

ร้านค้าสำหรับแอปพลิเคชันการประมวลผลภาพซึ่งจัดการงานบนระบบ
ประมวลผลแบบกลุ่ม

**Marketplace for Image Processing Application using Task
Management on Cluster Computing System**



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2565

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ร้านค้าสำหรับแอปพลิเคชันการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผลแบบกลุ่ม

Marketplace for Image Processing Application using Task Management on Cluster Computing System

ผู้จัดทำ

1. นายพศิน จันทรทัน รหัสนักศึกษา 63015121
2. นายสุธี สาระพันธ์ รหัสนักศึกษา 63015190

 อาจารย์ที่ปรึกษา
(รศ. ดร.อรจักร จิตต์โสภักตร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ร้านค้าสำหรับแอปพลิเคชันการประมวลผลภาพซึ่งจัดการงานบน ระบบประมวลผลแบบกลุ่ม

นายพศิน	จันทร์ทัน	63015121
นายสุธี	สาระพันธ์	63015190
รศ.ดร.อรฉัตร	จิตต์โสภักตร์	อาจารย์ที่ปรึกษา
ปีการศึกษา 2565		

บทคัดย่อ

ในปัจจุบันการประมวลผลภาพเป็นสิ่งสำคัญที่ใช้ทรัพยากรที่มากขึ้นเนื่องจากการใช้งานอย่างแพร่หลายในหลากหลายกลุ่มอุตสาหกรรม อย่างไรก็ตาม การใช้บริการงานประมวลผลบนอินเทอร์เน็ต (Cloud Processing) ที่มีค่าใช้จ่ายสูงอาจไม่เป็นทางเลือกที่เหมาะสมสำหรับบางบริษัทหรือบุคคลที่ต้องการประมวลผลภาพจำนวนมาก ดังนั้น โครงการนี้จึงได้ถูกจัดทำขึ้นมาเพื่อให้บริการการประมวลผลภาพให้สะดวกสบายและเหมาะสมกับทุกกลุ่มลูกค้า

โครงการร้านค้าสำหรับแอปพลิเคชันการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผลแบบกลุ่ม มีจุดประสงค์เพื่อสร้างระบบให้บริการใช้งานการประมวลผลภาพและแสดงผลงานการงานที่ได้จัดทำขึ้น โดยผู้ใช้สามารถเข้าถึงแอปพลิเคชันและเลือกใช้งานตามความต้องการได้หลากหลายเช่น การปรับแต่งภาพ, การจัดการกับรูปแบบไฟล์ภาพ และ ซือขายน้ำหนัของโมเดล นอกจากนี้ ผู้ใช้งานยังสามารถแสดงผลงานการประมวลผลที่ได้จัดทำขึ้นมา เพื่อให้ผู้ใช้งานอื่นๆ ได้เข้าถึง และ ซือขายกันได้ ในระบบ โดยทั้งนี้ การใช้งานร้านค้านี้จะเหมาะสมกับกลุ่มผู้ใช้งานที่ต้องการประมวลผลภาพจำนวนมาก และ ไม่ต้องการเสียค่าใช้จ่ายที่สูง

ร้านค้าสำหรับแอปพลิเคชันการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผลแบบกลุ่ม เป็นโครงการที่มุ่งเน้นการพัฒนาระบบให้มีความสามารถในการจัดการและประมวลผลภาพอย่างมีประสิทธิภาพและเร็วขึ้น โดยใช้ระบบประมวลผลแบบกลุ่มที่มีประสิทธิภาพสูงเป็นพื้นฐาน ซึ่งจะช่วยลดต้นทุนและเพิ่มประสิทธิภาพในการทำงานของระบบให้สูงขึ้น

Marketplace for Image Processing Application using Task Management on Cluster Computing System

Mr. Pasin Chantharathan 63015121

Mr. Sutee Saraphan 63015190

Assoc.Prof.Dr. Orachat Chitsobhuk Advisor

Academic Year 2022

ABSTRACT

Currently, image processing is becoming increasingly important as it is widely used in various industries. However, using cloud processing services, which are expensive, may not be a suitable option for some companies or individuals who do not need to process a large amount of images. Therefore, this project was created to provide image processing services that are convenient and suitable for all customers.

The project Marketplace for Image Processing Application using Task Management on Cluster Computing System. The objective is to create a system that provides image processing services and displays the work performed. Users can access the application and choose the services they need, such as image customization, file format management, and buying and selling of model weights. Additionally, users can showcase their processed images for others to view and buy within the system. This shop is suitable for users who do not need to process a large number of images and do not want to pay high costs.

The Marketplace for Image Processing Application using Task Management on Cluster Computing System is a project that focuses on developing a system that efficiently manages and processes images using a high-performance group-based processing system as the foundation. This system will help reduce costs and increase the efficiency of the system's performance. Users will benefit from the system's fast and accurate image processing capabilities.

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สามารถดำเนินการจนประสบความสำเร็จ เนื่องจากได้รับความอนุเคราะห์จาก รศ.ดร.อรฉัตร จิตต์โสภักดิ์ อาจารย์ที่ปรึกษาปริญญาบัตรที่กรุณาให้คำแนะนำ คำปรึกษาเพื่อปรับปรุง แก้ไขในข้อผิดพลาด และ เสนอแนวทางการทำงาน รวมไปถึงการให้อุปกรณ์สิ่งของต่างๆ และความรู้ แนวทางการศึกษา ที่จำเป็นต่อการดำเนินการศึกษาและค้นคว้าเพื่อจัดทำปริญญาบัตรฉบับนี้ คณะผู้จัดทำขอกราบขอบพระคุณเป็นอย่างไว้ ณ ที่นี้

ขอขอบพระคุณอาจารย์สรยุทธ กลมกล่อม อาจารย์ผู้ประสานงานที่ให้ข้อมูลและคำปรึกษาแก่คณะผู้จัดทำปริญญาบัตรจนสามารถสำเร็จลงด้วยดี

คณะผู้จัดทำมีความซาบซึ้งในความกรุณาของทุกท่านที่กล่าวถึงและผู้ไม่ได้เอ่ยนามในที่นี้ที่มีส่วนช่วยให้ความช่วยเหลือและสนับสนุนด้วยดีตลอดมา จึงขอกราบขอบพระคุณด้วยความจริงใจ

พศิน จันทรทัต

สุธี สาระพันธ์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูปภาพ.....	VII
สารบัญตาราง.....	XIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาของปัญหา.....	1
1.2 วัตถุประสงค์ของ โครงการ.....	2
1.3 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.4 ขอบเขตการวิจัย.....	3
บทที่ 2 เอกสารและทฤษฎีที่เกี่ยวข้อง.....	4
2.1 ทฤษฎีที่เกี่ยวข้อง.....	4
2.2 งานที่เกี่ยวข้อง.....	7
2.3 เครื่องมือที่ใช้งานในการพัฒนาระบบ.....	9
2.4 งานวิจัยที่เกี่ยวข้อง.....	12
บทที่ 3 การออกแบบและการพัฒนา.....	16
3.1 รายละเอียดของระบบ.....	16
3.2 ภาพรวมของระบบ.....	17
3.3 แผนภาพผังงาน (Flowchart).....	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
3.4 แผนภาพยูสเคส (Use Case Diagram)	23
3.5 แผนภาพแสดงองค์ประกอบ (Component Diagram)	28
3.6 แผนภาพแสดงความสัมพันธ์ของข้อมูล (E-R Diagram)	29
3.7 แผนภาพการทำงานแบบลำดับปฏิสัมพันธ์ (Sequence Diagram)	33
3.8 การออกแบบส่วนต่อประสานกับผู้ใช้ (User Interface)	37
บทที่ 4 ผลการทดลอง / ความก้าวหน้า	46
4.1 ส่วนของหน้าลงทะเบียนสมัครเข้าใช้งาน	46
4.2 ส่วนของหน้าลงทะเบียนเข้าสู่ระบบใช้งาน	48
4.3 ส่วนของหน้าแก้ไขข้อมูลผู้ใช้งาน	51
4.4 ส่วนของหน้าจัดการไฟล์.....	53
4.5 ส่วนของหน้าสั่งงานประมวลผล.....	57
4.6 ส่วนของหน้าตลาดซื้อขาย	60
4.7 ส่วนของหน้าประวัติการใช้งานของผู้ใช้.....	64
4.8 ส่วนของหน้าประวัติการใช้งานของสินค้า.....	65
4.9 ส่วนของหน้าร้านค้าแสดงสินค้า	67
4.10 ส่วนของหน้าแสดงข้อมูลเฉพาะสินค้า.....	69
4.11 ส่วนของการแบ่งภาระงาน	70
4.12 ส่วนของ Application Image Processing.....	74
บทที่ 5 สรุปผลและข้อเสนอแนะ	77
5.1 สรุปผลที่จากโครงการ	77

สารบัญ(ต่อ)

	หน้า
5.2 ปัญหาและอุปสรรค.....	80
5.3 แนวทางการพัฒนาต่อไปในอนาคต	80
บรรณานุกรม	59
ภาคผนวก ก	82
รายละเอียดของ diagram.....	82
ก.1 Component diagram(Detail).....	82
ภาคผนวก ข	89
Test case	89
ข.1 Frontend test.....	89
ข.2 Backend test	93

สารบัญตาราง

ตาราง	หน้า
3.1 รายละเอียดของ Use case Register.....	24
3.2 รายละเอียดของ Use case Login.....	24
3.3 รายละเอียดของ Use case Upload image.....	25
3.4 รายละเอียดของ Use case enchanted image.....	25
3.5 รายละเอียดของ Use case Load balance config.....	25
3.6 รายละเอียดของ Monitoring hardware.....	26
3.7 รายละเอียดของ Using statistics report.....	26
3.8 รายละเอียดของ Subscribe VIP plan.....	27
3.9 รายละเอียดของ Buy weight for model.....	27
3.10 รายละเอียดของ Payment log.....	27
3.11 แสดงรายละเอียดตารางข้อมูลผู้ใช้ (user).....	30
3.12 แสดงรายละเอียดตารางข้อมูลไฟล์รูปภาพ (Image_file).....	30
3.13 แสดงรายละเอียดตารางข้อมูลของโฟลเดอร์เก็บรูปภาพ (folder_img).....	30
3.14 แสดงรายละเอียดตารางเก็บข้อมูลการเข้าสู่ระบบ (login_log).....	31
3.15 แสดงรายละเอียดตารางข้อมูลการชำระเงิน (payment).....	31
3.16 แสดงรายละเอียดตารางข้อมูลของสินค้า (product).....	31
3.17 แสดงรายละเอียดตารางข้อมูลการสั่งงานประมวลผล (job).....	32
ก.1 UserManagement component.....	83
ก.2 DriveManagement component.....	83
ก.3 Marketplace component.....	83

สารบัญตาราง(ต่อ)

ตาราง	หน้า
ก.4 History component.....	84
ก.5 ImageProcessingApplicationPage component.....	84
ก.6 user_authentication component.....	85
ก.7 drive_management component.....	85
ก.8 product_management component.....	86
ก.9 user_management component.....	86
ก.10 image_processing_application component.....	86
ก.11 payment_management component.....	87
ก.12 preview_process component.....	87
ก.13 image_processing_process component.....	88
ก.14 load balancing component.....	88
ก.15 Database component.....	89
ก.16 File storage(NAS)component.....	89
ข.1 Frontend testcases.....	90
ข.2 Backend testcase.....	94

สารบัญรูปภาพ

รูป	หน้า
2.1 Cluster Computing.....	5
2.2 การประมวลผลแบบกลุ่มเมฆ (Cloud Computing).....	5
2.3 แผนผังรูปแบบการใช้งานการแบ่งภาระงาน(Load Balance).....	6
2.4 ตัวอย่างของงานประมวลผลภาพบน AnimeFilter.....	7
2.5 ตัวอย่างของ Film Emulator.....	8
2.6 ตัวอย่างของ aws marketplace.....	8
2.7 Component system ของ Vue.js.....	9
2.8 รูปแบบการทำงานของ Kubernetes.....	10
2.9 แผนผังแสดงระบบที่ใช้ในการทดสอบIngress controller.....	12
2.10 รูปผลลัพธ์จากการปรับค่าความคมชัดด้วยวิธีที่แตกต่างกัน.....	13
2.11 แผนผังรูปแบบการกระจายงานในช่วงที่สองที่มีการกระจายภาระงานระหว่าง CPU และ GPU....	14
2.12 แผนผัง System Architecture ของ ProCon.....	15
3.1 ภาพรวมของระบบ Overview.....	17
3.2 ภาพรวมของระบบ Level 0.....	18
3.3 ภาพรวมของระบบ Level 1.....	18
3.4 แผนภาพผังงานของการสั่งงานประมวลผลภาพ (Web Application).....	19
3.5 แผนภาพผังงานของการสั่งงานประมวลผลภาพบน Marketplace (Web Application).....	20
3.6 แผนภาพผังงานของการจัดเก็บไฟล์ภาพไว้ใน Drive (Web Application).....	21
3.7 แผนภาพผังงานของการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train.....	22
3.8 แผนภาพยูสเคส.....	23

สารบัญรูปภาพ(ต่อ)

รูป	หน้า
3.9 แผนภาพแสดงองค์ประกอบ.....	28
3.10 แผนภาพแสดงความสัมพันธ์ของข้อมูล.....	29
3.11 Sequence diagram ในส่วนของการสั่งงานประมวลผลภาพ.....	33
3.12 Sequence diagram ในส่วนของการสั่งงานประมวลผลภาพบน Marketplace.....	34
3.13 Sequence diagram ในส่วนของการจัดเก็บไฟล์ภาพไว้ใน Drive.....	35
3.14 รูปภาพแสดงการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train.....	36
3.15 การออกแบบส่วนลงชื่อผู้ใช้.....	37
3.16 การออกแบบส่วนลงทะเบียนสมัครสมาชิก.....	37
3.17 การออกแบบส่วนหน้าข่าวสาร.....	38
3.18 การออกแบบส่วนพื้นที่เก็บบันทึกรูปภาพ.....	38
3.19 การออกแบบส่วนพื้นที่แสดงรูปภาพที่เก็บบันทึกไว้.....	39
3.20 การออกแบบส่วนแอปพลิเคชันประมวลผลภาพ.....	39
3.21 การออกแบบส่วนพื้นที่ซื้อขาย.....	40
3.22 การออกแบบส่วนตั้งค่าบัญชี.....	40
3.23 การออกแบบส่วนตั้งค่าและดูบันทึกการใช้ทรัพยากร.....	41
3.24 การออกแบบส่วนดูบันทึกการใช้งานของผู้ใช้.....	41
3.25 การออกแบบส่วนดูบันทึกการขายของสินค้า.....	42
3.26 การออกแบบส่วนของการแสดงผลตัวอย่างงานประมวลผล Basic App.....	43
3.27 การออกแบบส่วนของการแสดงผลตัวอย่างงานประมวลผล Yolo Model.....	44
3.28 การออกแบบส่วนของการแสดงผลตัวอย่างงานประมวลผล GAN Model.....	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการ X วิชาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

รูป	หน้า
3.29 การออกแบบส่วนของการบีบอัดไฟล์ Zip API.....	45
4.1 ข้อมูลที่ส่งจาก Web application เทียบกับข้อมูลที่ส่วน Backend ได้รับ.....	47
4.2 ข้อมูลผู้ใช้ทำการสมัครสำเร็จถูกเพิ่มในฐานข้อมูล.....	47
4.3 ข้อความที่แสดงหากอีเมลที่ใช้สมัครนั้นมีอยู่แล้วในฐานข้อมูล.....	48
4.4 ข้อมูลที่ส่งจาก Web application เทียบกับข้อมูลที่ส่วน Backend ได้รับ.....	49
4.5 ข้อมูล Cookie ของผู้ใช้หากทำการ เข้าสู่ระบบสำเร็จ.....	50
4.6 ข้อความที่แสดงหากรหัสผ่านที่กรอกเข้าสู่ระบบนั้น ไม่ถูกต้อง.....	50
4.7 ข้อความที่แสดงหากสามารถเปลี่ยนข้อมูลผู้ใช้ได้สำเร็จ.....	51
4.8 ข้อความที่แสดงหากเกิดข้อผิดพลาดขึ้นขณะเปลี่ยนแปลงข้อมูลผู้ใช้.....	52
4.9 ข้อความที่แสดงหากสามารถเปลี่ยนข้อมูลผู้ใช้ได้สำเร็จ.....	53
4.10 ข้อความที่แสดงหากเกิดข้อผิดพลาดขึ้นขณะเปลี่ยนแปลงข้อมูลผู้ใช้.....	53
4.11 ข้อความที่แสดงหากสามารถเพิ่ม Folder สำหรับเก็บรูปภาพได้.....	54
4.12 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะเพิ่ม Folder สำหรับเก็บรูปภาพ.....	55
4.13 ข้อความที่แสดงหากสามารถเพิ่มรูปภาพสำเร็จ.....	56
4.14 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะเพิ่มรูปภาพ.....	56
4.15 รูปที่ผ่านการประมวลผลที่แสดงหากประมวลผลสำเร็จ.....	57
4.16 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะสร้างไฟล์งานประมวลผล.....	58
4.17 ข้อความที่แสดงหากสามารถสร้างไฟล์งานประมวลผลสำเร็จ.....	59
4.18 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะสร้างไฟล์งานประมวลผล.....	59
4.19 ข้อความที่แสดงหากสามารถเพิ่มสินค้าได้สำเร็จ.....	60

