบบ CSD

การออกแบบ Use Case Diagram ของผู้ใช้งานระบบ CSD มีผู้ใช้งาน คือ Registrar User และแสดงความสัมพันธ์ในการเข้าใช้งานโดยใช้เส้นเชื่อม ดังนี้



จากรูปที่ 3.4 เป็น Use Case Diagram ของระบบ CSD จากรูปแสดงโมดูลต่างๆที่สามารถเข้าใช้งานได้ภายในระบบ CSD ทั้งหมด โดยแยกประเภทออกเป็นเมนูต่างๆ ซึ่งจะประกอบไปด้วยโมดูลย่อย ได้แก่

โมดูลเมนู Quick Access ที่สามารถเลือกโมดูลที่ผู้ใช้งานต้องการเข้าถึงง่าย และต้องการใช้งานบ่อย

โมดูลเมนู Registrar ประกอบไปด้วย โมดูลย่อย Add Warrant, Attached Release Letter, DID, Spilt Consolidate, Change Security Name และ Fund Registrar

โมดูลเมนู Depository ประกอบไปด้วย โมดูลย่อย New Maintenance, Unblock Shares, Fees และ Book Closing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมดูลเมนู Inquiry/Report ประกอบไปด้วยโมดูลย่อย Static Data Inquiry, Download File และ New Maintenance

โมดูลเมนู Utility ประกอบไปด้วยโมดูลย่อย Personal Setting, Contact และ Static Data



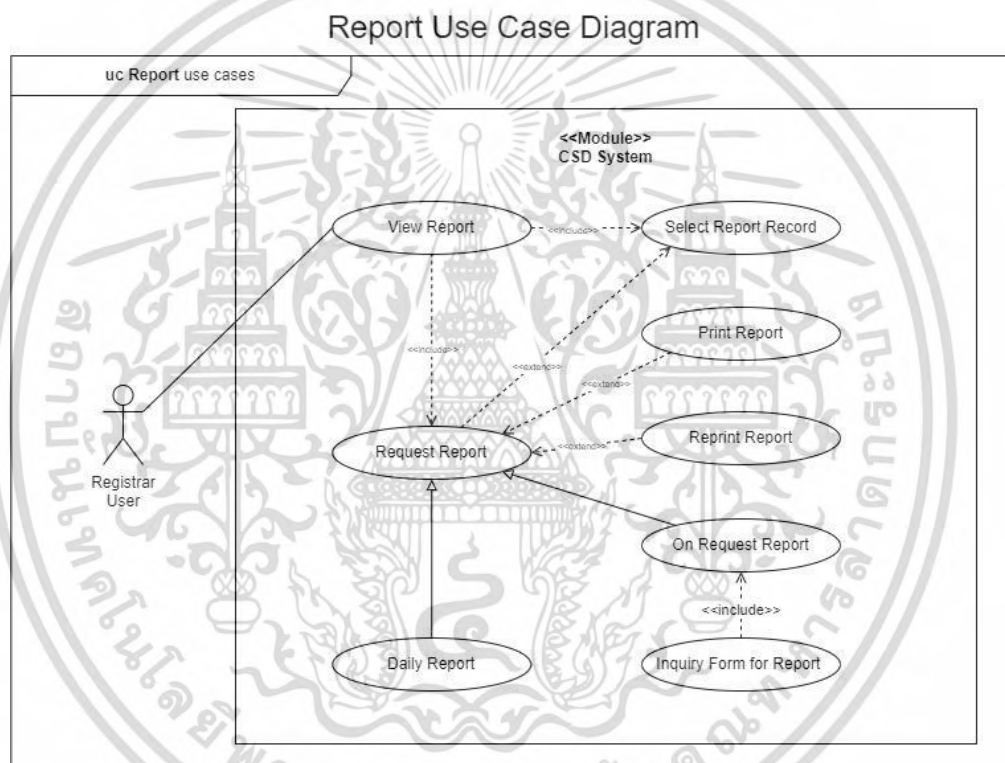
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 แผนภาพแสดงการทำงานของระบบ (Use Case Diagram)

แผนภาพแสดงการทำงานของระบบจะแสดงการทำงานในส่วนต่างๆ เฉพาะหน้าที่ผู้พัฒนาได้รับมอบหมายให้พัฒนา โดยอธิบายในรูปแบบ Use Case Diagram สามารถแบ่งได้ 5 ประเภทได้แก่

3.3.1 หน้าประเภท Report

หน้า Report เป็นหน้าสร้างรายงานสถานะของบริการต่างๆ เพื่อให้ผู้ใช้งานสามารถดูรายละเอียดได้ ได้พัฒนาใน 3 โมดูลย่อยทั้งหมด 3 หน้า ดังนี้ โมดูลย่อย Attached Release Letter, โมดูลย่อย Unblock Shares และโมดูลย่อย Fees โดยมี Use Case Diagram ดังนี้



รูปภาพที่ 3.5 Use Case Diagram ของ Report

จากรูปภาพที่ 3.5 Use Case Diagram ของ Report มีผู้ใช้งานเป็น Registrar User ที่แสดงการทำงานของผู้ใช้งานเมื่อต้องการดูรายงาน ต้องการรายงานที่ต้องการ โดยต้องเลือกประเภทรายงานแบบรายวัน (Daily Report) หรือรายครั้ง (On Request Report) หากต้องการแบบรายครั้งต้องกรอกเงื่อนไขในการค้นหารายงาน เลือกชนิดของรายงานประเภทสั่งพิมพ์อีกครั้ง (Reprint Report) หรือสั่งพิมพ์ปกติ (Print Report) และเลือกชื่อรายงานที่มีอยู่ในฐานข้อมูล

โดยสามารถอธิบาย Use Case Diagram ของ Report ในรูปแบบตารางความสัมพันธ์ จาก Use Case CSD-01 ถึง CSD-07 ได้ดังนี้

ตารางที่ 3.1 ตารางอธิบาย View Report Use Case

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Use Case Number	CSD-01
Use Case Name	View Report
Actors	Registrar User
Description	เพื่อเลือกรายการรายงานของโมดูลย่อยที่ต้องการ
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
	ต้องแจ้งความต้องการรายงาน และระบุเงื่อนไขของ Report ที่จำเป็นต่อการค้นหา
	ต้องเลือกรายการในหน้ารายการค้นหา หรือมีอยู่ในระบบก่อนหน้าแล้ว
Post-Condition	แสดงหน้า PDF Modal สำหรับ View
	สามารถดาวน์โหลดไฟล์ไปใช้งานได้

ตารางที่ 3. 2 ตารางอธิบาย Request Report Use Case

Use Case Number	CSD-02
Use Case Name	Request Report
Actors	Registrar User
Description	เพื่อแจ้งความต้องการรายงานจากรายการที่มีในระบบของโมดูลย่อย และระบุเงื่อนไขที่ต้องระบุ ของรายการที่ต้องการ
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
	ต้องระบุชนิดของรายงาน
	ถ้าหากชนิดรายงานใบนำส่ง เป็น On Request Report ต้องเลือกฟังก์ชันการทำงานระหว่าง Print หรือ Reprint
Post-Condition	เลือกรายการที่มีในระบบของโมดูลย่อยที่ต้องการ

ตารางที่ 3. 3 ตารางอธิบาย Select Report Use Case

Use Case Number	CSD-03
Use Case Name	Select Report
Actors	Registrar User
Description	เพื่อเลือกรายการรายงานที่ต้องการค้นหาในระบบของแต่ละโมดูลย่อย เพื่อจะแสดงรายงานจากข้อมูลในตารางที่แสดงหลังค้นหา
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
	ต้องเลือกชนิดของรายงานใบนำส่งก่อน (On Request Report หรือ Daily Schedule)
	ถ้าหากชนิดรายงานใบนำส่งเป็น On Request Report ต้องเลือกฟังก์ชันการทำงานระหว่าง Print หรือ Reprint
Post-Condition	กรอกข้อมูลตามแบบฟอร์มของแต่ละรายการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3. 4 ตารางอธิบาย Print Report Use Case

Use Case Number	CSD-04
Use Case Name	Print Report
Actors	Registrar User
Description	เป็นฟังก์ชันการทำงานที่ต้องการจะ Print รายงานใบนำส่ง
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
	ต้องเลือกชนิดของรายงานใบนำส่งก่อน (On Request Report หรือ Daily Schedule)
	ถ้าหากชนิดรายงานเป็น On Request Report ฟังก์ชันการทำงานระหว่าง Print หรือ Reprint
Post-Condition	เลือกรายการรายงานที่ต้องการค้นหาในระบบของแต่ละโมดูล และกรอกเงื่อนไขในการแสดงรายงาน

ตารางที่ 3. 5 ตารางอธิบาย Reprint Report Use Case

Use Case Number	CSD-05
Use Case Name	Reprint Report
Actors	Registrar User
Description	เพื่อแจ้งความต้องการรายงาน และระบุรายละเอียดที่ต้องการของรายงาน Attached Release Letter ที่ต้องการ
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
	ต้องระบุชนิดของรายงาน
	ต้องเลือกชนิดของรายงานใบนำส่งก่อน (On Request Report หรือ Daily Schedule)
Post-Condition	เลือกรายการรายงานที่ต้องการค้นหาในระบบของแต่ละโมดูล และกรอกเงื่อนไขในการแสดงรายงาน

ตารางที่ 3. 6 ตารางอธิบาย Daily Report Use Case

Use Case Number	CSD-06
Use Case Name	Daily Report
Actors	Registrar User
Description	เป็นชนิดของรายงานที่ต้องระบุเวลา โดยจะสามารถหาได้จาก Transaction Date ซึ่ง Daily Schedule Report เป็นรายงานตามตารางเวลารายวัน
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
Post-Condition	เลือกรายการรายงานที่ต้องการค้นหาในระบบของแต่ละโมดูล และกรอกเงื่อนไขในการแสดงรายงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3. 7 ตารางอธิบาย On Request Report Use Case

Use Case Number	CSD-07
Use Case Name	On Request Report
Actors	Registrar User
Description	เป็นชนิดของรายงานใบนำส่งที่ต้องระบบ
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
Post-Condition	เลือกรายการรายงานที่ต้องการค้นหาในระบบของแต่ละโมดูล และกรอกเงื่อนไขในการแสดงรายงาน
	สามารถดาวน์โหลดไฟล์ไปใช้งานได้

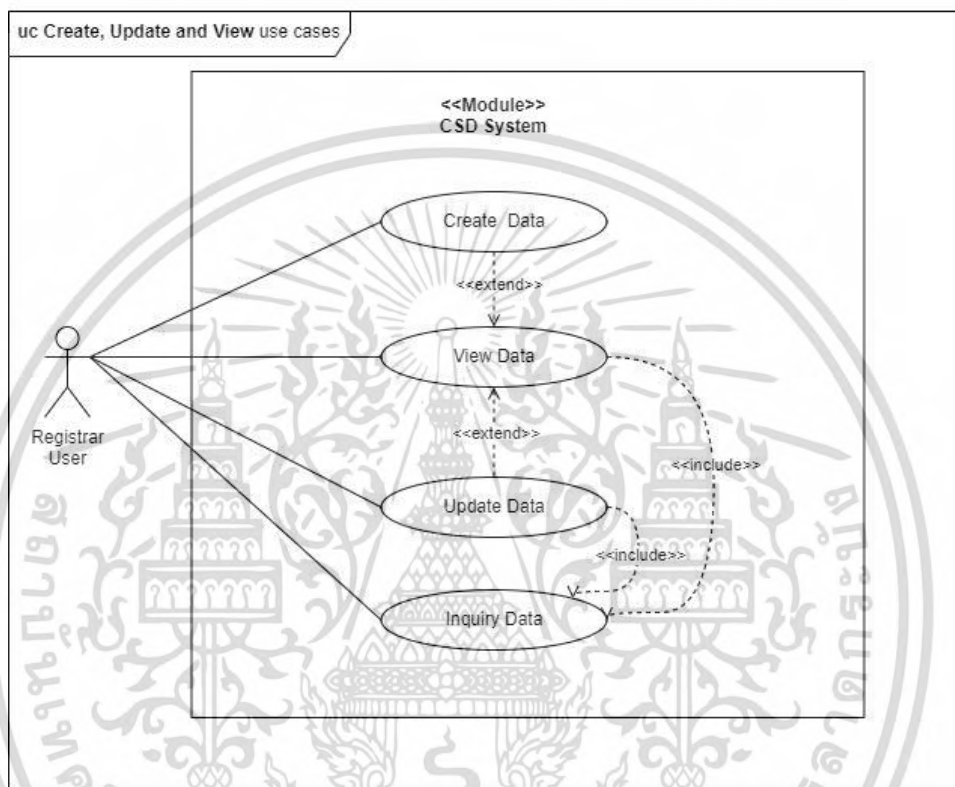


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 หน้าประเภท Create, Update และ View

หน้า Create, Update และ View เป็นหน้าที่มีรูปแบบ Drawer ที่ขยายออกมาจากด้านขวาเมื่อต้องการใช้งาน ใช้สำหรับสร้าง อัปเดต และดูข้อมูลที่ต้องการค้นหา ได้พัฒนาใน 1 โมดูลย่อย ทั้งหมด 2 หน้า ดังนี้ โมดูลย่อย Fees หน้า Product Group และ Fee Type

Create, Update and View Use Case Diagram



รูปภาพที่ 3. 6 Use Case Diagram ของ Create, Update และ View

จากรูปภาพที่ 3.6 Use Case Diagram ของ Create, Update และ View มีผู้ใช้งานเป็น Registrar User ที่แสดงการทำงานของผู้ใช้งานเมื่อต้องการสร้าง อัปเดต และดูข้อมูลที่ผู้ใช้งานต้องการ การอัปเดตข้อมูล (Update Data) และดูข้อมูล (View Data) ต้องสอบถามข้อมูลก่อน (Inquiry Data) แต่การสร้างข้อมูล (Create Data) สามารถสร้างได้โดยกรอกตามเงื่อนไขที่กำหนดของแต่ละหน้า ซึ่งการสร้างข้อมูล และอัปเดตข้อมูลมีผลกระทบต่อการดูข้อมูล (View Data)

โดยสามารถอธิบาย Use Case Diagram ของ Create, Update และ View ในรูปแบบตารางความสัมพันธ์ จาก Use Case CSD-11 ถึง CSD-14 ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3. 8 ตารางอธิบาย Create Use Case Diagram

Use Case Number	CSD-11
Use Case Name	Create Data
Actors	Registrar User
Description	เพื่อสร้างข้อมูลเพิ่ม โดยสามารถกรอกรายละเอียดการเพิ่มได้จาก Drawer ที่แสดงขึ้นมาหลังจากกดปุ่ม Create
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
Post-Condition	ข้อมูลถูกเพิ่มในตารางผลลัพธ์ด้านล่างของแบบฟอร์มของการค้นหา

ตารางที่ 3. 9 ตารางอธิบาย View Data Use Case Diagram

Use Case Number	CSD-12
Use Case Name	View Data
Actors	Registrar User
Description	เพื่อแสดงรายการที่เลือกจากแถวที่แสดงในตารางหลังสอบถามรายละเอียด โดยแสดงบน Drawer
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
	ต้องกรอกแบบฟอร์มสอบถาม หรือค้นหา พร้อมระบุเงื่อนไขที่ค้นหา เพื่อค้นหา ก่อน
Post-Condition	-

ตารางที่ 3. 10 ตารางอธิบาย Update Data Use Case Diagram

Use Case Number	CSD-13
Use Case Name	Update Data
Actors	Registrar User
Description	เพื่อแก้ไขข้อมูล โดยสามารถกรอกรายละเอียดการเพิ่มได้จาก Drawer ที่แสดงขึ้นมาหลังจากกดปุ่ม Update ที่อยู่ในคอลัมภ์ Action แต่ละแถว หากมีการแก้ไขข้อมูลใน View ก็จะต้องถูก Update ด้วย และสามารถแก้ไขได้แค่ช่องที่อนุญาตแก้ไขเท่านั้น
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
	ต้องกรอกแบบฟอร์มสอบถาม หรือค้นหารายละเอียด เพื่อค้นหา ก่อน
Post-Condition	ข้อมูลแก้ไขใหม่ในตาราง และใน View Drawer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3. 11 ตารางอธิบาย Inquiry Data Use Case Diagram

Use Case Number	CSD-14
Use Case Name	Inquiry Product Group
Actors	Registrar User
Description	เพื่อสอบถามข้อมูล หรือค้นหาข้อมูลในโมดูลย่อย
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
Post-Condition	ข้อมูลแก้ไขใหม่ในตารางด้านล่างแบบฟอร์มค้นหา หรือสอบถามข้อมูล

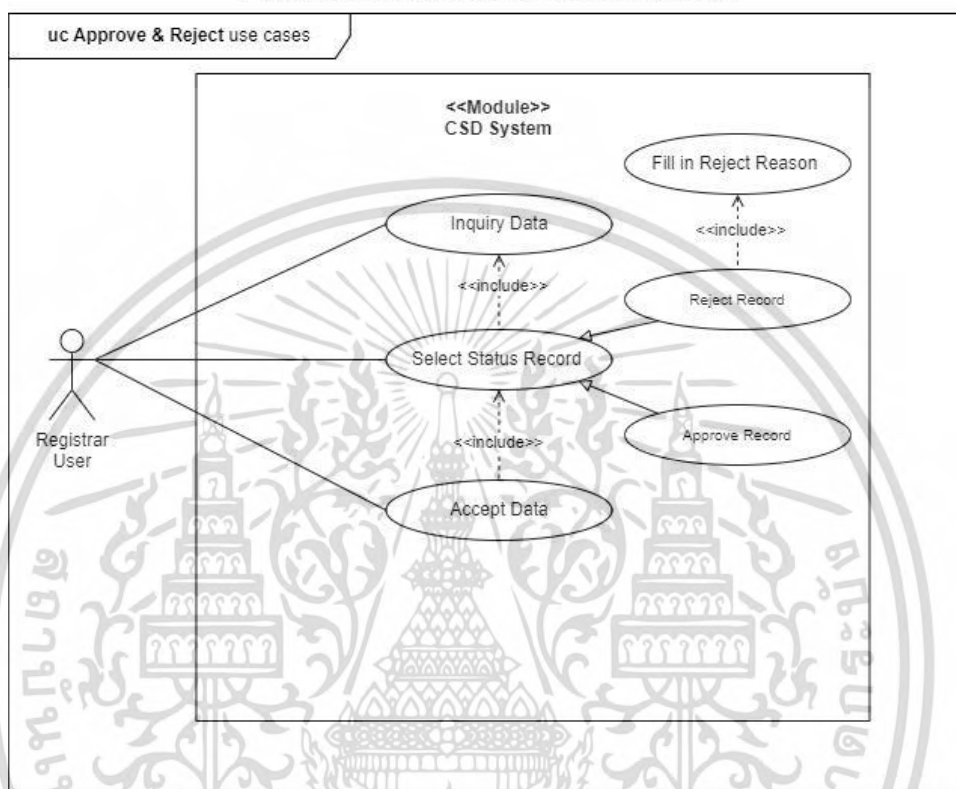


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 หน้าประเภท Approve และ Reject

หน้า Approve & Reject เป็นหน้าอนุมัติ และปฏิเสธรายการธุรกรรม ได้พัฒนาใน 1 โมดูลย่อย ทั้งหมด 1 หน้า ดังนี้ โมดูลย่อย Fees หน้า Fee Rate Custom

Approve & Reject Use Case Diagram



รูปภาพที่ 3. 7 Use Case Diagram ของ Approve และ Reject

จากรูปภาพที่ 3. 7 Use Case Diagram ของ Approve และ Reject มีผู้ใช้งานเป็น Registrar User ที่แสดงการทำงานของผู้ใช้งานเมื่อต้องการอนุมัติข้อมูล (Approve Record) และปฏิเสธข้อมูล (Reject Record) โดยผู้ใช้งานจำเป็นต้องมีสิทธิในการอนุมัติ และปฏิเสธ ผู้ใช้งานสามารถต้องสอบถามข้อมูล (Inquiry Data) และเลือกข้อมูลที่ต้องการเปลี่ยนสถานะ (Select Status Record) ก่อนที่จะสามารถยืนยันสถานะข้อมูลได้ ถ้ายืนยันสถานะปฏิเสธข้อมูล (Reject Record) ต้องกรอกเหตุผลที่ต้องปฏิเสธ (Fill in Reject Reason) โดยดึงข้อมูลจะดึงมาจากฐานข้อมูล เพื่อดึงรายการธุรกรรมที่ต้องการอนุมัติ (Approve Record) หรือปฏิเสธ (Reject Record)

โดยสามารถอธิบาย Use Case Diagram ของ Approve และ Reject ในรูปแบบตารางความสัมพันธ์ จาก Use Case CSD-31 ถึง CSD-36 ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3. 12 ตารางอธิบาย Inquiry Data Use Case

Use Case Number	CSD-31
Use Case Name	Inquiry Data
Actors	Registrar User
Description	เพื่อสอบถาม หรือค้นหาข้อมูลที่ต้องการอนุมัติ หรือปฏิเสธ ของโมดูลย่อย
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
Post-Condition	ข้อมูลจะแสดงในตารางผลลัพธ์ด้านล่างแบบฟอร์มกรอกข้อมูล

ตารางที่ 3. 13 ตารางอธิบาย Select Record Use Case

Use Case Number	CSD-32
Use Case Name	Select Record
Actors	Registrar User
Description	เพื่อเลือกรายการที่ต้องการอนุมัติ หรือปฏิเสธ ของโมดูล
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
	ต้องสอบถาม หรือค้นหาข้อมูลก่อน
Post-Condition	ยืนยันข้อมูลที่เลือกจากตาราง
	หากเลือกปฏิเสธต้องกรอกหมายเหตุก่อนกดยืนยันที่เลือก

ตารางที่ 3. 14 ตารางอธิบาย Accept Details Use Case

Use Case Number	CSD-33
Use Case Name	Accept Data
Actors	Registrar User
Description	เพื่อยืนยันข้อมูลที่อนุมัติ หรือปฏิเสธ ที่เลือกจากตาราง
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
	ต้องสอบถาม หรือค้นหาข้อมูลก่อน
Post-Condition	-

ตารางที่ 3. 15 ตารางอธิบาย Fill in Reject Reason Use Case

Use Case Number	CSD-34
Use Case Name	Fill in Reject Reason
Actors	Registrar User
Description	เพื่อกรอกเหตุผลในการปฏิเสธ Fee Rate Custom หากไม่มีเหตุผล จะไม่สามารถยกเลิกข้อมูลได้
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
	ต้องเลือกเป็นปฏิเสธ หรือ Reject ก่อน
Post-Condition	ยืนยันข้อมูลที่เลือกจากตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3. 16 ตารางอธิบาย Reject Record Use Case

Use Case Number	CSD-35
Use Case Name	Reject Fee Rate Custom Record
Actors	Registrar User
Description	เป็นชนิดของที่ชื่อว่า "ปฏิเสธ" การเลือกรายการ โดยสามารถเลือกได้ 2 แบบ คือ อนุมัติ และปฏิเสธ หากเลือกปฏิเสธจะต้องกรอกเหตุผลที่ปฏิเสธ
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
	ต้องสอบถามข้อมูล (Inquiry) โดยการกรอกฟอร์มการค้นหาก่อน
Post-Condition	กรอกเหตุผลที่ต้องการปฏิเสธ

ตารางที่ 3. 17 ตารางอธิบาย Approve Record Use Case

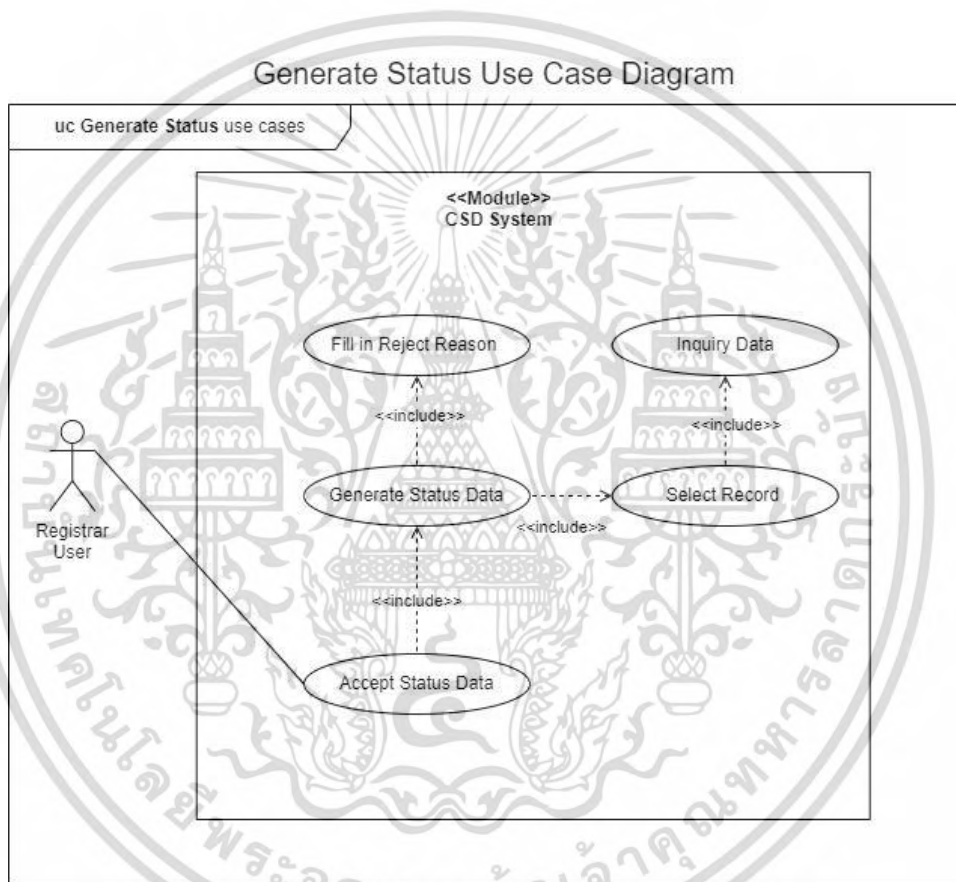
Use Case Number	CSD-36
Use Case Name	Approve Fee Rate Custom Record
Actors	Registrar User
Description	เป็นชนิดของที่ชื่อว่า "ยืนยัน" การเลือกรายการ โดยสามารถเลือกได้ 2 แบบ คือ อนุมัติ และปฏิเสธ
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
	ต้องสอบถาม หรือค้นหาข้อมูลก่อน โดยการกรอกฟอร์มการค้นหาก่อน
Post-Condition	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.4 หน้าประเภท Generate Status

หน้า Generate Status เป็นหน้าที่มีฟังก์ชันที่เปลี่ยนสถานะของรายการธุรกรรมนั้น ได้พัฒนาใน 2 โมดูลย่อย ทั้งหมด 3 หน้า ดังนี้

- สถานะ Reject โมดูลย่อย Fees หน้า Maintenance Adjust Fee Daily
- สถานะ Re-Auto โมดูลย่อย Book Closing หน้า Re-Auto Approve for Member
- สถานะ Generate โมดูลย่อย Book Closing หน้า Generate Shareholder Entitlement List and Right Revoke



รูปภาพที่ 3. 8 Use Case Diagram ของ Generate Status

จากรูปภาพที่ 3. 8 Use Case Diagram ของ Generate Status มีผู้ใช้งานเป็น Registrar User ที่แสดงการทำงานของผู้ใช้งานเมื่อต้องการเปลี่ยนสถานะ โดยผู้ใช้งานจำเป็นต้องมีสิทธิเข้าถึงการเปลี่ยนสถานะนั้น ก่อนยืนยันการเปลี่ยนสถานะของข้อมูล (Accept Status Data) ต้องสอบถามข้อมูล (Inquiry Data) แล้วเลือกรายการธุรกรรมที่ต้องการยืนยัน (Select Record) และกรอกเหตุผลที่ต้องการยืนยันสถานะ (Fill in Reason) และจะสามารถสร้างสถานะของรายการธุรกรรมได้ (Generate Status Data) โดยดึงข้อมูลจะดึงมาจากฐานข้อมูล เพื่อแสดงรายการธุรกรรมที่สามารถเปลี่ยนสถานะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสามารถอธิบาย Use Case Diagram ของ Generate Status ในรูปแบบตารางความสัมพันธ์ จาก Use Case CSD-41 ถึง CSD-45 ได้ดังนี้

ตารางที่ 3. 18 ตารางอธิบาย Accept Generate Status Use Case

Use Case Number	CSD-41
Use Case Name	Accept Status Data
Actors	Registrar User
Description	เพื่อยืนยันสถานะของ Record ที่เลือก
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
	เลือก Record ที่ต้องการสร้างสถานะ
Post-Condition	-

ตารางที่ 3. 19 ตารางอธิบาย Generate Status Use Case

Use Case Number	CSD-42
Use Case Name	Generate Status
Actors	Registrar User
Description	กวดตรงรายการเพื่อสร้างสถานะข้อมูลในตารางผลลัพธ์ค้นหา สามารถมี 3 สถานะ ในหน้าที่แตกต่างกัน คือ Reject, Re-Auto และ Generate
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
	ต้องสอบถามข้อมูล (Inquiry) ก่อน
Post-Condition	ยืนยันสถานะของ Record ที่เลือก

ตารางที่ 3. 20 ตารางอธิบาย Fill in Reject Reason Use Case

Use Case Number	CSD-43
Use Case Name	Fill in Reject Reason
Actors	Registrar User
Description	เพื่อกรอกเหตุผลในการปฏิเสธ บางหน้าหากไม่มีเหตุผล จะไม่สามารถยกเลิกข้อมูลได้ แต่บางหน้าก็ไม่จำเป็นต้องกรอกเหตุผล
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
	ต้องเลือกเป็นสร้างสถานะก่อน
Post-Condition	ยืนยันสถานะของข้อมูลที่เลือกจากตารางผลลัพธ์การค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3. 21 ตารางอธิบาย Select Record Use Case

Use Case Number	CSD-44
Use Case Name	Select Record
Actors	Registrar User
Description	เพื่อเลือกรายการที่ต้องการสร้างสถานะ จากตารางผลลัพธ์ด้านล่างแบบฟอร์มสอบถาม หรือค้นหา
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
	ต้องสอบถาม หรือค้นหาข้อมูลก่อน โดยการกรอกฟอร์มการค้นหาก่อน
Post-Condition	สร้างสถานะข้อมูลในตารางผลลัพธ์ค้นหา

ตารางที่ 3. 22 ตารางอธิบาย Inquiry Data Use Case

Use Case Number	CSD-45
Use Case Name	Inquiry Data
Actors	Registrar User
Description	ค้นหา หรือสอบถาม ข้อมูลเพื่อให้แสดงผลพื้ในตาราง
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
Post-Condition	ข้อมูลจะแสดงในตารางผลลัพธ์ด้านล่างแบบฟอร์มสอบถาม หรือค้นหาข้อมูล

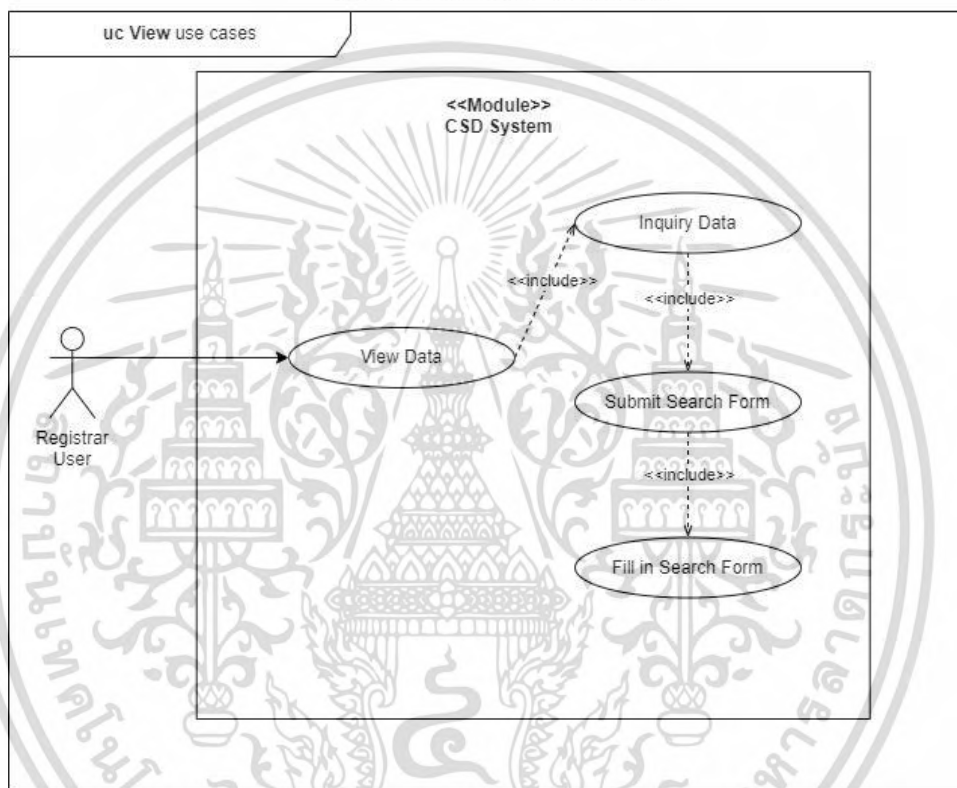
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5 หน้าประเภท View