สารบัญรูปภาพ(ต่อ)

รูป	หน้า
4.20 ข้อความที่แสดงหากไม่สามารถเพิ่มสินค้าได้.....	61
4.21 ข้อความที่แสดงหากสามารถแก้ไขข้อมูลสินค้าได้สำเร็จ.....	62
4.22 ข้อความที่แสดงหากไม่สามารถแก้ไขสินค้าได้.....	62
4.23 ข้อมูลสินค้าที่สามารถส่งมาทดลองใช้งานที่หน้าสั่งงานประมวลผล.....	63
4.24 ข้อความที่แสดงหากไม่สามารถส่งข้อมูลสินค้ามาที่หน้าสั่งงานประมวลผลได้.....	63
4.25 ประวัติรายการของผู้ใช้ที่มีการเรียงลำดับ โดยการแสดงรายการล่าสุดก่อน.....	64
4.26 ประวัติรายการของผู้ใช้ที่มีการเรียงลำดับ โดยการแสดงรายการเก่าที่สุดก่อน.....	65
4.27 ประวัติรายการของผู้ใช้ที่มีค้นหาโดยใช้วันที่ ที่กำหนดเอาไว้.....	65
4.28 ประวัติรายการของสินค้าที่มีการเรียงลำดับ โดยการแสดงรายการล่าสุดก่อน.....	66
4.29 ประวัติรายการของสินค้าที่มีการเรียงลำดับ โดยการแสดงรายการเก่าที่สุดก่อน.....	66
4.30 ประวัติรายการของสินค้าที่มีค้นหาโดยใช้วันที่ ที่กำหนดเอาไว้.....	67
4.31 รายการของสินค้าที่มีการเรียงลำดับ โดยการแสดงสินค้าที่มีการแก้ไขล่าสุด.....	68
4.32 รายการของสินค้าที่มีการเรียงลำดับ โดยการแสดงสินค้าที่มีการแก้ไขเก่าสุด.....	68
4.33 รายการของสินค้าที่มีการกรองเลือกเฉพาะประเภท Object detections.....	68
4.34 แสดงรายละเอียดของสินค้าที่ถูกบันทึกเอาไว้ในฐานข้อมูล.....	69
4.35 แสดงข้อความแจ้งเตือนหากทำหาสินค้าที่ไม่มีอยู่ในฐานข้อมูล.....	70
4.36 ผลการทดลองส่ง Request เข้ามาใช้งาน.....	71
4.37 กราฟของผลการทดลองเมื่อสั่งงานประมวลผลด้วยการประมวลผลด้วยเครื่องเดียว.....	72
4.38 กราฟของผลการทดลองเมื่อสั่งงานประมวลผลด้วยระบบคลัสเตอร์ด้วย Round Robin.....	72
4.39 กราฟของผลการทดลองเมื่อสั่งงานประมวลผลด้วยระบบคลัสเตอร์ด้วย Least Connection.....	73

สารบัญรูปภาพ(ต่อ)

รูป	หน้า
4.40 Repositories ของ DockerHub ที่ทำการเก็บ Application ไว้.....	74
4.41 รูปภาพต้นฉบับ และ รูปภาพที่ผ่านการประมวลผลด้วย Application ASCII.....	75
4.42 รูปภาพต้นฉบับ และ รูปภาพที่ผ่านการประมวลผลด้วย Application BlackWhite.....	75
4.43 รูปภาพต้นฉบับ และ รูปภาพที่ผ่านการประมวลผลด้วย Application PixelArt.....	76
4.44 รูปภาพต้นฉบับ และ รูปภาพที่ผ่านการประมวลผลด้วย Application Mosaic.....	76
ก.1 แผนภาพองค์ประกอบส่วน Frontend.....	82
ก.2 แผนภาพองค์ประกอบส่วน Backend.....	85
ก.3 แผนภาพองค์ประกอบส่วน Server.....	87
ก.4 แผนภาพองค์ประกอบส่วน Storage.....	88

บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

ในปัจจุบันผู้ให้บริการแอปพลิเคชันการประมวลผลภาพนั้นมีข้อจำกัดเรื่องงานการประมวลผล ใช้ทรัพยากรของเครื่องสูง ไม่มีประสิทธิภาพ หรือ ยังไม่มีการกำหนด จำกัดหน่วยประมวลผลของงานในแต่ละงานประมวลผล และ ไม่รองรับการประมวลผลภาพจำนวนมากพร้อม ๆ กัน พร้อมด้วยแอปพลิเคชันการประมวลผลภาพนั้นมีตัวเลือกค่อนข้างน้อยในแอปพลิเคชันเดียวให้ใช้งาน และเนื่องจากงานวิจัยด้าน Image Processing ต่าง ๆ นั้นมีการ Train ข้อมูลของ Weight Model เข้ามาช่วยในงานประมวลผลดังตัวอย่างในวิชา Image Processing ภายในสถาบันที่มีการให้นักศึกษา ศึกษาโครงการ Train Weight Model ภายในวิชาขึ้นมาเพื่อมาทดลองภายในห้องทดลอง แต่ยังคงพื้นที่สำหรับการให้บริการด้าน Model ของ Image Processing ที่นักศึกษาได้ทำการ Train Weight ข้อมูลไว้เพื่อให้ได้ทดสอบภายในห้องทดลองเนื่องจากใช้ทรัพยากรของเครื่องสูงในการประมวลผลของ Model และ ให้นักทดลองนอกได้เข้าถึงด้วย

ดังนั้น โครงการ ร้านค้าสำหรับแอปพลิเคชันการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผลแบบกลุ่ม มีจุดประสงค์เพื่อสร้างระบบการแบ่งการจัดลำดับงานประมวลผลของผู้ให้บริการแอปพลิเคชันเพื่อเป็นอีกทางเลือกหนึ่งในการใช้งานการประมวลผลภาพในรูปแบบต่าง ๆ อย่างมีประสิทธิภาพ และรองรับการประมวลผลภาพจำนวนมาก พร้อมด้วยรองรับแอปพลิเคชันที่สามารถนำเข้ามาเพิ่มในอนาคตได้อีกด้วย และ ในส่วนของ Weight Model ที่นักศึกษาได้ทำการ Train มานั้นก็สามารถนำเข้ามาภายในแอปพลิเคชันเพื่อเปิดให้บริการในแอปพลิเคชันของเราได้เพื่อทำการซื้อขาย หรือ เป็นที่แสดงผลงานของนักศึกษา และ เป็นอีกทางเลือกให้นักศึกษามีรายได้

โดยสรุปได้ว่า การตกแต่งรูปภาพ หรือ การปรับแต่งภาพ นั้นมีการใช้อยู่แพร่หลาย และ คนส่วนใหญ่มีการใช้งานเรื่องการประมวลผลภาพกันเป็นปกติ การนำ Weight Model มาใช้งาน หรือ จัดแสดงผลงาน และการซื้อขาย ซึ่งทั้งหมดที่กล่าวมาคือส่วนมากใช้งานบนแอปพลิเคชันต่างๆ ผ่านอินเทอร์เน็ตเราจึงพัฒนาโครงการบนพื้นฐานของเว็บแอปพลิเคชันซึ่งสอดคล้อง และ รองรับกับปัญหาดังกล่าว โครงการ “ร้านค้าสำหรับแอปพลิเคชันการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผลแบบกลุ่ม” นี้ถูกจัดทำขึ้นมาเพื่อแก้ไขในส่วนของกระบวนการนี้ โดยมีการแบ่งเป็น 3 ส่วนคือ

- 1) ส่วนการประมวลผลงานแบบการจัดลำดับงาน (Server) Task Scheduling การแบ่งจัดลำดับการทำงานประมวลผลแต่ละงานไปแต่ละเครื่อง โดยมีการกำหนดทรัพยากรที่ใช้ในแต่ละงานประมวลผลเพื่อให้ได้ประสิทธิภาพเพียงพอต่อความต้องการของงานประมวลผล และใช้ทรัพยากรของหน่วยประมวลผลให้คุ้มค่ากับทรัพยากรที่มีอยู่ของหน่วยประมวลผล
- 2) ส่วนแอปพลิเคชันการประมวลผลภาพ (User) Image Processing Application ส่วนระบบสำหรับประมวลผลภาพที่ระบบได้มีการเตรียมรูปแบบประมวลผลภาพต่าง ๆ ที่หลากหลายให้ผู้ใช้ได้เข้ามาใช้งานการประมวลผลภาพจำนวนมากที่ทางเราได้เตรียมจัดทำไว้
- 3) ส่วนแอปพลิเคชันให้บริการซื้อขาย Weight โมเดล (User) Marketplace ส่วนระบบสำหรับให้ผู้ใช้มีการซื้อ Weight Model และ นำเข้า Weight Model ที่ได้ทำการเทรนนิ่งไว้แล้วมาเปิดให้บริการบนแอปพลิเคชันเพื่อทำการ ขาย หรือ เพื่อทดลองงานต่าง ๆ โดยเป็นจุดแสดงผลงานของตัว Weight Model ที่ได้ทำการเทรนนิ่งไว้ได้อีกด้วย

1.2 วัตถุประสงค์ของโครงการ

- 1) เพื่อจัดสรรการใช้งานประมวลผลของแต่ละงานประมวลผลให้มีประสิทธิภาพเพียงพอ ต่อความต้องการของงาน
- 2) เพื่อให้งานในการประมวลผลภาพนั้นมีการจัดลำดับการประมวลผล และมีประสิทธิภาพมากยิ่งขึ้นต่อทรัพยากรของเครื่องประมวลผล
- 3) เพื่อนำไปประยุกต์ต่อยอดทางธุรกิจ

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้รับประสบการณ์ในการพัฒนาเว็บด้วย Vue.js และ DJANGO
- 2) ได้รับประสบการณ์ในการออกแบบฐานข้อมูลด้วย MongoDB (NOSQL)
- 3) ได้รับประสบการณ์ในการพัฒนาระบบ Cluster Computing ด้วย Kubernetes
- 4) ได้รับประสบการณ์ในการพัฒนาระบบ Server สำหรับงาน Cluster Computing
- 5) ได้รับประสบการณ์ในการพัฒนาแอปพลิเคชันการประมวลผลภาพที่นิยมในปัจจุบัน

1.4 ขอบเขตการวิจัย

- 1) ระบบ Cluster Computing เชื่อมต่อผ่านระบบเครือข่ายท้องถิ่น (LAN)
- 2) ระบบ Cluster Computing ในแอปพลิเคชันการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่ายนี้เป็นการแบ่งงานในแต่ละเครื่องเพื่อให้ใช้ทรัพยากรที่จำกัด และ จำกัดทรัพยากรในแต่ละงานประมวลผล
- 3) การจัดการลำดับงานบนระบบขึ้นอยู่กับประสิทธิภาพฮาร์ดแวร์ของ Node ที่พร้อมใช้งานในช่วงเวลานั้น
- 4) แอปพลิเคชันในส่วนของไฟล์ข้อมูล Structures ของ Directory แต่ละ User นั้นการแตกระดับชั้นของ Directory ได้ 1 ชั้น
- 5) แอปพลิเคชันในส่วนของการประมวลผลนั้นสามารถทำการประมวลผลภาพจำนวนมาก จึงปรับค่า Parameter ได้ 1 ค่าในงานประมวลผลภาพจำนวนมาก
- 6) ส่วนของ Marketplace ในการนำเข้า Weight นั้นจะต้องรองรับกับตัว Model ที่มีให้บริการภายในแอปพลิเคชันเท่านั้น

บทที่ 2

เอกสารและทฤษฎีที่เกี่ยวข้อง

ในการจัดทำโครงการ ผู้จัดทำได้ทำการศึกษาทฤษฎีที่เกี่ยวข้อง, งานที่เกี่ยวข้อง, เครื่องมือที่ใช้ งานในการพัฒนาระบบ และ งานวิจัยที่เกี่ยวข้อง เพื่อนำมาปรับใช้ และ เป็นแนวทางในการทำโครงการ ดังต่อไปนี้

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 Digital Image Processing

การประมวลผลภาพดิจิทัล หรือ Digital Image Processing คือ การใช้คอมพิวเตอร์ดิจิทัลในการประมวลผลภาพดิจิทัลผ่านอัลกอริทึม ภาพในที่นี่รวมความหมายถึงสัญญาณในระบบดิจิทัล 2 มิติ ภาพนิ่ง และ ภาพเคลื่อนไหว หรือเป็นชุดของภาพนิ่ง ที่เรียกว่า เฟรม (Frame) ซึ่งนับเป็นภาพดิจิทัล 3 มิติได้ เช่น ภาพทางการแพทย์ หรือ ภาพ 3 มิติหลายชนิด (Multimodal image)

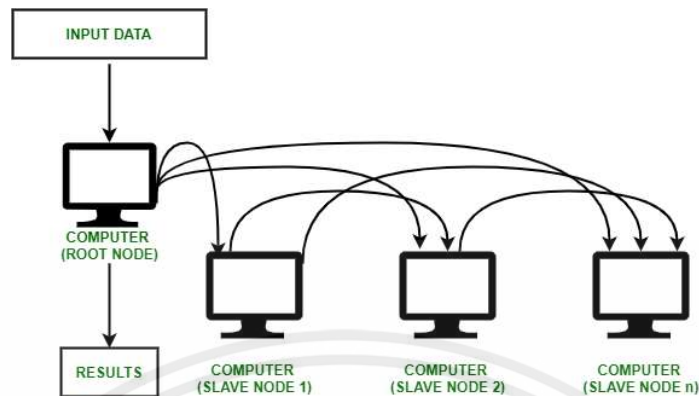
วัตถุประสงค์ของการประมวลผลภาพแบ่งได้ออกเป็น 5 กลุ่มหลักดังนี้

- 1) การสร้างภาพ (Visualization) จากวัตถุที่มองด้วยตาไม่เห็น สามารถทำให้มองเห็นภาพได้
- 2) การปรับความชัด และ การฟื้นฟูภาพ (Image sharpening and Restoration) ใช้สำหรับการปรับความละเอียดของภาพให้ดีขึ้น
- 3) การดึงภาพ (Image retrieval) ทำให้จุดที่น่าสนใจของภาพให้เด่นชัด
- 4) การวัดรูปแบบของภาพ (Measurement of pattern) วัตถุในภาพทั้งหมดนั้นจะถูกวัดค่ามาเพื่อหารูปแบบของวัตถุภายในภาพ
- 5) การรับจดจำภาพ (Image Recognition) สามารถแยกแยะวัตถุในภาพได้

2.1.2 Cluster Computing

ระบบคลัสเตอร์ หรือ คลัสเตอร์ริง เป็นการเชื่อมต่อระบบการทำงานของกลุ่มคอมพิวเตอร์เข้าด้วยกันภายใต้ระบบเครือข่าย ทำให้สามารถกระจายงานประมวลผลที่ทำไปยังเครื่องอื่นๆ ภายในระบบ เพื่อให้การประมวลผลใช้ทรัพยากรที่มีอยู่ได้อย่างคุ้มค่าและมีประสิทธิภาพการประมวลผลที่สูงขึ้น โดยถูกคิดค้นขึ้นมาเพื่อใช้ในการประมวลผลงานที่มีความซับซ้อนสูงโดยเฉพาะงานด้านวิทยาศาสตร์ เช่น การจำลองโครงสร้างของโมเลกุลทางเคมี, การวิเคราะห์เกี่ยวกับตำแหน่งการเกิดพายุสุริยะ, การประมวลผลภาพ เป็นต้น ดังรูป 2.1

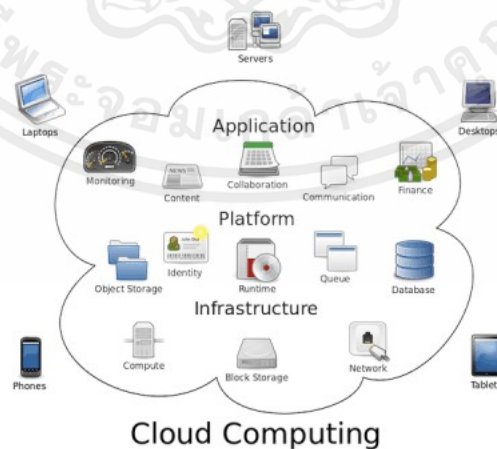
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.1 Cluster Computing

2.1.3 การประมวลผลแบบกลุ่มเมฆ (Cloud Computing)

การประมวลผลแบบกลุ่มเมฆ เป็นการที่ ผู้ให้บริการนั้นมีทรัพยากรที่ใช้สำหรับการประมวลผลอยู่ โดยเมื่อมีผู้ใช้บริการเข้ามาร้องขอต้องการใช้บริการก็จะทำการเปิดให้ผู้เข้าใช้บริการนั้นเข้าถึงทรัพยากรของผู้ให้บริการผ่านระบบอินเทอร์เน็ตทำให้ผู้ใช้บริการไม่จำเป็นต้องเป็นเจ้าของทรัพยากรจริงซึ่งทำให้ผู้ใช้บริการไม่จำเป็นต้องติดตั้งหรือตั้งค่า และ รู้การทำงานเบื้องหลัง โดยผู้ใช้จะสามารถเลือกได้ว่าต้องการใช้ทรัพยากรมากน้อยเท่าไร เพื่อให้เหมาะสมกับงานที่ต้องการประมวลผลมากที่สุด และสามารถเข้าถึงทรัพยากรจากที่ไหนก็ได้ และ ซึ่งโดยส่วนใหญ่จะมีการคิดค่าใช้จ่ายในลักษณะเช่าเป็นรายเดือน หรือ รายปี ขึ้นอยู่กับแต่ละผู้ให้บริการ โดยผู้ให้บริการก็มียูทิลิตี้หลายให้เลือกใช้บริการ เช่น Google Cloud Platform, Microsoft azure เป็นต้น ซึ่งรูปแบบหรือแผนผังตัวอย่างการประมวลผลแบบกลุ่มเมฆจะแสดงในรูปที่ 2.2



รูป 2.2 การประมวลผลแบบกลุ่มเมฆ (Cloud Computing)

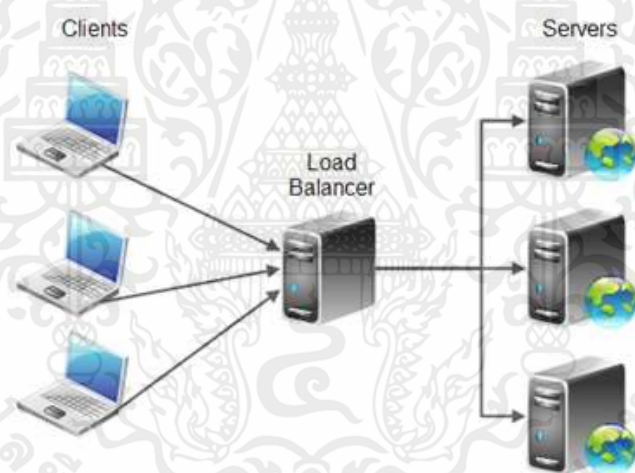
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 การแบ่งภาระงาน (Load Balance)

การแบ่งภาระงาน หรือ Load Balance คือเทคนิควิธีการทางด้าน Network ที่ถูกคิดค้นขึ้นมา เพื่อช่วยลดการเกิดปัญหา Server ไม่สามารถรองรับงานประมวผลจำนวนมากได้จน Overload และหยุดการทำงานไป โดยวิธีการคือการนำเครื่อง server ที่สามารถทำงานได้ในระดับเดียวกัน และสามารถทำงานแบบเดียวกัน มาทำงานร่วมกันเพื่อช่วยกระจายปริมาณงานที่เข้ามา โดยจะแบ่งงานไปตามวิธีการที่ผู้ใช้ได้กำหนดเอาไว้ ซึ่งวิธีการแบ่งงานที่นิยมใช้ก็มีต่อไป ดังนี้

1. Round Robin เป็นการแบ่งงานให้กับ Server ในกลุ่มแบบเรียงลำดับ เช่น 1,2,3,1,2,...
2. Sticky เป็นการแบ่งงานให้กับ Server โดยอิงจาก Sessions ที่ผู้ใช้ เคยเข้าไปใช้ก่อนหน้า
3. Workload เป็นการแบ่งงานให้กับ Server โดยการตัดสินใจจากประสิทธิภาพของ Server ในกลุ่ม

ทั้งนี้การเลือกรูปแบบการแบ่งภาระงานนั้นขึ้นอยู่กับความต้องการของผู้ใช้ที่มีต่อระบบหรืองานนั้นๆ โดยรูปที่ 2.3 จะแสดงถึงแผนผังรูปแบบการใช้งานการแบ่งภาระงาน



รูป 2.3 แผนผังรูปแบบการใช้งานการแบ่งภาระงาน(Load Balance)

2.2 งานที่เกี่ยวข้อง

2.2.1 Image Processing Application

เป็นบริการประมวลผลภาพที่อยู่ภายใน Application ไม่ว่าจะเป็นการตกแต่งรูปภาพ การเปลี่ยนสีภาพ หรือ การปรับแต่งขนาดรูปภาพ ก็ถือว่าเป็นการทำงานของ Image Processing ที่อยู่ในผู้ให้บริการ Application นั้น ๆ ว่ามีการให้บริการอย่างไร

2.2.2 AnimeFilter

AnimeFilter หรือ AnimeFilter.com เป็นผู้ให้บริการ Application การประมวลผลภาพโดยการให้ผู้ใช้งานนำเข้าภาพของตนเองที่ต้องการเพื่อให้ Application นำภาพนั้นไปประมวลผลโดยมีการประมวลผลดังนี้ จั๋ววัตถุของภาพที่ต้องการนำมาแปลงภาพให้อยู่ในรูปแบบ Model ที่ทางผู้ให้บริการได้เปิดให้ผู้ใช้งานเข้ามาใช้บริการแสดงดังรูปที่ 2.4



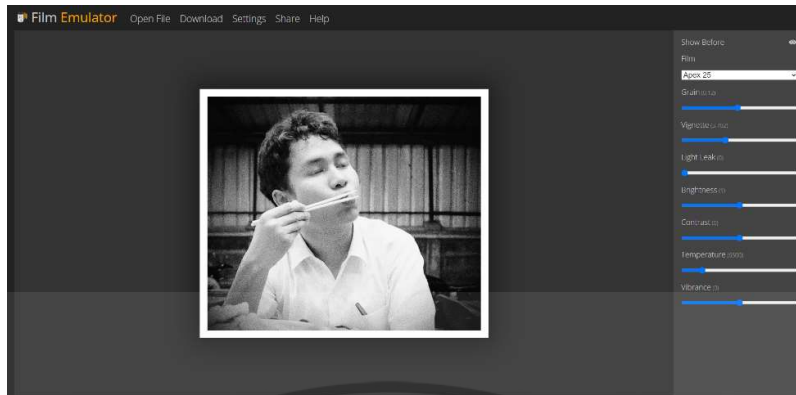
รูป 2.4 ตัวอย่างของงานประมวลผลภาพบน AnimeFilter

สิ่งที่ AnimeFilter ยังไม่มี หรือ นำมาปรับปรุง มาพัฒนาต่อในแอปพลิเคชันการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย

1. การนำเข้า Weight Model ที่ผู้ใช้ได้ Train มาเพื่อเปลี่ยน Filter เป็นแบบฉบับที่ผู้ใช้ได้ทำการ Train Weight ที่ต้องการมาใช้งาน
2. ให้บริการ Filter ที่หลากหลายแบบจากที่ผู้ใช้นำเข้ามาให้บริการใน Marketplace

2.2.2 Film Emulator

Film Emulator หรือ <https://29a.ch/film-emulator> เป็น ผู้ให้บริการ Application การประมวลผลภาพโดยการให้ผู้ใช้งานนำเข้าภาพของตนเองที่ต้องการเพื่อให้ Application นำภาพนั้นไปประมวลผลโดยมีการประมวลผลดังนี้ ปรับแต่งสีของภาพตามที่ใช้ต้องการ โดยมี parameter ที่กำหนดมาให้โดยมี Preset จากกล้องฟิล์มให้ผู้ใช้งานเข้ามาใช้บริการ ตัวอย่างแสดงดังรูปที่ 2.5



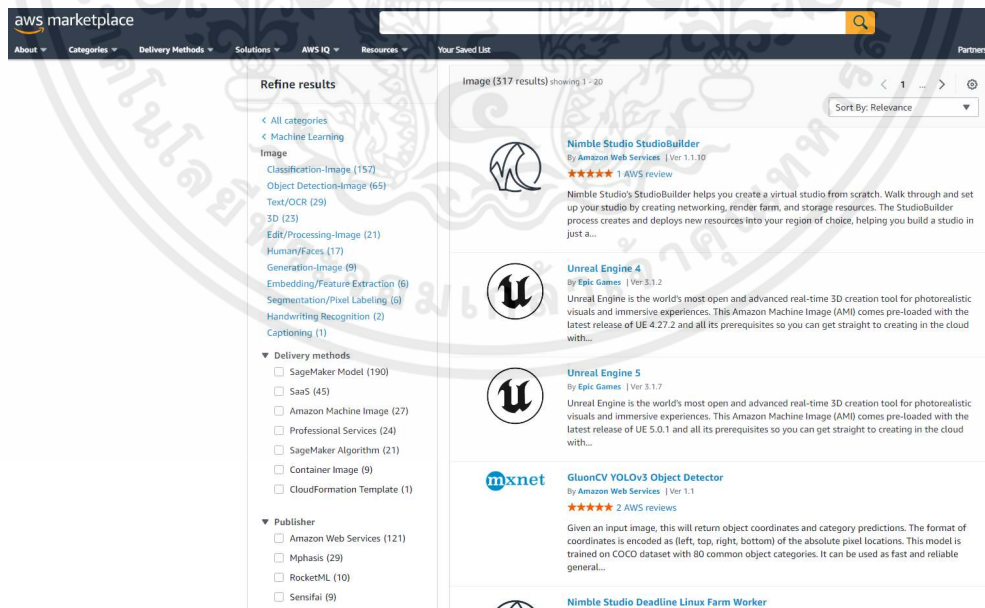
รูป 2.5 ตัวอย่างของ Film Emulator

สิ่งที่ Film Emulator ยังไม่มี หรือ นำมาปรับปรุง มาพัฒนาต่อในแอปพลิเคชันการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย

1. การรองรับการสั่งงานประมวลผลภาพจำนวนมาก
2. การรองรับการประมวลผลภาพจำนวนมากในการปรับค่า Parameter เดียวกัน
3. มี Application ในการปรับแต่งภาพหลากหลายแบบมากกว่า

2.2.3 aws marketplace

aws marketplace หรือ aws.amazon.com/marketplace/ เป็นผู้ให้บริการระบบ Cloud ที่ให้ผู้ใช้เลือกใช้บริการ Model ต่าง ๆ ที่ทาง aws รองรับโดยให้ผู้ใช้จ่ายค่าบริการเป็นรายชั่วโมงเพื่อเข้าใช้บริการงานต่าง ๆ เช่น YOLO, Unreal Engine 4 และ อื่น ๆ อีกมากมาย ดังรูป 2.6



รูป 2.6 ตัวอย่างของ aws marketplace

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งที่ aws marketplace ยังไม่มี หรือ นำมาปรับปรุง มาพัฒนาต่อในแอปพลิเคชันการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย

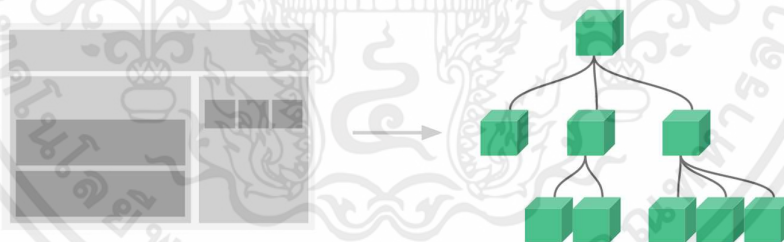
1. การให้บริการเป็นจำนวนภาพที่ประมวลผลแทนการให้บริการรายชั่วโมง
2. ให้ผู้ใช้รายย่อยอื่น ๆ นำเข้า Weight เพื่อมาซื้อขายได้จาก Model ที่ Application มีให้เหมือนกัน

2.3 เครื่องมือที่ใช้งานในการพัฒนาระบบ

2.3.1 Vue.js

Vue.js เป็น JavaScript Framework ที่ใช้สำหรับการพัฒนา หน้าจอแสดงผลผู้ใช้ (User Interface) เป็นหลัก โดยที่ library หลักของ Vue.js นั้นมุ่งเน้นไปที่การสร้าง View layer ของ Web application และ Vue.js ยังสามารถสร้าง Web application แบบ Single-Page Application ที่ไม่จำเป็นต้องมีการโหลดเมื่อสลับเปลี่ยนหน้าของ Web application

Vue.js มีส่วนสำคัญอย่างหนึ่งนั่นคือ component system ที่ช่วยให้เราสามารถสร้าง Web application ขนาดใหญ่ที่ประกอบไปด้วย components ขนาดเล็ก และสามารถนำ component เหล่านั้นมาใช้งานซ้ำได้อีกด้วย โดยตัวอย่างการทำงานของ Component stem ของ Vue.js แสดงดังรูปที่ 2.7



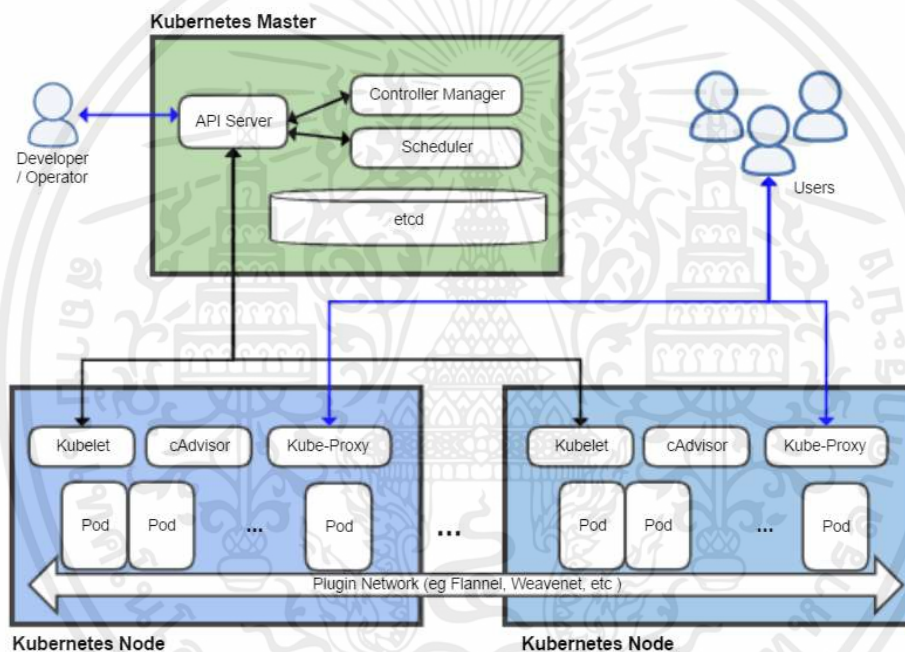
รูป 2.7 Component system ของ Vue.js

2.3.2 Docker

Docker คือชุด Platform ที่ใช้การจำลองเสมือนระดับ OS เพื่อดำเนินการทำงานส่วนแพ็คเกจที่เรียกว่าคอนเทนเนอร์ (Container) ซึ่งในแต่ละคอนเทนเนอร์แยกจากกันภายในแต่ละ คอนเทนเนอร์จะประกอบด้วยซอฟต์แวร์ ไบโบริ และ ไฟล์กำหนดค่าของตนเอง (Environment) อีกทั้งในแต่ละ Container ก็ยังสามารถสร้างช่องทางสื่อสารกันภายใน Container เพื่อให้ช่วยในงานประมวลผลโดยมีการใช้ทรัพยากรที่น้อย เมื่อเทียบเท่ากับการใช้งานบน OS ปกติทั่วไป และ ลดปัญหาในการเกิดข้อผิดพลาดของ Environment ในการใช้งาน

2.3.3 Kubernetes

คือ ถูกพัฒนาโดย Google เพื่อนำมา จัดการระบบคลัสเตอร์ (Cluster management software) สำหรับ Docker Container โดยตัว Kubernetes นั้นใช้สำหรับการ จัดการปล่อยซอฟต์แวร์อัตโนมัติ (automating software deployment) และ ทำการ Scaling ตรวจสอบความถูกต้องของ Container บนระบบ Cluster ได้ตลอดเวลา ตัว Kubernetes นั้นเนื่องจากเป็น Open-Source จึงมีการนำไปดัดแปลงแก้ไขเป็น Kubernetes เวอร์ชันอื่น ๆ มากอีกมากมายไม่ว่าจะเป็น MicroK8S, K3D, MiniKube โดยทั้งหมดที่กล่าวมามีเจือปน และ การทำงาน การใช้งานอยู่ในรูปแบบเดียวกัน โดยมีหลักการทำงานพื้นฐานเหมือนกันดัง รูป 2.8