หน้า View เป็นหน้ารูปแบบ Drawer สำหรับแสดงข้อมูลที่ผู้ใช้งานต้องการดู ได้พัฒนาใน 2 โมดูลย่อย ทั้งหมด 3 หน้า ดังนี้

- โมดูลย่อย Fees หน้า Adjust Fee for Participant และ Adjust Fee for Registrar
- โมดูลย่อย Depository Message Inquiry หน้า DIF Message Inquiry

View Use Case Diagram



รูปภาพที่ 3. 9 Use Case Diagram ของ View

จากรูปภาพที่ 3. 9 Use Case Diagram ของ View มีผู้ใช้งานเป็น Registrar User ที่แสดงการทำงานของผู้ใช้งานเมื่อต้องการดูรายละเอียดของรายการธุรกรรม (View Data) ก่อนที่จะดูรายการธุรกรรมที่ต้องการ ต้องกรอกข้อมูลเพื่อค้นหารายการธุรกรรม (Fill in Search Form) แล้วยืนยันการค้นหา (Submit Search Form) และสอบถามข้อมูล (Inquiry Datas) โดยดึงข้อมูลที่สอบถามมาจากฐานข้อมูล เพื่อแสดงรายการธุรกรรมที่ต้องการดู

โดยสามารถอธิบาย Use Case Diagram ของ Generate Status ในรูปแบบตารางความสัมพันธ์ จาก Use Case CSD-51 ถึง CSD-55 ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3. 23 ตารางอธิบาย View Data Use Case

Use Case Number	CSD-51
Use Case Name	View Data
Actors	Registrar User
Description	เพื่อดูข้อมูลของโมดูลย่อยแต่ละหน้า มีรูปแบบเป็น Drawer และ JSON Modal
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
	ต้องกรอกข้อมูลที่ต้องการค้นหา หรือสอบถาม
Post-Condition	ข้อมูลจะแสดงในตารางผลลัพธ์ด้านล่างฟอร์มสอบถาม หรือค้นหาข้อมูล

ตารางที่ 3. 24 ตารางอธิบาย Inquiry Use Case

Use Case Number	CSD-52
Use Case Name	Inquiry Data
Actors	Registrar User
Description	ค้นหา หรือสอบถาม ข้อมูลเพื่อให้เห็นผลลัพธ์ในตาราง
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
Post-Condition	ข้อมูลจะแสดงในตารางผลลัพธ์ด้านล่างแบบฟอร์มสอบถาม หรือค้นหาข้อมูล

ตารางที่ 3. 25 ตารางอธิบาย Submit Use Case

Use Case Number	CSD-53
Use Case Name	Submit Search Form
Actors	Registrar User
Description	ยืนยันแบบฟอร์มที่ต้องการค้นหา หรือสอบถามข้อมูล
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
	ต้องสอบถาม หรือค้นหาข้อมูลก่อน
Post-Condition	ข้อมูลจะแสดงในตารางผลลัพธ์ด้านล่างแบบฟอร์มสอบถาม หรือค้นหาข้อมูล

ตารางที่ 3. 26 ตารางอธิบาย Fill in Search Form Use Case

Use Case Number	CSD-54
Use Case Name	Fill in Search Form
Actors	Registrar User
Description	กรอกเงื่อนไขที่ต้องการค้นหาลงในแต่ละฟิลด์ ตามเงื่อนไขของแต่ละฟิลด์ โดยฟิลด์ที่แสดงเครื่องหมาย * หมายถึง มีความจำเป็นที่ต้องกรอกเพื่อค้นหา
Pre-Condition	Actors ต้องลงชื่อเข้าใช้ระบบ CSD
Post-Condition	ข้อมูลจะแสดงในตารางผลลัพธ์ด้านล่างแบบฟอร์มสอบถาม หรือค้นหาข้อมูล

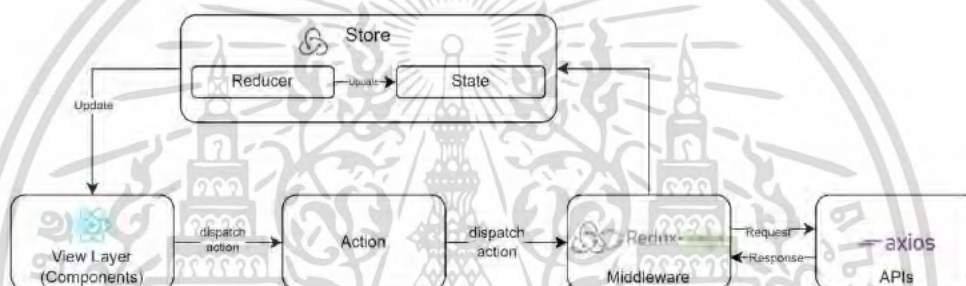
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ขั้นตอนในการพัฒนา User Interface

ระบบ CSD เป็นระบบที่จัดเป็นระบบขนาดใหญ่เนื่องจากในระบบยังมีระบบย่อยๆ และมีการเรียกใช้ APIs หลาย APIs ทำให้ต้องเลือกใช้หาเทคโนโลยีที่เหมาะสม พร้อมนำเทคโนโลยีที่หลากหลายมาช่วยควบคุม และจัดการ ดังนี้

3.4.1 Workflow ของเทคโนโลยีที่ใช้

แผนผัง Workflow ของเทคโนโลยีที่ใช้ เพื่ออธิบายเทคโนโลยีที่นำมาใช้พัฒนา User Interface และแสดงถึงขั้นตอนในการทำงานของ State ตั้งแต่ขั้นตอนที่เห็นทางหน้าจอ ถึงการใช้งาน APIs ดังนี้



รูปภาพที่ 3. 10 Workflow ของเทคโนโลยีที่ใช้

จากรูป Workflow ของระบบ CSD จะเห็นได้ว่ามีการทำงานในการพัฒนา User Interface มีการใช้ ซึ่งเทคโนโลยีหลักที่นำมาพัฒนา คือ React version 17.0.2 ซึ่งใช้ Middleware ชื่อว่า Redux Saga ควบคู่กับ Redux ในการจัดการ State ควบคุม และจัดการกับการรับส่ง HTTP Request และ HTTP Response APIs โดยใช้ Axios ซึ่งสามารถอธิบาย Workflow ได้ดังนี้

1) View Layer

จะใช้ React เวอร์ชัน 17.0.2 ในการพัฒนา โดยมีการใช้ TypeScript ประกอบไปด้วย Component ต่าง โดยจะมีการ Render และแสดงออกมาบนเบราว์เซอร์ ในส่วนนี้จะมีการนำ Component จาก Library ของ Material-UI และ Ant Design มาใช้เพื่อการออกแบบที่สวยงาม และง่ายต่อการพัฒนา มีรูปแบบการเขียนโปรแกรม ดังนี้

```

const Component: FC<IProps> = () => {
  return (
    <Card>
      { /* Fields */ }
    </Card>
  )
}
export default Component

```

รูปภาพที่ 3. 11 Component Code โดยใช้ React

จากรูปภาพที่ 3. 11 Component Code โดยใช้ React เป็นการเขียนแบบที่ใช้ Arrow Function มีคำสั่ง FC ใช้ระบุว่าฟังก์ชันเป็น Function Component

2) Action

จะเป็นส่วนที่เชื่อมต่อกับ View Layer โดยในจะส่ง State ไป และในบางการทำงานจะส่งมีการรับ Response จาก API ก็อาจจะต้องมีการเพิ่มอากิวเมนต์ ไปให้ Action ของโมดูลนั้นๆ โดยมี Payload ที่จะส่งไป Saga และมีการใช้ type ในการเชื่อมต่อระหว่าง Action กับ Saga มีรูปแบบการเขียนโปรแกรม ดังนี้

```

const Component: FC<IProps> = () => {
  const onSearch: SubmitHandler<IProps> = (state) => {
    dispatch(
      ModuleAction.onSearch(state)
    )
  }
  return (
    <Card>
      { /* Fields for Search */ }
    </Card>
  )
}
export default Component

```

รูปภาพที่ 3. 12 View เชื่อมต่อไป Action โดยใช้ React

จากรูปที่ 3.12 เป็นรูปแสดงการเขียนโปรแกรมในหน้า View ที่เป็นส่วนของ View Layer ในที่เชื่อมต่อไป Action โดยมีการใช้คำสั่ง “dispatch” ในการส่ง state ไปยังส่วนของ Action จากตัวอย่างส่ง Action แบบ Search ที่ใช้ในการค้นหาไปที่ Action

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

const onSearch = (
  data: IComponentState
): IModuleSagaRequest => {
  return {
    type: ModuleType.ON_SEARCH,
    payload: {
      data
    }
  }
}

export default {
  onSearch,
}

```

รูปภาพที่ 3. 13 การสร้าง Action โดยใช้ React

จากรูปที่ 3.13 เป็นการเขียนโปรแกรมในหน้า Action ที่มีฟังก์ชันรับ state ที่ส่งมาจาก View ในรูปที่ 3.12 และมีการ return ค่ากลับเป็น type ที่ใช้บอกชนิดของ Action และ Payload เก็บข้อมูลที่ต้องการส่ง

3) Middleware

ใช้ Redux-Saga ในการจัดการ side-effect ที่เกิดจากการดึงข้อมูลจาก API ใน Saga จะมีการเชื่อมค่าจาก State ที่ได้รับจากหน้าจอเชื่อมกับตัวแปรที่ API ได้รับ ในส่วนของ Saga จะมีการเชื่อมต่อระหว่าง Saga กับ Action โดยใช้ Type ที่สร้างไว้ มีรูปแบบการเขียนโปรแกรม ดังนี้

```

const ModuleType = {
  ON_SEARCH: 'module/ON_SEARCH'
}

export {ModuleType}

```

รูปภาพที่ 3. 14 การสร้าง Types โดยใช้ React

จากรูปที่ 3.14 เป็นการเขียนโปรแกรมกำหนดชนิดของ Action จะแยกออกมาอีกไฟล์ เนื่องจากใช้เชื่อมต่อกับ Saga ในรูปที่ 3.15

```

function* onSearch({
  payload: { data }
}): IModuleSagaRequest {
  try {
    // Show Loading.gif
    yield put(LoaderAction.startLoading())

    const header: IHeader = {
      channel,
      functionName
    }

    const response = yield call(
      ModuleService.functionName,
      header,
      data
    )

    // Stop Loading.gif
    yield put(LoaderAction.stopLoading())
    // Forward API response to View Layer
    onSuccess(response)
  } catch (err) {
    yield put(LoaderAction.stopLoading())
    // Show modal dialog with error
    yield put(
      Dialog.exceptionHandler(err)
    )
  }
}

takeLatest(ModuleTypes.ON_SEARCH, onSearch)

```

รูปภาพที่ 3. 15 การสร้าง Saga โดยใช้ React

จากรูปที่ 3.15 เป็นการเขียนโปรแกรมในส่วนของ Saga ที่เป็น Middleware ไว้ใช้เชื่อมต่อและจัดจังหวะการดึงข้อมูลโดยใช้ API จากรูปได้มีการเริ่มการทำงานโดยใช้คำสั่ง put และข้างกำหนดการเป็น Start หมายถึงเริ่ม และมีการกำหนด IHeader ตั้งชื่อเป็น header ที่ไว้ใช้ใส่ฟังก์ชัน Code เนื่องจากแต่ละบริการมีสิทธิในการเข้าถึงไม่เหมือนกัน ในการเรียกใช้ข้อมูลจะใช้คำสั่ง call ที่ใส่พารามิเตอร์เป็น ชื่อฟังก์ชัน, header และข้อมูลที่ถูกส่งมาจาก Action ในรูปที่ 3.13 และมีการหยุดการทำงานก่อนส่งข้อมูลกลับโดยใช้คำสั่ง put เหมือนตอนเริ่มแต่เปลี่ยน Load Action เป็น stop บรรทัดสุดท้ายที่ใช้คำสั่ง takeLatest เป็นการใช้คำสั่งเชื่อมต่อกับ Action โดยใช้ชนิดที่กำหนด ในรูปภาพที่ 3.14

```

export interface IHeader {
  action?: string
  channel: string
  functionName: string
}

```

รูปภาพที่ 3. 16 ชนิดตัวแปร Interface ของ IHeader

จากรูปที่ 3.16 เป็นการเขียนโปรแกรมกำหนด Interface ของ IHeader ที่ใช้เป็นชนิดข้อมูลของ header ในหน้า Saga จากรูปภาพที่ 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) Store

จะทำการเก็บ state ทั้งหมดไว้ในนี้ ถ้าหากต้องการเปลี่ยน Action จำเป็นต้องเปลี่ยนใน Store เท่านั้น ซึ่งประกอบไปด้วย Reducer ซึ่งจะเป็นขั้นตอนที่ใช้ในการนำ State ที่ได้กลับมาจากการดึงข้อมูล มากำหนดตามเงื่อนไขที่จะใช้ในการส่ง State กลับไปยัง Component ซึ่งในโพรเจกต์นี้ไม่ได้ใช้ Reducer ในทุก Module เพราะส่วนมากจะเป็นการส่ง Response ไปยังหน้าจอบนโดยตรง เพราะสามารถนำค่า Response ไปใช้ได้เลย มีรูปแบบการเขียนโปรแกรม ดังนี้

```
const functionReducer = (
  state = initialState,
  action: AnyAction
): IFunctionState => {
  // Check condition before update state in component
  switch (action.type) {
    case functionType.condition1: {
      const state1 = doSomething()
      return {
        state1
      }
    }
    case functionType.condition2: {
      const state2 = doSomething()
      return {
        state2
      }
    }
    default:
      return state
  }
}
export default functionReducer
```

รูปภาพที่ 3. 17 การสร้าง Reducer โดย React

จากรูปที่ 3.17 เป็นการเขียน Reducer ที่มีการใช้คำสั่ง Switch เพื่อแยกกรณีตามชนิดฟังก์ชันที่ได้รับเข้ามา และมีการกำหนดการทำงานหลังจากนั้นที่แตกต่างกันตามกรณีมา ก่อนจะ return ค่ากลับเป็น state

5) APIs

เป็นส่วนที่มีการใช้ Axios เป็น Promised-based HTTP Client ที่มีความสามารถสร้าง HTTP Request จากเบราว์เซอร์ โดยในการพัฒนาครั้งนี้ได้นำมาใช้เพื่อเรียก APIs ในส่วนการเรียก APIs มี get กับ post มีรูปแบบการเขียนโปรแกรม ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

const apiName = (
  header: IHeader,
  data: any,
) => {
  return
  AxiosPromise.post<IResponseDetails<IFindAdjustFeeParti
  cipantData>>(
    url,
    {
      header,
      detail: data
    }
  )
}

export default {apiName}

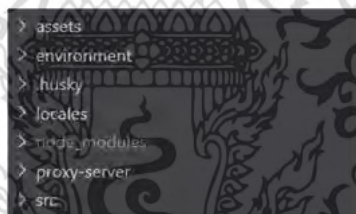
```

รูปภาพที่ 3. 18 API Service Call โดย React

จากรูปภาพที่ 3.18 เป็นการเขียนโปรแกรมเพื่อใช้เรียก APIs โดยตัวอย่างใช้คำสั่งส่งข้อมูลแบบ post มีพารามิเตอร์เป็น url และ object ที่มี header และ ใส่ข้อมูล โดยในฟังก์ชัน apiName มีการตอบกลับค่าที่ได้รับจาก Axios ชนิด Post

3.4.2 โครงสร้างการเขียนโปรแกรม

จาก Workflow ข้างต้น จะเห็นได้ว่าหลายๆ ในการเขียนเว็บแอปพลิเคชันนี้ ได้มีการใช้ React Library ซึ่งมีโฟลเดอร์ย่อย ดังต่อไปนี้



รูปภาพที่ 3. 19 โฟลเดอร์ที่สำคัญ

จากรูปภาพที่ 3.19 มีการแบ่งโฟลเดอร์ (Folder) หลักๆ ออกได้เป็น ดังต่อไปนี้

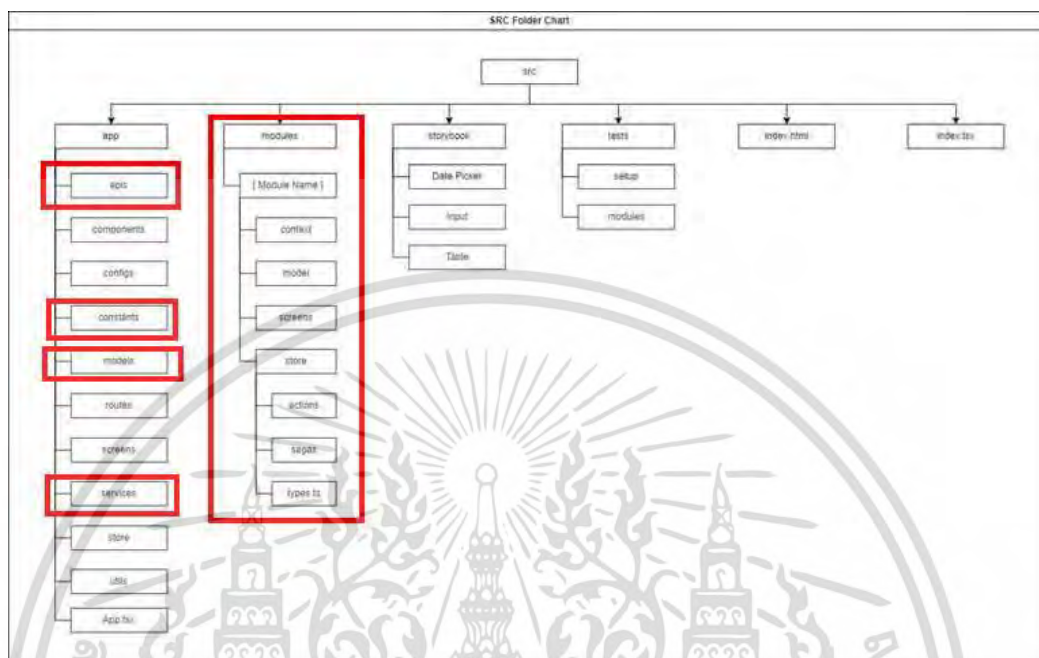
- assets เป็นโฟลเดอร์ไว้เก็บ ไฟล์ .css ฟอนต์ทั้งหมดที่ใช้ในการพัฒนา และรูปภาพต่างๆ เช่น โลโก้ เป็นต้น
- environment เป็นที่ไว้เก็บไฟล์ .env (dotenv) ซึ่ง .env มีไว้เพื่อเพิ่มความปลอดภัยในกับระบบในการเชื่อมต่อ API โดยไฟล์ .env จะเอาไว้เก็บรหัสที่มีความสำคัญ เช่น API_KEY เป็นต้น เพราะฉะนั้น .env จะถูกเพิ่มในโปรเจกต์ แต่จะไม่ถูกเพิ่มใน Source Control เนื่องจากการพัฒนาครั้งนี้มีระบบย่อยๆ ในการเรียกเยอะ จึงทำให้มีการแยกออกมาสร้าง dotenv เป็นโฟลเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- .husky เป็น package ที่มีใน npm เป็นประโยชน์สำหรับทุกๆ ครั้งที่ต้องการนำขึ้น Version Control โดย package นี้จะมีการสั่งให้ใช้คำสั่ง npm test จาก test Package โดยอัตโนมัติ
- locales มีไว้สำหรับเก็บ ข้อความ หรือ text ที่ใช้ทั้งหมดในโปรเจกต์ โดยแยกตามภาษาที่ใช้ ในโปรเจกต์ มี 2 ภาษา ได้แก่ อังกฤษ และไทย อีกทั้งยังง่ายต่อการเพิ่มภาษาในอนาคต
- node_modules เป็นโฟลเดอร์ที่ถูกสร้างมาหลังจากใช้คำสั่ง npm โดยเป็น Global Library โดยจะถูกสร้างทุกครั้งที่มีการใช้คำสั่ง npm
- proxy-server มีไว้เพื่อในบางกรณีที่มีการเก็บของ Client และ Backend ไว้ต่าง Server กัน Proxy จึงมีไว้ ไว้เป็นทางเชื่อมต่อ เปรียบง่ายๆ เป็นท่อที่เอาไว้ใช้ระหว่าง Client และ Server เมื่อมีการเรียกการร้องขอต่างๆ เมื่อมีการขอ Request ไปที่ Server หรือมีการรับ Response กลับจาก Server
- src เป็นหัวใจหลักของ React Application เริ่มดำเนินการทำงานโดย webpack ที่ถูกเขียนขึ้นในโฟลเดอร์ src ถูกสร้างขึ้นตั้งแต่ใช้คำสั่ง npx create-react-app [name] ซึ่งเป็นหนึ่งในคำสั่งเรียกใช้ package ที่ถูกพัฒนาโดยนักพัฒนาโดยทีม react อีกทั้งยังมีโฟลเดอร์ src จะมีโฟลเดอร์ย่อยๆ ที่สำคัญ โดยโฟลเดอร์ย่อยภายในโฟลเดอร์ src จะมีการสร้างที่ต่างกันไปตามความถนัดของนักพัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่กล่าวถึงโครงสร้างโฟลเดอร์ของโปรเจกต์ไป เนื่องจาก src เป็นโฟลเดอร์หลัก จึงจะอธิบายรายละเอียดภายในโฟลเดอร์ src โดยภาพแผนภูมิ ดังต่อไปนี้



รูปภาพที่ 3. 20 แผนภูมิโครงสร้างโฟลเดอร์ src

จากรูปภาพแผนภูมิโครงสร้างโฟลเดอร์ src ด้านบน จะมีโฟลเดอร์หลักๆ ที่ได้เข้าไปเพิ่มตามกรอบสีแดง ดังนี้

1) app เป็นโฟลเดอร์มีไว้เก็บโฟลเดอร์ย่อยหลักๆ เก็บ service เก็บ API สำหรับการเขียนแอป โดยสามารถเขียนได้กำหนดแยกย่อยโฟลเดอร์ได้ตามความถนัดของนักพัฒนา

- apis เป็นโฟลเดอร์มีไว้สำหรับเก็บเส้นของ API เพื่อเชื่อมต่อพูดคุยขอข้อมูลจากเส้นต่าง ๆ ซึ่งในชื่อไว้ตามโมดูล ในนี้มีทั้ง API ที่เรียกสำหรับ Lookup และหน้าปกติ

2) modules เป็นโฟลเดอร์ที่มีเนื่องจากงานที่พัฒนามีการแยกเป็นหมวดหมู่ จะเรียกว่า โมดูล (Module) โดยในโปรเจกต์นี้จะมีโมดูลย่อยลงไปอีก เนื่องจากเป็นระบบใหญ่ และมีฟังก์ชันเยอะ และมี Module ย่อยตามชื่อโดยในโครงสร้างเขียนว่า [Module Name] เป็นโฟลเดอร์ที่จะตั้งชื่อตามโมดูลย่อยที่อยู่ย่อยลงไปโมดูล โมดูลย่อยชื่ออะไรก็จะเป็นชื่อแทนใน [] ภายในโมดูลจะมีโฟลเดอร์ย่อย ดังต่อไปนี้

- Context เป็นโฟลเดอร์มีไว้เขียนดึงพวก List ต่างๆที่ต้องเรียก API เพื่อที่พามาเข้าหน้าจอที่ใช้งาน จะได้มีการโหลดไว้อ่อน ป้องกันไม่ให้หน้าจอแสดงผล Render ก่อน ที่ List จะดึงข้อมูลเสร็จเรียบร้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

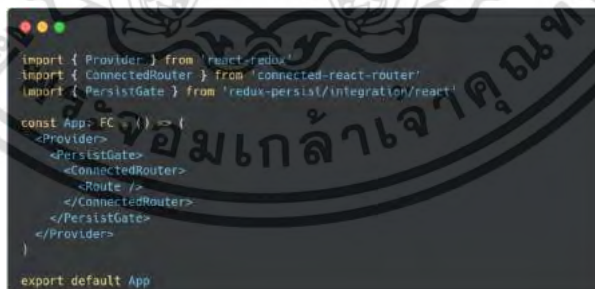
- Model เป็นโพลเดอร์มีไว้เก็บชนิดตัวแปรที่สร้างเอง ในที่นี้เก็บเป็น Object โดยสร้าง Object Type ที่เอามาไว้ใช้กำหนด Props ภายในหน้าจอนั้น รวมถึงกำหนด Object Type ที่ใช้ในการขอ Request ไปยัง API
- Screens เป็นโพลเดอร์ไว้เก็บไฟล์หน้าจอไฟล์ .tsx เนื่องจากในโพลเดอร์นี้จะแยกตามหน้าจอ
- Store เป็นโพลเดอร์เอาไว้เก็บที่ไว้เชื่อมกับ Middleware ในที่นี้เก็บ ไฟล์ Action และ Saga
- Actions เป็นโพลเดอร์เอาไว้ใช้เชื่อมต่อระหว่างหน้าจอที่เก็บไว้ใน Screens และ Sagas ทุกครั้งที่ต้องใช้งานถึง Middleware จำเป็นต้องเขียนผ่าน Action ภายใน Action จะมีการกำหนดชนิดตัวแปรที่ส่งไป Sagas หรือฟังก์ชันที่ตอบกลับจาก Saga โดย Sagas เป็นโพลเดอร์ที่เอาไว้เชื่อมต่อระหว่าง Middleware ออกไปยัง API ที่เขียนเชื่อมต่อโดยใช้ Axios และ types.ts เป็นไฟล์เอาไว้เก็บตัวแปรที่ถูกกำหนดไว้ใช้เชื่อมต่อระหว่าง Action และ Saga

3.4.3 เทคนิคการเขียนโปรแกรมที่ใช้

การเขียนเว็บ หรือเว็บแอปพลิเคชัน จะมีการเขียนที่แตกต่างกัน ขึ้นอยู่กับชนิดของโปรเจกต์ และเทคนิคที่นักพัฒนาได้นำมาใช้ อีกทั้งปัจจุบันมี Package ใหม่ของ npm ที่ออกมาช่วยในการเขียนมากมาย ดังนั้นจะแสดงตัวอย่างรูปแบบการเขียนโปรแกรมบางส่วนของโครงสร้างโมดูลในโปรเจกต์ ดังนี้

1) เทคนิคการใช้ Route

สร้างเส้นทางใช้ Component ชื่อ Route จาก Package ชื่อ react-router-dom โดยสร้างไฟล์ .tsx หนึ่งเป็น Function Component มีรูปแบบการเขียนโปรแกรมดังต่อไปนี้



```

import { Provider } from 'react-redux'
import { ConnectedRouter } from 'connected-react-router'
import { PersistGate } from 'redux-persist/integration/react'

const App: FC = () => {
  <Provider>
    <PersistGate>
      <ConnectedRouter>
        <Route />
      </ConnectedRouter>
    </PersistGate>
  </Provider>
}

export default App

```

รูปภาพที่ 3. 21 โปรแกรมหน้า App.tsx

จากรูปที่ 3.21 เป็นการเขียนโปรแกรมในหน้า App.tsx ที่เขียนมีการใช้ Route Tag ในการเชื่อมต่อไปยังหน้าต่างๆ ในโปรเจกต์ที่สร้าง ซึ่ง Route ก่อนจะเข้า Route จะมีการใช้ PersistGate เพื่อลดการเรนเดอร์ของ UI รอให้ Redux ได้รับสถานะก่อน และ Provider เพื่ออนุญาตให้ใช้ Nested Component ได้ และ ConnectedRouter ต้องมีการใช้งานคู่กับ Provider

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

const Route: FC = () => {
  return (
    <MuiProvider>
      <Switch>
        <Route
          exact
          path="/"
          component={() => <Redirect to={contextRoot} />}
        />
        <Route
          exact
          path={contextRoot}
          component={LoginScreen}
        />
        <Route path={path} component={ModuleRoute} />
      </Switch>
    </MuiProvider>
  )
}
export default Route

```

รูปภาพที่ 3. 22 โปรแกรมหน้า Route.tsx

จากรูปภาพที่ 3.22 เป็นการเขียนโปรแกรมส่วนเชื่อมต่อกับ Route Tag จากรูปภาพที่ 3.21 เป็นการดัดแปลง Router เนื่องจากโปรเจกต์มีการแยก Route ไปยังหลายโมดูล ซึ่งในรูปแบบต่อไปยัง ModuleRoute โดยใช้ Route Tag

```

const ModuleRoute: FC = () => {
  return (
    <Switch>
      <Route
        path={path as string}
        component={component}
      />
      <Route component={PageNotFound} />
    </Switch>
  )
}
export default ModuleRoute

```

รูปภาพที่ 3. 23 โปรแกรมหน้า ModuleRoute.tsx

จากรูปภาพที่ 3.23 เป็นการเขียนโปรแกรมของ ModuleRoute ใช้ Function Component โดยจะมีการใช้เงื่อนไข Switch เพื่อตรวจสอบการหาหน้ามาสามารถเชื่อมต่อไปยังหน้าดังกล่าวได้ไหม ถ้าเชื่อมต่อไม่ได้จะ return Page Not Found Page

จากทั้งหมดการใช้ Route Component โดยเขียนเริ่มจาก App.tsx ของโปรเจกต์ Route จะใช้เพื่อเปลี่ยนเส้นทางของหน้าจอไปยังหน้าที่ต้องการจะใช้ โดยใน Route Tag สามารถเปลี่ยน Component เป็นชื่อไฟล์ .tsx ได้

2) เทคนิคการเขียนเมนูย่อย

เนื่องจากการสร้างโมดูลเมนูจำเป็นต้องมีการเช็ค Permission โดยเทียบกับ SystemCode ที่เก็บไว้เป็นความลับ ส่วนใหญ่มักจะเก็บไว้ในไฟล์ .env มีการใช้ lazy จาก Package ชื่อว่า React มีไว้เพื่อรับ return ที่เป็น เพื่อทำการ Code Splitting เพื่อแยก component ต่างๆออกเป็นอิสระต่อกัน เมื่อทำ lazy อยู่จะแสดงผลเป็นจอสีขาว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import { lazy } from 'react'

const ModuleMapping = {
  key: 'SECOND_SUB_MODULE',
  component: lazy(() => import('@Modules/ModuleName/screens/ModuleName'))
}

export default ModuleMapping

```

รูปภาพที่ 3. 24 โปรแกรมหน้า ModuleMapping.ts

จากรูปที่ 3.24 เป็นการเขียนโปรแกรมอ้างอิงส่วนที่กล่าวไป โดยใช้ lazy โดยฟังก์ชันนี้ กำหนดค่าที่ Mapping เป็น Key และ Component

```

const check = (allows: Array<SystemCode> | 'ALL', menu: IMenu | IMenuItem) => {
  if (allows === 'ALL' || allows.includes(systemName as SystemCode)) {
    return menu
  }
  return undefined
}

const permission = (allows: Array<SystemCode> | 'ALL') => (
  menu: IMenu
): IMenu | undefined => {
  return check(allows, menu) as IMenu
}

export const ModuleMenu = permission(['SYSTEMCODE'])({
  path: '/module-path',
  menuName: 'Module',
  menus: createMenuList([
    createMenuItem(['CSD'])({
      createMenuItem(['CSD'])({
        code: Module.SUB_MODULE,
        key: 'SUB_MODULE',
        menuName: 'module:menus:subModule',
        path: '/sub-module-path',
        context(ComponentKey: 'SUB_MODULE',
          subMenu: createMenuList([
            createMenuItem(['CSD'])({
              key: 'SECOND_SUB_MODULE',
              menuName: string,
              path: '/second-sub-module',
              lazyComponentKey: ModuleMapping.SECOND_SUB_MODULE.key,
              permission: ['FUNCTION_CODE'],
              permissionType: 'FUNC'
            })
          ])
        })
      ])
    ])
  ])
}

```

รูปภาพที่ 3. 25 โปรแกรมหน้า ModuleMapping.ts

จากรูปภาพที่ 3.25 จะเห็นได้ว่าการตรวจสอบว่าสามารถเข้าใช้ได้หรือไม่ เพื่อแบ่งตามระบบที่เขียนขึ้นมา ในการเขียนเมนูย่อยมีการ Item Object มาช่วย เพราะในการเขียนโมดูลมีมากกว่า 1 โมดูลย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) เทคนิคการเขียนเชื่อมต่อ API

จะใช้ไฟล์ Saga ใช้ในการเรียกใช้ Services แล้วทำการต่อกับเส้นของ API ดังนี้

```
import { AnyAction } from 'redux'
import { loaderType } from '@Store/types'

const startLoading = (): AnyAction => {
  return {
    type: loaderType.REQUEST_START
  }
}

const stopLoading = (): AnyAction => {
  return {
    type: loaderType.REQUEST_STOP
  }
}

export default {
  startLoading,
  stopLoading
}
```

รูปภาพที่ 3. 26 โปรแกรมหน้า LoaderAction.ts

จากรูปภาพที่ 3.26 เป็นหน้า Loader Action เอาไว้เชื่อมต่อกับภายนอกกับ Saga โดยมีชนิดตัวแปรกำหนดเพื่อเชื่อมต่อกับฟังก์ชันประเภท Generate ที่ Saga

```
import { takeLatest, put } from 'redux-saga/effects'
import { loaderType } from '@Store/types'

// start loading action
function* startLoading() {
  yield put({ type: loaderType.START })
}

// stop loading action
function* stopLoading() {
  yield put({ type: loaderType.STOP })
}

// dialog saga
const dialogSaga = [
  takeLatest(loaderType.REQUEST_START, startLoading),
  takeLatest(loaderType.REQUEST_STOP, stopLoading)
]

export default dialogSaga
```

รูปภาพที่ 3. 27 โปรแกรมหน้า LoaderSaga.ts

จากรูปภาพที่ 3.27 เป็นหน้า Loader Saga มีเอาไว้เพื่อกำหนดให้ฟังก์ชัน Generate สามารถเริ่มทำงานหลังมีการกระทำเข้ามา มีการใช้ ของ Redux Saga

- select เป็นคำสั่งใช้เรียกใช้ state ของ Reducer ที่ค่าเปลี่ยนแปลงไม่ได้
- put เป็นคำสั่งให้ทำงานที่กำหนด
- call เป็นคำสั่งเรียกใช้ Service ที่แยกกันตามโพลเดอร์ เพื่อเรียกไว้เรียก API สามารถเรียกได้ครั้งละ 1 API
- all เป็นคำสั่งเรียกใช้ Service ที่แยกกันตามโพลเดอร์ เพื่อเรียกไว้เรียก api สามารถเรียกได้ครั้งละหลายๆ API โดยใช้รูปแบบ Object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- takeLatest เป็นคำสั่งให้ยกเลิกการกระทำก่อนหน้าทั้งหมดหากมีรูปแบบที่ประกาศไว้ตรงกันกับใน Action ที่ถูกเรียกใช้จากหน้าจอ เช่น เมื่อมีการกดปุ่ม Create และกดปุ่ม Search การกระทำที่เกิดขึ้นจะเป็น Search

```

import { call, put, takeLatest } from "redux-saga/effects"

function* onSearch({
  payload: { state, pagination, onSuccess },
}: IDIFMessageInquirySagaRequest) {
  try {
    yield put(LoaderAction.startLoading())

    const header: IHeader = {
      channel,
      functionCode: 'PBCTCODE',
    }

    const condition: ISearchRequest = {
      field: state.value
    }

    const response = (yield call(
      ModuleService.moduleSearch,
      header,
      condition,
      pagination
    )) as ISearchData

    yield put(LoaderAction.stopLoading())
    onSuccess(response)
  } catch (err) {
    yield put(LoaderAction.stopLoading())
    yield put(console.log('Error', err))
  }
}

const SagaModuleSaga = takeLatest([subModule.REQUEST_SEARCH], onSearch)
export default IDIFMessageInquirySaga

```

รูปภาพที่ 3. 28 โปรแกรมหน้า SearchSaga.ts

จากรูปที่ 3.28 เป็นการเขียนโปรแกรม Saga ของการทำงาน Search ที่มีการใช้คำสั่ง put และ call โดยมีการทำงานตั้งแต่เริ่มต้น Start ที่เรียกใช้จากรูปภาพที่ 3.27 และระหว่างคำสั่ง put ที่มีพารามิเตอร์ LoadAction.startLoading() กับ LoadAction.stopLoading() มีการกำหนด Header Condition และค่ามาจากการส่ง Request โดยเก็บไว้ใน response และสุดท้ายมีการใช้ takeLatest

```

export interface IHeader {
  action?: string
  channel: string
  functionCode: string
}

```

รูปภาพที่ 3. 29 ชนิดตัวแปร Interface ของ IHeader

จากรูปภาพเป็นการกำหนดชนิด Interface ของ IHeader ใช้อ้างอิงก่อนส่งไปขอ Request โดยเทียบกับตัวแปรชื่อ header ในรูปภาพที่ 3.28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

const moduleSearch = async (
  header: IHeader,
  condition: ISearchRequest,
  paging: IPagination
): Promise<ISearchRequestData> => {
  const response = await moduleAPI.moduleSearch(
    header,
    condition,
    paging
  )

  return {
    details: response.details,
    paging: response.paging
  }
}

export default {moduleSearch}

```

รูปภาพที่ 3. 30 โปรแกรมหน้า ModuleService.ts

จากรูปภาพที่ 3.30 moduleSearch มีการจัดจ้งหะการทำงานโดยใช้ async และ await เนื่องจากการเรียกใช้ API อาจจะใช้เวลานาน ทำให้อาจจะมีการทำงานอื่นก่อน เลยต้องใช้ await เพื่อรอให้คำสั่งนี้เสร็จก่อน ค่อยไปทำลำดับถัดไป

```

const moduleSearch = async (
  header: IHeader,
  condition: ISearchRequest,
  paging: IPagination
): Promise<ISearchRequestData> => {
  const response = await moduleAPI.moduleSearch(
    header,
    condition,
    paging
  )

  return HttpUtils.post<ISearchData>(
    `${apiEndpoint}/web/module/moduleSearch`,
    {
      header,
      details: {
        ...condition,
        paging
      }
    }
  )
}

export default {moduleSearch}

```

รูปภาพที่ 3. 31 โปรแกรมหน้า ModuleAPI.ts

จากรูปภาพที่ 3.31 จากการเรียกใช้ Service จาก Saga โดยส่งเงื่อนไขมาเป็น Condition, Header ที่เก็บรหัสฟังก์ชันที่ไม่ซ้ำไว้ให้ของแต่ละฟังก์ชันการทำงาน และ Paging เอาไว้เก็บ จำนวนแถวในแต่ละหน้าที่ใช้แสดง กำหนดหน้าที่เท่าไร โดยนำทั้งสามมารวมเป็นก้อนข้อมูลที่ส่งไป เนื่องจากใช้ .post มีการเรียกใช้ HttpUtils ที่มีการเขียน post และ get ที่รู้จักกันว่าเป็น Http Request โดย Get สำหรับร้องขอไปที่ API โดยตรงโดยไม่จำเป็นต้องใส่เงื่อนไข Request อะไร เข้าไปก็สามารถได้ Response กลับมา และ Post สำหรับร้องขอไปที่ API โดยตรงโดยต้องใส่เงื่อนไข Request จึงได้ Response กลับมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

const post = async <T>({
  url: string,
  data?: unknown,
  config?: IConfigAxios
}): Promise<T> => {
  const response = await callApi<T>('POST', url, data, config)
  if (response.status === 200) {
    return response.data
  }
  return {} as T
}

const get = async <T>(url: string, config?: IConfigAxios): Promise<T> => {
  const response = await callApi<T>('GET', url, null, config)
  if (response.status === 200) {
    return response.data
  }
  return {} as T
}

export default {post, get}

```

รูปภาพที่ 3. 32 โปรแกรมหน้า HttpUtils.ts ฟังก์ชัน Get และ Post

จากรูปภาพที่ 3.32 ที่จะมีการเช็คเมื่อมีการเรียกใช้แล้วได้ HTTP Status Code กลับมาเป็น 200 จึงจะสามารถส่ง response กลับมาได้

ในการเขียน API Get กับ Post จะมีการส่ง เรียก API ผ่านฟังก์ชัน Call API ที่มีการนำ axios มาใช้จัดการกับการดึง และบันทึกข้อมูลของการใช้ API ที่ติดตั้งเข้ามาในโปรเจกต์ เนื่องจากเป็น Library ของ JavaScript

```

const callApi = async <T>({
  method: Method,
  url: string,
  payload: any,
  config: IConfigAxios = {
    ignoreTokens: false
  }
}): Promise<AxiosResponse<T>> => {
  const headers: any = getHeaders(config)
  const options = {
    method,
    headers,
    responseType: config.responseType,
    data: payload ? JSON.stringify(payload) : undefined,
    url,
    validateStatus
  } as AxiosRequestConfig
  return axios(options)
}

```

รูปภาพที่ 3. 33 โปรแกรมหน้า HttpUtils.ts ฟังก์ชัน callApi

จากรูปภาพที่ 3.33 เป็นฟังก์ชัน callApi ที่นำเอา Axios มาจัดการกับการดึงข้อมูล โดย IConfig ก็เป็นชนิดตัวแปรที่ดึงมาจาก Axios จากที่เห็นบรรทัด data ในตัวแปร option จะเช็คกว่าชนิดข้อมูลที่ส่งไป เป็น payload หรือเปล่า ถ้าเป็น payload ให้แปลงเป็น JSON ถ้าไม่ให้ส่งไปแบบ Undefined โดยใน option มีการกำหนดชนิดเป็น AxiosRequestConfig ที่มาจาก Axios Library และมีชนิด Interface เป็น AxiosResponse ตามรูปภาพที่ 3.34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

export interface AxiosResponse<T = any> {
  data: T;
  status: number;
  statusText: string;
  headers: any;
  config: AxiosRequestConfig;
  request?: any;
}

```

รูปภาพที่ 3. 34 ชนิดตัวแปร Interface ของ Axios Response

จากรูปภาพที่ 3.34 เป็นการกำหนดชนิด Interface เป็น AxiosResponse อ้างอิงค่าที่รับมา ฟังก์ชัน callAPI ตามรูปภาพที่ 3.33

4) เทคนิคการเขียน React Hook

มีไว้เพื่อใช้กับ Function Component มีการสร้างที่ตาม React Life Cycle โดย React Hook อนุญาตให้มีการเข้าถึง State และพีเจอร์ React อื่นๆ เมื่อใช้ Function Component

- useState สำหรับการใช้งาน state ต่างๆ ทั้งจากนอก และใน Component จำเป็นต้องมีการเรียกใช้ useState โดยจะประกาศตัวแปรที่ใช้เป็น 2 ค่า คือค่าที่ใช้เป็น State กับค่าที่นำไปแก้ไขจะขึ้นต้นด้วย set พร้อมทั้งกำหนดค่าเริ่มต้นใน () ส่วนของ TypeScript จำเป็นต้องมีการประกาศชนิดของตัวแปร

```

const [viewJSONData, setViewJSONData] = useState<{
  open: boolean
  data: string | null
}>({
  open: false,
  data: null
})

```

รูปภาพที่ 3. 35 ตัวอย่าง useState

จากรูปภาพที่ 3.35 เป็นตัวอย่างการประกาศใช้ State แบบใช้ Object ในการกำหนดชนิดตัวแปร โดยใน Object มี 2 ชนิดคือ open เป็น Boolean และ data เป็น String โดยมีค่าเริ่มต้นเป็น Object โดย open เป็น false และ data เป็น null

- useEffect ใช้ในกรณีที่เกิดผลข้างเคียงต่างๆ เช่น เมื่อมีการแก้ไขข้อมูลส่งมาใหม่ทำให้จำเป็นต้องมีการดึงข้อมูลใหม่ ในการประกาศใช้งานจะประกาศเป็น `useEffect(<function>, <dependency>)` function เอาไว้เขียนการทำงานต่างๆที่จะเกิดทำเมื่อเกิดผลข้างเคียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

const [pagination, setPagination] = useState<{
  pageNo: number,
  rowPerPage: number
}>({
  pageNo: 0,
  rowPerPage: 0
})

useEffect(() => {
  if (dataList) {
    setPagination({ ...pagination, rowCount:
dataList.length })
  } else {
    setPagination(defaultPagng)
  }
}, [dataList])

```

รูปภาพที่ 3. 36 ตัวอย่าง useEffect

จากรูปภาพที่ 3.36 เป็นการใช้ useEffect ในการตรวจสอบ dataList ที่ดึงค่ามาจาก API โดยถ้ามีการเปลี่ยนแปลงของข้อมูลชื่อ dataList จะทำการตั้ง State ที่ชื่อ pagination ใหม่ โดยแก้ไขค่าของ setPagination ให้มีค่าทุกอย่างคงเดิม แต่เปลี่ยนค่าจำนวนแถวตามความกว้างของ dataList

- useContext ตามหลักที่ในการเขียน React เป็นไปได้ที่หากเขียนไปหลายๆ Component อาจเกิดการทับซ้อนของ Component มาก จนทำให้เกิดเป็น Nested Component ถ้าหากต้องการที่จะใช้ State ใน Component แรกๆ จำเป็นต้องมีการส่ง State ผ่าน Component ในการส่งข้ามระหว่าง Component เรียกว่า Props จำนวนมากเพื่อให้ได้ใช้ State นั้น ในการแก้ไขปัญหาเรียกใช้ State จากด้านบนสุดของ Stack จึงมี useContext เพื่อประกาศ State แบบ Global ที่ทุกๆ Component สามารถเรียกใช้ได้โดยใช้แค่คำสั่ง

```

// Context.ts
export const BookClosingContext = React.createContext(values)
const BookClosingContext:FC = () => {
  const [bookClosingTypeList, setBookClosingTypeList] = useState<
Array<IBookClosingTypeData>
>([])
  ...
}

// Component.tsx
const Component:FC = () => {
  const { txnStatusIdList } = useContext(BookClosingContext)
  ...
}

```

รูปภาพที่ 3. 37 ตัวอย่าง useContext

จากรูปภาพที่ 3.37 มีการประกาศ BookClosingContext เอาไว้เก็บ State ต่างๆที่ต้องการเก็บเป็น Global ใช้คำสั่ง React.createContext โดยในตัวอย่างมีการประกาศ bookClosingList

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ useState ใน Function Component ชื่อ bookClosingContext ในการเรียกใช้ useContext สามารถเรียกใช้ได้ในไฟล์อื่นๆ โดยใช้คำสั่งเป็น useContext

- useCallback มีไว้ตรวจสอบการเปลี่ยนแปลงของฟังก์ชันที่ถูกเรียกใช้งาน เพื่อตัดปัญหาไม่ให้มีการ Render ซ้ำทุกครั้งเมื่อไม่มีการเปลี่ยนแปลง โดยค่าที่ส่งกลับของ useCallback จะส่งกลับเป็นฟังก์ชัน

```
const onChangeBookClosingDateTo = useCallback(
  (date: MaterialUIPickersDate) => {
    setValue('bookClosingDateTo', date)
  }, []
)

return (
  <div>
    <KeyboardDatePicker
      onAccept={onChangeBookClosingDateTo}
      onChange={onChangeBookClosingDateTo}
    />
  </div>
)
```

รูปภาพที่ 3. 38 ตัวอย่าง useCallback

จากรูปภาพที่ 3.38 มีการเรียกใช้ useCallback ในการตรวจสอบว่าตัวเลือกวันเดือนปีมีการเปลี่ยนแปลงหรือไม่ ถ้ามีการเปลี่ยนแปลงเกิดขึ้นให้ตั้งค่าตัวแปรที่ชื่อ bookClosingDateTo เป็นค่าใหม่ที่ได้รับมา

- useMemo มีไว้ตรวจสอบการเปลี่ยนแปลงเหมือนกับ useCallback แต่ค่าที่ส่งกลับมาจะเป็นรูปแบบค่า ไม่ใช่ฟังก์ชัน และจะมีการ Render ใหม่ก็ต่อเมื่อมีการเปลี่ยนแปลงค่าในตัวที่ประกาศของ useMemo เท่านั้น

```
const column = useMemo(
  (): ColumnType<APIResponseData> => [
    {
      title: 'column name',
      dataIndex: 'apiValue'
    },
    {...}], []
)

return (
  <Table<APIResponseData>
    columns = {column}
    ...
  />
)
```

รูปภาพที่ 3. 39 ตัวอย่าง useMemo

จากรูปภาพที่ 3.39 เป็นการใช้งาน useMemo ในการจดจำแถวของตาราง มีค่าที่ส่งกลับเป็นรูปแบบ Array ใน Table Tag เมื่อมีการเปลี่ยนแปลงค่าเมื่อไหร่จะมีการ Render ใหม่อีกครั้ง หากไม่มีการเปลี่ยนแปลงค่าก็จะใช้ค่าเดิมที่มีอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) เทคนิคการเขียนใช้ React Hook Form

ในการพัฒนาครั้งนี้ได้นำ React Hook Form มาช่วยเนื่องจากโพรเจกต์นี้มีการใช้งานในรูปแบบฟอร์มที่เยอะ ทำให้การนำ Library มาช่วยเพื่อตรวจสอบความถูกต้องได้ง่ายขึ้น และลดจำนวนการเขียนโปรแกรมให้น้อยลงได้ React Hook Form ที่นำมาใช้งานมีทั้งหมด ดังนี้

```
const { control } = useForm<IProps>
({defaultValues})
return (
  <>
    { /* Field */ }
    <Grid item sm={3}>
      <Controller
        control={control}
        name='fileType'
        render={({ field }) => (
          <Select {...field}>
            <MenuItem value='PDF' key='PDF'>
              PDF
            </MenuItem>
          </Select>
        )
      }
    </Grid>
  </>
)
```

รูปภาพที่ 3. 40 ตัวอย่าง React Hook Form Control

จากรูปภาพที่ 3.40 เป็นการใช้ control เพื่อควบคุม Field ที่สร้างขึ้น ส่วนมากมักใช้กับ Field ที่จำเป็นต้องกรอกเพื่อนำไปใช้ยืนยันฟอร์มในการค้นหาต่อ ตัวอย่างในการควบคุม เช่น สามารถนำมากำหนดค่าที่แสดงข้อผิดพลาดภายใต้ Field ได้ มักใช้ควบคู่กับ Controller ที่ดึงมาจาก Library ของ react-hook-form

```
const { watch } = useForm<IProps>
({defaultValues})
const watchValue = watch('value')

return (
  { /* Field */ }
  </>
)
```

รูปภาพที่ 3. 41 ตัวอย่าง React Hook Form Watch

จากรูปภาพที่ 3.41 เป็นการใช้ watch ไว้ใช้ดูค่าที่ต้องการให้เก็บการ Render ไว้ทุกครั้ง หรือก็คือหากมีการเปลี่ยนแปลงค่าก็จะสามารถเก็บค่านี้ไว้เพื่อนำไปใช้ต่อได้ โดย watch ใน React Hook Form สามารถแก้ปัญหาการตรวจจับทั้งฟอร์ม เมื่อมีการเปลี่ยนแปลงค่าเป็นให้สามารถตรวจจับได้เฉพาะบาง Field ที่ต้องการให้ตรวจจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

const { setValue } = useForm<IProps>({defaultValues})

const onChangeDate = useCallback((date: MaterialUIPickersDate) => {
  setValue('date', date)
}, [])

return (
  < /* Field */ >
</>
)

```

รูปภาพที่ 3. 42 ตัวอย่าง React Hook Form Set Value

จากรูปภาพที่ 3.42 เป็นการใช้ setValue ใช้เพื่อกำหนดค่าที่ต้องการกำหนดโดย Field แต่
 ละตัวจะสามารถกำหนดค่าได้ โดยใช้ setValue จากตัวอย่างมีการกำหนดค่า date ลงไปในตัวแปร
 ชื่อ date เมื่อมีการเปลี่ยนแปลงค่านี้จะถูกแทนที่ค่าเดิมในตัวแปรที่ชื่อ date

```

const { handleSubmit } = useForm<IProps>({defaultValues})

const onSubmit: SubmitHandler<IProps> = (state) => {
  dispatch(
    action_dispatch(state)
  )
}

return (
  < /* Field */ >
  <Button onClick=
  {handleSubmit(onSearch)}>Search</Button>
</>
)

```

รูปภาพที่ 3. 43 ตัวอย่าง React Hook Form Handle Submit

จากรูปภาพที่ 3.43 เป็นการใช้ handleSubmit มีไว้เพื่อระบุงการกระทำในการยืนยัน
 แบบฟอร์ม handleSubmit จะส่งค่าของ Field ทุกตัวที่อยู่ในแบบฟอร์มไป หากเป็นปุ่มที่ต้องการให้
 ยืนยันแบบฟอร์มจำเป็นต้องมี handleSubmit กับ SubmitHandler

```

const { reset } = useForm<IProps>
({defaultValues})

const onClear = () => {
  reset(defaultValues)
}

return (
  < /* Field */ >
  <Button onClick={onClear}>Clear</Button>
</>
)

```

รูปภาพที่ 3. 44 ตัวอย่าง React Hook Form Reset

จากรูปภาพที่ 3.44 เป็นการใช้ reset เป็นการระบุงค่าที่ต้องการจะให้ป็น จากตัวอย่างจะ
 เอกสารนี้เป็เห็นได้ว่านำไปใช้ระบุงการล้างข้อมูลทุกค่าที่อยู่ในแบบฟอร์มเมื่อกดปุ่ม Clear ดังนั้นเมื่อมีการกค
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปุ่มจะล้างข้อมูลให้กลับไปเป็นตาม Object ที่กำหนดเป็นชื่อ defaultValue เอาไว้ นอกจากการล้างข้อมูลแล้ว ยังสามารถกำหนดข้อมูลลงไปแบบฟอร์มเป็นแบบอื่นได้ตามสถานการณ์ที่ต้องการ

6) เทคนิคการเขียนใช้ Material User Interface

การเขียนหน้าจอแสดงผลจะเขียนโดยใช้ชื่อไฟล์ในเขียนอยู่ที่ @Modules/ModuleName/screens/ModuleName.tsx ในหน้านี้จะมีการใช้ React Hook เพื่อมาจัดการ state และช่วยให้ทำงานได้ดีขึ้น โดยเทคนิคการแสดงวันที่จะใช้ มีการใช้ useForm ในการช่วยจัดการแบบฟอร์ม เนื่องจากโปรเจกต์นี้ใช้แบบฟอร์มเยอะ จึงนำมาช่วยจัดการ Field ต่าง ๆ

```

import { Grid } from '@material-ui/core'
import {
  MuiPickersUtilsProvider,
  KeyboardDatePicker
} from '@material-ui/pickers'

export const ModuleName: FC<Props> = () => {
  const { t } = useTranslation('Namespace.Module')
  const { control } = useForm({
    initialValues: { date: null },
  })

  const onDateChange = (date, value) => {
    setDate(date)
    setInputValue(value)
  }