รูป 2.8 รูปแบบการทำงานของ Kubernetes

2.3.4 Python

Python เป็นภาษาในการเขียนโปรแกรมที่ใช้อย่างแพร่หลายในการพัฒนาซอฟต์แวร์ เว็บ แอปพลิเคชัน วิทยาศาสตร์ข้อมูล การประมวลผลภาพ และ แมชชีนเลิร์นนิง (ML) มีการเรียนรู้ง่าย และสามารถทำงานบนแพลตฟอร์มได้มากมายไม่ว่าจะเป็นบน Window, MacOS, Linux ทั้งนี้ Python เป็นซอฟต์แวร์ที่สามารถดาวน์โหลด และ ใช้งานได้ฟรี

2.3.5 Django

Django เป็น Framework ที่ถูกเขียนด้วยภาษา Python สำหรับพัฒนาเว็บแอปพลิเคชัน โดยที่สามารถพัฒนาได้อย่างรวดเร็ว และมีประสิทธิภาพ เหมาะกับการพัฒนาเว็บที่มีการใช้งานบ่อย เช่น การตรวจสอบการดึงข้อมูลจากฐานข้อมูล การจัดการคุกกี้ และ การทำงานควบคู่ไปกับ Python ซอฟต์แวร์อื่น ๆ มาปรับให้ใช้งานภายในเว็บแอปพลิเคชันได้ ตัว Django เป็น open-source ที่สามารถนำมาใช้งานได้ฟรี และ พัฒนาได้ฟรีโดยมี Community ที่ช่วยดูแลรักษาซอฟต์แวร์ตัวนี้ และมีองค์กรไม่แสวงหาผลกำไรชื่อว่า Django Software Foundation ที่ทำการพัฒนา ปรับปรุง และ บำรุงรักษา ตัว Django ให้มีคุณภาพสูง และ ฟีเจอร์อีกมากมายให้ใช้งาน โดยไม่มีค่าใช้จ่ายตลอดมา

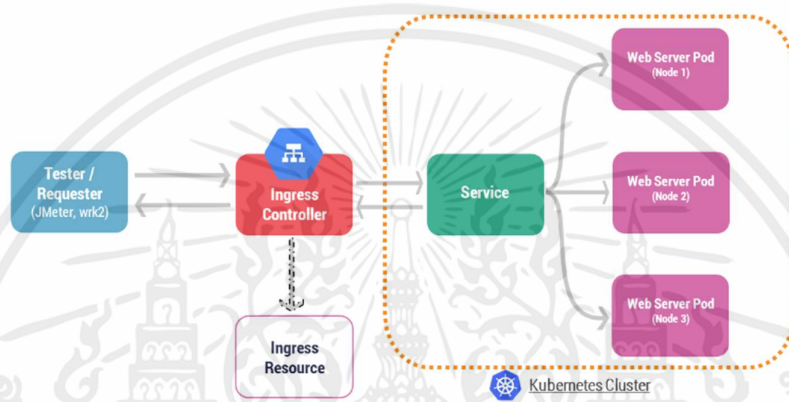
2.3.6 MongoDB

MongoDB เป็น NoSQL Database ที่สามารถใช้งานข้าม Platform ต่าง ๆ ได้โดยตัว MongoDB ได้ใช้ NoSQL เป็นการเข้าถึงข้อมูลโดยใช้เทคนิคของการดึงข้อมูลผ่าน Key Pair Value โดยเหมาะกับการทำงานที่ต้องการเข้าถึงข้อมูลได้อย่างรวดเร็ว และ เหมาะกับการทำงานงาน Big Data ตัวของ MongoDB สามารถที่จะสร้างเป็น Cluster เพื่อตอบสนองความต้องการเป็น High Availability (HA) ได้ และสามารถทำการ Auto Scale ไม่ว่าจะมีการใช้งานมาก หรือ น้อยแค่ไหนก็สามารถปรับแต่ง Environment นั้น ๆ ได้ให้เข้ากับการใช้งาน

2.4 งานวิจัยที่เกี่ยวข้อง

2.4.1 Performance evaluation and comparison of ingress controllers on Kubernetes cluster

งานวิจัยนี้จัดทำในปี พ.ศ. 2561 โดย คุณอาชิป พวงลำไย และ คุณชัยพร เขมระภาคะพันธ์ กล่าวถึงเรื่องการทดลองเพื่อประเมินสมรรถนะ และเปรียบเทียบประสิทธิภาพของ ระบบ Load balance แบบ Layer – 7 ที่ถูกใช้งานในสภาพแวดล้อมการทำงานที่อยู่ในรูปแบบของ Container cluster ด้วย Kubernetes โดยแผนผังการวางระบบในการวิจัยจะเป็นดังรูป 2.9



รูป 2.9 แผนผังแสดงระบบที่ใช้ในการทดสอบ Ingress controller

โดยโครงการนี้ได้นำเอกสารนี้มาศึกษาเพื่อใช้ในการพัฒนาส่วน Cluster Computing เพื่อให้การจัดลำดับงานและการประมวลผลใน Cluster มีประสิทธิภาพมากขึ้น

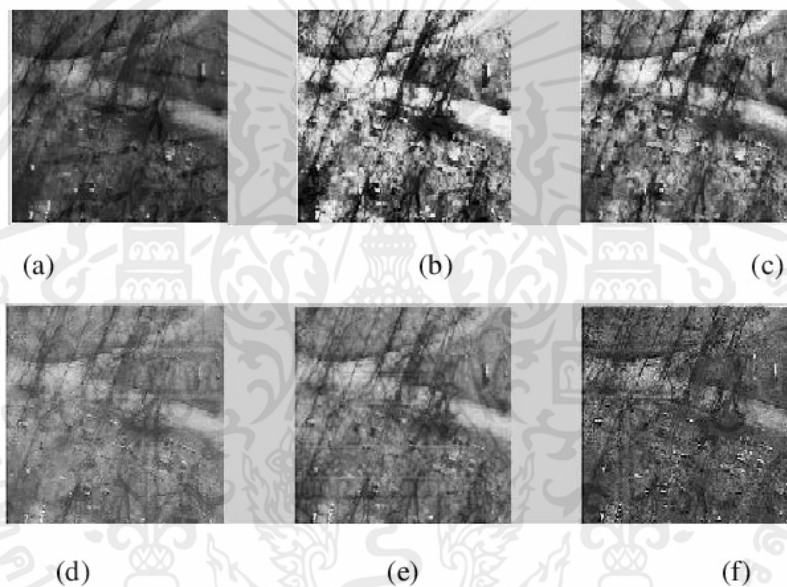
2.4.2 A modified gray-level difference algorithm for analyzing Gaussian Blurred texture images

งานวิจัยนี้จัดทำขึ้นในปี 2011 โดย Rui Zhang, Xiang Qian และ Datian Ye กล่าวถึงเรื่องการกระบวนกรแยกแยะพื้นผิวของรูปภาพโดยการใช้อัลกอริทึม gray-level difference (GLD) โดยจะมี 4 ขั้นตอนคือ 1. ประเมินความแตกต่างของพื้นผิวรูปภาพโดยเปรียบเทียบข้อมูลที่เตรียมเอาไว้ก่อนหน้า 2. ใช้ตัวกรองแบบ Wiener เพื่อลบ Gaussian blur noise ที่ที่อาจเกิดขึ้นหากค่าความแตกต่างมีน้อย 3. ทำซ้ำวิธีการที่ 2. ไปจนกระทั่งค่าความแตกต่างสูงกว่าข้อมูลที่เตรียมเอาไว้ 4. ใช้อัลกอริทึม gray-level difference แบบดั้งเดิม เพื่อเก็บข้อมูลพื้นผิวจากรูปภาพที่ผ่านการประมวลผลแล้ว เพื่อนำมาวัดประสิทธิภาพเทียบกับอัลกอริทึม gray-level difference แบบใหม่

โดยโครงการนี้ได้นำเอกสารนี้มาเพื่อศึกษาใช้ในการพัฒนาส่วนของการพัฒนา Image processing application เพื่อสร้าง Remove background หรือ ลบพื้นหลัง เพื่อมาเป็น 1 ใน Image processing application ที่จะอยู่ใน Web application

2.4.3 Contrast Stretching Enhancement in Remote Sensing Image

งานวิจัยนี้จัดทำขึ้นในปี 2011 โดย Salem Saleh Al-amri กล่าวถึงเรื่องการปรับแต่งรูปภาพที่ถูกถ่ายมาจากดาวเทียมในส่วนของความคมชัดของภาพด้วยเทคนิค Local contrast stretching (LCS) โดยจะทำการ sliding window ผ่านรูปภาพแล้วปรับความคมชัดของภาพไปที่ละส่วน โดยสามารถปรับให้ความคมชัดของตัวรูปภาพนั้นเพิ่มขึ้นและลดลงได้ตามความต้องการ และนอกจากนี้ก็ยังนำเสนอวิธีการอื่นที่ใช้ในการปรับแต่งความคมชัดเช่น the bi-histogram equalization (BHE) และ Global histogram equalization เป็นต้น โดยรูปที่ 2.10 จะเป็นรูปผลลัพธ์จากการปรับต่างด้วยวิธีที่แตกต่างกัน



รูป 2.10 รูปผลลัพธ์จากการปรับต่างความคมชัดด้วยวิธีที่แตกต่างกัน โดย (a) Original Image Contrast , (b) Histogram Equalization Contrast Enhancement , (c) Adaptive Histogram Equalization , (d) LPF Homomorphic Contrast Enhancement , (e) HPF Homomorphic Contrast Enhancement และ (f) Unsharp Mask Contrast Enhancement

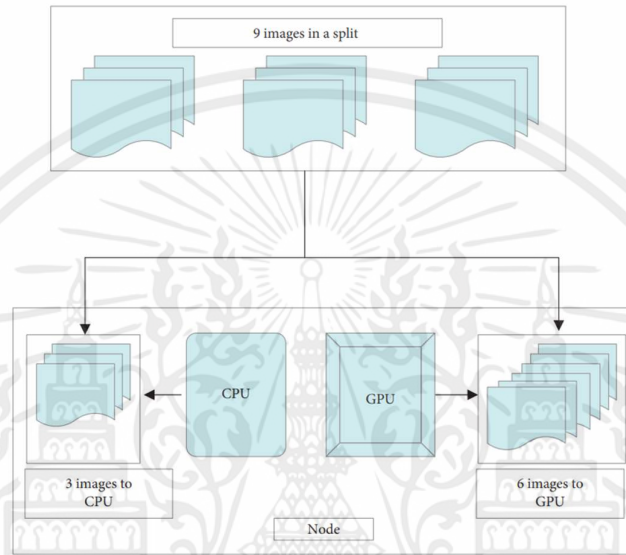
โดยโครงการนี้ได้นำเอกสารนี้มาศึกษาเพื่อใช้ในการพัฒนาส่วนของ Image processing application เพื่อสร้าง Filtering และการปรับความคมชัดของรูปภาพที่จะถูกปรับแต่งภายใน Image processing application

2.4.4 Efficient Processing of Image Processing Applications on CPU/GPU

งานวิจัยนี้จัดทำขึ้นในปี 2020 โดย Najia Naz, Abdul Haseeb Malik, Abu Bakar Khurshid, Furqan Aziz, Bader Alouffi, M. Irfan Uddin และ Ahmed AlGhamdi กล่าวถึงเรื่องการกระบวนการ การแบ่งงานที่มีประสิทธิภาพสำหรับ Image processing application จากปกติการที่ประมวลผลงานด้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

image processing สิ่งที่สามารถพบเจอได้เป็นปกติคือ การกระจายภาระงานที่ไม่สมดุล ซึ่งทำให้ประสิทธิภาพของการทำงานนั้นลดลง โดยการแบ่งนั้นจะแบ่งออกเป็น 2 ช่วง ช่วงแรกจะเป็นการแบ่งและแยกแยะข้อมูลของรูปภาพเป็นส่วนๆ ที่เหมาะสมต่อการกระจายไปยังหน่วยประมวลผลต่างๆ จากนั้นในช่วงที่สอง ทำการกระจายเพิ่มเติมไปยัง CPU และ GPU ขึ้นอยู่กับความเร็วในการประมวลผล ดังรูป 2.11

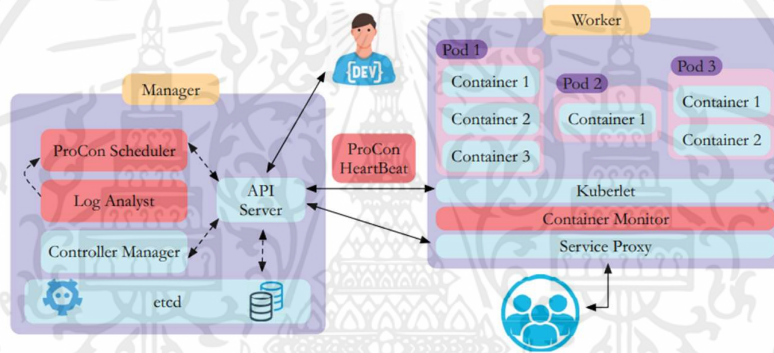


รูป 2.11 แผนผังรูปแบบการกระจายงานในช่วงที่สองที่มีการกระจายภาระงานระหว่าง CPU และ GPU

โดยโครงการนี้ได้นำเอกสารนี้มาศึกษาเพื่อใช้ในการพัฒนาส่วน Load balance ที่จะทำหน้าที่แบ่งงานภาระงาน Image processing ไปยัง CPU และ GPU

2.4.5 Progress-based Container Scheduling for Short-lived Applications in a Kubernetes Cluster

งานวิจัยนี้จัดทำขึ้นในปี 2019 โดย Yuqi Fu, Shaolun Zhang, Jose Terrero, Ying Mao, Guangya Liu, Sheng Li และ Dingwen Tao กล่าวถึงเรื่องการจัดวางรูปแบบ คอนเทนเนอร์ ที่ตั้งชื่อไว้ว่า ProCon โดยปกติแล้วจะมี Manager node เป็นตัวที่เลือกใช้งาน Worker node แต่ละตัวตามอัลกอริทึมที่ตั้งค่าเอาไว้ โดยที่ ProCon จะทำหน้าที่ในการกำหนด input คอนเทนเนอร์แล้ว พิจารณาการใช้ทรัพยากรทันที รวมไปถึงการประเมินทรัพยากรในอนาคตด้วย และขณะที่ประมวลผลนั้น ProCon ก็จะทำการจัดสมดุลของทรัพยากรภายใน Cluster ไปด้วย โดยรูปแบบแผนผังของ System Architecture ของ ProCon จะเป็นดังรูปที่ 2.12



รูป 2.12 แผนผัง System Architecture ของ ProCon

โดยโครงการนี้ได้ร่วมงานวิจัยมาศึกษาเพื่อสร้างรูปแบบการจัดการคอนเทนเนอร์ที่มีการจัดทรัพยากรให้แต่ละ Worker อย่างมีประสิทธิภาพ

บทที่ 3

การออกแบบและการพัฒนา

การออกแบบและพัฒนาระบบ แอปพลิเคชันการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย แบ่งออกเป็น 3 ส่วน คือ Web Application ส่วนของ Image Processing Application และ ส่วนของ Server โดยมีรายละเอียดส่วนต่าง ๆ ดังนี้

3.1 รายละเอียดของระบบ

1) Web Application

- 1) ผู้ใช้สามารถสมัครสมาชิก เข้าสู่ระบบได้
- 2) สามารถเปลี่ยนรหัสผ่านได้
- 3) สามารถจัดเก็บไฟล์รูปภาพได้
- 4) สามารถส่งงานประมวลผลภาพได้
- 5) สามารถทดลองดูตัวอย่างงานประมวลผลภาพได้
- 6) สามารถจัดเก็บ Weight Model ได้
- 7) สามารถซื้อขาย Weight Model ได้
- 8) สามารถส่งงานประมวลผลภาพบน Marker Place ได้
- 9) สามารถดูสถิติการประมวลผลภาพได้