  const dateFormatter = (str) => {
    return str
  }

  return (
    <Grid>
      <Grid item sm={12}>
        <Grid container spacing={1} direction='row'>
          <Grid item sm={12} className='text-right'>
            <InputLabel className='text-input-label' required>
              {t('text')}
            </InputLabel>
          </Grid>
          <Grid item sm={12}>
            <Controller
              control={control}
              name='date'
              render={({ field, fieldState }) => (
                <MuiPickersUtilsProvider>
                  <KeyboardDatePicker
                    { ...field }
                    autoOk={true}
                    showTodayButton={true}
                    value={selectedDate}
                    format='YYYY-MM-DD'
                    inputValue={inputValue}
                    onChange={onDateChange}
                    timeFormat={dateFormatter}
                  />
                </MuiPickersUtilsProvider>
              )}
            />
          </Grid>
        </Grid>
      </Grid>
    </Grid>
  )
}

```

รูปภาพที่ 3. 45 โปรแกรมหน้า ModuleName.tsx

จากรูปภาพที่ 3.45 เป็นตัวอย่างการนำ KeyboardDatePicker มาใช้ ซึ่งเป็น Component ที่ถูกสร้างไว้แล้วใน Library ที่ชื่อว่า @material-ui/picker ในการนำมาใช้สามารถกำหนดค่าของ attribute ต่างๆ ต้องรอกตามเงื่อนไขของ Library ที่ใช้งานว่าจำเป็นต้องมีค่าอะไรบ้าง โดยประกาศภายใต้ return ของ Function Component

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการดำเนินงานและการอภิปรายผล

บทนี้จะแสดงผลการดำเนินงานจากการพัฒนาเว็บแอปพลิเคชันของโปรแกรม CSD Refresh Technology โดยมีผลลัพธ์ดังต่อไปนี้

4.1 ภาพหน้าจอที่ได้รับผิดชอบในการพัฒนาของระบบ CSD

เนื่องจากระบบ CSD เป็นระบบที่มีขนาดใหญ่ จึงมีผู้ร่วมพัฒนาในส่วนของหน้าจอหลายใน ส่วนของการพัฒนา User Interface เนื่องจากระบบ CSD มีการแยกสำหรับนายทะเบียน (Registrar) และนายทะเบียนนอก (Other Registrar) อีกทั้งยังช่วยลดเวลาในการ build และลดการใช้ทรัพยากร ของเครื่อง ซึ่งระบบนี้จะใช้ CSS Library เพื่อเปลี่ยนการออกแบบให้ดูใช้งานง่ายมากขึ้น และทันสมัย ขึ้น ซึ่งยังคงเค้าโครงของเนื้อหา และข้อมูลเดิม ในหลายๆ แสดงเทียบหน้าจอรระบบเก่า และระบบ ใหม่ เพื่อให้เห็นความชัดเจนในการพัฒนา รวมถึงแต่ละโมดูลก็มีการแสดงผลหน้าจการทำงานที่ คล้ายกันเนื่องจากการทำงานเป็นฟังก์ชันเดียวกัน อาจทำให้มีคำอธิบายของหน้าจอที่คล้ายกัน ดังนี้

4.1.1 โมดูล Registrar

เป็นโมดูลหลักในระบบที่มีระบบย่อย โดยมีการพัฒนาโมดูลเปรียบเทียบกับหน้าจอเก่า ดังต่อไปนี้

1) โมดูลย่อย Attached Release Letter หน้า Report

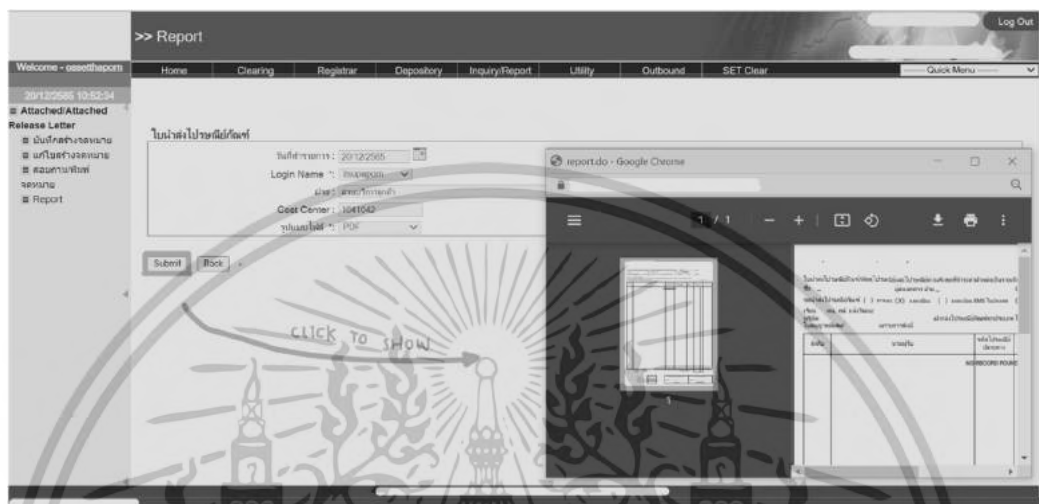
ระบบเดิม เป็นระบบที่สร้างมาก่อนหน้าที่มีการออกแบบดั้งเดิม มีหน้าจอแสดงดังต่อไปนี้



รูปภาพที่ 4. 1 On Request หน้า Attached Release Letter Report ระบบเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

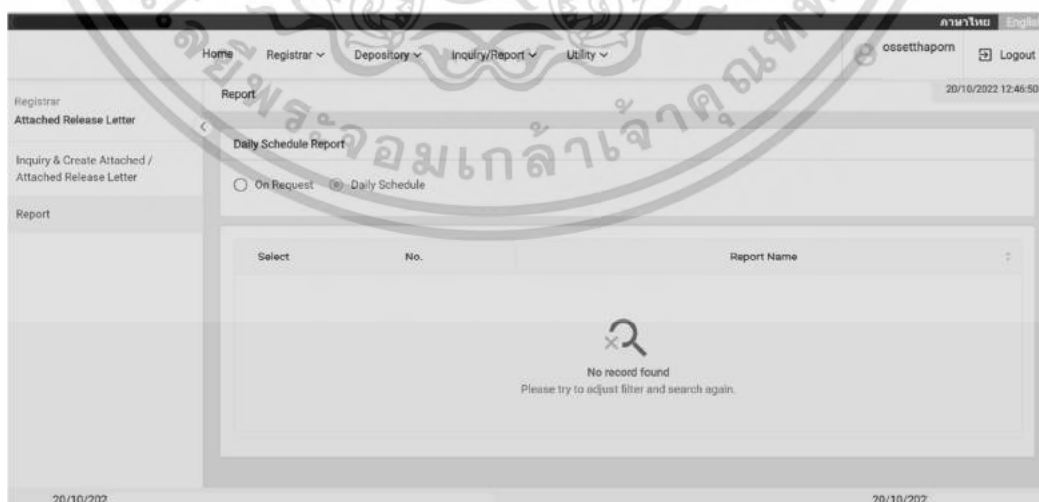
จากรูปภาพที่ 4.1 เป็นหน้าการทำงานของ Attached Release Letter Report เป็นหน้าแสดงข้อมูลที่แสดงรายการ Report ที่สามารถ Export ออกมาได้ เมื่อกดเข้าไปที่รายการที่แสดงในตารางแสดงผลค้นหา จะแสดงอีกหน้าที่มีไว้เพิ่มเงื่อนไขในการค้นหา ทำให้ทุกครั้งที่ต้องการค้นหาตามรายการจะมีการเปิดหน้าต่างใหม่ ดังนี้



รูปภาพที่ 4. 2 On Request ของ Search Conditions หน้า Attached Release Letter Report ระบบเดิม

จากรูปภาพที่ 4.2 เป็นหน้าการทำงานของ Attached Release Letter Report แสดงหลังจากกดปุ่ม Submit เป็นหน้าจอที่แยกออกมาเป็น PDF Frame

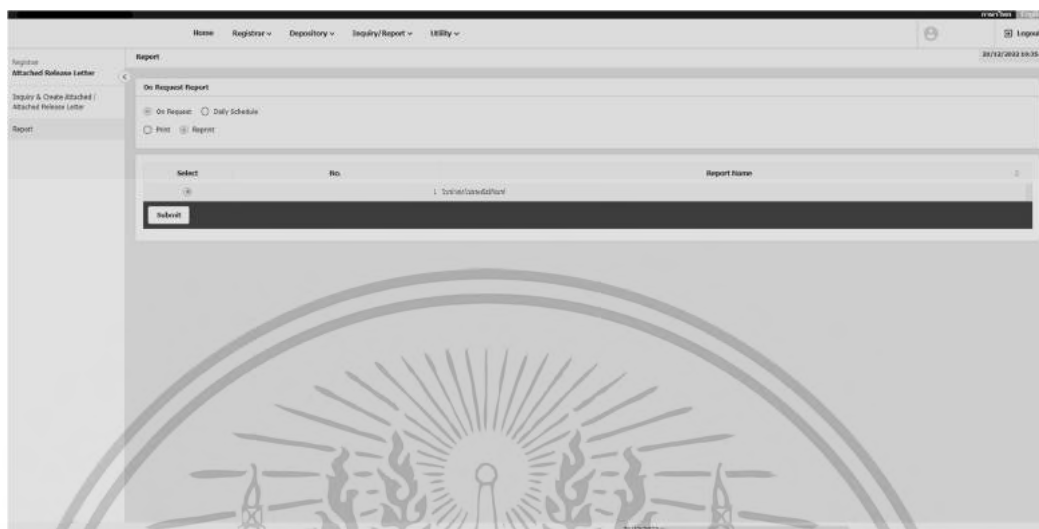
ระบบใหม่ มีการออกแบบโดยมีการจัด Layout ที่ดีขึ้น รวมหน้าค้นหา และหน้าผลลัพธ์ไว้ในหน้าเดียวกัน มีการกำหนด Modal ใหม่จาก CSS Library ที่ใช้ โดยจะแสดงรายงานออกมาเป็น PDF Modal ที่สามารถดาวน์โหลด หรือปริ้นต์ได้ มีการแสดงผลหน้าจอทั้งหมด ดังต่อไปนี้



รูปภาพที่ 4. 3 Daily Schedule หน้า Attached Release Letter Report

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปภาพที่ 4.3 เป็นหน้าที่แสดง Daily Schedule เป็นชนิดรายงานแบบที่พิมพ์ตามรอบเวลาที่กำหนดไว้ในระบบของแต่ละวัน



รูปภาพที่ 4. 4 Search Report List ของ Reprint หน้า Attached Release Letter Report

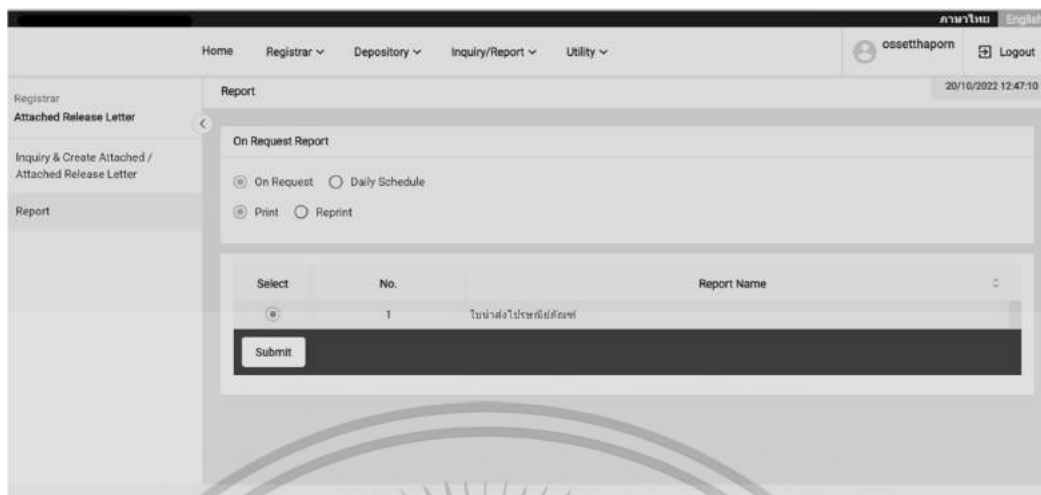
จากรูปภาพที่ 4.4 เป็นหน้าที่แสดง On Request เป็นชนิดรายงานที่ขอรายงานตามที่ต้องการ โดยมีเงื่อนไขเพื่อค้นหา ก่อนจะสามารถเปิดเป็นผลลัพธ์ของ PDF แบบ Modal ได้ อีกทั้งใน On Request ยังสามารถเลือกพิมพ์ แบบ Print ที่ต้องกรอกเงื่อนไขในการค้นหา หรือ Reprint ที่จะแสดงไฟล์ก็ต่อเมื่อมีการขอ PDF ไว้แล้ว โดยจะใส่วันที่เป็นเงื่อนไข จากรูปเป็นการเลือก Reprint



รูปภาพที่ 4. 5 Search Report List ของ Reprint Modal หน้า Attached Release Letter Report

จากรูปภาพที่ 4.5 เป็นการแสดงหน้า Modal หลังจากเลือกรายการจากรูปภาพ 4.4 โดยหลังจากแสดง Modal ให้เลือกวันที่ของรายการธุรกรรมที่ถูกสร้างไว้ และเลือกรายการที่สามารถทำการพิมพ์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



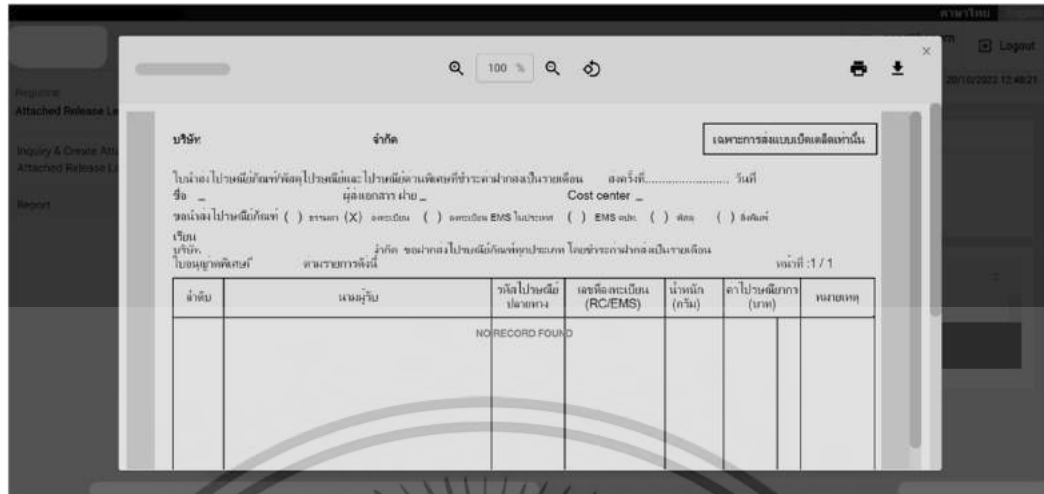
รูปภาพที่ 4. 6 Search Report List ของ Print หน้า Attached Release Letter Report
จากรูปภาพที่ 4.6 เป็นหน้าที่แสดง On Request ชนิด Print หลังจากที่ได้เลือกแล้วจะแสดงรายการด้านล่าง ที่สามารถเลือกรายการได้ และจะแสดงปุ่ม Submit หลังจากเลือกรายงานที่แสดงแล้ว



รูปภาพที่ 4. 7 Search Report List Modal ของใบนำส่งไปรษณีย์ หน้า Attached Release Letter Report

จากรูปภาพที่ 4.7 เป็นหน้าที่แสดง Modal หลังจากกดปุ่ม Submit ในหน้า 4.6 โดยจะต้องกรอกเงื่อนไขต่างๆ โดยจะมี Textbox ที่จำเป็นต้องกรอก และกดปุ่ม Submit เพื่อร้องขอรายงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 8 PDF Modal หน้า Attached Release Letter Report

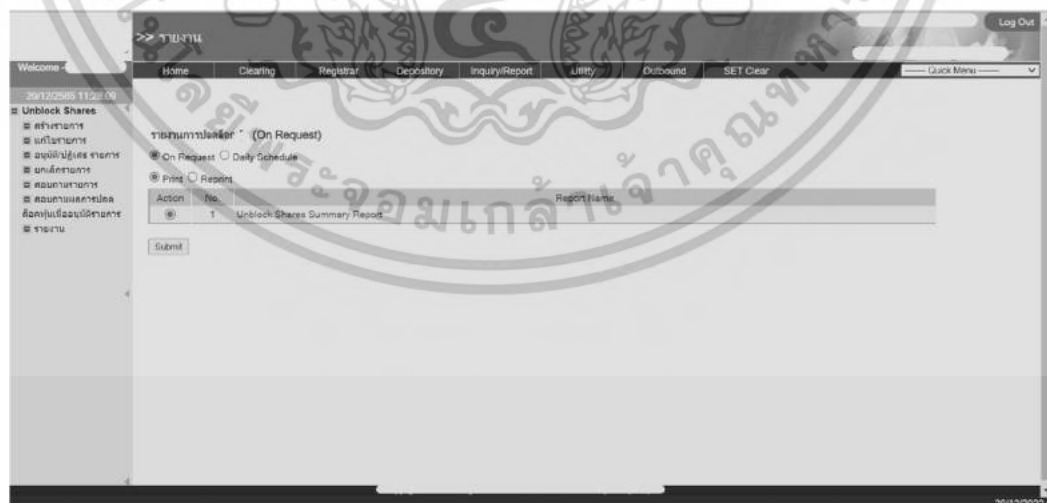
จากรูปภาพที่ 4.8 เป็นหน้าที่แสดง PDF Report Modal ที่ได้หลังจากการกดปุ่ม Submit ในรูปภาพที่ 4.7 โดยสามารถกดปุ่ม Download และ Print ได้ตรงที่ไอคอนบนขวาของ Modal

4.1.2 โมดูล Depository

เป็นโมดูลหลักในระบบที่มีระบบย่อย โดยมีการพัฒนาโมดูลเปรียบเทียบกับหน้าจอเก่า ดังต่อไปนี้

1) โมดูลย่อย Unblock Shares หน้า Report

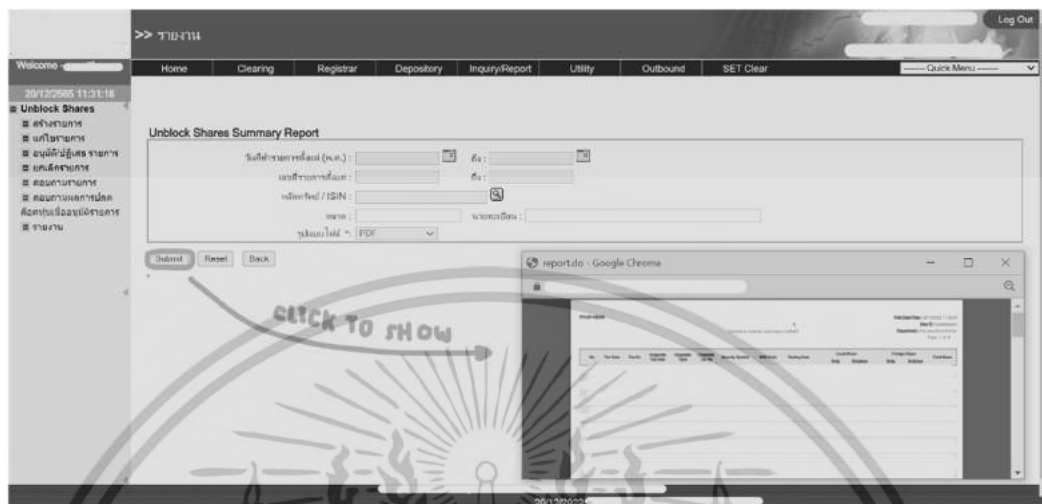
ระบบเดิม เนื่องจากได้มีการเขียนไว้เมื่อนานมาแล้ว ทำให้ระบบมีการออกแบบที่เก่า และไม่มีการลดจำนวนหน้าของการเขียน มีการแสดงผลหน้าจอทั้งหมด ดังต่อไปนี้



รูปภาพที่ 4. 9 Search หน้า Unblock Shares Report ระบบเดิม

จากรูปภาพที่ 4.9 เป็นหน้าที่แสดงหน้า Unblock Shares Report เป็นหน้าแสดงข้อมูลที่เอกสารนี้เป็นรายการ Report ที่สามารถ Export ออกมาได้ เมื่อกดเข้าไปที่รายการที่แสดงในตารางแสดงผล ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลัพท์ค้นหา จะแสดงอีกหน้าที่มีไว้เพิ่มเงื่อนไขในการค้นหา ทำให้ทุกครั้งที่ต้องการค้นหาตามรายการ จะมีการเปิดหน้าต่างใหม่



รูปภาพที่ 4. 10 Submit หน้า Unlock Shares Report ระบบเดิม

จากรูปภาพที่ 4.10 เป็นหน้าการทำงานของ Unlock Shares Report แสดงหลังจากกดปุ่ม Submit เป็นหน้าจอที่แยกออกมาเป็น PDF Frame

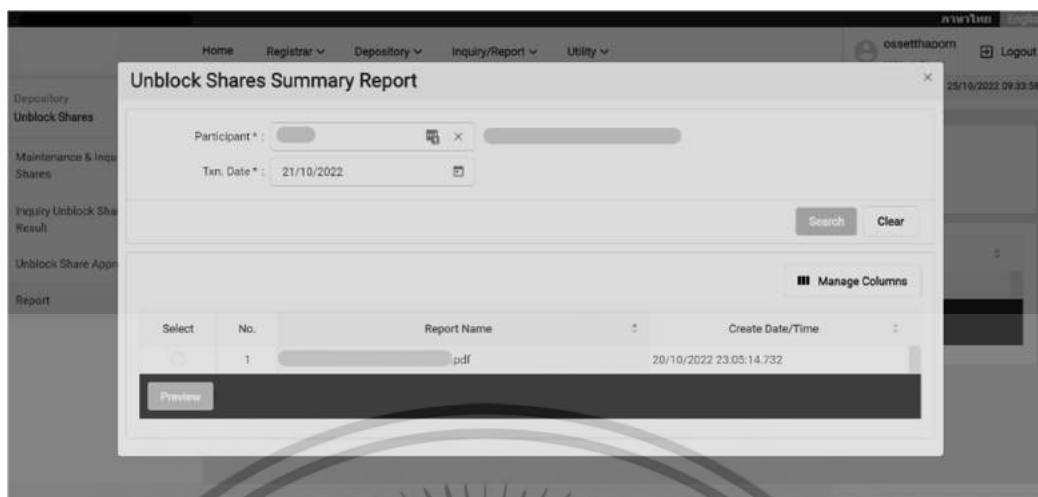
ระบบใหม่ มีการออกแบบโดยมีการจัด Layout ที่ดีขึ้น รวมหน้าค้นหา และหน้าผลลัพธ์ไว้ในหน้าเดียวกัน มีการกำหนด Modal ใหม่จาก CSS Library ที่ใช้ โดยจะแสดงรายงานออกมาเป็น PDF Modal ที่สามารถดาวน์โหลด หรือปริ้นต์ได้ มีการแสดงผลหน้าจอทั้งหมด ดังต่อไปนี้



รูปภาพที่ 4. 11 Daily Schedule ของหน้า Unlock Shares

จากรูปภาพที่ 4.11 เป็นหน้าที่แสดง Daily Schedule เป็นชนิดรายงานแบบที่ปริ้นต์ตามรอบเวลาที่กำหนดไว้ในระบบของแต่ละวัน ซึ่งรายการที่นำมาแสดงทั้งหมดได้ดึงมาจาก API ที่ถูกสร้างไว้ จากรูปมีรายการให้เลือก 1 รายการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 12 Unblock Shares Summary Report ของ Daily Schedule หน้า Unblock Shares

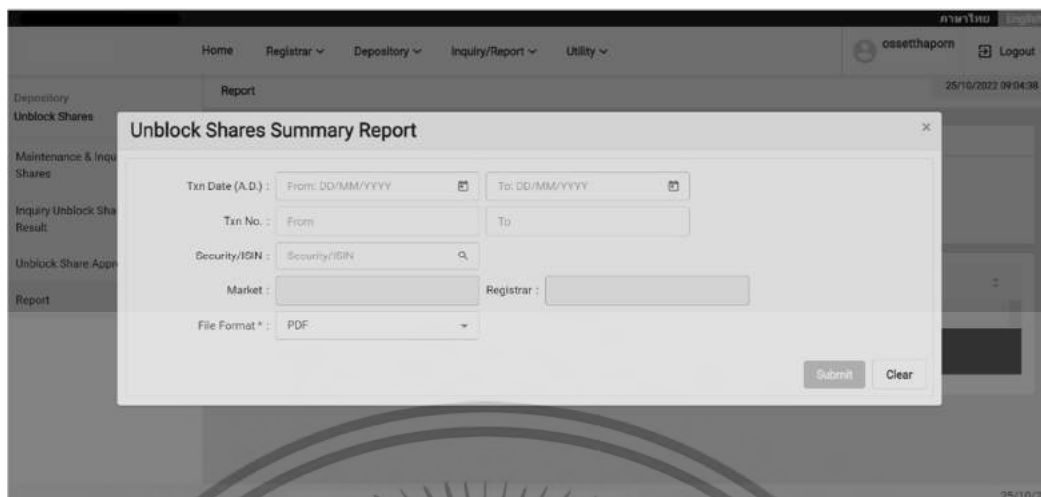
จากรูปภาพที่ 4.12 เป็นหน้าที่แสดง Unblock Shares Summary Report จะแสดงหลังจากที่เลือกรายการจากรูปภาพที่ 4.11 และจะมีการแสดงรายการที่ดึงข้อมูลมาจาก API หากกดที่รายการที่แสดงจะมี PDF Modal แสดงขึ้นมาให้สามารถเลือกการ Export ได้



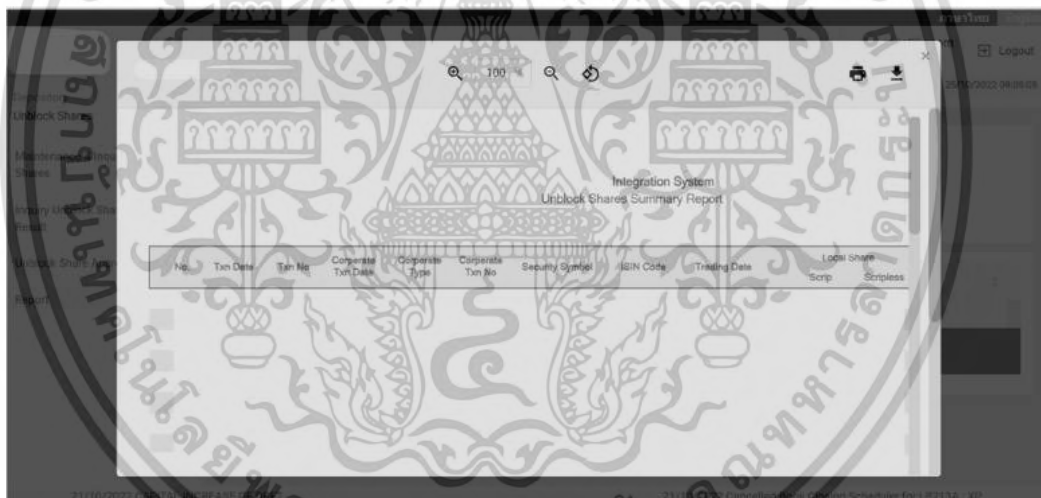
รูปภาพที่ 4. 13 Search Report ของ Print หน้า Unblock Shares

จากรูปภาพที่ 4.13 เป็นหน้าที่แสดง On Request เป็นชนิดรายงานที่ขอรายงานตามที่ต้องการ โดยมีเงื่อนไขเพื่อค้นหา ก่อนจะสามารถเปิดเป็นผลลัพธ์ของ PDF แบบ Modal ได้ อีกทั้งใน On Request ยังสามารถเลือกพิมพ์ แบบ Print ที่ต้องกรอกเงื่อนไขในการค้นหา หรือ Reprint ที่จะแสดงไฟล์ก็ต่อเมื่อมีการขอ PDF ไว้แล้ว โดยจะใส่วันที่เป็นเงื่อนไข จากรูปเป็นการเลือก Print

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



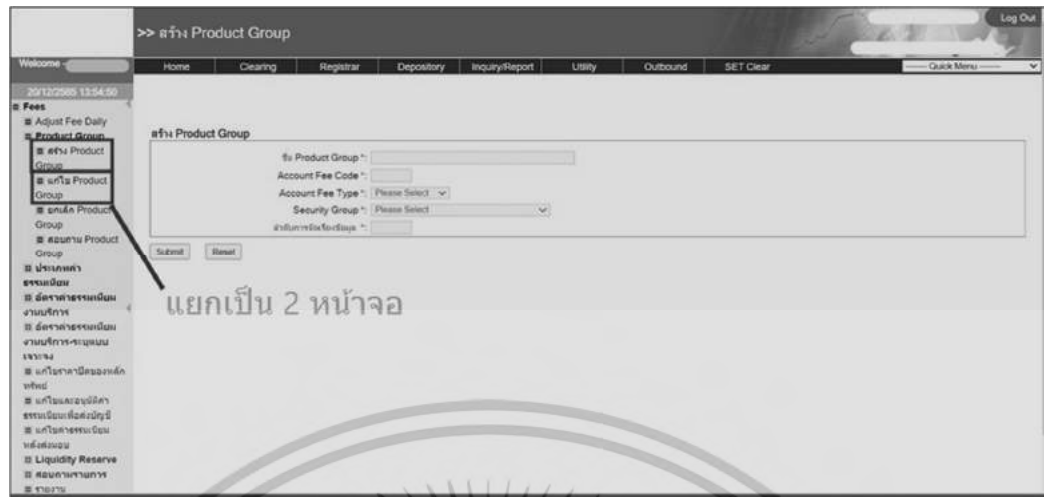
รูปภาพที่ 4. 14 Submit Report ของ Unblock Shares Summary หน้า Unblock Shares
จากรูปภาพที่ 4.14 เป็นหน้าที่แสดง Modal หลังจากกดปุ่ม Submit ในหน้า 4.13 โดย
จะต้องกรอกเงื่อนไขต่างๆ โดยจะมี Textbox ที่จำเป็นต้องกรอก และกดปุ่ม Submit เพื่อร้องขอ
รายงานได้



รูปภาพที่ 4. 15 PDF Report Modal หน้า Unblock Shares
จากรูปภาพที่ 4.15 เป็นหน้าที่แสดง PDF Report Modal ที่ได้หลังจากการกดปุ่ม Submit
ในรูปภาพที่ 4.7 โดยสามารถกดปุ่ม Download และ Print ได้ตรงที่ไอคอนบนขวาของ Modal

2) โมดูลย่อย Fees หน้า Create, Update และ View Product Group รูปแบบ Drawer
ระบบเดิม เนื่องจากได้มีการเขียนไว้เมื่อนานมาแล้ว ทำให้ระบบมีการออกแบบที่เก่า และไม่
มีการลดจำนวนหน้าของการเขียน มีการแสดงผลหน้าจอทั้งหมด ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 16 Create หน้า Product Group ระบบเดิม

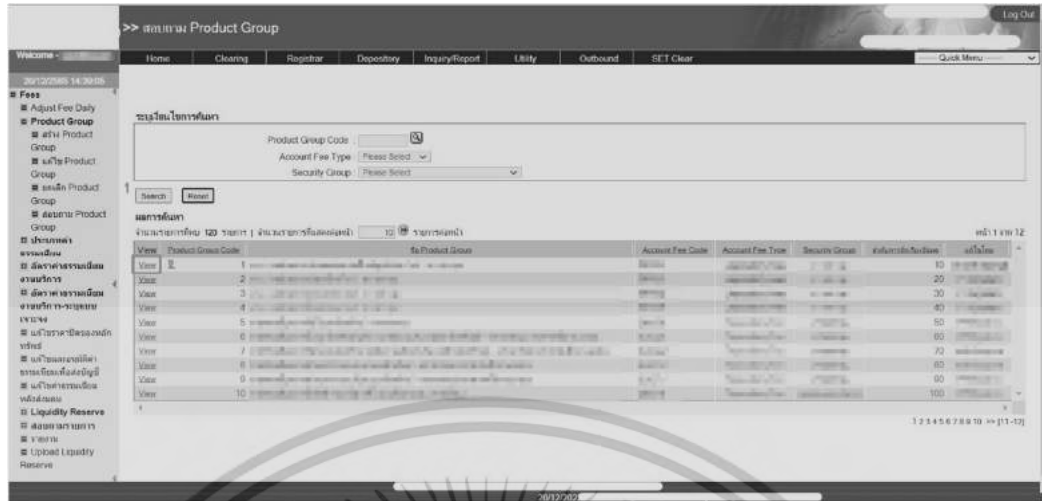
จากรูปภาพที่ 4.16 เป็นหน้าแสดง Create สำหรับสร้างข้อมูล Product Group ซึ่งสามารถสร้างได้แค่ผู้ที่มีสิทธิของการสร้างเท่านั้น ในการสร้างข้อมูลต้องกรอกเงื่อนไขต่างๆ ใน Text Field ในทุก Text Field ที่ต้องการให้กรอกข้อมูล หลังจากนั้นต้องกดปุ่ม Submit ซึ่งจะแสดงในหน้าใหม่ที่ถูก Render ขึ้นมา



รูปภาพที่ 4. 17 Search หน้า Product Group ระบบเดิม

จากรูปภาพที่ 4.17 เป็นหน้าแสดง Search สำหรับค้นหาข้อมูลที่ต้องการแก้ไข ซึ่งสามารถค้นหาได้แค่ผู้ที่มีสิทธิในการแก้ไขเท่านั้น โดยสามารถค้นหาได้จากการระบุเงื่อนไข หรือไม่ต้องระบุเงื่อนไข ก่อนกดปุ่ม Submit เพื่อแสดงตารางการค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 18 ปุ่ม View หน้า Product Group ระบบเดิม

จากรูปภาพที่ 4.18 เป็นหน้าแสดง View สำหรับแสดงข้อมูล Product Group ที่ต้องการดูข้อมูล จะมีข้อมูลได้มาจากการดึงมาจาก API สามารถเรียงลำดับตามจากน้อยไปมาก หรือมากไปน้อย หลังจากรายการแสดงขึ้นมาสามารถกดปุ่ม View เพื่อถ่ายสำหรับดูรายละเอียด จึงมีการแยกอีกหน้า



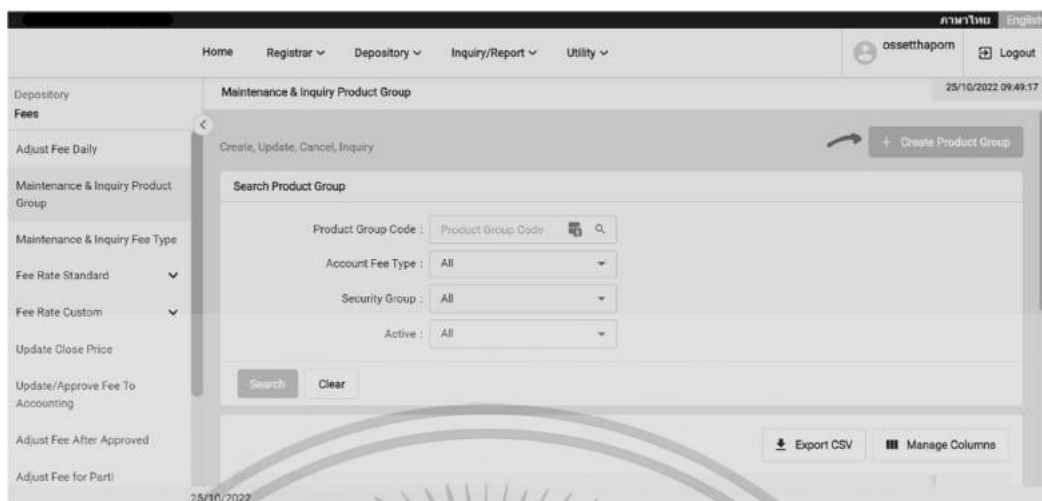
รูปภาพที่ 4. 19 View ของหน้า Product Group ระบบเดิม

จากรูปภาพที่ 4.19 เป็นหน้าแสดง View สำหรับหลังจากเลือกการให้แสดงข้อมูล Product Group ซึ่งแสดงออกมาในรูปแบบ Disable Text Field

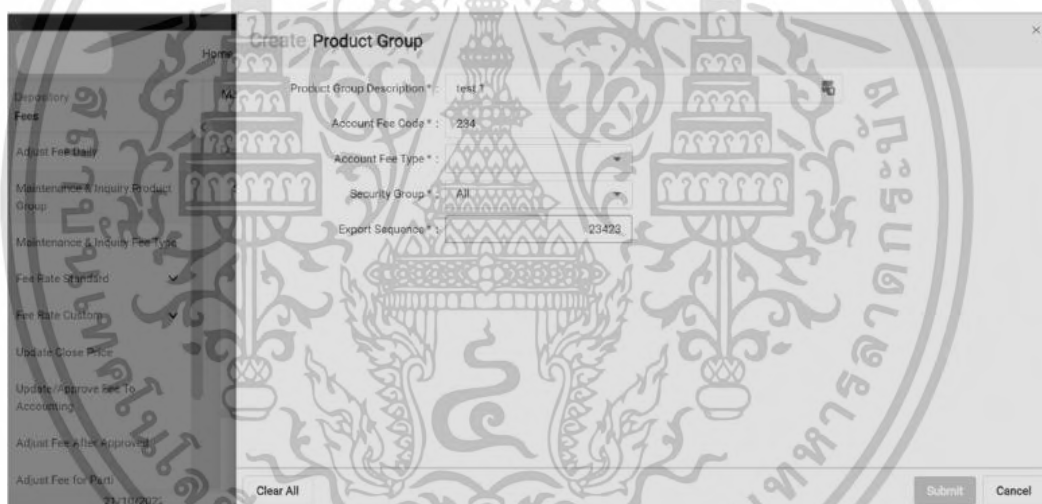
โดยจากรูปภาพที่ 4.16 - 4.19 จะเห็นว่าทั้งสองหน้าจะมี Text Field ที่มีลักษณะที่เหมือนกัน ดังนั้นจึงสามารถใช้ Text Field ร่วมกันได้จากการแบ่งเขียนหลายหน้า

ระบบใหม่ มีการเปลี่ยนแปลงให้รวมเป็น Drawer ในทั้ง 3 ประเภท ซึ่งนำการเขียนแบบแยกประเภท Action เพื่อให้ Create, View และ Update รวมกันเป็น Component เดียวกัน และใช้การแยกประเภท Action ที่ส่งเป็น Props เพื่อแยกประเภท และช่วยลดการเขียนโปรแกรมให้น้อยลงดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

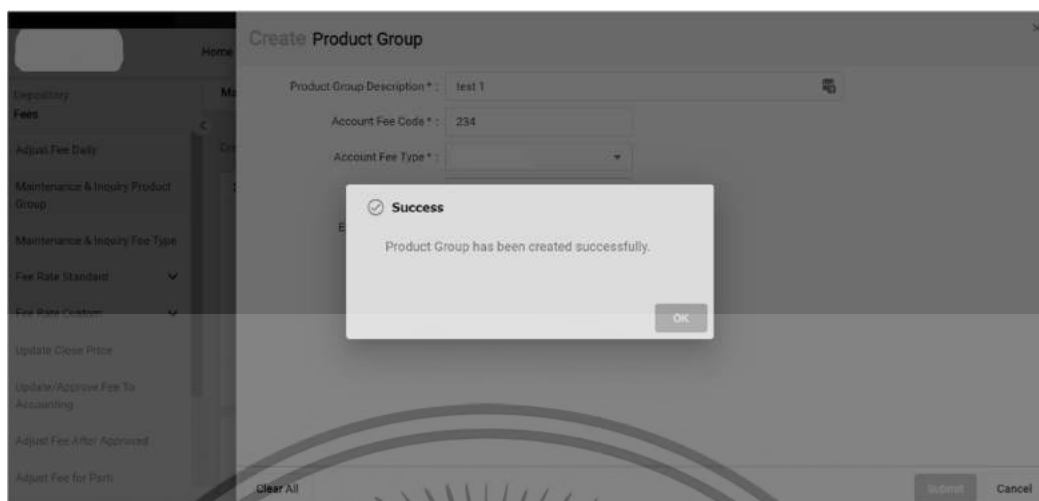


รูปภาพที่ 4. 20 ปุ่ม Create หน้า Product Group
จากรูปภาพที่ 4.20 เป็นหน้าแสดงปุ่ม Create เพื่อเข้าถึงการสร้างข้อมูล Product Group ได้ โดยกดปุ่มบนขวาด้านบน



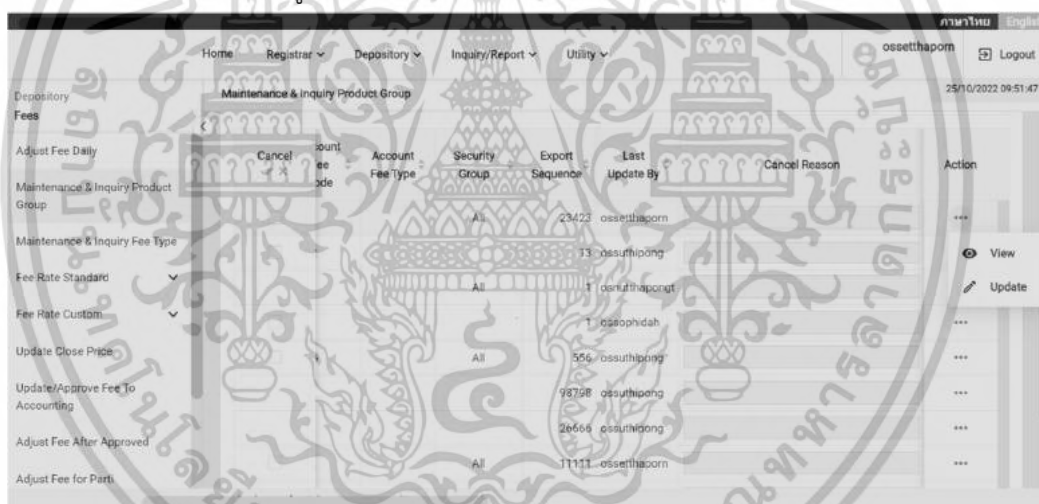
รูปภาพที่ 4. 21 Create Drawer หน้า Product Group
จากรูปภาพที่ 4.21 เป็นหน้าแสดง Create รูปแบบ Drawer เพื่อสร้าง หรือเพิ่มข้อมูล Product Group โดยต้องกรอกเงื่อนไขในแต่ละ Text Field ที่จำเป็นต้องกรอก โดยแต่ละ Text Field จะมีการกำหนดข้อมูลที่กรอกไม่เหมือนกัน เช่น String Number หรือ Dropdown ที่ต้องไปดึงข้อมูลที่แสดงมาจาก API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 22 Success Create Drawer หน้า Product Group