2) Image Processing Application

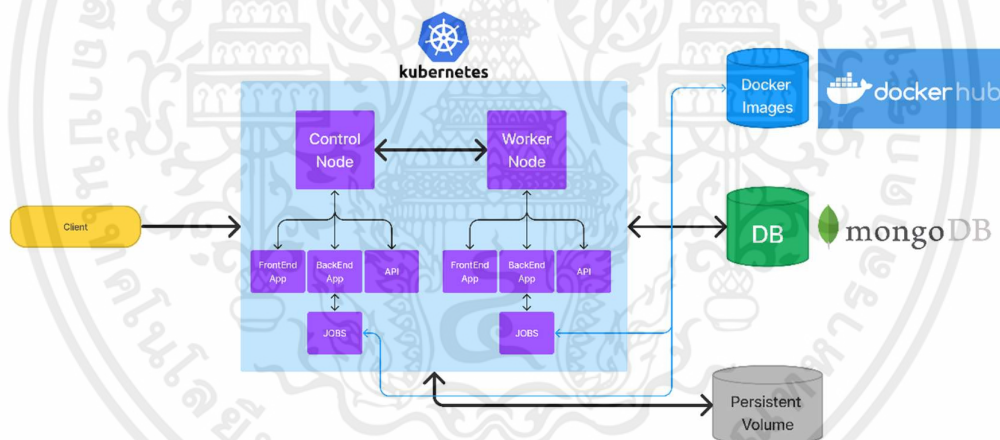
- 1) สามารถประมวลผลภาพเป็นภาพ ASCII ได้
- 2) สามารถประมวลผลภาพเป็นภาพ PixelArt ได้
- 3) สามารถประมวลผลภาพเป็นภาพ Mosaic ได้
- 4) สามารถประมวลผลภาพเป็นภาพขาวดำได้
- 5) สามารถประมวลผลภาพเป็นภาพเพื่อลบพื้นหลังภาพได้
- 6) สามารถประมวลผลภาพเป็นภาพจาก Model และ Weight ได้

3) Server

- 1) สามารถ Task Management งานประมวลผลได้
- 2) สามารถ Load Balance การ Request เข้าถึง Service ได้
- 3) สามารถจัดเก็บข้อมูลไว้แยกจากตัว Server ได้
- 4) สามารถ High Availability Server ได้
- 5) สามารถ Scale Service บน Server ได้

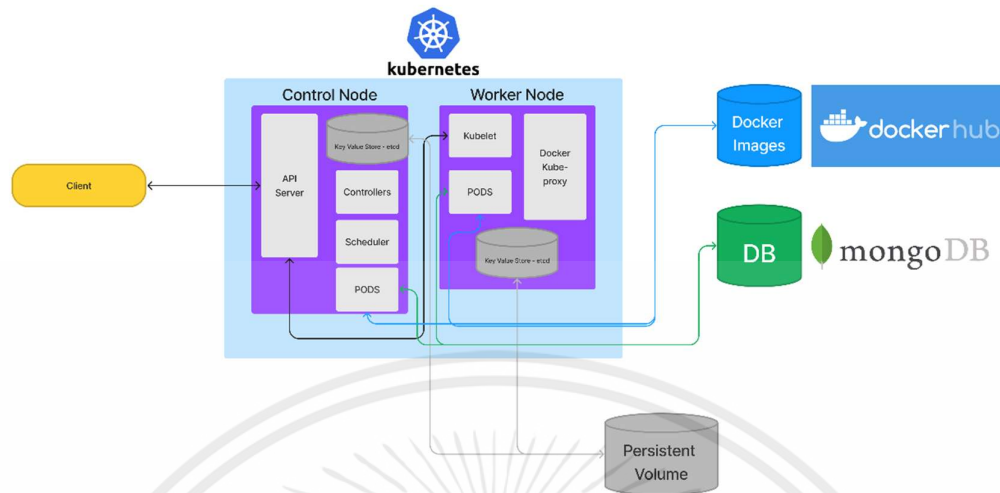
3.2 ภาพรวมของระบบ

ภาพรวมในการพัฒนาระบบที่แสดงให้เห็นส่วนต่าง ๆ ที่อยู่ในแผนการพัฒนาโครงการที่ประกอบไปด้วยส่วนประกอบหลัก ๆ คือ Kubernetes ที่ครอบระบบทุกอย่างอยู่โดยจะมี Node เป็น Control ที่ใช้ควบคุมการทำงานต่าง ๆ และ รับคำสั่งต่าง ๆ ให้กับ Worker โดยที่มีการเชื่อมต่อกับ DockerHub MongoDB และ สร้าง Persistent Volume แยกขึ้นมา จะแสดงดังรูปที่ 3.1



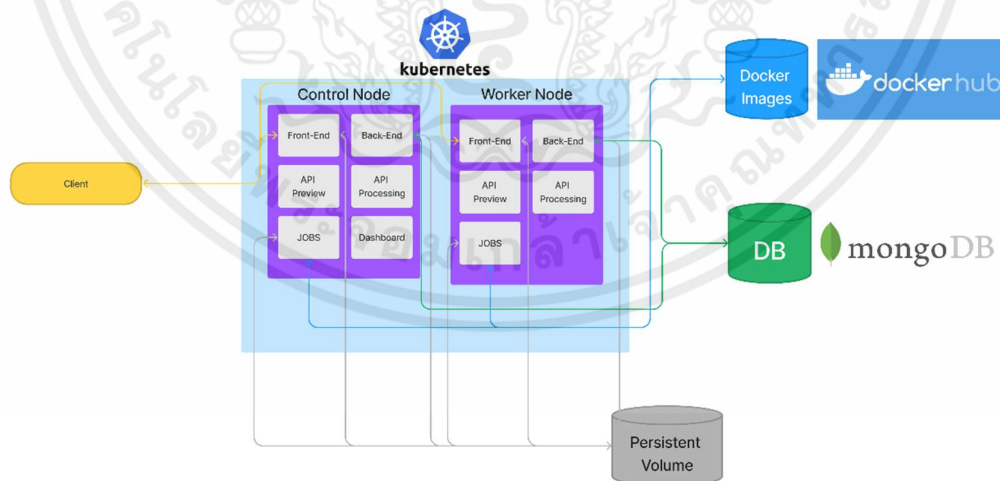
รูป 3.1 ภาพรวมของระบบ Overview

โดยภาพรวมของระบบใน Level 0 นั้นจะกล่าวถึงการเชื่อมต่อของ Control Node และ Worker Node ที่มีการ Scheduler ผ่าน API-server เพื่อทำการ Load Balance งานที่ Control ได้รับไปยัง Worker แต่ละเครื่องตามที่ได้ตั้งค่าไว้ใน Load Balancer ของ Kubernetes ดังรูปที่ 3.2



รูป 3.2 ภาพรวมของระบบ Level 0

โดยภาพรวมของระบบใน Level 1 นั้นจะกล่าวถึงการเชื่อมต่อของ User ไปยัง Web Service นั้นโดยมี Port การเชื่อมต่อไป ในส่วนของ Front-End และ เข้าใช้งานส่วนของการเรียกใช้อื่น ๆ ผ่านตัว Back-End ซึ่งจะมีการเปิดรับ API สำหรับทำการประมวลผลภาพเป็น Preview และ Processing สำหรับการประมวลผลงานส่วนอื่น ๆ มีตัว JOBS ที่เก็บงานประมวลผลที่ได้รับตั้งเข้ามา และ กำลังทำงานอยู่ รวมถึง หน้า Dashboard สำหรับจัดการ ฝ้าดูการทำงานของระบบโดยรวมได้ ดังรูปที่ 3.3



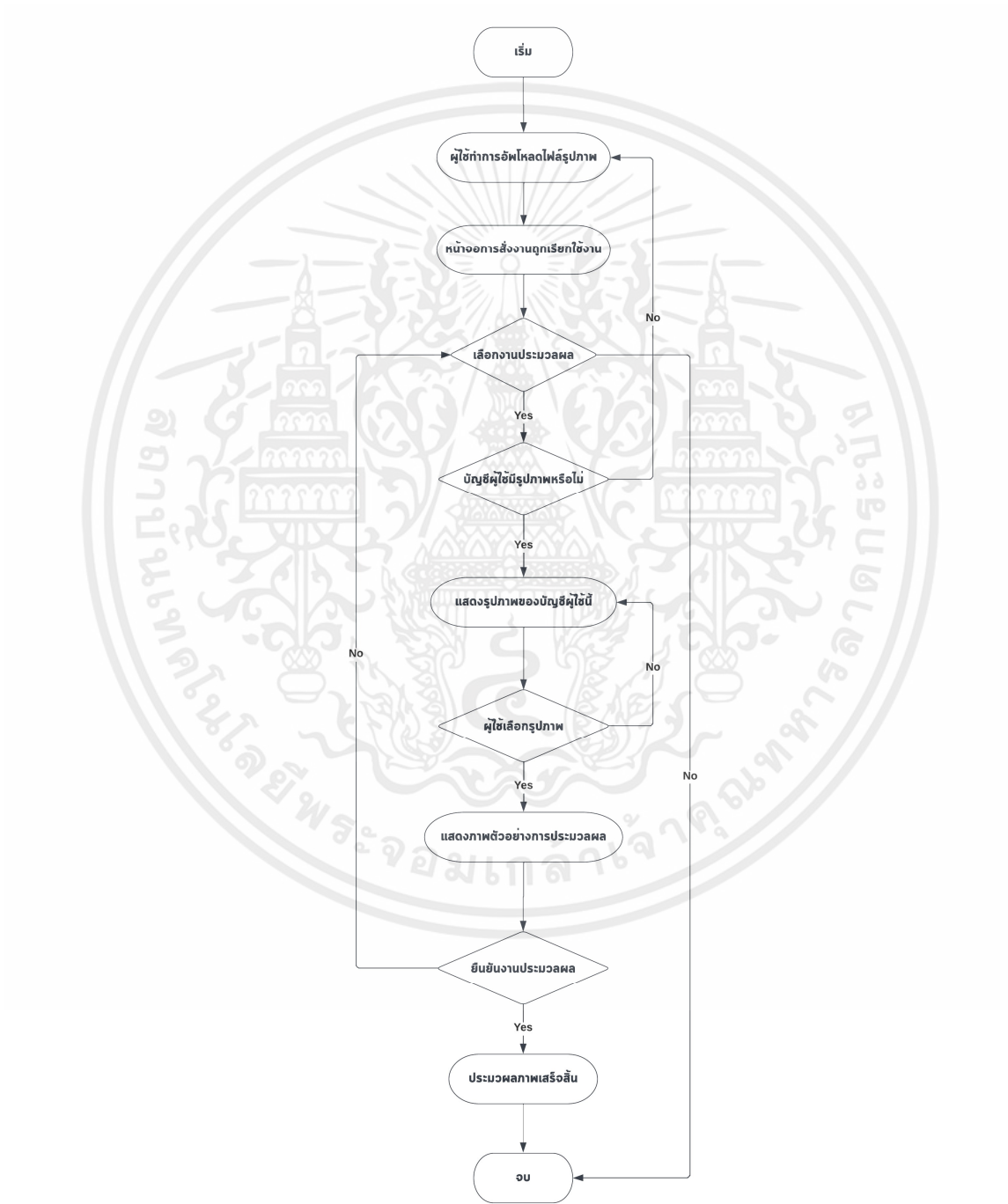
รูป 3.3 ภาพรวมของระบบ Level 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 แผนภาพผังงาน (Flowchart)

3.3.1 แผนภาพผังงาน (Flowchart) ของการสั่งงานประมวลผลภาพ (Web Application)

แผนภาพผังงานในส่วนของ Web Application บนเว็บที่ผู้ใช้งานเริ่มทำการใช้งานการสั่งงานประมวลผลภาพจนถึงการประมวลผลภาพเสร็จสิ้น โดยแผนภาพผังงานในส่วนของคำสั่งงานประมวลผลภาพ จะแสดงดังรูปที่ 3.4

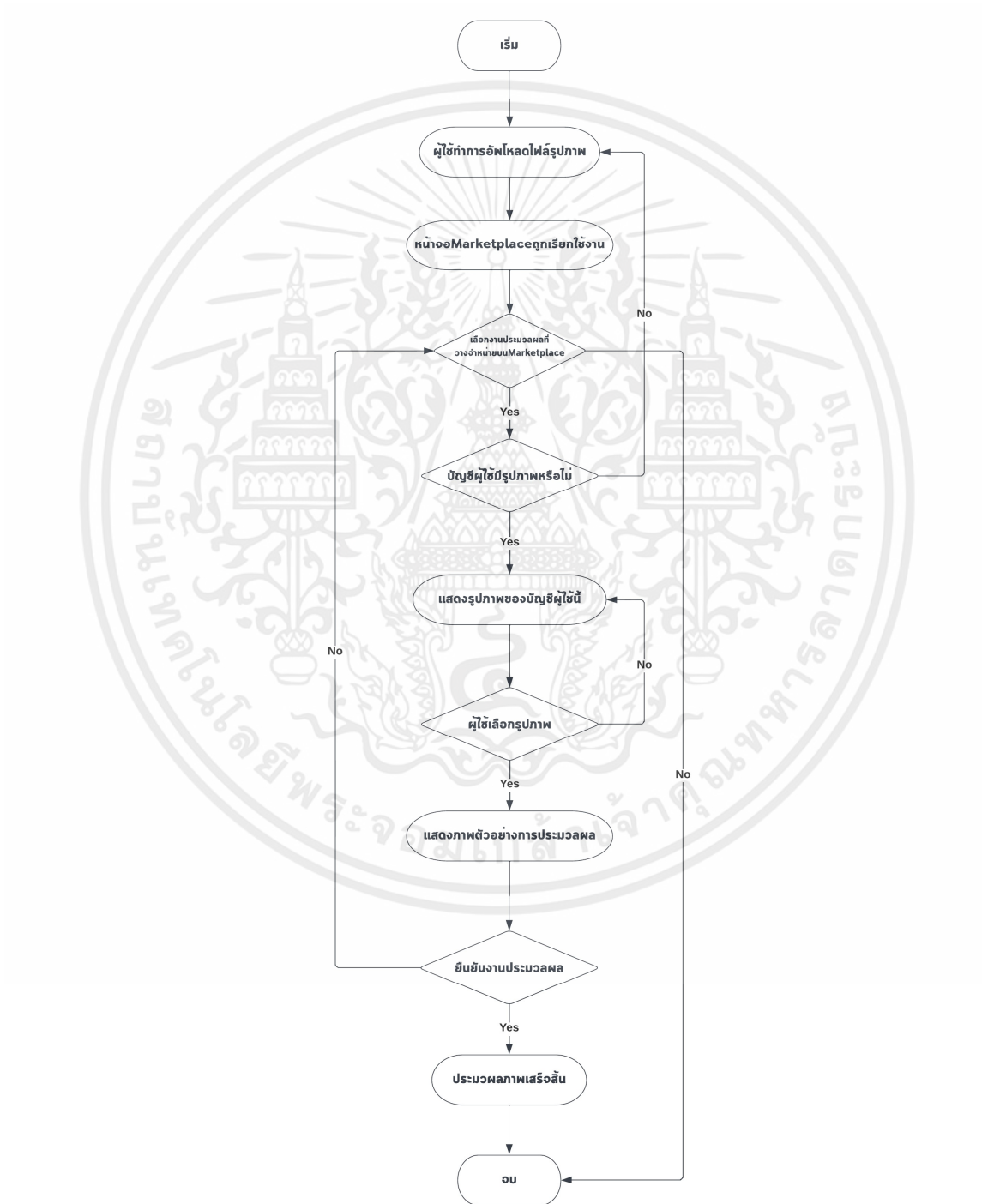


รูป 3.4 แผนภาพผังงานของการสั่งงานประมวลผลภาพ (Web Application)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 แผนภาพผังงาน (Flowchart) การสั่งงานประมวลผลภาพบน Marketplace (Web Application)

แผนภาพผังงานในส่วนของ Web Application บนเว็บที่ผู้ใช้งานเริ่มทำการใช้งานการสั่งงานประมวลผลภาพบน Marketplace จนถึงการประมวลผลภาพเสร็จสิ้น โดยแผนภาพผังงานในส่วนของการสั่งงานประมวลผลภาพ จะแสดงดังรูปที่ 3.5