จากรูปที่ 4.22 เป็นหน้าแสดง Success Create Drawer เพื่อปฏิสัมพันธ์กับผู้ใช้งานว่าข้อมูลที่ถูกสร้างสำเร็จตามเงื่อนไขครบถ้วน หากมีการกรอกข้อมูลไม่ผ่านตามเงื่อนไขจะแสดงขึ้นมาถึงข้อผิดพลาดในการกรอกข้อมูล



รูปภาพที่ 4. 23 ปุ่ม View และ Update หน้า Product Group

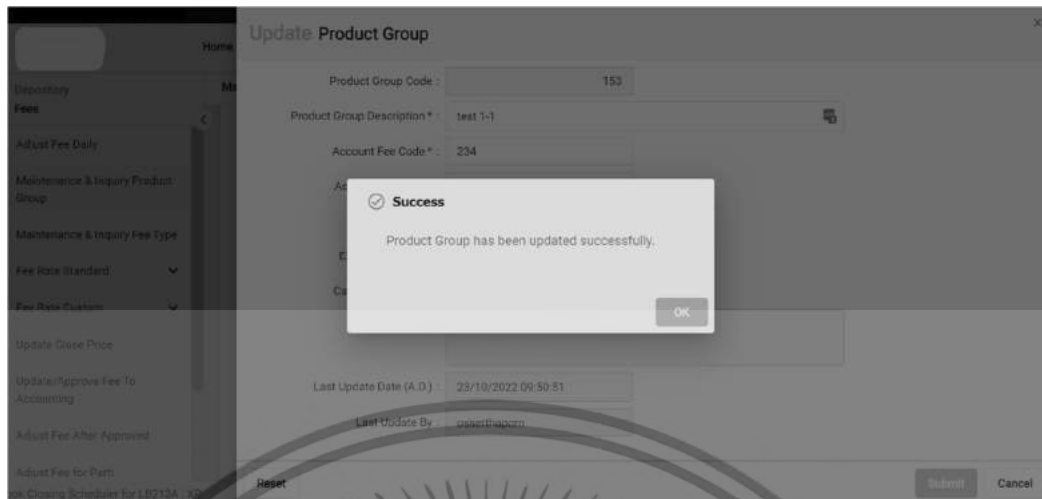
จากรูปภาพที่ 4.23 เป็นหน้าแสดงปุ่ม View และ Update เพื่อเข้าถึงการแก้ไข และดูข้อมูลของ Product Group หากต้องการดู และแก้ไขข้อมูลต้องค้นหาข้อมูลตามเงื่อนไขก่อน หลังจากนั้นจะแสดงข้อมูลแต่ละรายการในรูปแบบตาราง ซึ่งปุ่มดังกล่าวแสดงอยู่ Column สุดท้ายของตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปภาพที่ 4. 24 View Drawer หน้า Product Group
จากรูปภาพที่ 4.24 เป็นหน้าแสดง View ในรูปแบบ Drawer เพื่อแสดงรายละเอียดข้อมูล Product Group ทั้งหมดของธุรกรรมนั้น โดยแต่ละ Text Field จะมีรูปแบบเป็น Disable

รูปภาพที่ 4. 25 Update Drawer หน้า Product Group
จากรูปภาพที่ 4.25 เป็นหน้าแสดง Update รูปแบบ Drawer เพื่อแสดงรายละเอียดข้อมูล Product Group ที่สามารถแก้ไขได้ โดยจะแสดงเป็น Text Field ที่สามารถกดเพื่อแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 26 Success Update Drawer ของหน้า Product Group

จากรูปภาพที่ 4.26 เป็นหน้าแสดง Success Update Drawer เพื่อปฏิสัมพันธ์กับผู้ใช้กันว่า ข้อมูลที่ถูกแก้ไขสำเร็จตามเงื่อนไขครบถ้วน หากมีการกรอกข้อมูลไม่ผ่านตามเงื่อนไขจะแสดงขึ้นมาถึงข้อผิดพลาดในการกรอกข้อมูล

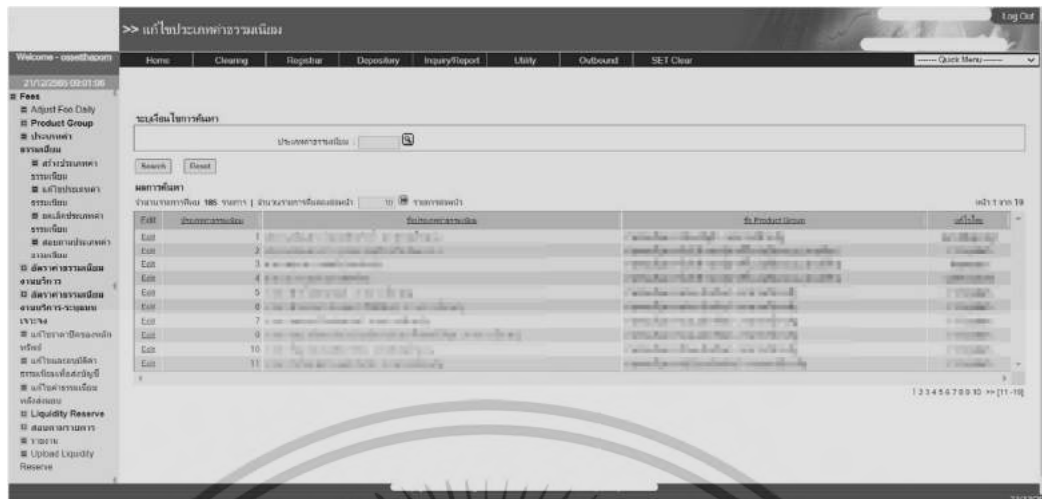
3) โมดูลย่อย Fees หน้า Create, Update และ View Fee Type รูปแบบ Drawer

ระบบเดิม เนื่องจากการเขียนไว้เมื่อนานมาแล้ว ทำให้ระบบมีการออกแบบที่เก่า และไม่มีการลดจำนวนหน้าของการเขียน มีการแสดงผลหน้าจอทั้งหมด ดังต่อไปนี้ ดังต่อไปนี้



รูปภาพที่ 4. 27 Create ของหน้า Fee Type ระบบเดิม

จากรูปภาพที่ 4.27 เป็นหน้าแสดง Create สำหรับสร้างข้อมูล Fee Type ซึ่งสามารถสร้างได้แค่ผู้ที่มีสิทธิของการสร้างเท่านั้น ในการสร้างข้อมูลต้องกรอกเงื่อนไขต่างๆ ใน Text Field ในทุก Text Field ที่ต้องการให้กรอกข้อมูล หลังจากนั้นต้องกดปุ่ม Submit ซึ่งจะแสดงในหน้าใหม่ที่ถูกเอกสารนี้เป็น Render ขึ้นมาไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 28 Search หน้า Fee Type ระบบเดิม

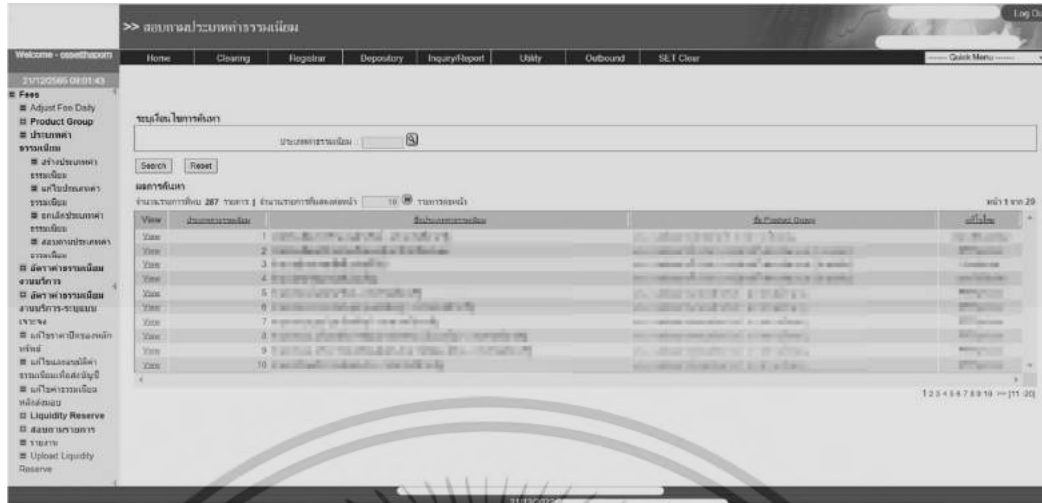
จากรูปภาพที่ 4.28 เป็นหน้าแสดง Search สำหรับค้นหาข้อมูลที่ต้องการแก้ไข ซึ่งสามารถค้นหาได้แก่ผู้ที่มีสิทธิในการแก้ไขเท่านั้น โดยสามารถค้นหาได้จากการระบุเงื่อนไข หรือไม่ต้องระบุเงื่อนไข ก่อนกดปุ่ม Submit เพื่อแสดงตารางการค้นหา



รูปภาพที่ 4. 29 Update หน้า Fee Type ระบบเดิม

จากรูปภาพที่ 4.29 เป็นหน้าแสดง Update สำหรับแก้ไขข้อมูล Fee Type เพื่อจะแสดงหน้าใหม่ ให้แก้ไขข้อมูลโดยต้องกรอกเงื่อนไขต่างๆ ใน Text Field ในทุก Text Field ที่ต้องการให้แก้ไขข้อมูล หลังจากนั้นต้องกดปุ่ม Submit เพื่อยืนยันข้อมูลที่ถูกแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 30 ปุ่ม View หน้า Fee Type ระบบเดิม

จากรูปภาพที่ 4.30 เป็นหน้าแสดง View สำหรับแสดงข้อมูล Fee Type ที่ต้องการดูข้อมูล จะมีข้อมูลได้มาจากการดึงมาจาก API สามารถเรียงลำดับตามจากน้อยไปมาก หรือมากไปน้อย หลังจากรายการแสดงขึ้นมาสามารถกดปุ่ม View เพื่อถ่ายสำหรับดูรายละเอียด จึงมีการแยกอีกหน้า



รูปภาพที่ 4. 31 View หน้า Fee Type ระบบเดิม

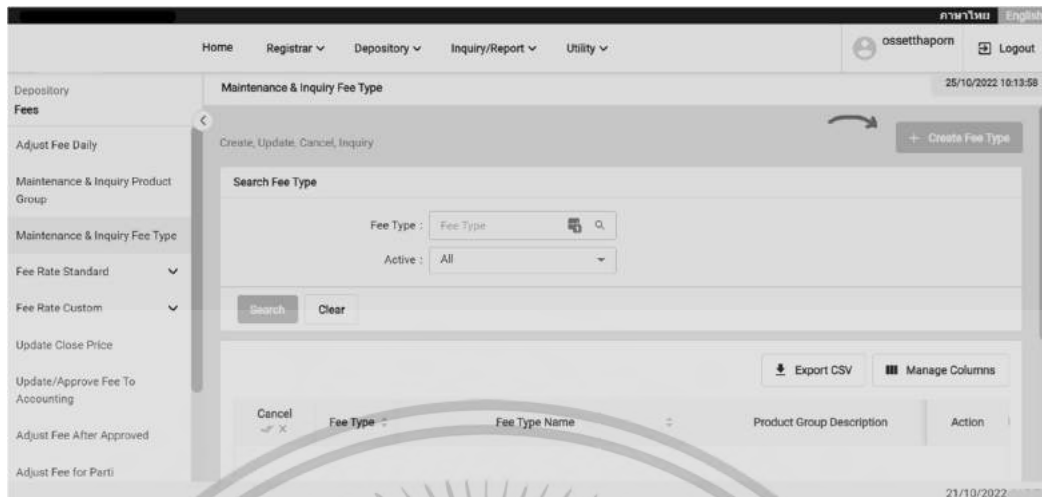
จากรูปภาพที่ 4.31 เป็นหน้าแสดง View สำหรับหลังจากเลือกรายการให้แสดงข้อมูล Product Group ซึ่งแสดงออกมาในรูปแบบ Disable Text Field

โดยจากรูปภาพที่ 4.27 - 4.31 จะเห็นว่าทั้งสองหน้าจะมี Text Field ที่มีลักษณะที่เหมือนกัน ดังนั้นจึงสามารถใช้ Text Field ร่วมกันได้จากการแบ่งเขียนหลายหน้า

ระบบใหม่ มีการเปลี่ยนแปลงให้รวมเป็น Drawer ในทั้ง 3 ประเภท ซึ่งนำการเขียนแบบแยกประเภท Action เพื่อให้ Create, View และ Update รวมกันเป็น Component เดียวกัน และใช้การแยกประเภท Action ที่ส่งเป็น Props เพื่อแยกประเภท และช่วยลดการเขียนโปรแกรมให้น้อยลง

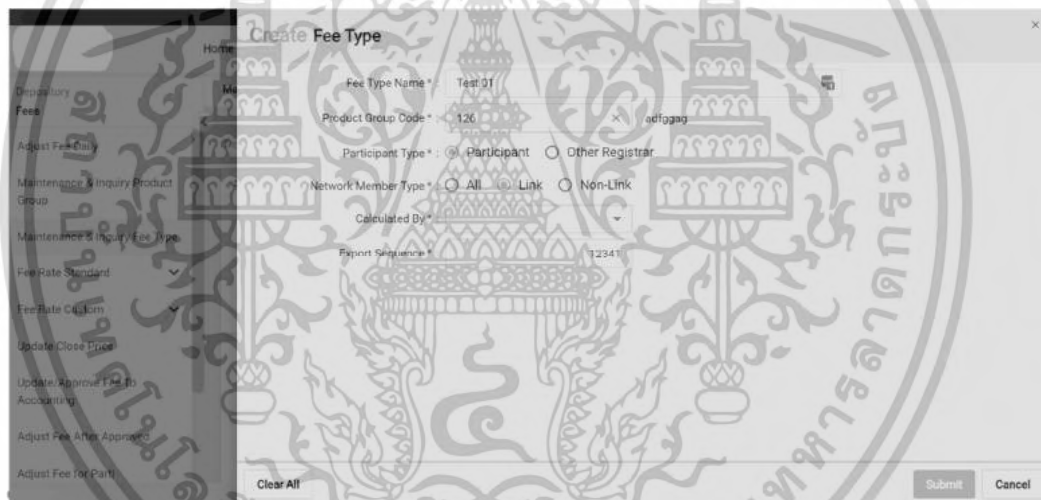
ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 32 ปุ่ม Create หน้า Fee Type

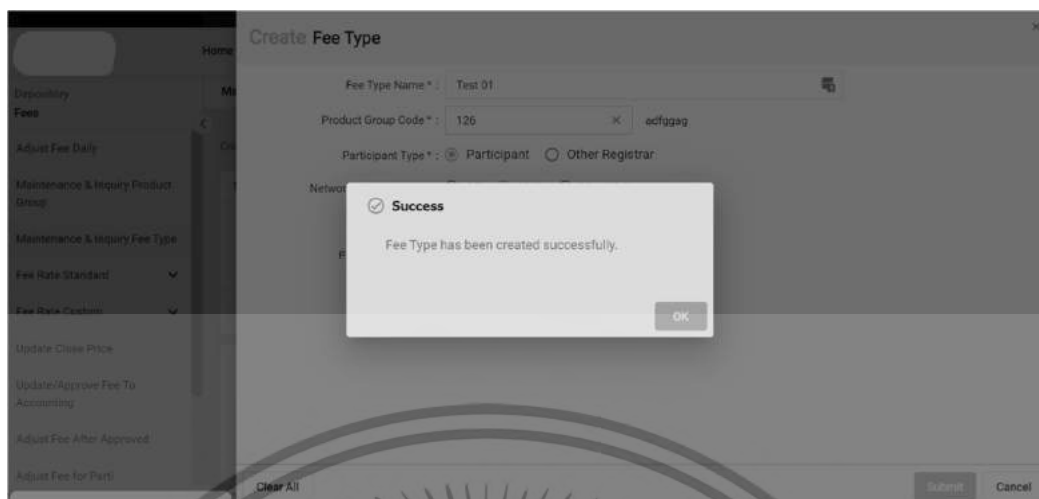
จากรูปภาพที่ 4.22 เป็นหน้าแสดงปุ่ม Create เพื่อเข้าถึงการสร้างข้อมูล Fee Type ได้ โดยกดปุ่มบนขวาด้านบน



รูปภาพที่ 4. 33 Create Drawer หน้า Fee Type

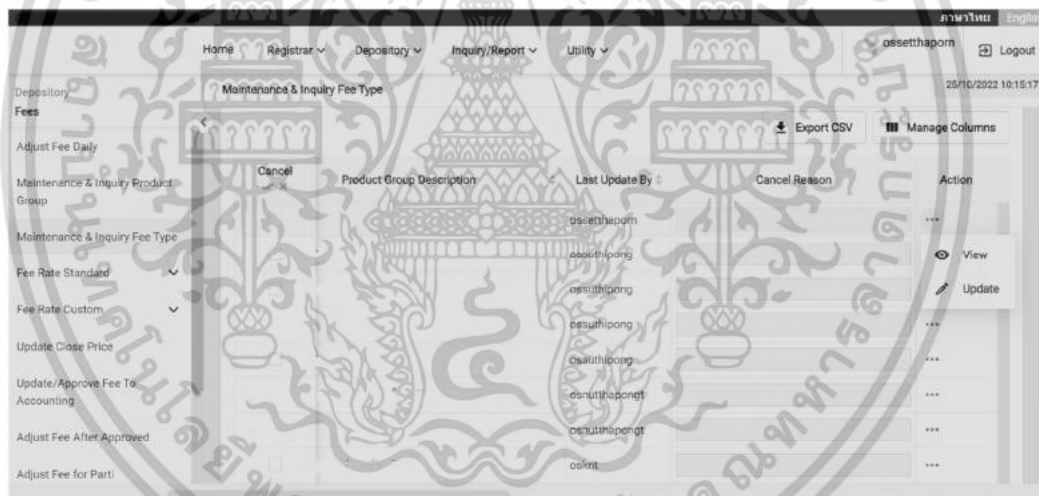
จากรูปภาพที่ 4.33 เป็นหน้าแสดง Create รูปแบบ Drawer เพื่อสร้าง หรือเพิ่มข้อมูล Fee Type โดยต้องกรอกเงื่อนไขในแต่ละ Text Field ที่จำเป็นต้องกรอก โดยแต่ละ Text Field จะมีการกำหนดข้อมูลที่กรอกไม่เหมือนกัน เช่น String Number หรือ Dropdown ที่ต้องไปดึงข้อมูลที่แสดงมาจาก API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 34 Success Create Drawer หน้า Fee Type

จากรูปที่ 4.34 เป็นหน้าต่างแสดง Success Create Drawer เพื่อปฏิสัมพันธ์กับผู้ใช้งานว่าข้อมูลที่ถูกรสร้างสำเร็จตามเงื่อนไขครบถ้วน หากมีการกรอกข้อมูลไม่ผ่านตามเงื่อนไขจะแสดงขึ้นมาถึงข้อผิดพลาดในการกรอกข้อมูล



รูปภาพที่ 4. 35 ปุ่ม View และ Update หน้า Fee Type

จากรูปภาพที่ 4.35 เป็นหน้าต่างแสดงปุ่ม View และ Update เพื่อเข้าถึงการแก้ไข และดูข้อมูลของ Fee Type หากต้องการดู และแก้ไขข้อมูลต้องค้นหาข้อมูลตามเงื่อนไขก่อน หลังจากนั้นจะแสดงข้อมูลแต่ละรายการในรูปแบบตาราง ซึ่งปุ่มดังกล่าวแสดงอยู่ Column สุดท้ายของตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

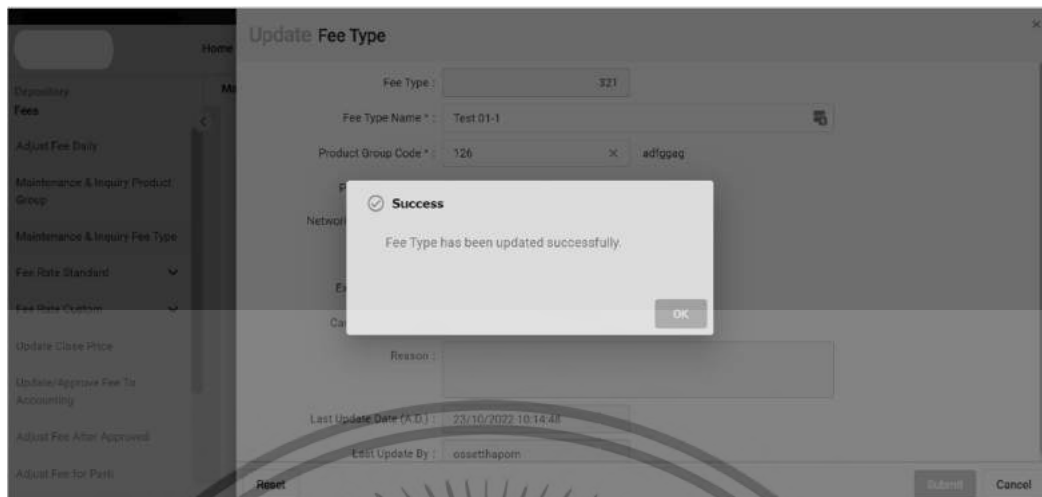
รูปภาพที่ 4. 36 View Drawer หน้า Fee Type

จากรูปภาพที่ 4.36 เป็นหน้าแสดง View ในรูปแบบ Drawer เพื่อแสดงรายละเอียดข้อมูล Fee Type ทั้งหมดของธุรกรรมนั้น โดยแต่ละ Text Field จะมีรูปแบบเป็น Disable

รูปภาพที่ 4. 37 Update Drawer ของหน้า Fee Type

จากรูปภาพที่ 4.37 เป็นหน้าแสดง Update รูปแบบ Drawer เพื่อแสดงรายละเอียดข้อมูล Fee Type ที่สามารถแก้ไขได้ โดยจะแสดงเป็น Text Field ที่สามารถกดเพื่อแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 38 Success Update Drawer หน้า Fee Type

จากรูปภาพที่ 4.38 เป็นหน้าแสดง Success Update Drawer เพื่อปฏิสัมพันธ์กับผู้ใช้งานว่า ข้อมูลที่ถูกแก้ไขสำเร็จตามเงื่อนไขครบถ้วน หากมีการกรอกข้อมูลไม่ผ่านตามเงื่อนไขจะแสดงขึ้นมาถึงข้อผิดพลาดในการกรอกข้อมูล

4) โมดูลย่อย Fees หน้า Approve & Reject Fee Rate Custom

ระบบเดิม มีการออกแบบจะแสดงผลพร้อมตารางเมื่อมีการค้นหา และตารางไม่สามารถดาวน์โหลดไฟล์ .csv ได้ พร้อมทั้งมีปุ่มสำหรับ Reset ตารางให้กลับไปเป็นตารางว่าง ดังต่อไปนี้



รูปภาพที่ 4. 39 Search Result หน้า Approve & Reject Fee Rate Custom ระบบเดิม

จากรูปภาพที่ 4.39 เป็นหน้าแสดง Search Result สำหรับทำการ Approve และ Reject ซึ่งต้องกดปุ่ม Search ก่อนเพื่อให้แสดงตารางที่มีรายการด้านล่าง ซึ่งในตารางมี Check Box เพื่อเลือก Approve หรือ Reject ถ้าหากเลือก Reject ต้องกรอกเงื่อนไขในการเลือกใน Column สุดท้ายของตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบใหม่ เปลี่ยนแปลงให้มีลักษณะเป็น Default ที่มีการค้นหา และมีผลลัพธ์แสดงในรูปแบบตาราง โดย Approve เพื่ออนุมัติรายการ และ Reject เพื่อปฏิเสธรายการ โดยรายการที่สร้างมาจากปุ่ม Create ที่อยู่ในหน้า Maintainace & Inquiry ดังต่อไปนี้

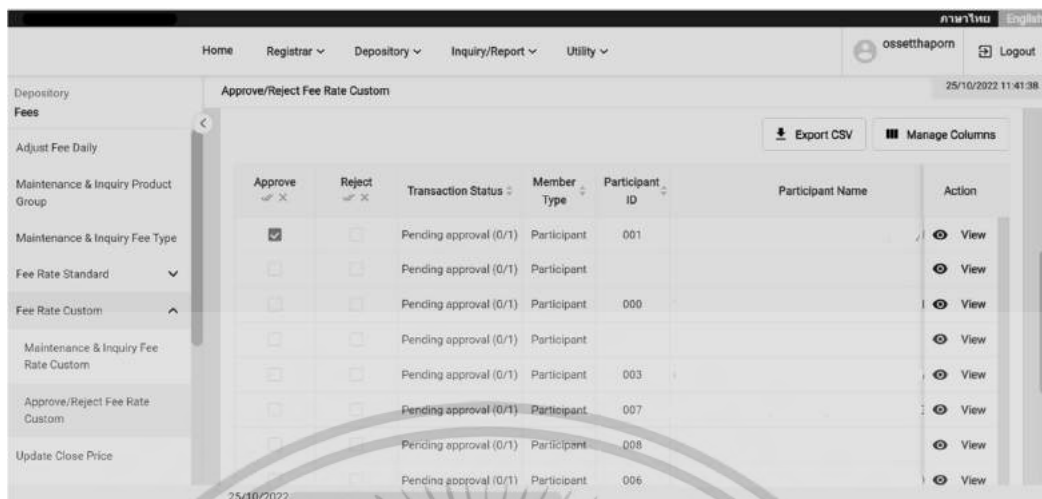
รูปภาพที่ 4. 40 Search ของ Participant หน้า Approve & Reject Fee Rate Custom

จากรูปภาพที่ 4.40 เป็นหน้า Search สำหรับค้นหาข้อมูลที่ต้องการ Approve หรือ Reject ซึ่งสามารถค้นหาได้แค่ผู้ที่มีสิทธิในการแก้ไขเท่านั้น โดยสามารถค้นหาได้จากการระบุเงื่อนไข หรือไม่ต้องระบุเงื่อนไข ก่อนกดปุ่ม Submit เพื่อแสดงตารางการค้นหา ซึ่งเงื่อนไขค้นหาจะแตกต่างกันตาม Member Type ที่เป็น Dropdown ให้เลือกประเภท Member Type ในรูปแบบ Participant

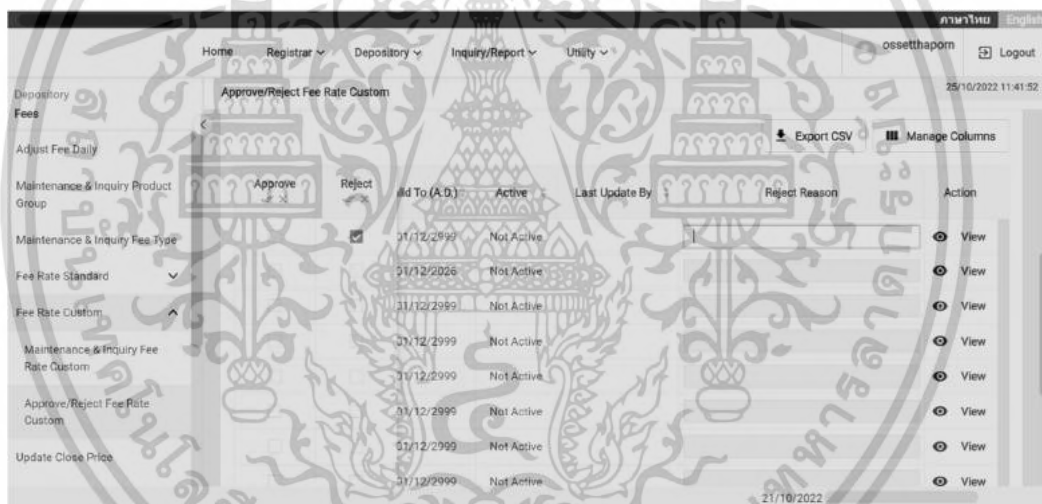
รูปภาพที่ 4. 41 Search ของ Other Registrar หน้า Approve & Reject Fee Rate Custom

จากรูปภาพที่ 4.41 เป็นหน้า Search ในเงื่อนไขของ Member Type ที่เป็น Dropdown ให้เลือกประเภท Member Type ในรูปแบบ Other Registrar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

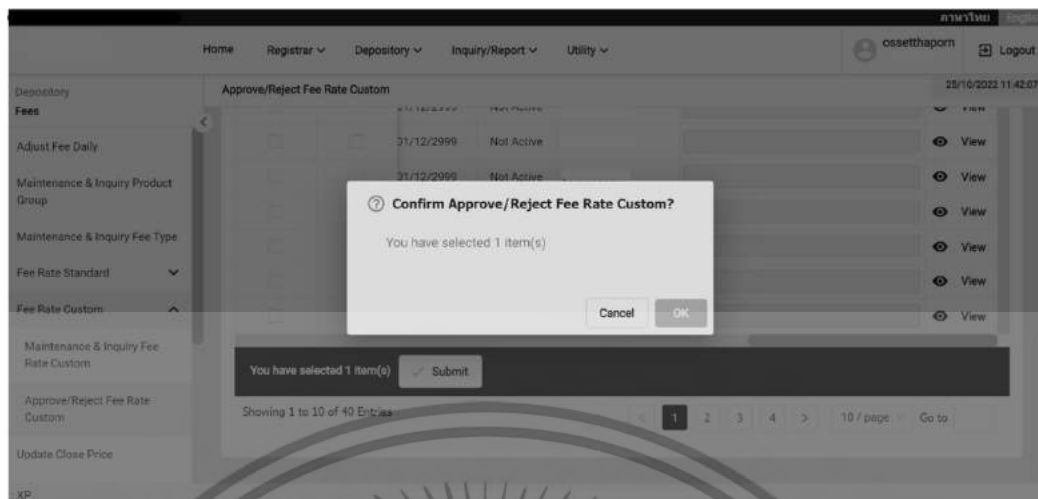


รูปภาพที่ 4. 42 Approve หน้า Approve & Reject Fee Rate Custom
จากรูปภาพที่ 4.42 เป็นหน้าแสดง Approve ซึ่งต้องกดปุ่ม Search ก่อน เพื่อแสดงตารางที่มีรายการด้านล่าง ซึ่งในตารางมี Check Box เพื่อเลือก Approve

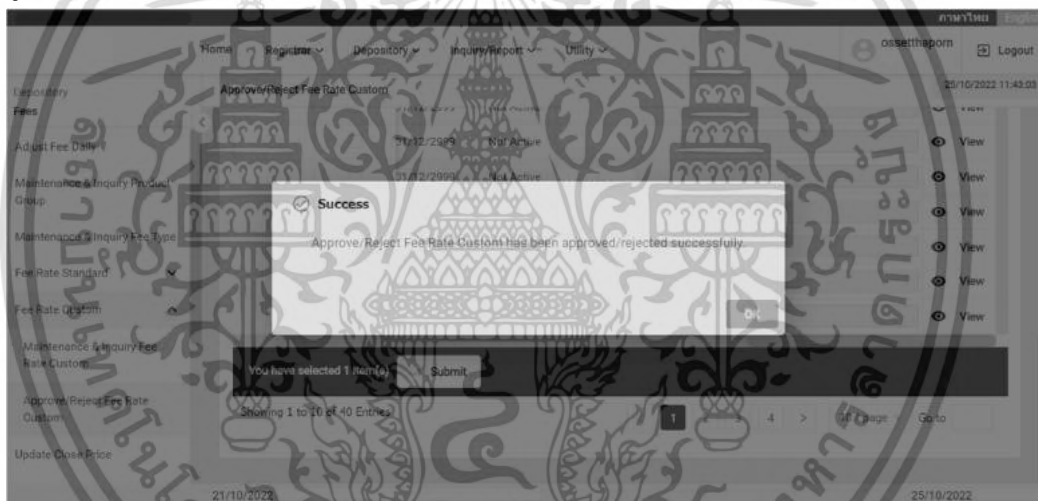


รูปภาพที่ 4. 43 Reject หน้า Approve & Reject Fee Rate Custom
จากรูปภาพที่ 4.43 เป็นหน้าแสดง Reject ซึ่งต้องกดปุ่ม Search ก่อน เพื่อแสดงตารางที่มีรายการด้านล่าง ซึ่งในตารางมี Check Box เพื่อเลือก Reject ถ้าหากเลือก Reject ต้องกรอกเงื่อนไขในการเลือกใน Column สุดท้ายของตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 44 Confirm หน้า Approve & Reject Fee Rate Custom
จากรูปภาพที่ 4.44 เป็นหน้าแสดง Confirm หลังจากกด Approve หรือ Reject จาก
รูปภาพที่ 4.42 – 4.43 โดยจะแสดงเป็น Pop up ขึ้นมาให้ยืนยันรายการ



รูปภาพที่ 4. 45 Success หน้า Approve & Reject Fee Rate Custom
จากรูปภาพที่ 4.45 เป็นหน้าแสดง Success หลังจากกด Confirm จากรูปภาพที่ 4.44 โดย
จะแสดงเป็น Pop up เพื่อแจ้งสถานะว่ารายการที่ทำไปสำเร็จ หรือไม่

5) โหมดล่อย Fees หน้า Adjust Fee Daily

ระบบเดิม มีการออกแบบจะแสดงผลบันทึงตารางเมื่อมีการค้นหา และตารางไม่สามารถ
ดาวน์โหลดไฟล์ .csv ได้ พร้อมทั้งมีปุ่มสำหรับ Reset ตารางให้กลับไปเป็นตารางว่าง ดังต่อไปนี้

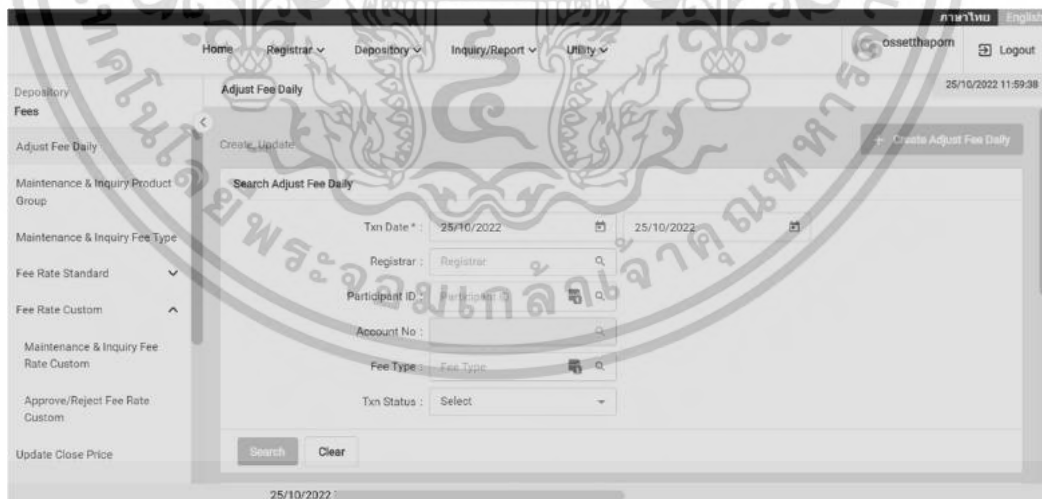
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 46 Search หน้า Adjust Fee Daily ระบบเดิม

จากรูปภาพที่ 4.46 เป็นหน้าแสดง Search สำหรับทำการ Reject ข้อมูล Adjust Fee Dairy ซึ่งมีการออกแบบเอการค้นหา และเพิ่มรายการข้อมูลมาต่อกัน หากต้องการ Reject ต้องค้นหาจากด้านบนก่อนกดปุ่ม Search และ Reject รายการข้อมูลในตาราง Result ด้านล่างที่จะแสดงรายการที่สามารถ Reject หากต้องการเพิ่มรายการสามารถเพิ่มในส่วนของ Adjust Fee Daily และกดปุ่ม Add

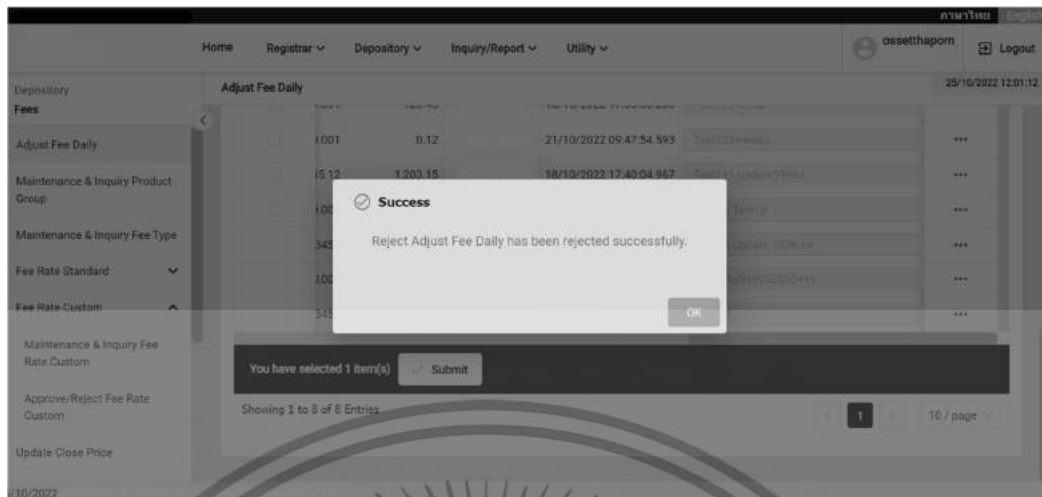
ระบบใหม่ เปลี่ยนแปลงให้มีลักษณะเป็น Default ที่มีการค้นหา และมีผลลัพธ์แสดงในรูปแบบตาราง โดย Reject มีไว้ เพื่อปฏิเสธ ซึ่งการค้นหารายการที่ Reject ต้องใส่เงื่อนไขตามที่ฟอร์มกำหนดให้ใส่ * ก่อนเพื่อค้นหา ดังต่อไปนี้



รูปภาพที่ 4. 47 Search หน้า Adjust Fee Daily

จากรูปภาพที่ 4.47 เป็นหน้า Search สำหรับค้นหาข้อมูลที่ต้องการ Reject ซึ่งสามารถค้นหาได้แก่ผู้ที่มีสิทธิในการแก้ไขเท่านั้น โดยสามารถค้นหาได้จากกราระบุเงื่อนไข หรือไม่ต้องระบุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 50 Success หน้า Adjust Fee Daily

จากรูปภาพที่ 4.50 เป็นหน้าแสดง Success หลังจากกด Confirm จากรูปภาพที่ 4.49 โดยจะแสดงเป็น Pop-up เพื่อแจ้งสถานะว่ารายการที่ทำไปสำเร็จ หรือไม่

6) โมดูลย่อย Fees หน้า Adjust Fee for Parti

ระบบใหม่ เปลี่ยนแปลงให้มีลักษณะเป็น Default ที่มีการค้นหา และมีผลลัพธ์แสดงในรูปแบบตาราง ซึ่งการค้นหารายการต้องใส่เงื่อนไขตามที่ฟอร์มกำหนดให้ใส่ * ก่อนเพื่อค้นหาดังต่อไปนี้



รูปภาพที่ 4. 51 Search ของหน้า Adjust Fee for Parti

จากรูปภาพที่ 4.51 เป็นหน้าแสดง Search เพื่อค้นหาโดยกรอกเงื่อนไข ก่อนทำการเลือกข้อมูลที่ต้องการ View เช่น Dropdown ของ Fee of Month และ Adjustment Type ที่ดึงมาจาก API , Registrar รูปแบบเมื่อกดให้แสดง Lookup, Export File Date รูปแบบที่กดเป็น Date Calendar เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Adjustment Type	Export File Date (A.D)	Participant ID	Account No.	Fee Type Name	Action
Adjust After Export File	12/09/2022	008	C		View
Adjust After Export File	13/09/2022	001			View
Adjust After Export File	13/09/2022	001			View
Adjust After Export File	14/09/2022	001			View
Adjust After Export File	14/09/2022	001			View
Adjust After Export File	07/10/2022	001			View

รูปภาพที่ 4. 52 Search Result หน้า Adjust Fee for Parti

จากรูปภาพที่ 4.52 เป็นหน้าแสดง Search Result สำหรับทำการ View ข้อมูล Adjust Fee for Parti หากต้องการ View ต้องค้นหาจากด้านบนก่อนกดปุ่ม Search จากรูปภาพที่ 4.52 เพื่อให้รายการแสดงที่ตาราง และเลือกรายการที่ต้องการ View