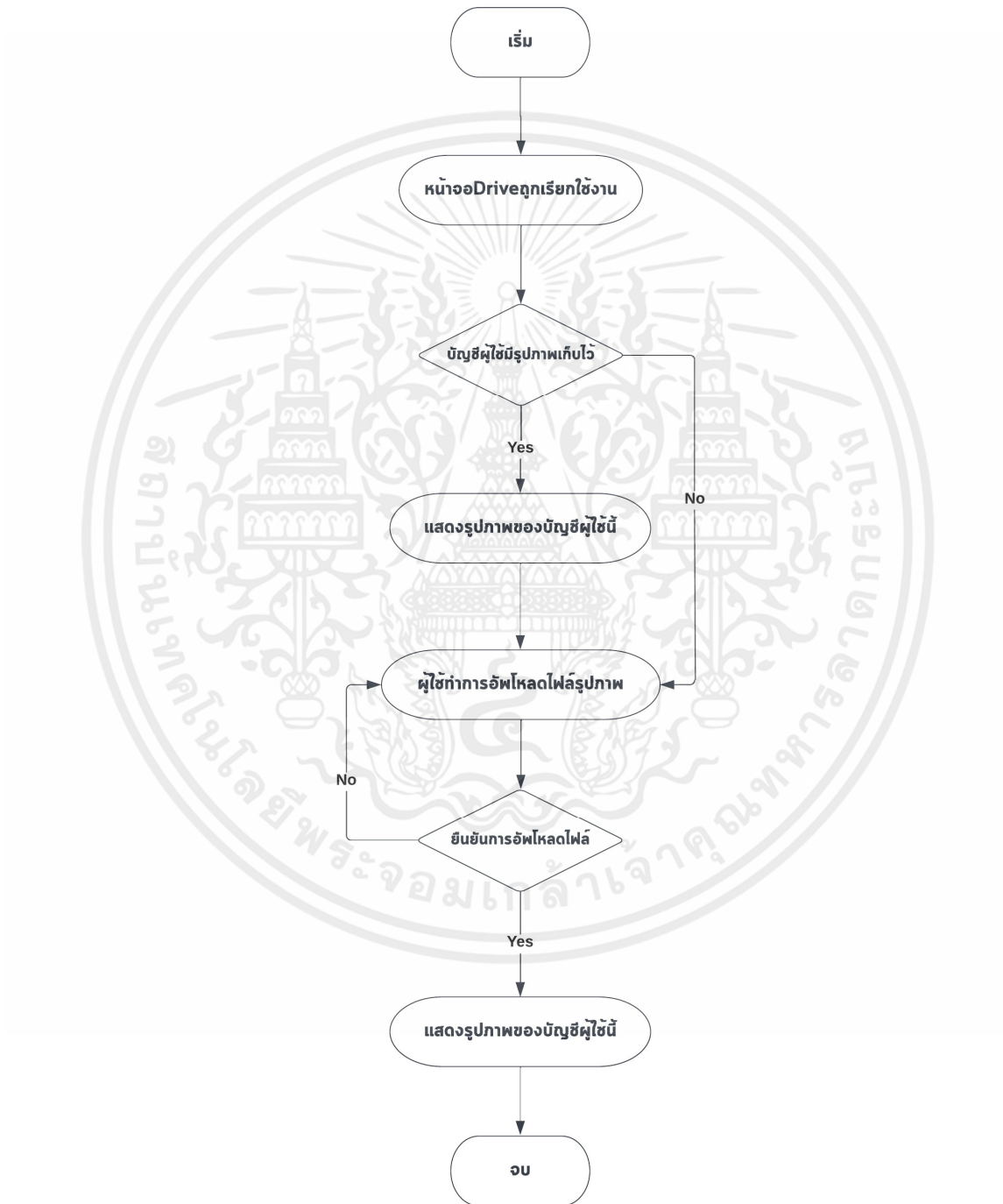


รูป 3.5 แผนภาพผังงานของการสั่งงานประมวลผลภาพบน Marketplace (Web Application)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 แผนภาพผังงาน (Flowchart) การจัดเก็บไฟล์ภาพไว้ใน Drive (Web Application)

แผนภาพผังงานในส่วนของ Web Application บนเว็บที่ผู้ใช้งานเริ่มทำการจัดเก็บไฟล์ภาพไว้ใน Drive จนถึงผลการจัดเก็บไฟล์ภาพเสร็จสิ้น โดยแผนภาพผังงานในส่วนของการจัดเก็บไฟล์ภาพไว้ใน Drive จะแสดงดังรูปที่ 3.6

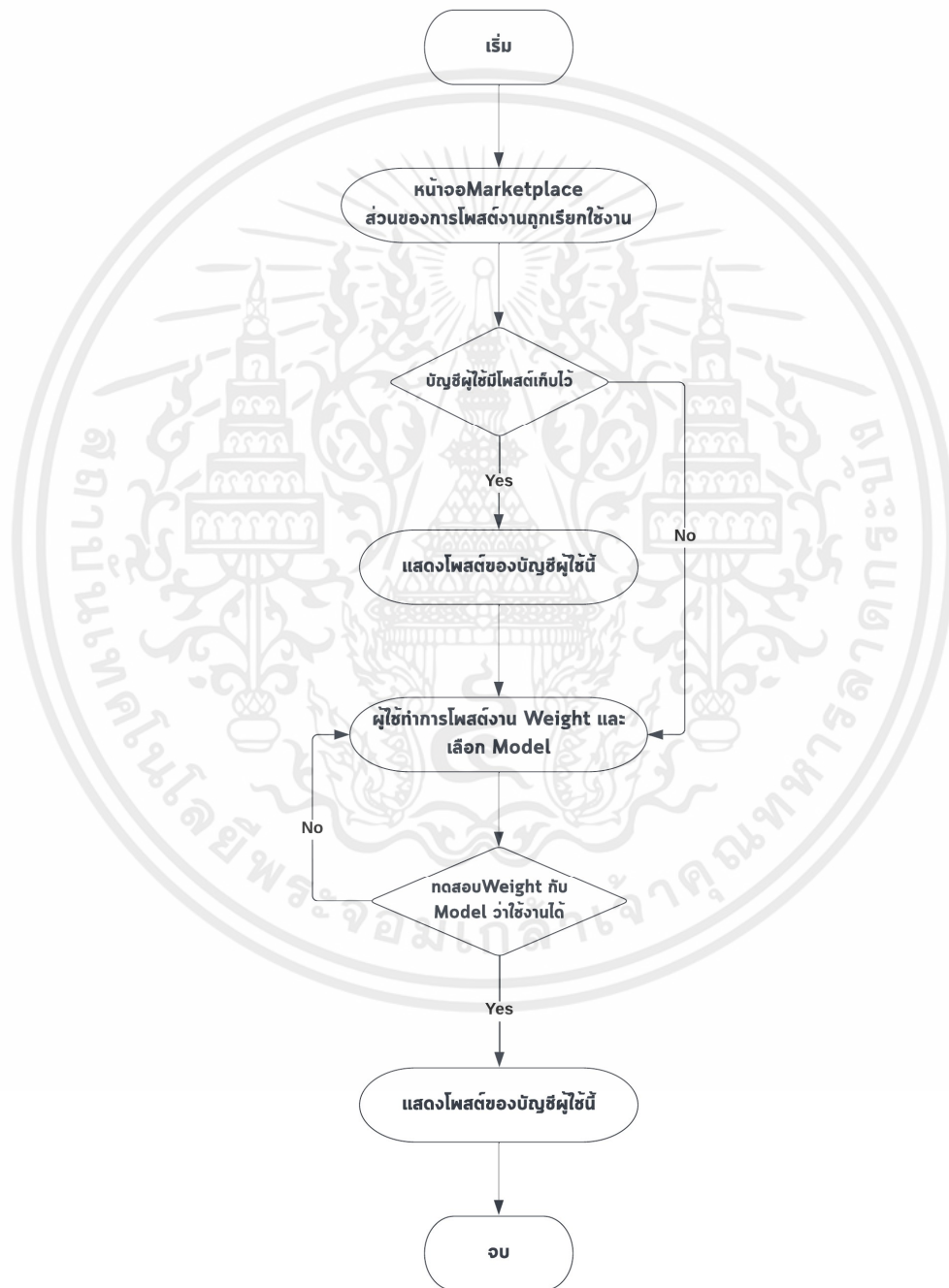


รูป 3.6 แผนภาพผังงานของการการจัดเก็บไฟล์ภาพไว้ใน Drive (Web Application)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.4 แผนภาพผังงาน (Flowchart) การใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train (Web Application)

แผนภาพผังงานในส่วนของ Web Application บนเว็บที่ผู้ใช้งานเริ่มทำการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train จนถึงการโพสต์งานเสร็จสิ้น โดยแผนภาพผังงานในส่วนของ การขาย Weight ที่ผู้ใช้ได้ทำการ Train จะแสดงดังรูปที่ 3.7

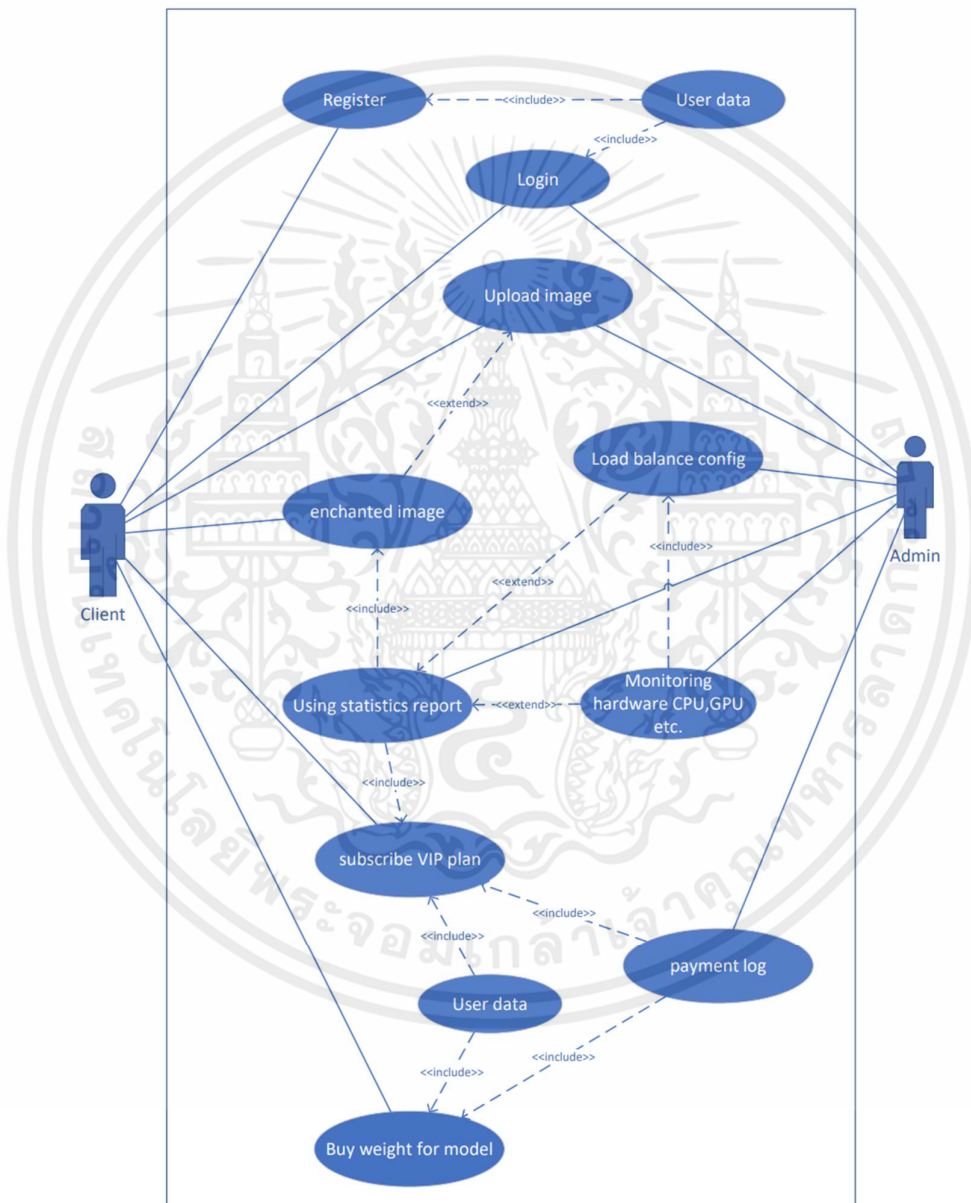


รูป 3.7 แผนภาพผังงานของการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 แผนภาพยูสเคส (Use Case Diagram)

แผนภาพยูสเคสจะแสดงให้เห็นถึงผู้ใช้ที่ถูกแบ่งเป็น 2 ประเภท คือ Client คือ ผู้ใช้ที่เข้ามาในรูปแบบลูกค้าที่ต้องการใช้บริการงานประมวลผลภาพต่างๆ และ Administer คือผู้ใช้ที่เป็นผู้ดูแลในส่วน of ข้อมูลในระบบ และ ตั้งค่า ในส่วนของการประมวลผลภาพ รวมไปถึงการตรวจสอบการ payment ต่างๆ โดยแผนภาพ ยูสเคส จะแสดงดังรูปที่ 3.8



รูป 3.8 แผนภาพยูสเคส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย ส่วนรายละเอียดของ Use case จะแสดงดังตารางที่ 3.1 ถึง 3.10

ตาราง 3.1 รายละเอียดของ Use case Register

Use Case Title : Register	Use case ID: 1
Description : ลงทะเบียนสมัครเข้าใช้งาน	
Actor : Client	
Precondition : เข้าใช้งานเว็บไซต์	
Postcondition : ได้รับผลการยืนยันการสมัคร	
Fail End Condition : แจ้งเตือนข้อความ “การสมัครบัญชีล้มเหลว เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
Trigger : กดปุ่มคลิกยืนยันการสมัครใช้งาน	

ตาราง 3.2 รายละเอียดของ Use case Login

Use Case Title : Login	Use case ID: 2
Description : ลงชื่อเข้าใช้งานเว็บไซต์	
Actor : Client, Administer	
Precondition : เข้าใช้งานเว็บไซต์และสมัครเข้าใช้งานแล้ว	
Postcondition : ได้รับผลการยืนยันตัวตน	
Fail End Condition : แจ้งเตือนข้อความ “การลงชื่อเข้าใช้ล้มเหลว เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
Trigger : กดปุ่มคลิกลงชื่อเข้าใช้	

ตาราง 3.3 รายละเอียดของ Use case Upload image

Use Case Title : Upload image	Use case ID: 3
Description : อัปโหลดรูปภาพเพื่อเก็บไว้ในฐานข้อมูล	
Actor : Client, Administer	
Precondition : ยืนยันตัวตนลงชื่อเข้าใช้	
Postcondition : ได้รับผลอัปโหลดรูปและเก็บรูปภาพไว้ในฐานข้อมูล	
Fail End Condition : แจ้งเตือนข้อความ “การอัปโหลดรูปภาพล้มเหลว เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
Trigger : เลือกรูปภาพที่จะอัปโหลดและกดปุ่มอัปโหลด	

ตาราง 3.4 รายละเอียดของ Use case enchanted image

Use Case Title : Enchanted image	Use case ID: 4
Description : ปรับแต่งรูปภาพ	
Actor : Client	
Precondition : ยืนยันตัวตนลงชื่อเข้าใช้	
Postcondition : ได้รับรูปภาพที่ผ่านการปรับแต่ง	
Fail End Condition : แจ้งเตือนข้อความ “การปรับแต่งรูปภาพรูปภาพล้มเหลว เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
Trigger : เลือกรูปภาพที่จะปรับแต่ง เลือกฟังก์ชันการปรับแต่ง และกดปุ่มปรับแต่ง	

ตาราง 3.5 รายละเอียดของ Use case Load balance config

Use Case Title : Load balance config	Use case ID: 5
Description : ปรับแต่งการทำงานของ Load balance	
Actor : Administer	
Precondition : ยืนยันตัวตนลงชื่อเข้าใช้	
Postcondition : ได้ผลยืนยันการปรับแต่ง Load balance	
Fail End Condition : แจ้งเตือนข้อความ “การปรับแต่งรูปภาพรูปภาพล้มเหลว เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
Trigger : เลือกฟังก์ชันการปรับแต่ง และกดปุ่มปรับแต่ง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.6 รายละเอียดของ Monitoring hardware

Use Case Title : Monitoring hardware	Use case ID: 6
Description : คุรระดับการทำงานของฮาร์ดแวร์ต่างๆ เช่น CPU ,GPU เป็นต้น	
Actor : Administer	
Fail End Condition :	
Precondition : ยืนยันตัวตนลงชื่อเข้าใช้	
Postcondition : ได้ผลการทำงานของฮาร์ดแวร์	
Fail End Condition : แจ้งเตือนข้อความ “ไม่สามารถเข้าถึงระดับการทำงานของฮาร์ดแวร์ได้ เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
Trigger : เลือกฟังก์ชันคุรระดับการทำงาน	

ตาราง 3.7 รายละเอียดของ Using statistics report

Use Case Title : Using statistics report	Use case ID: 7
Description : คุสติการใช้งานของผู้ใช้ที่อยู่ในระดับ Client	
Actor : Administer	
Precondition : ยืนยันตัวตนลงชื่อเข้าใช้	
Postcondition : ได้สติการใช้งาน	
Fail End Condition : แจ้งเตือนข้อความ “ไม่สามารถเข้าถึงสติผู้ใช้ได้ เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
Trigger : เลือกฟังก์ชันคุสติการใช้งาน	

ตาราง 3.8 รายละเอียดของ Subscribe VIP plan

Use Case Title : Subscribe VIP plan	Use case ID: 8
Description : สมัครเข้าเป็นผู้ใช้ Client ระดับ VIP	
Actor : Client	
Precondition : ยืนยันตัวตนลงชื่อเข้าใช้	
Postcondition : ได้สถานะเป็น Client ระดับ VIP	
Fail End Condition : แจ้งเตือนข้อความ “การสมัครล้มเหลว เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
Trigger : เลือกสมัครเข้าเป็นผู้ใช้ Client ระดับ VIP และชำระเงิน	