7) โมดูลย่อย Fees หน้า Adjust Fee for Registrar

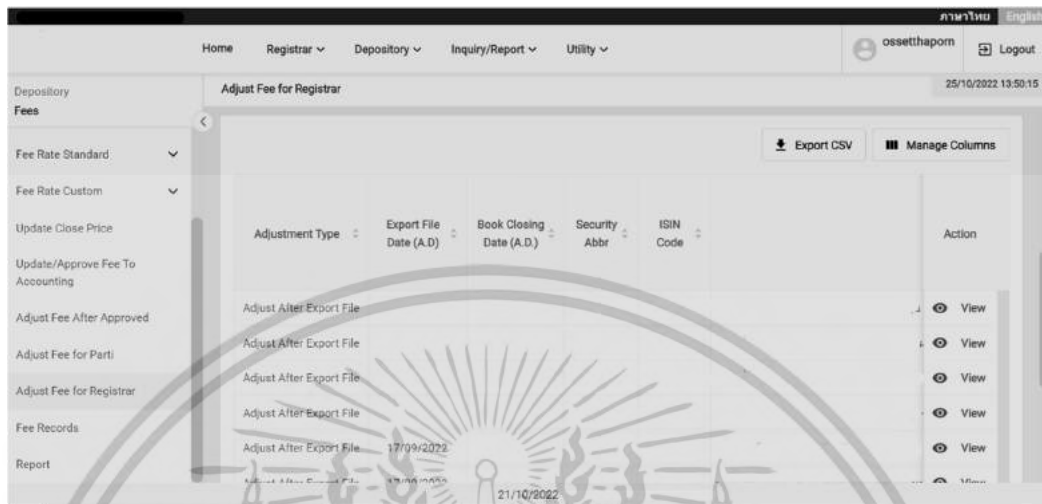
ระบบใหม่ เปลี่ยนแปลงให้มีลักษณะเป็น Default ที่มีการค้นหา และมีผลลัพธ์แสดงในรูปแบบตาราง ซึ่งการค้นหารายการต้องใส่เงื่อนไขตามที่ฟอร์มกำหนดให้ใส่ * ก่อนเพื่อค้นหาดังต่อไปนี้

รูปภาพที่ 4. 53 Search หน้า Adjust Fee for Registrar

จากรูปภาพที่ 4.53 เป็นหน้าแสดง Search เพื่อค้นหาโดยกรอกเงื่อนไข ก่อนทำการเลือกข้อมูลที่ต้องการ View โดยแต่ละ Text Field จะมีเงื่อนไขในการกรอกแตกต่างกัน เช่น Dropdown

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ Fee of Month และ Adjustment Type ที่ดึงมาจาก API , Registrar รูปแบบเมื่อกดให้แสดง Lookup, Export File Date รูปแบบที่กดเป็น Date Calendar เป็นต้น



รูปภาพที่ 4. 54 Search Result หน้า Adjust Fee for Registrar

จากรูปภาพที่ 4.54 เป็นหน้าแสดง Search Result สำหรับทำการ View ข้อมูล Adjust Fee for Registrar หากต้องการ View ต้องค้นหาจากด้านบนก่อนกดปุ่ม Search จากรูปภาพที่ 4.53 เพื่อให้รายการแสดงที่ตาราง และเลือกรายการที่ต้องการ View

8) โมดูลย่อย Fees หน้า Report

ระบบเดิม เนื่องจากได้มีการเขียนไว้เมื่อนานมาแล้ว ทำให้ระบบมีการออกแบบที่เก่า และไม่มีการลดจำนวนหน้าของการเขียน มีการแสดงผลหน้าจอทั้งหมด ดังต่อไปนี้



รูปภาพที่ 4. 55 Search Report หน้า Fees Report ระบบเดิม

จากรูปภาพที่ 4.55 เป็นหน้าที่แสดงหน้า Unblock Shares Report เป็นหน้าแสดงข้อมูลที่

แสดงรายการ Report ที่สามารถ Export ออกมาได้ เมื่อกดเข้าไปที่รายการที่แสดงในตารางแสดงผล เอกสารนี้เป็นเอกสารที่ส่งมอบให้บริษัทฯ เพื่อดูการแก้ไข ให้เห็น เมื่อผู้ดูแลระบบแก้ไขหน้าให้กรค่า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลัพท์ค้นหา จะแสดงอีกหน้าที่มีไว้เพิ่มเงื่อนไขในการค้นหา ทำให้ทุกครั้งที่ต้องการค้นหาตามรายการ จะมีการเปิดหน้าต่างใหม่



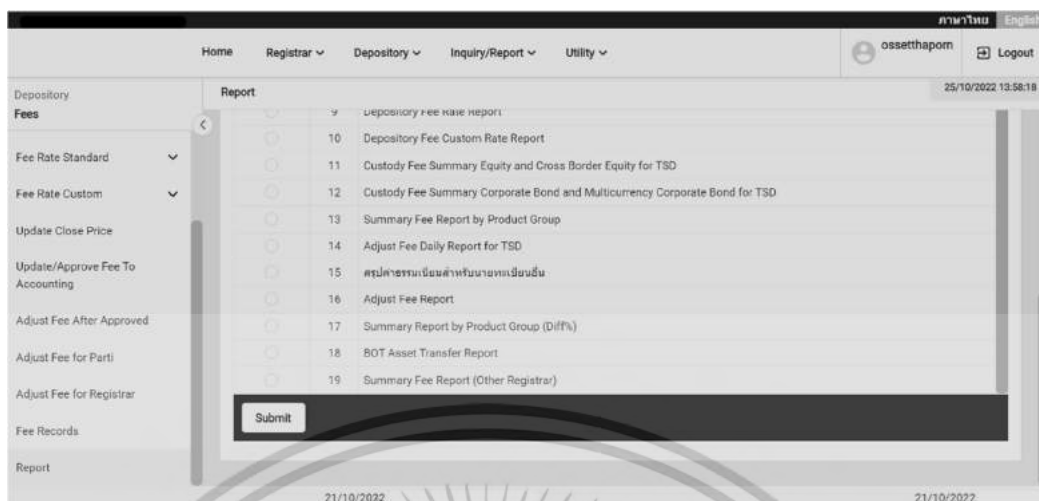
รูปภาพที่ 4. 56 Search Report Submit ของหน้า Fees Report ระบบเดิม
จากรูปภาพที่ 4.56 เป็นหน้าต่างการทำงานของ Fees Report แสดงหลังจากกดปุ่ม Submit เป็นหน้าต่างที่แยกออกมาเป็น PDF Frame

ระบบใหม่ มีการออกแบบโดยมีการจัด Layout ที่ดีขึ้น รวมหน้าค้นหา และหน้าผลลัพธ์ไว้ในหน้าเดียวกัน มีการกำหนด Modal ใหม่จาก CSS Library ที่ใช้ โดยจะแสดงรายงานออกมาเป็น PDF Modal ที่สามารถดาวน์โหลด หรือพรีนต์ได้ มีการแสดงผลหน้าจอทั้งหมด ดังต่อไปนี้



รูปภาพที่ 4. 57 Search Report List ของหน้า Fees Report
จากรูปภาพที่ 4.57 เป็นหน้าที่แสดง On Request เป็นชนิดรายงานที่ขอรายงานตามที่ต้องการ โดยมีเงื่อนไขเพื่อค้นหา ก่อนจะสามารถเปิดเป็นผลลัพธ์ของ PDF แบบ Modal ได้ อีกทั้งใน On Request ยังสามารถเลือกพริ้นต์ แบบ Print ที่ต้องกรอกเงื่อนไขในการค้นหา หรือ Reprint ที่จะ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ใช้งานเห็นใบแจ้งระบบงานด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 58 Submit Report List หน้า Fees Report

จากรูปภาพที่ 4.58 เป็นหน้าแสดง Submit ของ Report List เพื่อค้นหา Report ที่เลือกไป จากภาพที่ 4.57 และกดปุ่ม Submit เพื่อไปต่อหน้ากรอกเงื่อนไข ดังต่อไปนี้

8.1) รายงานสถิติ

ระบบเดิม เนื่องจากได้มีการเขียนไว้เมื่อนานมาแล้ว ทำให้ระบบมีการออกแบบที่เก่า และไม่มีการลดจำนวนหน้าของการเขียน แยกหน้าออกมาจากหน้า Submit ดังต่อไปนี้

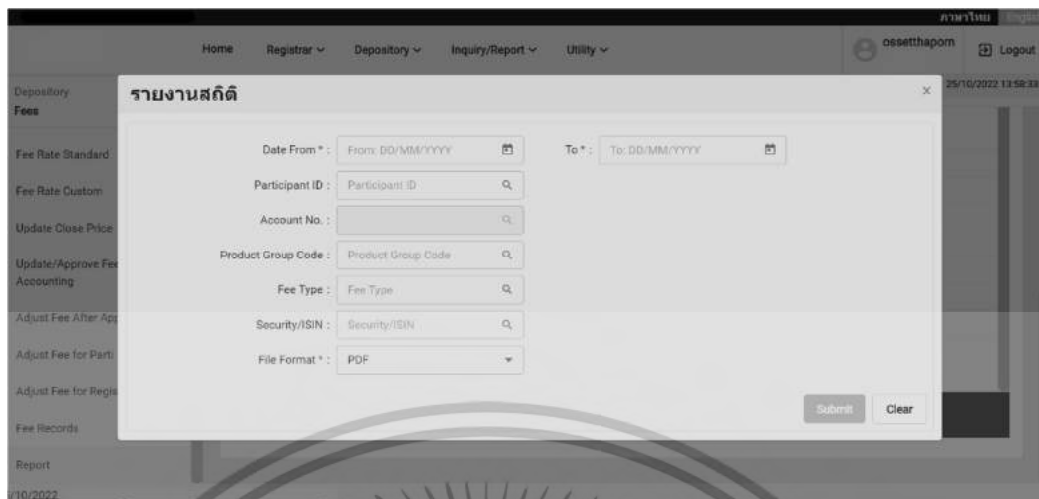


รูปภาพที่ 4. 59 รายงานสถิติ หน้า Fees Report ระบบเดิม

จากรูปภาพที่ 4.59 เป็นหน้ารายงานสถิติ ต่อจากหน้า Search Report List จากภาพ 4.55 เพื่อค้นหารายงานสถิติ โดยกรอกเงื่อนไขในการค้นหา และกดค้นหา เพื่อแสดงเป็น PDF ใน Window ใหม่

ระบบใหม่ เปลี่ยน PDF แบบ Window เป็น Modal ลดการ Render หน้าใหม่ ซึ่งจะแสดง ให้กรอกเงื่อนไขหลังจากเลือกรายการที่ต้องการค้นหา ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 60 รายงานสถิติ หน้า Fees Report

จากภาพที่ 4.60 เป็นหน้ารายงานสถิติ ต่อจากหน้า Search Report List จากภาพ 4.57 เพื่อค้นหารายงานสถิติ โดยกรอกเงื่อนไขในการค้นหา และกดค้นหา เพื่อแสดงเป็น PDF แบบ Modal หน้าเดิม

8.2) รายงานสถิติ (Summary)

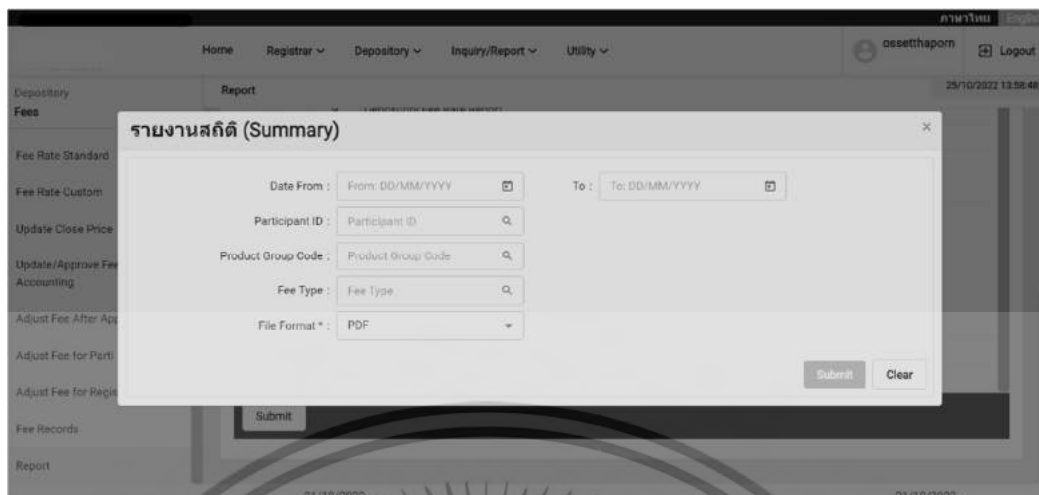
ระบบเดิม เนื่องจากได้มีการเขียนไว้เมื่อนานมาแล้ว ทำให้ระบบมีการออกแบบที่เก่า และไม่มีการลดจำนวนหน้าของการเขียน แยกหน้าออกมาจากหน้า Submit ดังต่อไปนี้



รูปภาพที่ 4. 61 รายงานสถิติ (Summary) หน้า Fees Report ระบบเดิม

จากภาพที่ 4.61 เป็นหน้ารายงานสถิติ (Summary) ต่อจากหน้า Search Report List จากภาพ 4.55 เพื่อค้นหารายงานสถิติแบบสรุป โดยกรอกเงื่อนไขในการค้นหา และกดค้นหา เพื่อแสดงเป็น PDF ใน Window ใหม่

ระบบใหม่ เปลี่ยน PDF แบบ Window เป็น Modal สดการ Render หน้าใหม่ ซึ่งจะแสดงเอกสารนี้เป็นการแจ้งเตือนหรือการแจ้งเตือนให้ทราบเกี่ยวกับเอกสารที่ส่งมอบให้ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 62 รายงานสถิติ (Summary) หน้า Fees Report

จากภาพที่ 4.62 เป็นหน้ารายงานสถิติ (Summary) ต่อจากหน้า Search Report List จากภาพ 4.57 เพื่อค้นหารายงานสถิติแบบสรุป โดยกรอกเงื่อนไขในการค้นหา และกดค้นหา เพื่อแสดงเป็น PDF แบบ Modal หน้าเดิม

8.3) รายงานสรุปค่าธรรมเนียม (รายเดือน)

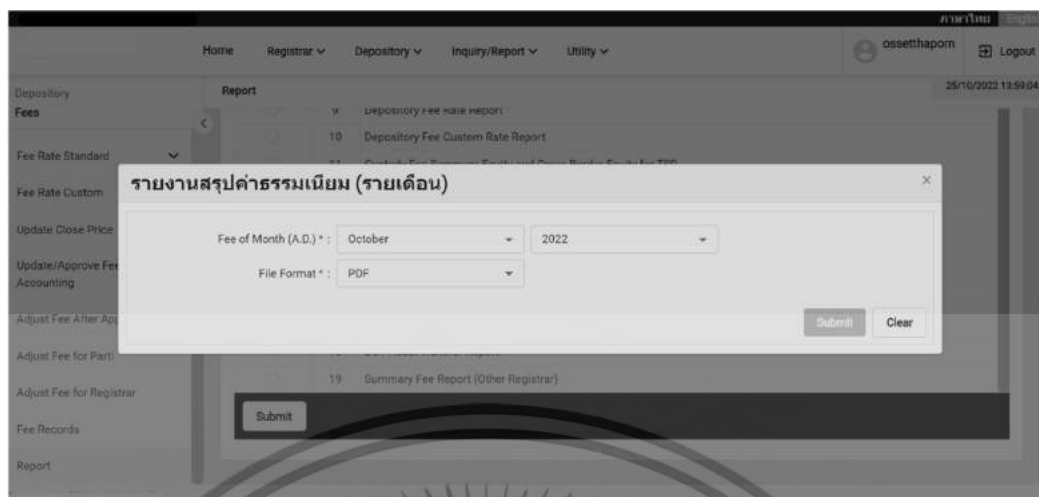
ระบบเดิม เนื่องจากได้มีการเขียนไว้เมื่อนานมาแล้ว ทำให้ระบบมีการออกแบบที่เก่า และไม่มีการลดจำนวนหน้าของการเขียน แยกหน้าออกมาจากหน้า Submit ดังต่อไปนี้



รูปภาพที่ 4. 63 รายงานสรุปค่าธรรมเนียม (รายเดือน) หน้า Fees Report ระบบเดิม

จากภาพที่ 4.63 เป็นหน้ารายงานสรุปค่าธรรมเนียม (รายเดือน) ต่อจากหน้า Search Report List จากภาพ 4.55 เพื่อค้นหาค่าธรรมเนียมแบบสรุปในรูปแบบรายเดือน โดยกรอกเงื่อนไขในการค้นหา และกดค้นหา เพื่อแสดงเป็น PDF ใน Window ใหม่

ระบบใหม่ เปลี่ยน PDF แบบ Window เป็น Modal ลดการ Render หน้าใหม่ ซึ่งจะแสดงเอกสารนี้เพื่อให้กรอกเงื่อนไขหลังจากเลือกรายการที่ต้องการค้นหา ดังต่อไปนี้ อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 64 รายงานสรุปค่าธรรมเนียม (รายเดือน) หน้า Fees Report จากภาพที่ 4.64 เป็นหน้ารายงานสรุปค่าธรรมเนียม (รายเดือน) ต่อจากหน้า Search Report List จากภาพ 4.57 เพื่อค้นหาค่าธรรมเนียมแบบสรุปในรูปแบบรายเดือน โดยกรอกเงื่อนไขในการค้นหา และกดค้นหา เพื่อแสดงเป็น PDF แบบ Modal หน้าเต็ม

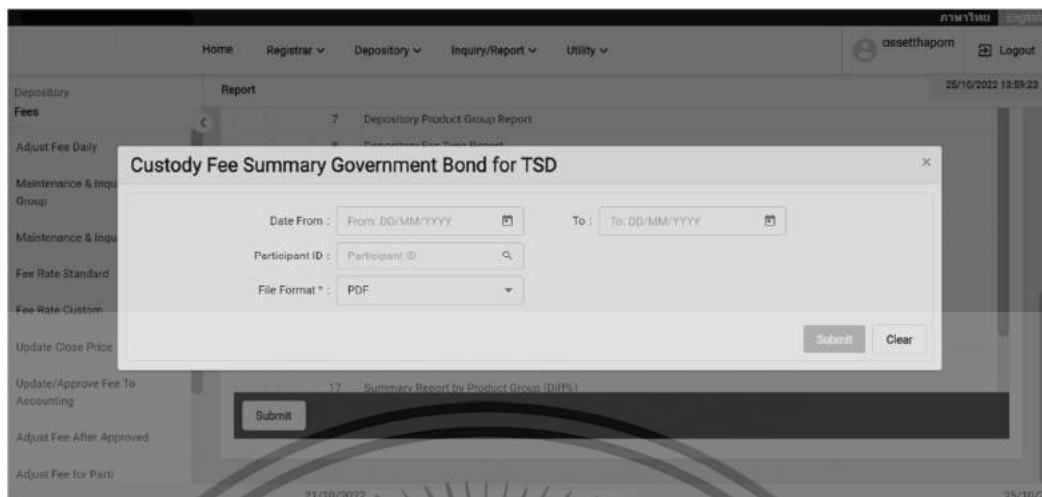
8.4) Custody Fee Summary Government Bond

ระบบเดิม เนื่องจากการเขียนไว้เมื่อนานมาแล้ว ทำให้ระบบมีการออกแบบที่เก่า และไม่มีการลดจำนวนหน้าของการเขียน แยกหน้าออกมาจากหน้า Submit ดังต่อไปนี้



รูปภาพที่ 4. 65 Custody Fee Summary Government Bond หน้า Fees Report ระบบเดิม จากภาพที่ 4.65 เป็นหน้า Custody Fee Summary Government Bond ต่อจากหน้า Search Report List จากภาพ 4.55 เพื่อค้นหาค่าธรรมเนียมการดูแลของพันธบัตรรัฐบาลแบบสรุป โดยกรอกเงื่อนไขในการค้นหา และกดค้นหา เพื่อแสดงเป็น PDF ใน Window ใหม่

ระบบใหม่ เปลี่ยน PDF แบบ Window เป็น Modal ลดการ Render หน้าใหม่ ซึ่งจะแสดงเอกสารนี้เพื่อให้กรอกเงื่อนไขหลังจากเลือกรายการที่ต้องการค้นหา ดังต่อไปนี้ อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 66 Custody Fee Summary Government Bond หน้า Fees Report จากภาพที่ 4.66 เป็นหน้า Custody Fee Summary Government Bond ต่อจากหน้า Search Report List จากภาพ 4.57 เพื่อค้นหาค่าธรรมเนียมการดูแลของพันธบัตรรัฐบาลแบบสรุป โดยกรอกเงื่อนไขในการค้นหา และกดค้นหา เพื่อแสดงเป็น PDF แบบ Modal หน้าเดิม

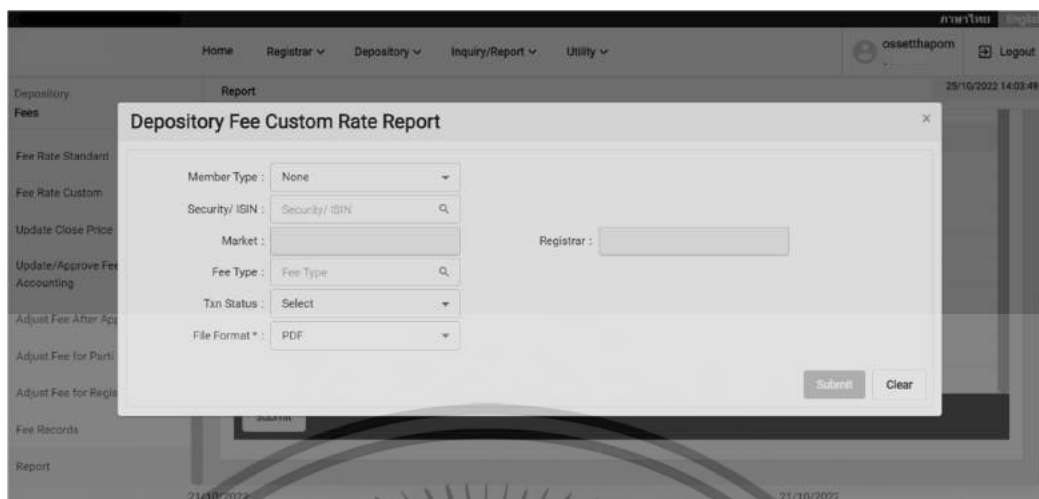
8.5) Deposit Fee Custom Rate Report

ระบบเดิม เนื่องจากได้มีการเขียนไว้เมื่อนานมาแล้ว ทำให้ระบบมีการออกแบบที่เก่า และไม่มีการลดจำนวนหน้าของการเขียน แยกหน้าออกมาจากหน้า Submit ดังต่อไปนี้



รูปภาพที่ 4. 67 Depository Fee Rate หน้า Fees Report ระบบเดิม จากภาพที่ 4.67 เป็นหน้า Deposit Fee Custom Rate Report ต่อจากหน้า Search Report List จากภาพ 4.55 เพื่อค้นหาอัตราค่าธรรมเนียมรับฝาก โดยกรอกเงื่อนไขในการค้นหา และกดค้นหา เพื่อแสดงเป็น PDF ใน Window ใหม่

ระบบใหม่ เปลี่ยน PDF แบบ Window เป็น Modal ลดการ Render หน้าใหม่ ซึ่งจะแสดงเอกสารนี้เพื่อให้กรอกเงื่อนไขหลังจากเลือกรายการที่ต้องการค้นหา ดังต่อไปนี้ อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 68 Depository Fee Rate หน้า Fees Report

จากภาพที่ 4.68 เป็นหน้า Deposit Fee Custom Rate Report ต่อจากหน้า Search Report List จากภาพ 4.57 เพื่อค้นหาอัตราค่าธรรมเนียมรับฝาก โดยกรอกเงื่อนไขในการค้นหา โดยกรอกเงื่อนไขในการค้นหา และกดค้นหา เพื่อแสดงเป็น PDF แบบ Modal หน้าเดิม

8.6) Adjust Fee Report

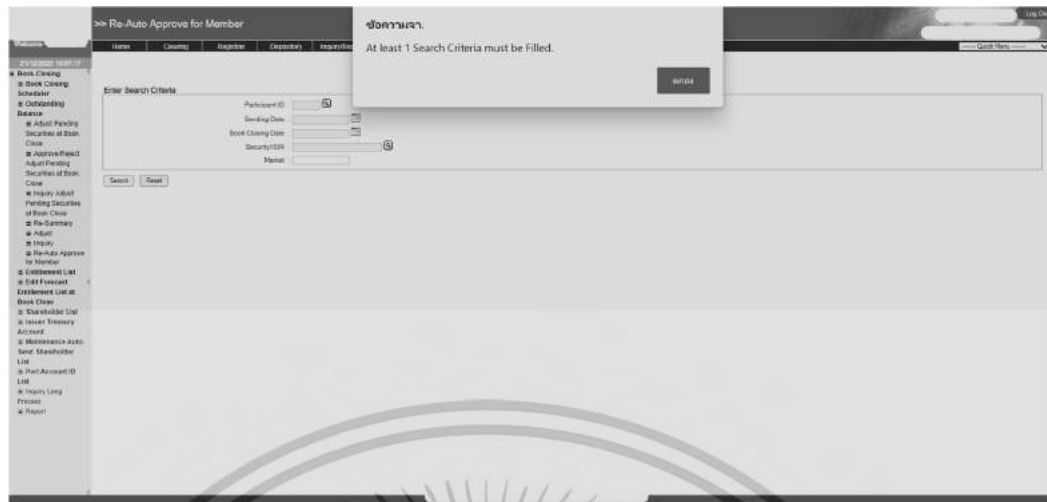
ระบบเดิม เนื่องจากการเขียนไว้เมื่อนานมาแล้ว ทำให้ระบบมีการออกแบบที่เก่า และไม่มีการลดจำนวนหน้าของการเขียน แยกหน้าออกมาจากหน้า Submit ดังต่อไปนี้



รูปภาพที่ 4. 69 Adjust Fee หน้า Fees Report ระบบเดิม

จากภาพที่ 4.69 เป็นหน้า Adjust Fee Report ต่อจากหน้า Search Report List จากภาพ 4.55 เพื่อค้นหาอัตราค่าปรับ โดยกรอกเงื่อนไขในการค้นหา และกดค้นหา เพื่อแสดงเป็น PDF ใน Window ใหม่

ระบบใหม่ เปลี่ยน PDF แบบ Window เป็น Modal ลดการ Render หน้าใหม่ ซึ่งจะแสดงเอกสารนี้เป้าให้กรอกเงื่อนไขหลังจากเลือกรายการที่ต้องการค้นหา ดังต่อไปนี้ อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



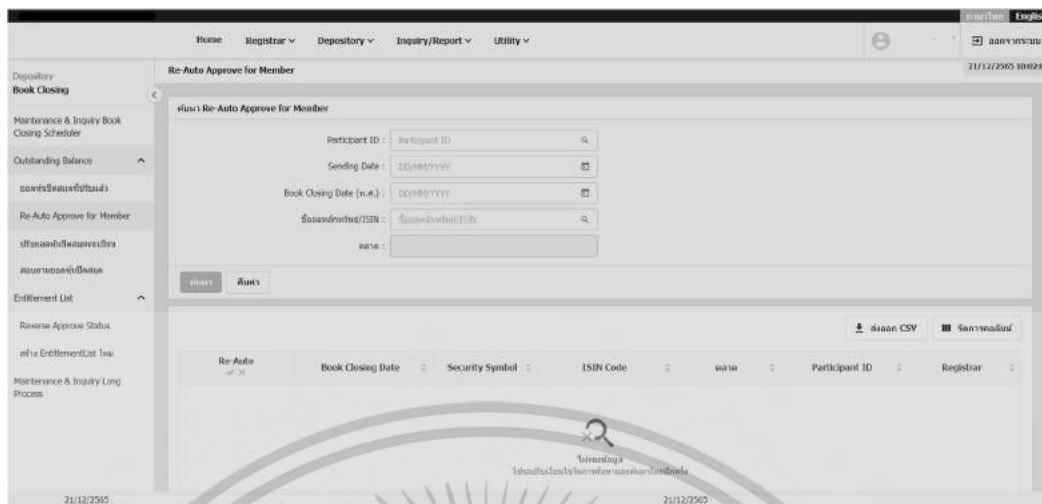
รูปภาพที่ 4. 72 Alert หน้า Re-Auto Approve for Member ระบบเดิม
จากรูปภาพที่ 4.72 เป็นหน้าแสดง Alert สำหรับแจ้งผู้ใช้งานให้กรอกเงื่อนไขอย่างน้อย 1 เงื่อนไขในการค้นหาข้อมูล Re-Auto Approve for Member หากต้องการระบุให้รายการมีสถานะเป็น Re-Auto ต้องค้นหารายการโดยกดปุ่ม Search เพื่อให้รายการแสดงที่ตาราง และเลือกรายการที่ต้องการเลือก



รูปภาพที่ 4. 73 Search Result หน้า Re-Auto Approve for Member ระบบเดิม
จากรูปภาพที่ 4.73 เป็นหน้าแสดง Search Result สำหรับทำการระบุให้รายการมีสถานะเป็น Re-Auto ต้องค้นหาจากด้านบนก่อนกดปุ่ม Search จากรูปภาพที่ 4.71 เพื่อให้รายการแสดงที่ตาราง และเลือกรายการที่ต้องการระบุสถานะ พร้อมกรอกเงื่อนไขที่ต้องการ Re-Auto ก่อนกด Submit เพื่อยืนยันรายการ

ระบบใหม่ เปลี่ยนแปลงให้มีลักษณะเป็น Default ที่มีการค้นหา และมีผลลัพธ์แสดงในรูปแบบตาราง โดย Re-Auto เพื่อเปลี่ยนสถานะเป็น Re-Auto ซึ่งการค้นหารายการที่ Re-Auto ต้องใส่เงื่อนไขตามที่ฟอร์มกำหนดให้ใส่ * ก่อนเพื่อค้นหา ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 74 Search หน้า Re-Auto Approve for Member

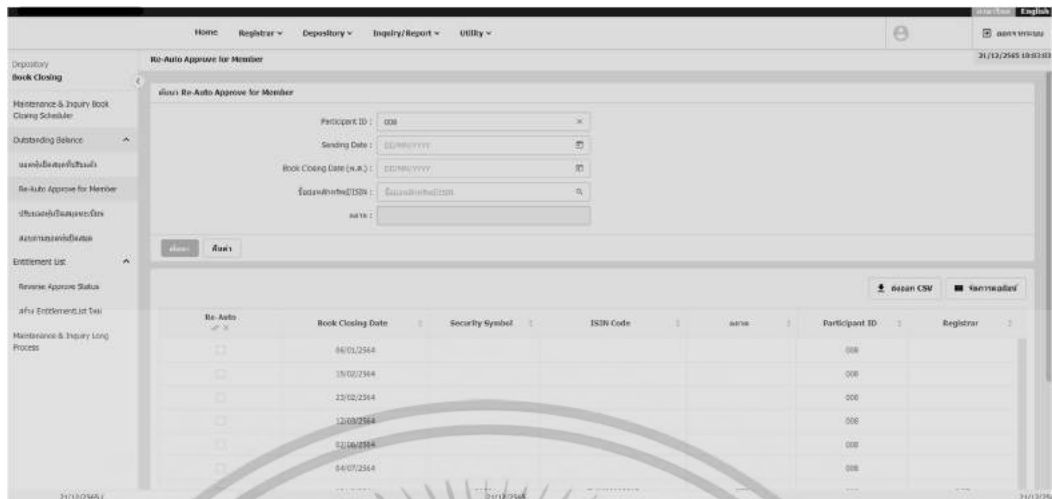
จากรูปภาพที่ 4.74 เป็นหน้า Search สำหรับค้นหาข้อมูลที่ต้องการ Re-Auto ซึ่งสามารถค้นหาได้แก่ผู้ที่มีสิทธิในการแก้ไขเท่านั้น โดยสามารถค้นหาได้จากกรอกรหัสเงื่อนไข หรือไม่ต้องการระบุเงื่อนไข ก่อนกดปุ่ม Submit เพื่อแสดงตาราง



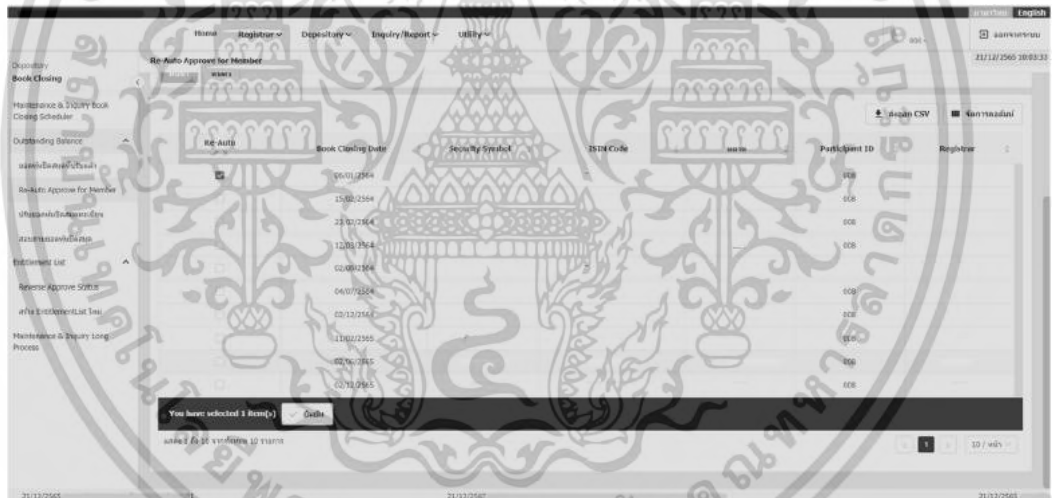
รูปภาพที่ 4. 75 Alert Search หน้า Re-Auto Approve for Member

จากรูปภาพที่ 4.75 เป็นหน้าแสดง Alert สำหรับแจ้งผู้ใช้งานให้กรอกเงื่อนไขอย่างน้อย 1 เงื่อนไขในการค้นหาข้อมูล Re-Auto Approve for Member หากต้องการระบุให้รายการมีสถานะเป็น Re-Auto ต้องค้นหารายการโดยกดปุ่ม Search เพื่อให้รายการแสดงที่ตาราง และเลือกรายการที่ต้องการเลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

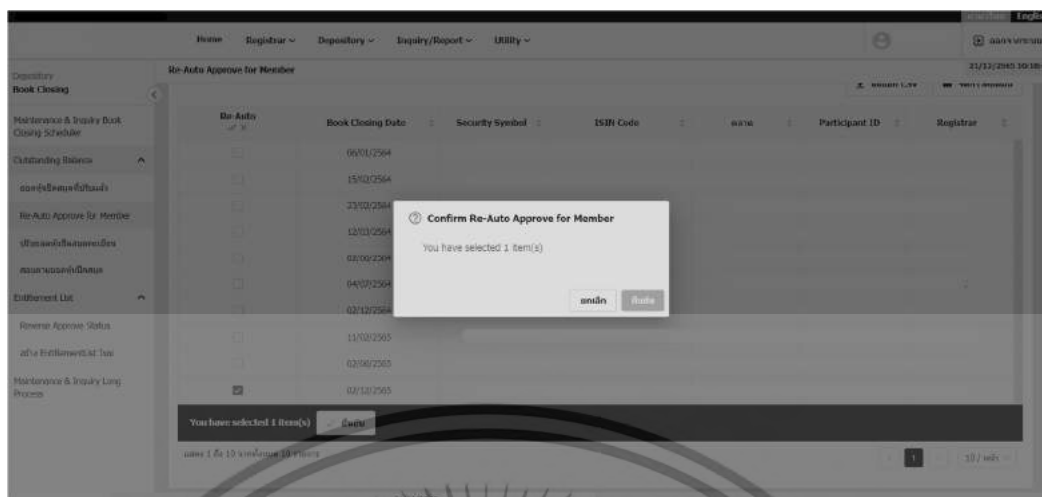


รูปภาพที่ 4. 76 Search Result หน้า Re-Auto Approve for Member
จากรูปภาพที่ 4.76 เป็นหน้าแสดงผลการค้นหา สำหรับเลือกรายการที่สามารถ Re-Auto ได้ ในหนึ่งครั้งสามารถเลือกได้หลายรายการที่ต้องการ Re-Auto โดยตารางจะแสดงข้อมูลที่ดึงมาจาก API หากต้องการ Re-Auto ต้องกดปุ่ม Submit ด้านล่างสุดของตารางจากรูปภาพที่ 4.77



รูปภาพที่ 4. 77 Re-Auto Submit หน้า Re-Auto Approve for Member
จากรูปภาพที่ 4.77 เป็นหน้าแสดงปุ่ม Submit สำหรับยืนยันว่าต้องการ Re-Auto จำนวนกี่รายการ หลังจากเลือกรายการที่ต้องการเสร็จ และกดปุ่ม Submit จะแสดง Alert Submit จากรูปภาพที่ 4.78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 78 Alert Submit หน้า Re-Auto Approve for Member
 จากรูปภาพที่ 4.78 เป็นหน้าต่างแสดง Alert Submit สำหรับยืนยันว่าจำนวนรายการที่ต้องการ Re-Auto ถูกต้อง และให้ดำเนินการต่อได้ หลังจากเลือกรายการที่ต้องการ หลัง Submit จะทำการตรวจสอบรายการ Re-Auto และแจ้ง Status จากรูปภาพที่ 4.79