ตาราง 3.9 รายละเอียดของ Buy weight for model

Use Case Title : Buy weight for model	Use case ID: 9
Description : ซื้อ weight สำหรับ Model เพื่อนำไปทำการประมวลผลภาพ	
Actor : Client	
Precondition : ยืนยันตัวตนลงชื่อเข้าใช้	
Postcondition : ได้รับ weight สำหรับ Model ไว้ในฐานข้อมูล	
Fail End Condition : แจ้งเตือนข้อความ “การซื้อขายล้มเหลว เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
Trigger : เลือก weight สำหรับ Model และชำระเงิน	

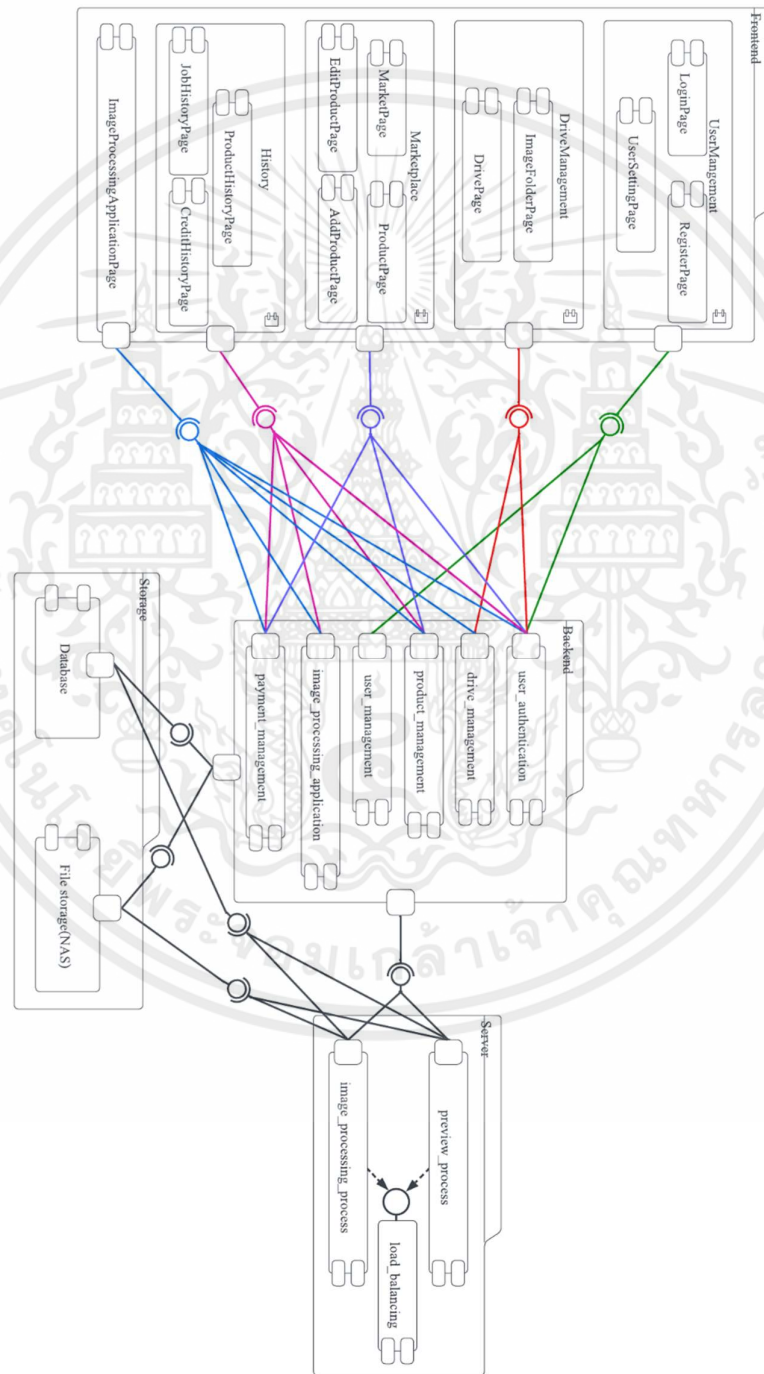
ตาราง 3.10 รายละเอียดของ Payment log

Use Case Title : Payment log	Use case ID: 10
Description : ดูบันทึกการชำระเงินต่างๆ	
Actor : Administer	
Precondition : ยืนยันตัวตนลงชื่อเข้าใช้	
Postcondition : ได้ผลบันทึกการชำระเงิน	
Fail End Condition : แจ้งเตือนข้อความ “ไม่สามารถเข้าถึงบันทึกการชำระเงินได้ เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
Trigger : เลือกฟังก์ชันดูบันทึกการชำระเงิน	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 แผนภาพแสดงองค์ประกอบ (Component Diagram)

แผนภาพแสดงโครงสร้างและความสัมพันธ์ระหว่างองค์ประกอบภายในโครงการเพื่อให้เกิดความเข้าใจและสามารถสื่อสารได้เข้าใจตรงกันภายในกลุ่มโครงการ และยังช่วยลดความซับซ้อนของระบบลงไปอีกด้วย โดยแผนภาพแสดงองค์ประกอบจะแสดงดังรูปที่ 3.9

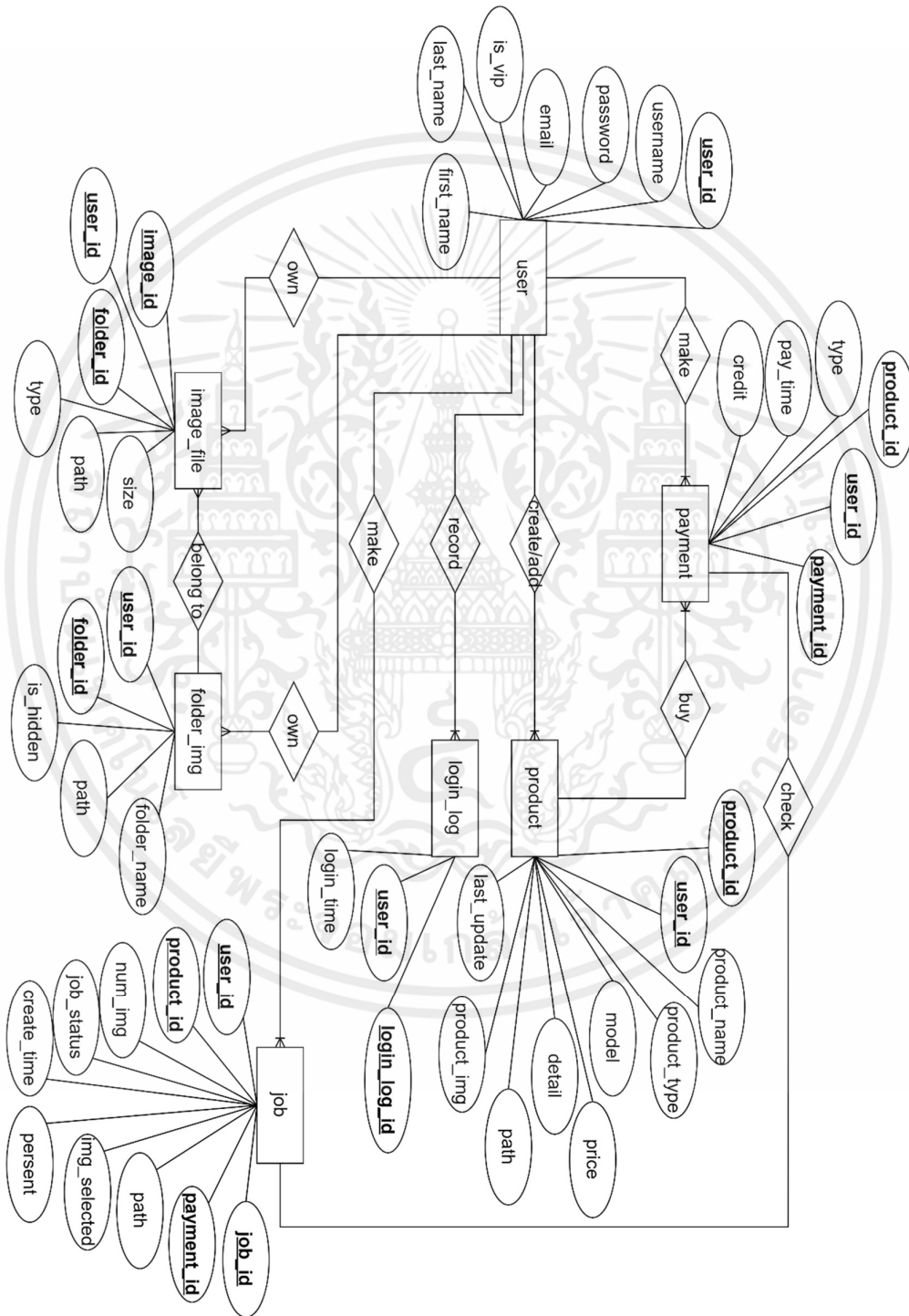


รูป 3.9 แผนภาพแสดงองค์ประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 แผนภาพแสดงความสัมพันธ์ของข้อมูล (E-R Diagram)

แผนภาพแสดงความสัมพันธ์ของข้อมูลที่ใช้ในการออกแบบระบบฐานข้อมูล ประกอบด้วย 9 tables คือ user, image_file, folder_img, img_app, job, login_log, product, payment และ feed_post โดยแผนภาพแสดงความสัมพันธ์ของข้อมูลจะแสดงดังรูปที่ 3.10



รูป 3.10 แผนภาพแสดงความสัมพันธ์ของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยรายละเอียดของแผนภาพแสดงความสัมพันธ์ของข้อมูลแสดงดังตารางที่ 3.11 ถึง 3.17

ตาราง 3.11 แสดงรายละเอียดตารางข้อมูลผู้ใช้ (user)

คีย์	ชื่อแอตทริบิวต์	ชนิดตัวแปร	คำอธิบาย
PK	user_id	Varchar(50)	รหัสประจำตัวของผู้ใช้งาน
-	username	Varchar(50)	ชื่อสำหรับแสดงผลของผู้ใช้
-	password	Varchar(50)	รหัสผ่านของผู้ใช้งาน
-	email	Varchar(50)	อีเมลของผู้ใช้งาน
-	is_vip	Boolean	สถานะระดับของผู้ใช้
-	first_name	Varchar(50)	ชื่อของผู้ใช้งาน
-	last_name	Varchar(50)	นามสกุลของผู้ใช้งาน

ตาราง 3.12 แสดงรายละเอียดตารางข้อมูลไฟล์รูปภาพ (Image_file)

คีย์	ชื่อแอตทริบิวต์	ชนิดตัวแปร	คำอธิบาย
PK	image_id	Varchar(50)	รหัสข้อมูลรูปภาพ
FK	user_id	Varchar(50)	รหัสประจำตัวของผู้ใช้งานเจ้าของรูปภาพ
FK	folder_id	Varchar(50)	รหัสข้อมูลโฟลเดอร์ที่เก็บรูปภาพ
-	type	Varchar(50)	ประเภทของรูปภาพ
-	path	Varchar(50)	เส้นทางการเข้าถึงไฟล์รูปภาพ
-	size	Integer	ขนาดของไฟล์รูปภาพ

ตาราง 3.13 แสดงรายละเอียดตารางข้อมูลของโฟลเดอร์เก็บรูปภาพ (folder_img)

คีย์	ชื่อแอตทริบิวต์	ชนิดตัวแปร	คำอธิบาย
PK	folder_id	Varchar(50)	รหัสข้อมูลโฟลเดอร์
FK	user_id	Varchar(50)	รหัสประจำตัวของผู้ใช้งานเจ้าของโฟลเดอร์
-	is_hidden	Varchar(50)	สถานะการแสดงผล
-	folder_name	Varchar(50)	ชื่อของโฟลเดอร์
-	path	Varchar(50)	เส้นทางการเข้าถึงโฟลเดอร์

ตาราง 3.14 แสดงรายละเอียดตารางเก็บข้อมูลการเข้าสู่ระบบ (login_log)

คีย์	ชื่อแอตทริบิวต์	ชนิดตัวแปร	คำอธิบาย
PK	login_log_id	Varchar(50)	รหัสข้อมูล Log การเข้าสู่ระบบ
FK	user_id	Varchar(50)	รหัสประจำตัวของผู้ใช้งานที่เข้าสู่ระบบ
-	login_time	Datetime	เวลาที่ทำการเข้าสู่ระบบ

ตาราง 3.15 แสดงรายละเอียดตารางข้อมูลการชำระเงิน (payment)

คีย์	ชื่อแอตทริบิวต์	ชนิดตัวแปร	คำอธิบาย
PK	payment_id	Varchar(50)	รหัสข้อมูลการชำระเงิน
FK	user_id	Varchar(50)	รหัสประจำตัวผู้ใช้ที่ชำระเงิน
FK	product_id	Varchar(50)	รหัสสินค้าที่ทำการซื้อ
-	type	Varchar(50)	ประเภทของการชำระเงิน
-	pay_time	Datetime	เวลาที่ทำการชำระเงิน
-	credit	Integer	จำนวนเครดิตที่ใช้

ตาราง 3.16 แสดงรายละเอียดตารางข้อมูลของสินค้า (product)

คีย์	ชื่อแอตทริบิวต์	ชนิดตัวแปร	คำอธิบาย
PK	product_id	Varchar(50)	รหัสประจำสินค้า
FK	user_id	Varchar(50)	รหัสประจำตัวผู้ใช้ที่เป็นเจ้าของสินค้า
-	product_name	Varchar(50)	ชื่อสินค้า
-	product_type	Varchar(50)	ประเภทสินค้า
-	model	Varchar(50)	โมเดลที่สินค้าใช้ในการประมวลผล
-	price	Integer	ราคาของสินค้า
-	detail	Vachar(1000)	รายละเอียดของสินค้า
-	path	Vachar(50)	เส้นทางเก็บไฟล์สินค้า
-	product_img	Vachar(50)	เส้นทางเก็บไฟล์รูปของสินค้า
-	last_update	Datetime	วันเวลาที่มีการเปลี่ยนแปลงรายละเอียดสินค้าล่าสุด

ตาราง 3.17 แสดงรายละเอียดตารางข้อมูลการสั่งงานประมวลผล (job)

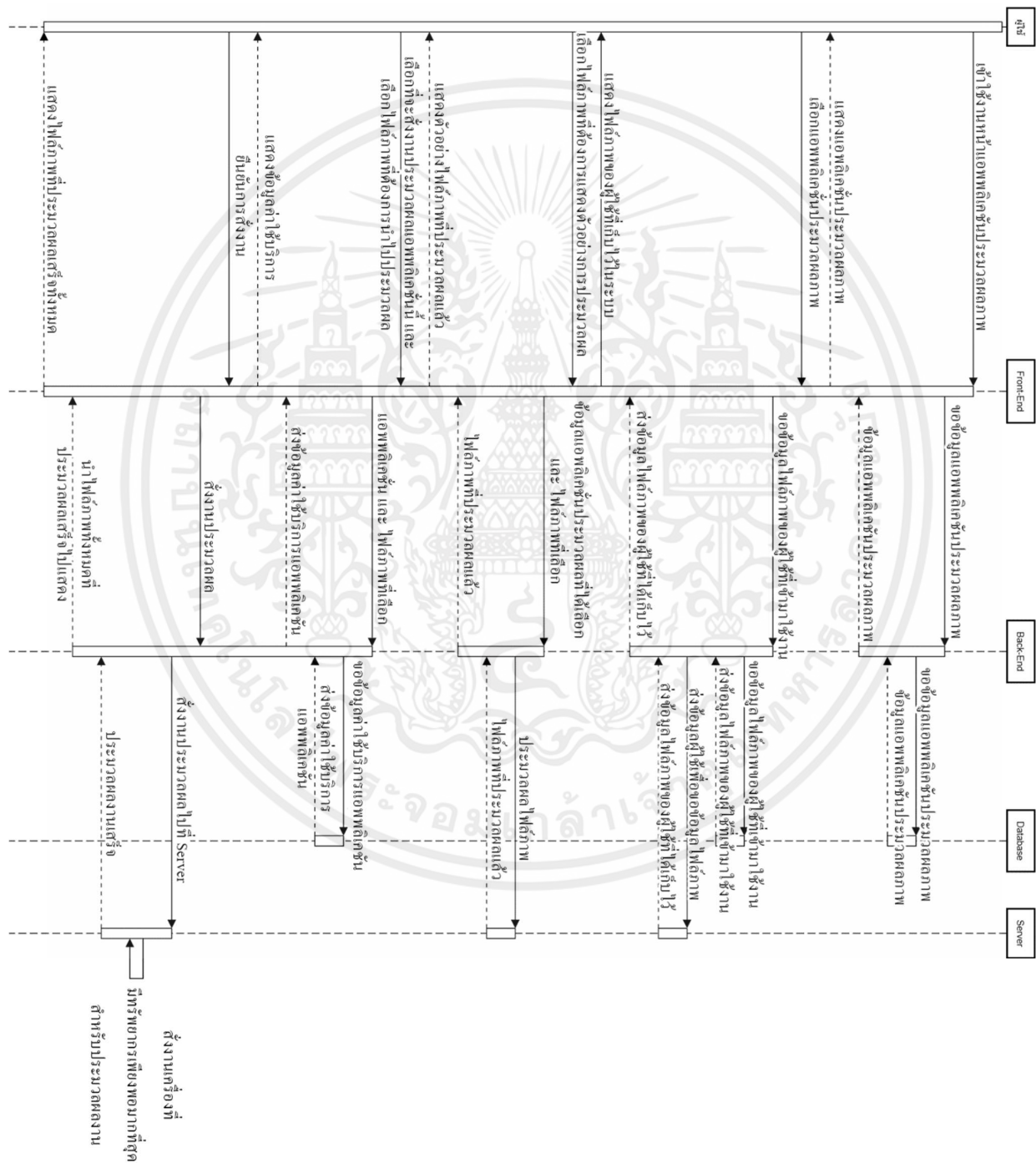
คีย์	ชื่อแอตทริบิวต์	ชนิดตัวแปร	คำอธิบาย
PK	job_id	Varchar(50)	รหัสระบุงานที่สั่ง
FK	user_id	Varchar(50)	รหัสประจำตัวของผู้ใช้งานที่สั่งงาน
FK	product_id	Varchar(50)	รหัสข้อมูล สินค้า ที่ต้องการนำมาใช้งานร่วม
FK	Payment_id	Varchar(50)	รหัสข้อมูลการชำระเงินที่เกิดขึ้นตอนสั่งงาน ประมวลผล
-	img_selected	Varchar(50)	ข้อมูลรูปที่สั่งงาน
-	path	Varchar(50)	เส้นทางของไฟล์เดอร์ที่เก็บไฟล์รูปที่จะสั่งงาน
-	num_img	Integer	จำนวนรูปภาพที่สั่งงาน
-	percent	Integer	เปอร์เซ็นต์แสดงผลการประมวลผล
-	job_status	Integer	สถานะของงานที่สั่ง
-	create_time	Datetime	เวลาที่สั่งงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 แผนภาพการทำงานแบบลำดับปฏิสัมพันธ์ (Sequence Diagram)