รูปภาพที่ 4. 79 Status หน้า Re-Auto Approve for Member
 จากรูปภาพที่ 4.79 เป็นหน้าต่างแสดง Status สำหรับแสดงสถานะของรายการ Re-Auto ซึ่งหากมีสถานะผิดพลาดจะแสดงรายการที่ผิดพลาดตามจำนวนรายการที่ผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10) โมดูลย่อย Book Closing หน้า Generate Shareholder Entitlement List and Right Revok ระบบเดิม มีการออกแบบจะแสดงผลพร้อมตารางเมื่อมีการค้นหา และตารางไม่สามารถดาวน์โหลดไฟล์ .csv ได้ พร้อมทั้งมีปุ่มสำหรับ Reset ตารางให้กลับไปเป็นตารางว่าง ดังต่อไปนี้



รูปภาพที่ 4. 80 Search หน้า Generate Entitlement ระบบเดิม
จากรูปภาพที่ 4.80 เป็นหน้าแสดง Search เพื่อค้นหาโดยกรอกเงื่อนไข ก่อนทำการเลือกข้อมูลที่ต้องการ View โดยแต่ละ Text Field จะมีเงื่อนไขในการกรอกแตกต่างกัน เช่น Participant ID และ Security ID รูปแบบเมื่อกดให้แสดง Lookup, Sending Date และ Book Closing Date รูปแบบที่กดเป็น Date Calendar เป็นต้น



รูปภาพที่ 4. 81 Search Result หน้า Generate Entitlement ระบบเดิม
จากรูปภาพที่ 4.81 เป็นหน้าแสดง Search Result สำหรับทำการระบุให้รายการมีสถานะเป็น Generate ต้องค้นหาจากด้านบนก่อนกดปุ่ม Search จากรูปภาพที่ 4.80 เพื่อให้รายการแสดงที่

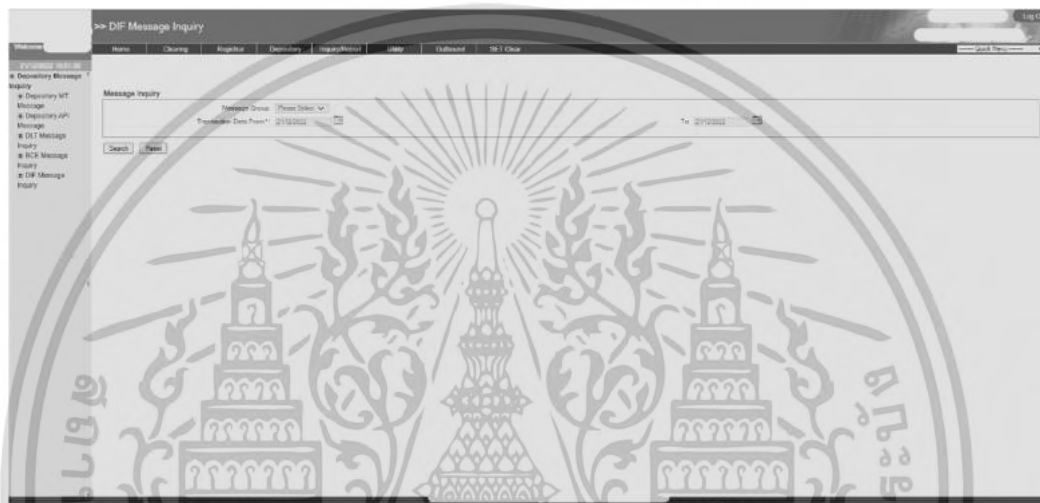
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 โมดูล Inquiry/ Report

เป็นโมดูลหลักในระบบที่มีระบบย่อย โดยมีการพัฒนาโมดูลเปรียบเทียบกับหน้าจอเก่า ดังต่อไปนี้

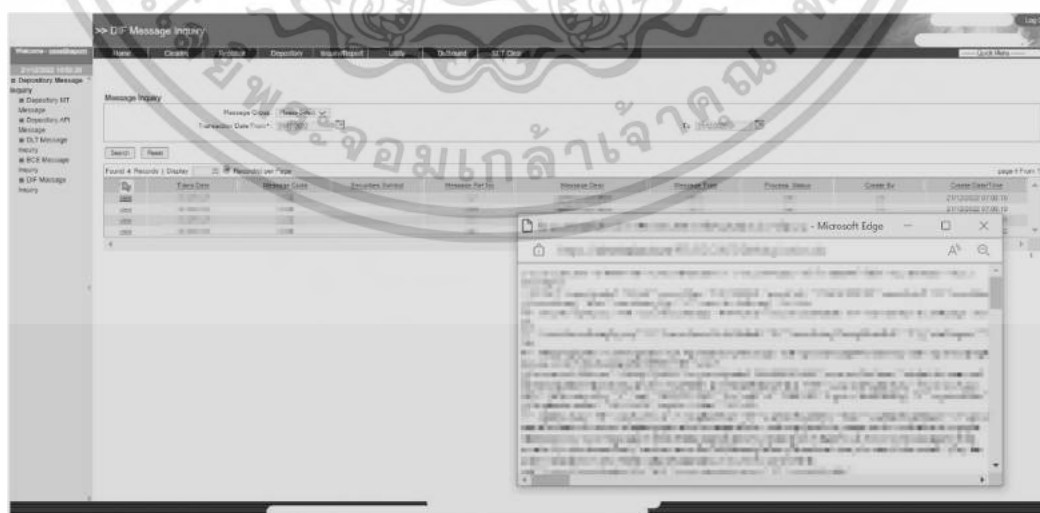
1) โมดูลย่อย Depository Message Inquiry หน้า DIF Message Inquiry

ระบบเดิม เนื่องจากได้มีการเขียนไว้เมื่อนานมาแล้ว ทำให้ระบบมีการออกแบบที่เก่า และไม่มีการลดจำนวนหน้าของการเขียน มีการแสดงผลหน้าจอทั้งหมด ดังต่อไปนี้



รูปภาพที่ 4. 86 Search หน้า DIF Message Inquiry ระบบเดิม

จากรูปภาพที่ 4.86 เป็นหน้าแสดง Search สำหรับค้นหาข้อมูลที่ต้องการแก้ไข ซึ่งสามารถค้นหาได้แก่ผู้ที่มีสิทธิในการแก้ไขเท่านั้น โดยสามารถค้นหาได้จากการระบุเงื่อนไข หรือไม่ต้องระบุเงื่อนไข ก่อนกดปุ่ม Submit เพื่อแสดงตารางการค้นหา

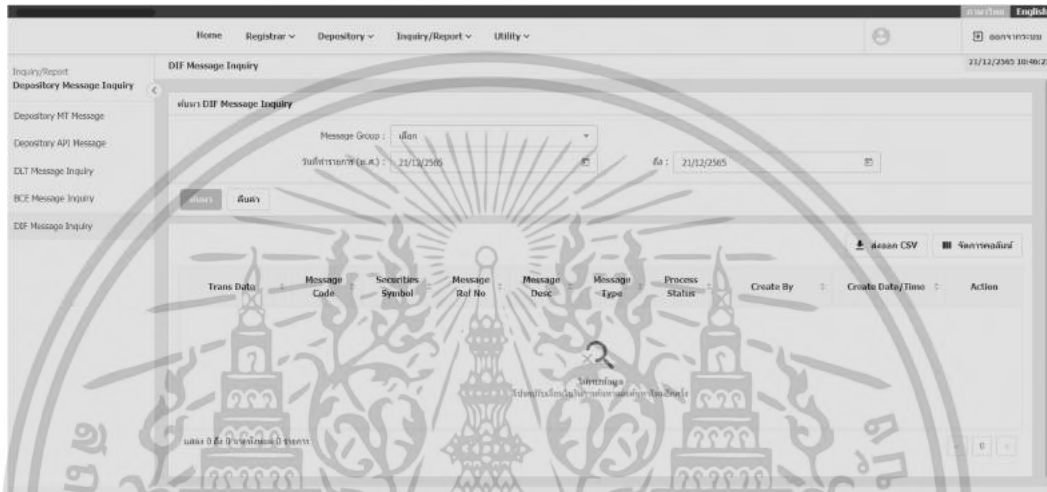


รูปภาพที่ 4. 87 View หน้า DIF Message Inquiry ระบบเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปภาพที่ 4.87 เป็นหน้าแสดง View สำหรับแสดงข้อมูล DIF Message Inquiry ที่ต้องการดูข้อมูลรูปแบบ JSON จะมีข้อมูลได้มาจากการดึงมาจาก API สามารถเรียงลำดับตามจากน้อยไปมาก หรือมากไปน้อย หลังจากรายการแสดงขึ้นมาสามารถกดปุ่ม View เพื่อง่ายสำหรับดูรายละเอียด จึงมีการแยกอีกหน้าต่าง

ระบบใหม่ มีการเปลี่ยนแปลงเป็น Modal ที่มีการออกแบบตารางผลลัพธ์แบบ Default ดังต่อไปนี้



รูปภาพที่ 4. 88 Search หน้า DIF Message Inquiry

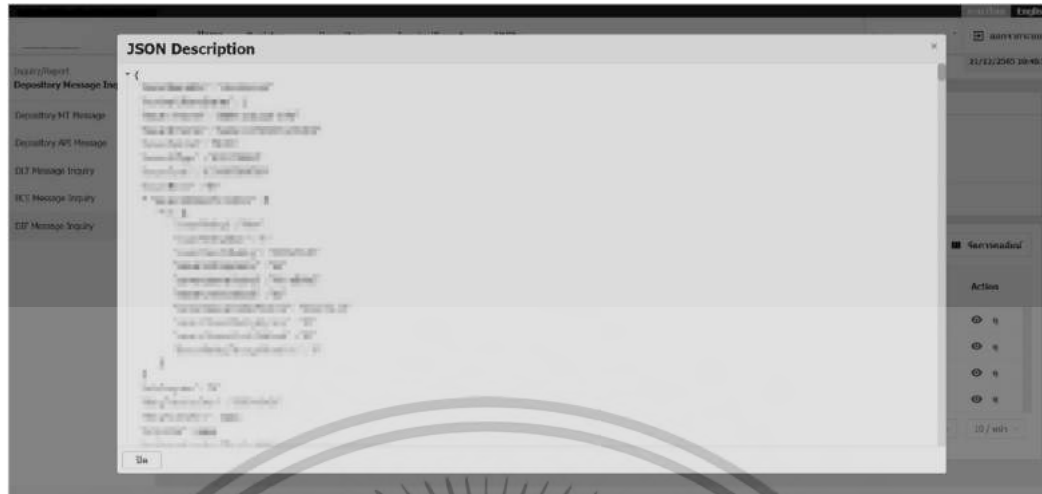
จากรูปภาพที่ 4.88 เป็นหน้า Search สำหรับค้นหาข้อมูลที่ต้องการดู โดยสามารถค้นหาได้แบบระบุเงื่อนไข หรือไม่ต้องระบุเงื่อนไข ก่อนกดปุ่ม Submit เพื่อแสดงตาราง



รูปภาพที่ 4. 89 Search Result หน้า DIF Message Inquiry

จากรูปภาพที่ 4.89 เป็นหน้าแสดงผลการค้นหา สำหรับเลือกรายการที่ต้องการ โดยตารางจะแสดงข้อมูลที่ดึงมาจาก API หากต้องการดู สามารถกดปุ่ม View ได้ในคอลัมน์สุดท้ายของ

ตารางผลลัพธ์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 90 JSON Modal หน้า DIF Message Inquiry

จากรูปภาพที่ 4.90 เป็นหน้าที่แสดง JSON Modal ที่ได้หลังจากการกดปุ่ม View ในรูปภาพที่ 4.89 จะแสดงข้อมูลในรูปแบบ JSON ที่มีการจัดรูปแบบให้อ่านง่าย

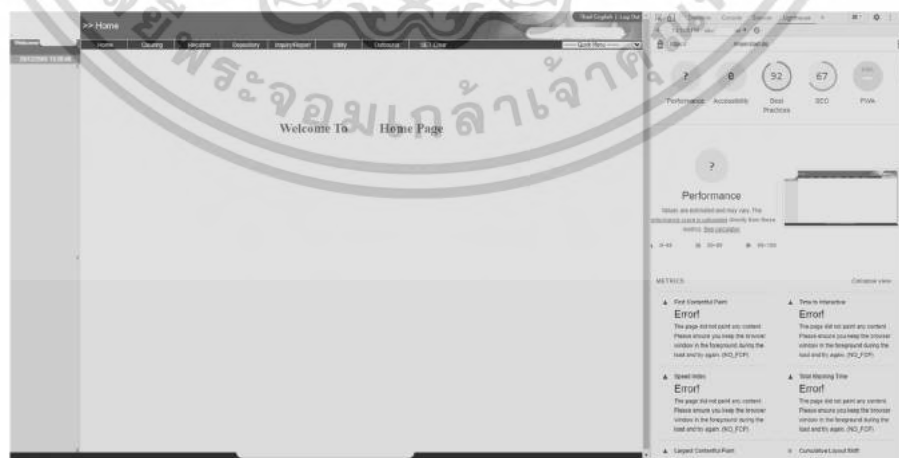
4.2 ผลการเปรียบเทียบประสิทธิภาพการทำงาน

การเปรียบเทียบจะใช้ Lighthouse ในการวิเคราะห์โดยทดสอบ 3 โหมด Performance, Accessibility, Best Practices, SEO และ PWA ได้ผลลัพธ์ดังต่อไปนี้

4.2.1 โหมด Navigation

มีการวัดค่าทั้งหมด 4 เกณฑ์ คือ Performance และ Best Practices ซึ่งสามารถวัดประสิทธิภาพของเว็บแอปพลิเคชัน ดังต่อไปนี้

ระบบเก่า เนื่องจากเป็นระบบที่มีการพัฒนานานแล้ว ส่งผลให้ประสิทธิภาพของเว็บแอปพลิเคชันไม่ตรงตามเกณฑ์ ทำให้มีค่าที่ไม่สามารถประเมินได้



รูปภาพที่ 4. 91 รายงานของโหมด Navigation ระบบเก่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปภาพที่ 4.91 เป็นหน้าที่แสดงรายงานของโหมด Navigation ของระบบเก่า ที่ถูกพัฒนาหน้า โดยสามารถแจกแจงผลลัพธ์แยกตามเกณฑ์ ดังนี้

? ไม่สามารถประเมินได้ หมายถึงไม่มีตามเงื่อนไขกำหนด

0 – 49 หมายถึง ไม่ดี

50 – 89 หมายถึง ต้องพัฒนาเพิ่มอีก

90 – 100 หมายถึง ดี

จากการเกณฑ์การประเมินระบบของ Lighthouse สามารถอธิบายวิธีการเพิ่มประสิทธิภาพจากการประเมินค่าของระบบเก่า ดังต่อไปนี้

1) ค่า Performance ไม่สามารถประเมินได้ เนื่องจากในเนื้อหาต่างๆของหน้าเว็บของระบบเก่าไม่ได้มีการโหลดไว้อ่อนที่จะเข้าหน้าเว็บเลยไม่สามารถประเมินได้ จากรูปภาพที่ 4.91 จะเห็นได้ว่าค่าใน Metrics ไม่สามารถวิเคราะห์ได้เลย ยกเว้นค่า CLS คือค่าเอาไววัดค่าความมั่นคงทางสายตา ตอนที่เกิดการกระทำต่างๆ แล้วเปลี่ยนหน้าจอ ถ้าหากยังมีค่า CLS เยอะยิ่งทำให้ User Experience ไม่ได้ โดยสามารถแสดงค่าการประเมิน Performance ตามหัวข้อ 6 หัวข้อรูปแบบตาราง เรียกว่า Metrics ซึ่งเกณฑ์การประเมินตามลำดับ คือ ดี ปานกลาง ไม่ดี มีเกณฑ์การเปรียบเทียบตามตารางที่ 4.1

ตารางที่ 4. 1 ตารางเปรียบเทียบค่า Metrics สำหรับค่า Performance

หัวข้อ	ดี (วินาที)	ปานกลาง (วินาที)	ไม่ดี (วินาที)
ค่าแรกที่ถูกวาดขึ้น (FCP)	< 0.95	< 1.6	>= 1.6
เวลาในการตอบสนองกลับหลังเข้าเว็บ (TTI)	< 2.5	< 4.53	>= 4.53
ความเร็วในการโหลดเนื้อหา (SI)	< 1.33	< 2.32	>= 2.32
Blocking Time รวม (รวมความเร็ว FCP และ Time to Interaction เมื่อการทำงานมากกว่า 50 ms)	< 0.15	< 0.356	>= 0.356
เวลาเมื่อวาดเนื้อหาที่ใหญ่ที่สุด (LCP)	< 1.2	< 2.42	>= 2.42
ค่าวัดการเคลื่อนไหวที่เห็นได้ (CLS)	< 0.1	< 0.25	>= 0.25

จากตาราง 4.1 เป็นตารางเปรียบเทียบเกณฑ์ของค่าที่นำมาคำนวณเป็น Metrics ที่นำมาคำนวณเป็นค่า Performance ซึ่งจะมีทั้งหมด 6 ค่าที่ใช้คำนวณ โดยแต่ละค่าจะเปรียบเทียบจากเวลาที่มีหน่วยวัดเป็นวินาที ยกเว้นค่าวัดการเคลื่อนไหวที่ไม่มีหน่วยวัด ซึ่งนำมาเปรียบเทียบกับระบบพัฒนาได้ผลลัพธ์ตามตารางที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4. 2 ตารางแสดง Performance ของระบบเก่า

ค่าแรกที่ถูกวาดขึ้น (FCT)	Error	ไม่สามารถประเมินได้
เวลาในการตอบสนองกลับหลังเข้าเว็บ (TTI)	Error	ไม่สามารถประเมินได้
ความเร็วในการโหลดเนื้อหา	Error	ไม่สามารถประเมินได้
Blocking Time (รวมความเร็ว FCP และ Time to Interaction เมื่อการทำงานมากกว่า 50 ms)	Error	ไม่สามารถประเมินได้
เวลาเมื่อวาดเนื้อหาที่ใหญ่ที่สุด (LCP)	Error	ไม่สามารถประเมินได้
ค่าวัดการเคลื่อนไหวที่เห็นได้ (CLS)	0.107	สูงกว่าเกณฑ์

จากตารางที่ 4. 2 แสดงการวัด Performance สรุปได้ว่า ค่าแรกที่ถูกวาดขึ้น (FCT), เวลาในการตอบสนองกลับหลังเข้าเว็บ (TTI), ความเร็วในการโหลดเนื้อหา, Blocking Time, เวลาเมื่อวาดเนื้อหาที่ใหญ่ที่สุด (LCP) ไม่สามารถประเมินค่าได้ หรือ Error เนื่องจากมีคุณสมบัติไม่ถึงเกณฑ์หลายเงื่อนไข โดยมีตัวอย่างปัญหา ดังนี้

รูปภาพที่ 4. 92 ตัวอย่าง Network ที่ใช้ HTTP/1 ระบบเก่า

จากรูปภาพที่ 4.92 เป็นภาพที่แสดงตัวอย่าง Network ที่ใช้ HTTP/1 ของระบบเก่า จะเห็นได้ว่ามีหลายไฟล์ที่ใช้งานดึงจาก Network ที่ใช้ HTTP/1 ซึ่งควรจะใช้เป็น HTTP/2 ที่มีความปลอดภัยมากขึ้น

URL	Cache TTL	Transfer Size
...scripts/bigdecimal.js	None	39 KiB
...scripts/clientForm.js	None	21 KiB
...scripts/calendar.js	None	15 KiB
...scripts/activeui.js	None	13 KiB

รูปภาพที่ 4. 93 ตัวอย่าง Cache Policy ระบบเก่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปภาพที่ 4.93 เป็นภาพแสดงตัวอย่าง Cache Policy ของระบบเก่า จะเห็นได้ว่ามีการเก็บทรัพยากรต่างๆ หรือ Asset ที่ไม่เป็นไปตาม Cache Policy กำหนดกฎที่ใช้ในการพิจารณาว่าสามารถดำเนินการตามคำขอโดยใช้ Cache Copy ของแหล่งข้อมูลร้องขอ โดยจากรูปพบทั้งหมด 43 ไฟล์ โดยการเก็บ Cache จะช่วยทำให้เว็บสามารถทำงานเร็วขึ้น

2) ค่า Accessibility ประเมินได้ 0 เนื่องจากปัญหา การตั้งชื่อ จาก Document ของ HTML ที่เขียนไม่มีการใช้ title tag <title>, frame <frame> และ iframe tag <iframe> ไม่มี title tag <title> กลุ่ม การประกาศแบบสากล ยังไม่มีการประกาศ [lang] ได้ <html>

3) ค่า Best Practices ประเมินได้ 92 หน้าที่สร้างขึ้นทั้งหมดที่เป็น HTML ขาด doctype ทำให้เข้าสู่โหมด quirks (quirks-mode)

4) ค่า SEO ประเมินได้ 67 เนื่องจากในการตรวจสอบความสามารถตอนใช้บนโทรศัพท์ หา meta tag <meta name="viewport"> ไม่พบทำให้ไม่สามารถกำหนดความกว้าง ความสูงได้ ด้านเนื้อหาของเว็บไม่พบ <title>

ระบบใหม่ มีการพัฒนาโดยนำเทคโนโลยีใหม่มาช่วย ทำให้สามารถจัดการปัญหาหลายๆส่วนที่ส่งผลให้ประสิทธิภาพของเว็บแอปพลิเคชันได้ ทำให้มีค่าประเมินตามเกณฑ์



รูปภาพที่ 4. 94 รายงานของโหมต Navigation ระบบใหม่

จากรูปภาพที่ 4.94 เป็นหน้าที่แสดงรายงานของโหมต Navigation ของระบบใหม่ ที่ถูกพัฒนา โดยสามารถแจกแจงผลลัพธ์แยกตามเกณฑ์ ดังนี้

? ไม่สามารถประเมินได้ หมายถึงไม่มีตามเงื่อนไขกำหนด

0 – 49 หมายถึง ไม่ดี

50 – 89 หมายถึง ต้องพัฒนาเพิ่มอีก

90 – 100 หมายถึง ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการเกณฑ์การประเมินระบบของ Lighthouse สามารถอธิบายวิธีการเพิ่มประสิทธิภาพจากการประเมินค่าของระบบเก่า ดังต่อไปนี้

1) ค่า Performance ประเมินได้ 76 อยู่ในเกณฑ์ที่ต้องการพัฒนาเพิ่มขึ้น โดยแก้ไขตามข้อเสนอแนะของ Lighthouse โดยสามารถแสดงค่าการประเมิน Performance ตามหัวข้อ 6 หัวข้อรูปแบบตาราง เรียกว่า Metrics โดยมีเกณฑ์การประเมินตามลำดับ คือ ดี ปานกลาง ไม่ดี มีเกณฑ์การเปรียบเทียบตามตารางที่ 4.3

ตารางที่ 4. 3 ตารางเปรียบเทียบค่า Metrics สำหรับค่า Performance

หัวข้อ	ดี (วินาที)	ปานกลาง (วินาที)	ไม่ดี (วินาที)
ค่าแรกที่ถูกวาดขึ้น (FCP)	< 0.95	< 1.6	>= 1.6
เวลาในการตอบสนองกลับหลังเข้าเว็บ (TTI)	< 2.5	< 4.53	>= 4.53
ความเร็วในการโหลดเนื้อหา (SI)	< 1.33	< 2.32	>= 2.32
Blocking Time รวม (รวมความเร็ว FCP และ Time to Interation เมื่อการทำงานมากกว่า 50 ms)	< 0.15	< 0.356	>= 0.356
เวลาเมื่อวาดเนื้อหาที่ใหญ่ที่สุด (LCP)	< 1.2	< 2.42	>= 2.42
ค่าวัดการเคลื่อนไหวที่เห็นได้ (CLS)	< 0.1	< 0.25	>= 0.25

จากตาราง 4.2 เป็นตารางเปรียบเทียบเกณฑ์ของค่าที่นำมาคำนวณเป็น Metrics ที่นำมาคำนวณเป็นค่า Performance ซึ่งจะมีทั้งหมด 6 ค่าที่ใช้คำนวณ โดยแต่ละค่าจะเปรียบเทียบจากเวลาที่มีหน่วยวัดเป็นวินาที ยกเว้นค่าวัดการเคลื่อนไหวที่ไม่มีหน่วยวัด ซึ่งนำมาเปรียบเทียบกับระบบพัฒนาได้ผลลัพธ์ตามตารางที่ 4.3

ตารางที่ 4. 4 ตารางแสดง Performance ของระบบใหม่

ค่าที่ประเมิน	เวลา (วินาที)	เกณฑ์
ค่าแรกที่ถูกวาดขึ้น (FCP)	1.5	ปานกลาง
เวลาในการตอบสนองกลับหลังเข้าเว็บ (TTI)	2.3	ดี
ความเร็วในการโหลดเนื้อหา	1.5	ปานกลาง
Blocking Time (รวมความเร็ว FCP และ Time to Interation เมื่อการทำงานมากกว่า 50 ms)	0.18	ปานกลาง
เวลาเมื่อวาดเนื้อหาที่ใหญ่ที่สุด (LCP)	2.3	ปานกลาง
ค่าวัดการเคลื่อนไหวที่เห็นได้ (CLS)	0	ดี

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ทางธุรกิจ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 4. 2 แสดงการวัด Performance เมื่อประเมินค่าของ Lighthouse จะเห็นได้ว่าหลายค่าที่ประเมินสามารถพัฒนาให้ถึงเกณฑ์ที่ได้ โดย Lighthouse ก็ได้แนะนำวิธีแก้ปัญหาตัวอย่าง ดังนี้

URL	Transfer Size	Potential Savings
/main.2c98f8e...js	729.1 KIB	482.1 KIB
chrome-extension://fmkadmapgofadopljbjfkapdkolenih/build/installHook.js	417.0 KIB	324.2 KIB
chrome-extension://fmkadmapgofadopljbjfkapdkolenih/build/react_devtools_backend.js	521.6 KIB	313.7 KIB
/material-ui.850cc7b...js	95.6 KIB	69.7 KIB

รูปภาพที่ 4. 95 ตัวอย่าง JavaScript ที่ไม่ได้ใช้งาน ระบบใหม่

จากรูปภาพที่ 4.95 เป็นภาพแสดงตัวอย่าง JavaScript ที่ไม่ได้ใช้งานของระบบใหม่ ที่ไม่ได้ใช้ เนื่องจากใช้เวลาถึง 0.96 วินาที และการลด JavaScript ที่ไม่ได้ใช้ หรือเลื่อนการโหลดไปจนกว่าจะมีการเรียกใช้งาน จะลดการใช้งานบนเครือข่ายมือถือได้มากขึ้น ในการใช้ React จะมีการ Render ในฝั่ง Client-side ทำให้ใช้ทรัพยากรในการโหลดไปเยอะ จากรูปด้านล่างเป็นไฟล์ JavaScript ที่มีขนาดใหญ่

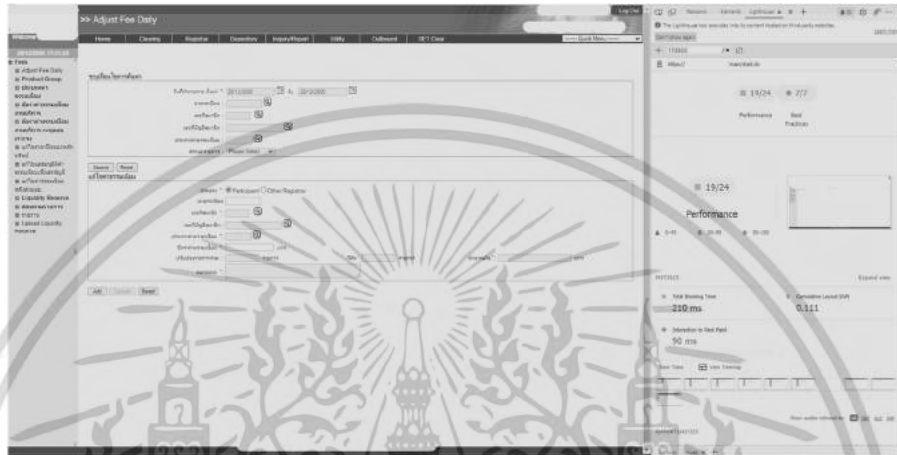
- 2) ค่า Accessibility ประเมินได้ 89 ตรวจจับปัญหาที่สามารถมาแก้ไขได้ เรื่องหมวดการตั้งชื่อ คือ ไม่มีตั้งชื่อให้ปุ่มที่มีไว้ให้ช่วยในการเข้าถึง
- 3) ค่า Best Practices ประเมินได้ 100 มีความน่าเชื่อถือ และปลอดภัยที่ดี โดยรวมของระบบดี มีการใช้ HTTPS, มีการอนุญาตให้วาง Password มีการวางรูปแบบภาพที่ถูกสัดส่วน หลีกเลี่ยงการใช้ API ที่เลิกใช้งาน และหน้าเพจมีการใช้ DOCTYPE เป็นต้น
- 4) ค่า SEO ประเมินได้ 90 เนื่องจากมีไม่สามารถหา <meta name="description"> พบ หากมีจะทำการค้นหาข้อมูลภายในเว็บที่สร้างง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 โหมด Timespan

มีการวัดค่าทั้งหมด 2 เกณฑ์ คือ Performance และ Best Practices ซึ่งสามารถวัดประสิทธิภาพของเว็บแอปพลิเคชัน ดังต่อไปนี้

ระบบเก่า สามารถวัดค่าออกมาได้ดี และมีระยะเวลาที่ดีในการเข้าถึงหน้าใหม่ๆ เนื่องจากมีการออกแบบที่เรียบง่าย ทำให้มี Element ไม่เยอะ



รูปภาพที่ 4. 96 รายงานของโหมด Timespan ระบบเก่า

จากภาพที่ 4.96 เป็นภาพแสดงรายงานของโหมด Timespan ที่พัฒนาในระบบเก่า โดยสามารถประเมินค่าได้ทั้ง 2 ค่า และ Lighthouse ได้เสนอวิธีการแก้ปัญหา ดังต่อไปนี้

- 1) Performance ประเมินได้ 19/24 แก้ปัญหาเพื่อเพิ่ม Performance โดยลด JavaScript และข้อมูลที่ดึงมาใช้ ควรใช้ HTTP/2
- 2) Best Practices ประเมินได้ 7/7 เนื่องจากการใช้ HTTPSs, มีการอนุญาตให้วาง Password มีการวางรูปแบบภาพที่ถูกสัดส่วน หลีกเลี่ยงการใช้ API ที่เลิกใช้งาน และหน้าเพจมีการใช้ DOCTYPE เป็นต้น

นอกจากวัดทั้งสองค่าแล้ว โหมด Timespan ยังสามารถวัดระยะเวลาในการเข้าถึงหน้าตัวอย่าง โดยสามารถสรุปค่าออกมาเป็น Chart ได้ตามรูปภาพที่ 4.97

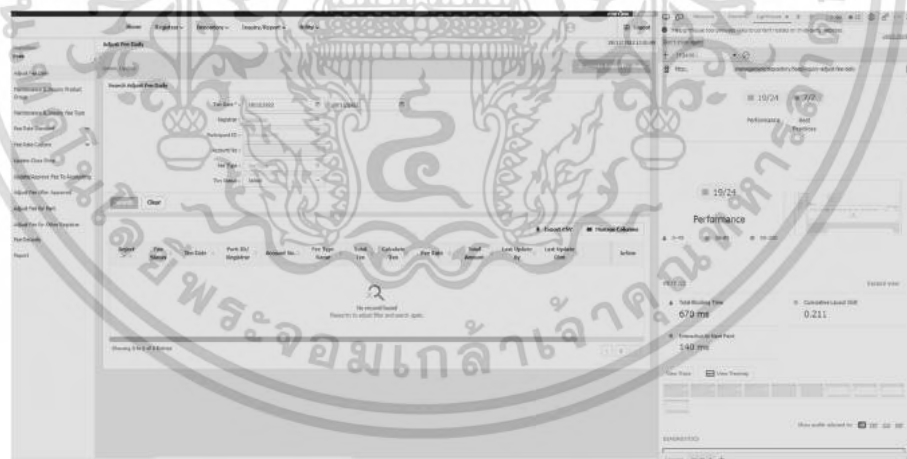
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 4. 97 ระยะเวลาการทำงานระบบเก่า

จากรูปที่ 4.97 เป็นภาพแสดงระยะเวลาการทำงานระบบเก่า หน้า Adjust Fee Daily จะเห็น Chart ที่แสดงระยะเวลาทั้งหมดตั้งแต่ที่เริ่มโหลดตอนแรก โหลดสคริปต์ การ Render การวาดเนื้อหาต่างๆ ขึ้นมา และการทำงานของระบบ และเวลาประมวลผลของส่วนที่ไม่ได้ใช้งาน โดยระยะเวลาที่แสดงในการทำงานต่าง ๆ ได้เปรียบเทียบกับตารางที่ 4.5

ระบบใหม่ มีการพัฒนาเพิ่มขึ้น แต่ยังคงการเขียนในหลาย ๆ ส่วนที่ยังไม่แก้ไขปัญหา ยังคงทำให้มีค่าประเมินที่เท่าเดิม หรือปัญหามากกว่าเดิม เนื่องจากการเปลี่ยนแปลงการออกแบบใหม่ เพื่อให้สวยงามมากขึ้น จึงใช้ระยะเวลาในบางการโหลดมากขึ้น



รูปภาพที่ 4. 98 รายงานของโหนด Timespan ระบบใหม่

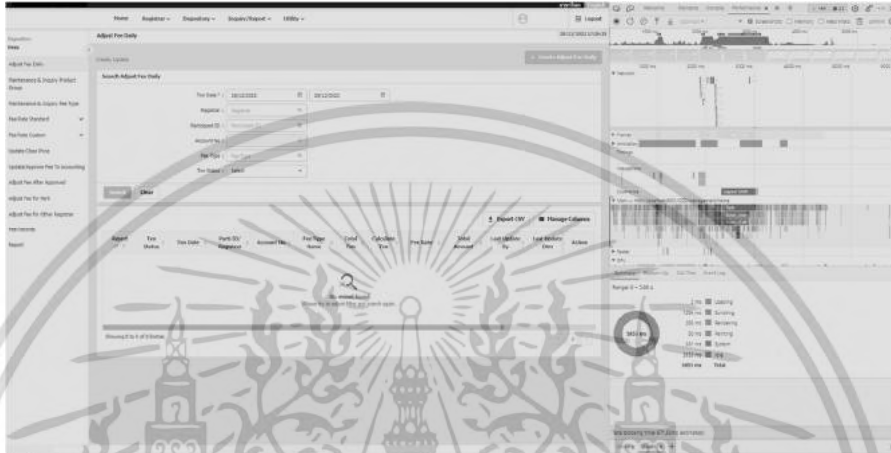
จากภาพที่ 4.98 เป็นภาพแสดงรายงานของโหนด Timespan ที่พัฒนาในระบบใหม่ โดยสามารถประเมินค่าได้ทั้ง 2 ค่า และ Lighthouse ได้เสนอวิธีการแก้ปัญหา ดังต่อไปนี้

1) Performance ประเมินได้ 19/24 เนื่องจากพบปัญหาของการไม่กำหนดขนาดความกว้าง และความสูงของรูปภาพ การเก็บ Asset ไม่ตาม Cache Policy ทำให้ต้องโหลดใหม่ทุกครั้ง เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) Best Practices ประเมินได้ 7/7 เนื่องจากการใช้ HTTPs, มีการอนุญาตให้วาง Password มีการวางรูปแบบภาพที่ถูกสัดส่วน หลีกเลี่ยงการใช้ API ที่เลิกใช้งาน และหน้าเพจมีการใช้ DOCTYPE เป็นต้น

นอกจากวัดทั้งสองค่าแล้ว โหมด Timespan ยังสามารถวัดระยะเวลาในการเข้าถึงหน้าตัวอย่าง โดยสามารถสรุปค่าออกมาเป็น Chart ได้ตามรูปภาพที่ 4.99



รูปภาพที่ 4. 99 ระยะเวลาการทำงานของระบบใหม่ จากรูปที่ 4.99 จะเห็น Chart ที่แสดงระยะเวลาทั้งหมดตั้งแต่ที่เริ่มโหลดตอนแรก โหลดสคริปต์ การ Render การวาดเนื้อหาต่างๆ ขึ้นมา และการทำงานของระบบ และเวลาประมวลผลของส่วนที่ไม่ได้ใช้งาน โดยระยะเวลาที่แสดงในการทำงานต่าง ๆ ได้เปรียบเทียบในตารางที่ 4.5

ตารางที่ 4. 5 ตารางแสดงผลการเปรียบเทียบระยะเวลาของระบบเก่า และใหม่

เวลา	ระบบเก่า (มิลลิวินาที)	ระบบใหม่ (มิลลิวินาที)
โหลดเนื้อหาทั้งหมด (Loading)	90	2
โหลดสคริปต์ (Scripting)	696	1284
การ Render	282	285
การวาดหน้าจอ Painting	64	50
ทำงานของระบบ System	372	301
ส่วนที่ไม่ได้ใช้งาน Idle	11406	3933
รวม	12910	5855

จากตารางที่ 4.5 สามารถสรุปได้ว่าระบบเก่ามีการใช้เวลามากกว่า ดังนี้ โหลดเนื้อหา เพราะว่าเป็นการโหลด DOM ทั้งหมดที่ใช้งานในโมดูลนี้ และส่วนไม่ได้ใช้งาน เนื่องจากโหลดทั้งหมดภายในโมดูลมาก่อนค่อยเลือกที่จะใช้งานทำให้เสียเวลาโหลดส่วนที่ไม่ได้ใช้งานด้วย การทำงานระบบใหม่ที่มีมากกว่า ดังนี้ การโหลดสคริปต์ เนื่องจากระบบใหม่มีการใช้ Library หลายตัวในการช่วยพัฒนาเอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ค่าที่ใกล้เคียงกันมีในเรื่องการ Render การวาดหน้าจอในส่วนต่างๆ ขึ้นมา และการทำงานในระบบ เนื่องจากมีลักษณะการทำงานที่เหมือนเดิม

4.2.3 โหมด Snapshot

มีเกณฑ์การวัดทั้งหมด 4 ค่า คือ Performance, Accessibility, Best Practices และ SEO ซึ่งสามารถวัดประสิทธิภาพของเว็บแอปพลิเคชัน ดังต่อไปนี้

ระบบเก่า สามารถวัดค่าออกมาได้ในทุกเกณฑ์ แต่มีบางค่าที่ยังสามารถปรับปรุงเพื่อให้ค่าของ Snapshot เพิ่มขึ้นได้



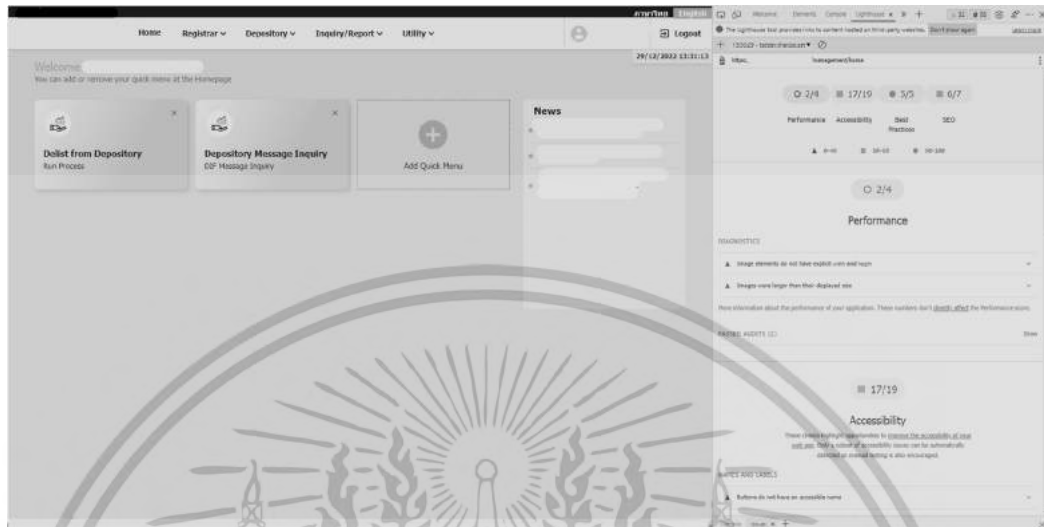
รูปภาพที่ 4. 100 รายงานของโหมด Snapshot ระบบเก่า

จากภาพที่ 4.100 เป็นภาพแสดงรายงานของโหมด Snapshot ที่พัฒนาในระบบเก่า โดยสามารถประเมินค่าได้ทั้ง 4 ค่า และ Lighthouse ได้เสนอวิธีการแก้ปัญหา ดังต่อไปนี้

- 1) Performance ประเมินได้ 2/3 ไม่มีการใช้ DOM ที่ใหญ่เกินไป รูปภาพมีการกำหนดสูง และกว้าง เหมาะกับการแสดงผล แต่พบปัญหาเรื่อง Viewport ที่ไม่ได้กำหนดความสูง และความกว้าง ทำให้ตอนใช้งานบนโทรศัพท์จะไม่ Responsive
- 2) Accessibility ประเมินได้ 0/3 เนื่องจากปัญหา การตั้งชื่อ จาก Document ของ HTML ที่เขียน ไม่มีการใช้ title tag <title>, frame <frame> และ iframe tag <iframe> ไม่มี title tag <title> กลุ่ม การประกาศแบบสากล ยังไม่มีการประกาศ [lang] ใต้ <html>
- 3) Best Practice ประเมินได้ 4/5 เนื่องจากหน้าที่สร้างขึ้นทั้งหมดที่เป็น HTML ขาด doctype ทำให้เข้าสู่โหมด quirks (quirks-mode)
- 4) SEO เนื่องจากในการตรวจสอบความสามารถตอนใช้บนโทรศัพท์ หา meta tag <meta name="viewport"> ไม่พบทำให้ไม่สามารถกำหนดความกว้าง ความสูงได้ ด้านเนื้อหาของเว็บไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบใหม่ มีค่าในการประเมินที่สูงหลายค่า และยังมีค่าที่สามารถปรับปรุงตามปัญหาที่พบจากการประเมิน Lighthouse เพื่อเพิ่มค่าให้สูงขึ้นได้



รูปภาพที่ 4. 101 รายงานของโหนด Snapshot ระบบใหม่

จากภาพที่ 4.101 เป็นภาพแสดงรายงานของโหนด Snapshot ที่พัฒนาในระบบใหม่ โดยสามารถประเมินค่าได้ทั้ง 4 ค่า และ Lighthouse ได้เสนอวิธีการแก้ปัญหา ดังต่อไปนี้

- 1) Performance ประเมินได้ 2/4 เนื่องจากองค์ประกอบภาพรูปภาพไม่มีการกำหนดความกว้าง และความสูงที่ชัดเจน จึงควรกำหนดเพื่อให้ค่า CLS ดีขึ้นเมื่อมีการเปลี่ยนแปลงของ Layout อีกทั้งรูปภาพที่ใช้ตอนเขียนขึ้นมา มีขนาดใหญ่กว่าภาพที่แสดงบนหน้าเว็บแอปพลิเคชัน หากมีรูปภาพขนาดเหมาะสมจะประหยัดอินเทอร์เน็ต และประหยัดเวลาในการโหลด
- 2) Accessibility ประเมินได้ 17/19 ตรวจจับได้ คือ หมวดการตั้งชื่อ คือ ไม่มีตั้งชื่อให้ปุ่มที่มีไว้ให้ช่วยในการเข้าถึง
- 3) Best Practices ประเมินได้ 5/5 มีการหลีกเลี่ยงใช้ Library ที่มีช่องโหว่ด้านความปลอดภัย, อนุญาตผู้ใช้ให้วางรหัสผ่านในช่องได้, แสดงรูปภาพที่มีสัดส่วนถูกต้อง, แสดงรูปภาพที่มีความละเอียดเหมาะสม และมีการใช้ DOCTYPE ใน HTML
- 4) SEO ประเมินได้ 6/7 เนื่องจากมีไม่สามารถหา <meta name="description"> พบ หากมีจะทำให้การค้นหาข้อมูลภายในเว็บที่สร้างง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการที่ได้สหกิจศึกษาของโครงการ CSD Refresh Technology ผู้จัดทำที่ได้เข้าร่วมทีมพัฒนาส่วนของหน้าประสานงานกับผู้ใช้งาน (User Interface) ที่นำ React v.7.0 มาใช้ในการพัฒนา ได้ลองเขียนหลากหลายฟังก์ชัน ทั้งฟังก์ชันค้นหา (Search), สร้าง(Create), แก้ไข(Update), ดู (View), อนุมัติ (Approve), ปฏิเสธ (Reject) และอื่นๆ ตามหลัก CRUD โดยใช้ Library ต่างๆ มาร่วมพัฒนาได้มีการเรียนรู้เริ่มจากการลองเขียน Component โดยใช้ Ant design และ Material UI Library ในการรองรับการเขียนหน้าจอให้สวยงาม และมีการใช้ React Hook ในการจัดการการทำงานบนหน้าจอต่างๆ เช่น การจัดการ State ของหน้าจอ การจัดการ Props ที่ส่งมาระหว่าง Component เป็นต้น ได้การเชื่อมต่อระหว่างหน้าจอไปยัง Middleware ที่ช่วยจัดการการเขียน Request และ Response รวมถึงเรื่อง Request Response Cycle ทั้งการใช้ Redux-Saga แทน Redux-Thunk ที่นำมาแก้ปัญหาเรื่อง Callback Hell ที่เมื่อใช้ Redux Thunk มีการ dispatch Callback Functions เรื่อยๆ การที่ได้ลองเชื่อมต่อกับ API การใช้ axios ในการดึง หรือร้องขอข้อมูลจาก API ได้มีการเรียนรู้การเขียนที่ง่าย และเหมาะกับการบำรุงรักษา โดยมีการจัดการไฟล์ที่เป็นระบบมาก มีการปรึกษาเสมอก่อนที่จะพัฒนาเสมอว่าควรรวม Component หรือแยกออกมาเพื่อให้เกิดความรวดเร็วได้การ Build มากที่สุด

ในการทำงานร่วมกับทีมอื่นมีการใช้ Jira ในการจัดการงานที่ได้รับมอบหมาย มีการกำหนดรายละเอียด หน้าที่ต้องพัฒนา และระบุตัวบุคคลที่ทำงานนั้นได้ อีกทั้งยังมีการทำ Daily Scrum แบบรวมทีมในทุกๆ วันที่ทำงาน และมีการประชุมเฉพาะทีมอีกครั้ง เพื่อปรึกษา และแก้ปัญหาที่เจอร่วมกัน

5.2 ข้อเสนอแนะ

1.) ควรปรับปรุง และพัฒนาเว็บแอปพลิเคชันให้มีค่าการประเมินของ Lighthouse ในทั้งสามโหมดให้สูงขึ้น และยึดเกณฑ์มาตรฐานของ Lighthouse เพื่อการทำงานในแต่ละส่วนมีประสิทธิภาพมากขึ้น

เอกสารอ้างอิง

พชร อารยะการกุล. 2562. การทำงานแบบ Agile อาวุธเด็ดในการสร้างองค์กรยุคดิจิทัล. [Online]. เข้าถึงได้จาก www.forbesthailand.com/commentaries/insights/สร้างองค์กร-การทำงานแบบ.

วิกิพีเดีย สารานุกรมเสรี. 2557. การเขียนโปรแกรมแบบโพรโทไทป์. [Online]. เข้าถึงได้จาก <https://th.wikipedia.org/wiki/การเขียนโปรแกรมแบบโพรโทไทป์>.

Alexander S. Gillis. 2019. just-in-time compiler (JIT). [Online]. <https://www.theserverside.com/definition/just-in-time-compiler-JIT>.

Barasa, M. 2021. How to Implement Material-UI in React. [Online]. <https://www.section.io/engineering-education/how-to-implement-material-ui-in-react>.

BstaR. 2022. Learn Typescript in 15 minutes. [Online]. <https://medium.com/front-end-weekly/learn-typescript-in-15-minutes-bf921cf355f5>.

Chaivanichanan, T. 2020. รู้จักกับ TypeScript – ประวัติของภาษาที่เติมโทป์ให้กับ JavaScript. [Online]. <https://www.centrilliontech.co.th/blog/2843/history-of-typescript/>.

Codebots. 2020. What is Axios?. [Online]. <https://codebots.com/docs/what-is-axios>.

DataexpressTH. 2020. รู้จักและเริ่มใช้งาน JIRA Project Management Software (1/4). [Online]. <https://dataexpress.co.th/รู้จักและเริ่มใช้งาน-jira-project-manage/>.

Elliott, E. 2020. The Missing Introduction to React. [Online]. <https://medium.com/javascript-scene/the-missing-introduction-to-react-62837cb2fd76>.

GeeksforGeeks. 2022. Ant Design. [Online]. <https://www.geeksforgeeks.org/ant-design/>.

Google Developers. 2022. Lighthouse. [Online]. <https://developer.chrome.com/docs/lighthouse/overview/>.

Jain, P. 2019. Folder Structure in React. [Online]. <https://www.c-sharpcorner.com/article/folder-structure-in-react/>.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Montha, S. 2019. ให้ SonarLint ช่วยคุณแก้ปัญหาคุณภาพของโค้ด ก่อนส่งมอบงานไปสู่สังคมกันเถอะ. [Online]. <https://iamgique.medium.com/ให้-sonarlint-ช่วยคุณแก้ปัญหาคุณภาพของโค้ด-ก่อนส่งมอบงานไปสู่สังคมกันเถอะ-562e4710dd1>.
- Nipa Technology. 2022. GitLab ซอฟต์แวร์ตัวเด็ดเทรนด์ใหม่ของ Opensource. [Online]. <https://www.nipa.cloud/blog/gitlab-software-opensource>.
- NPM. 2022. About npm. [Online]. <https://docs.npmjs.com/about-npm>.
- Olawanle, J. 2022. How to Get Started With React – A Beginner's Guide. [Online]. <https://www.freecodecamp.org/news/get-started-with-react-for-beginners/>.
- Redux-Saga. 2022. About Redux-Saga. [Online]. <https://redux-saga.js.org/docs/About/>.
- Saengow, J. 2018. [GitLab] คืออะไร เริ่มใช้งานเบื้องต้น. [Online]. <https://medium.com/jed-ng/gitlab-คืออะไร-เริ่มใช้งานเบื้องต้น-5ccffc430456>.
- Sheldon, R. 2022. Bytecode. [Online]. <https://www.theserverside.com/definition/just-in-time-compiler-JIT>.
- Typescript. 2022. The Basics. [Online]. <https://www.typescriptlang.org/docs/handbook/2/basic-types.html>.
- SmartBear Software. 2022. What Is OpenAPI?. [Online]. <https://swagger.io/docs/specification/about/>.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้