3.7.1 แผนภาพการทำงานในส่วนของงานประมวลผลภาพ (Web Application)

โดยรูปแบบลำดับปฏิสัมพันธ์นี้จะเริ่มตั้งแต่เหตุการณ์ที่ ผู้ใช้งานเริ่มทำการใช้งานการสั่ง งาน ประมวลผลภาพจนถึงการประมวลผลภาพเสร็จสิ้น โดยรูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์ โดยรวมในส่วนของงานประมวลผลภาพ จะแสดงดังรูปที่ 3.11

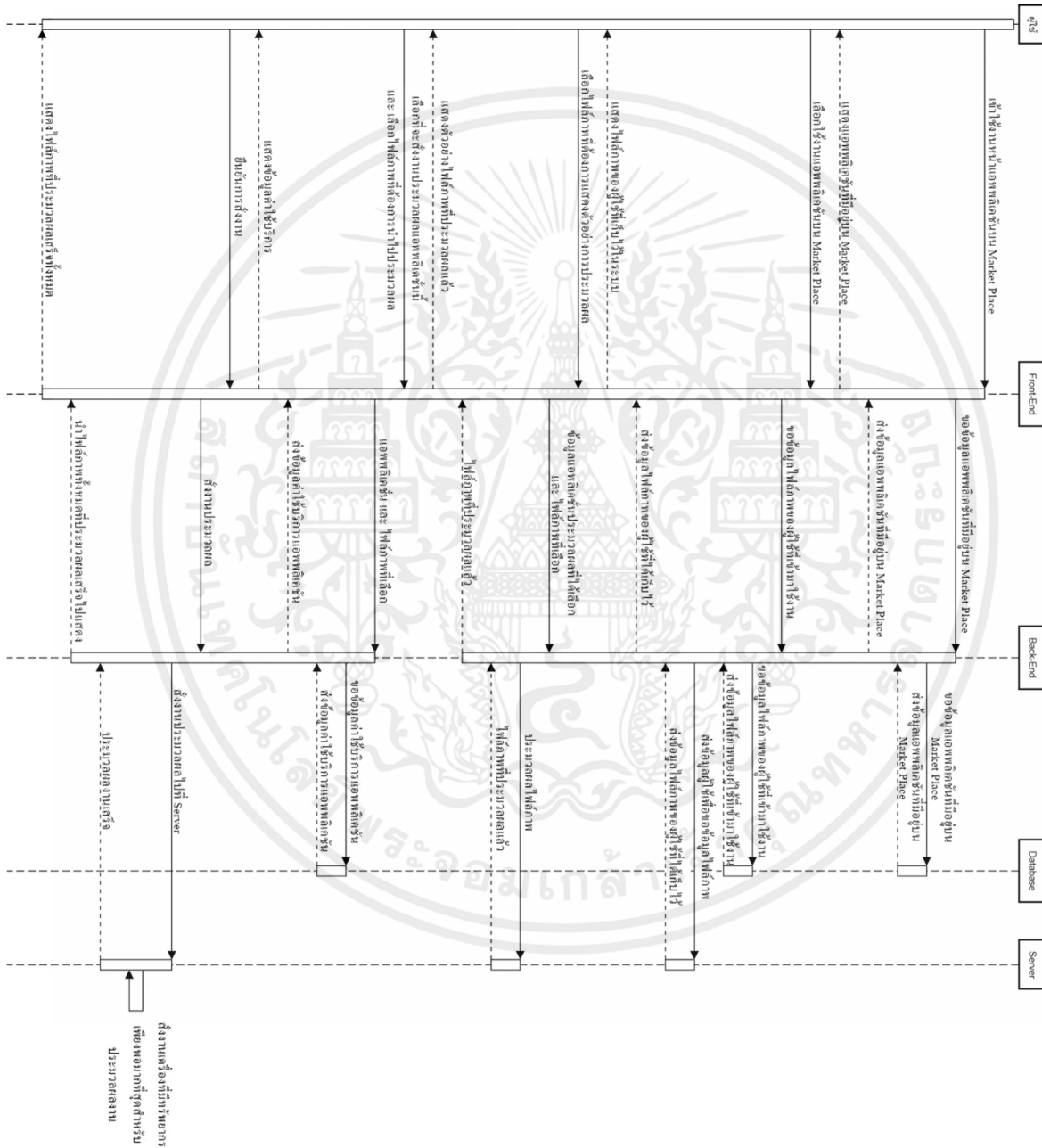


รูป 3.11 รูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของงานประมวลผลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.2 แผนภาพการทำงานในส่วนของการดำเนินงานประมวลผลภาพบน Marketplace (Web Application)

โดยรูปแบบลำดับปฏิสัมพันธ์นี้จะเริ่มตั้งแต่เหตุการณ์ที่ ผู้ใช้งานเริ่มทำการใช้งานการส่ง งานประมวลผลภาพบน Marketplace จนถึงการประมวลผลภาพเสร็จสิ้น โดยรูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการดำเนินงานประมวลผลภาพ จะแสดงดังรูปที่ 3.12

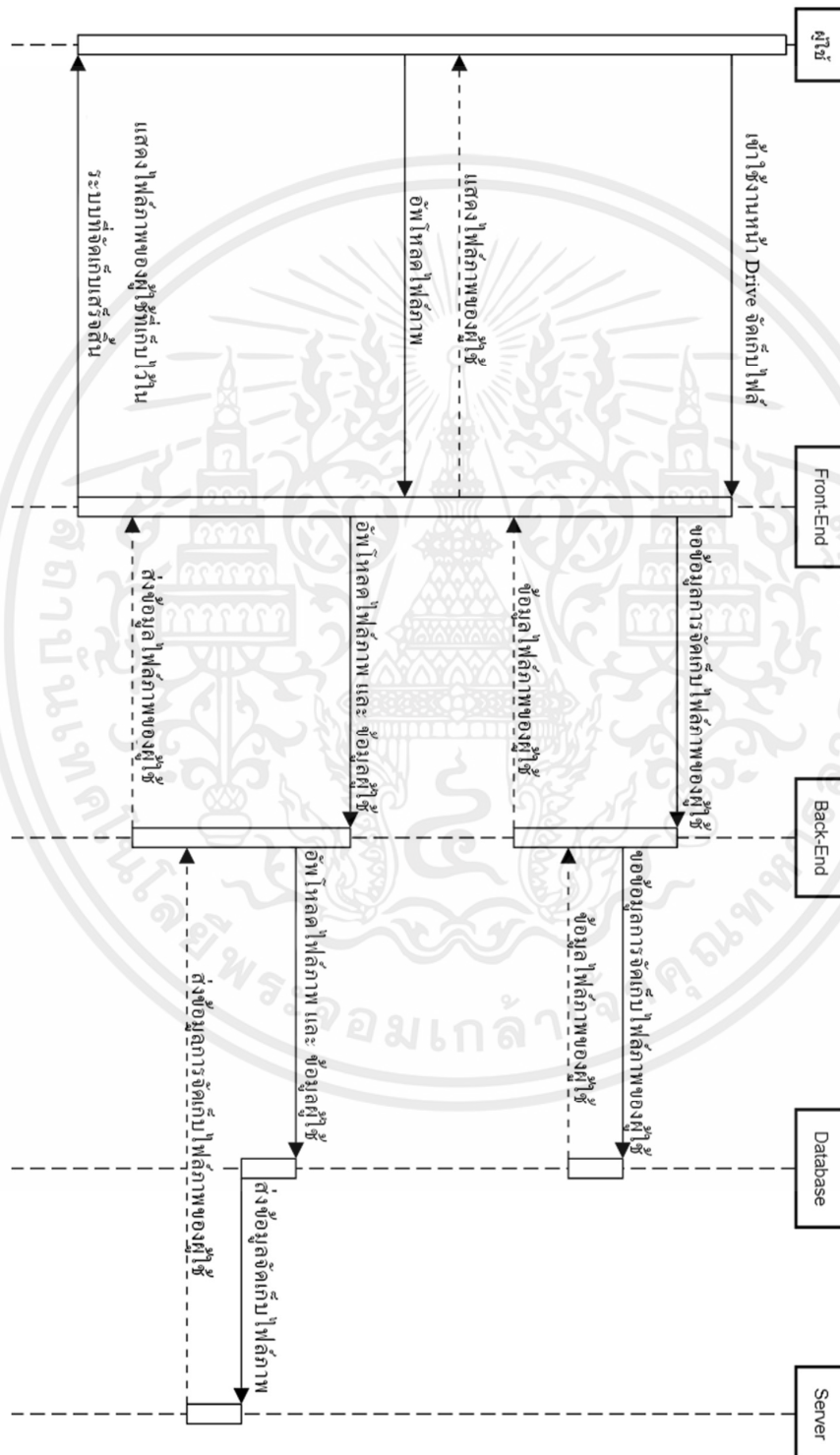


รูป 3.12 รูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการดำเนินงานประมวลผลภาพบน Marketplace

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.3 แผนภาพการทำงานในส่วนของการจัดเก็บไฟล์ภาพไว้ใน Drive (Web Application)

โดยรูปแบบลำดับปฏิสัมพันธ์นี้จะเริ่มตั้งแต่เหตุการณ์ที่ ผู้ใช้งานเริ่มทำการจัดเก็บไฟล์ภาพไว้ใน Drive จนถึงการทำงานการจัดเก็บไฟล์ภาพเสร็จสิ้น โดยรูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการจัดเก็บไฟล์ภาพไว้ใน Drive จะแสดงดังรูปที่ 3.13

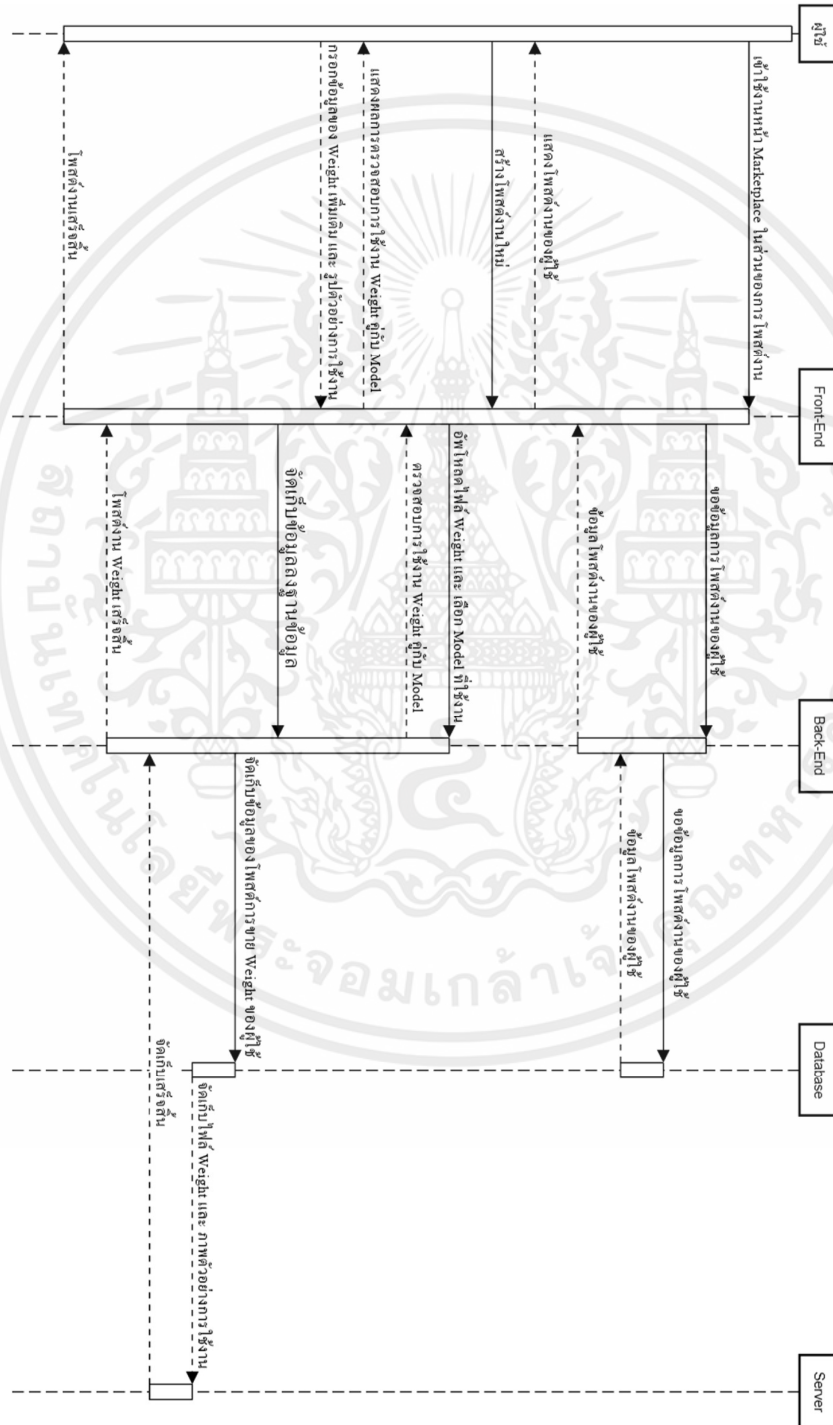


รูป 3.13 รูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการจัดเก็บไฟล์ภาพไว้ใน Drive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.4 แผนภาพการทำงานในส่วนของการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train (Web Application)

โดยรูปแบบลำดับปฏิสัมพันธ์นี้จะเริ่มตั้งแต่เหตุการณ์ที่ การใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train จนถึงการ โปสต์งานเสร็จสิ้น โดยรูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการ Weight ที่ผู้ใช้ได้ทำการ Train จะแสดงดังรูปที่ 3.14



รูป 3.14 รูปภาพแสดงการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 การออกแบบส่วนต่อประสานกับผู้ใช้ (User Interface)

3.8.1 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนลงชื่อเข้าใช้ (Login)

จะแสดงช่องข้อความสำหรับกรอก Email และ รหัสผ่านเพื่อที่จะนำไปยืนยันตัวตนสำหรับเข้าใช้งาน web application โดยการออกแบบในส่วนลงชื่อเข้าใช้ จะแสดงดังรูปที่ 3.15

รูป 3.15 การออกแบบส่วนลงชื่อเข้าใช้

3.8.2 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนลงทะเบียนสมัครสมาชิก (Register)

จะแสดงช่องข้อความสำหรับกรอก ชื่อ Username, Email, รหัสผ่าน, รหัสผ่านอีกครั้ง, ชื่อจริง และ นามสกุล เพื่อที่จะนำไปยืนยันตัวตนสำหรับเข้าใช้งาน web application โดยการออกแบบในส่วนลงทะเบียนสมัครสมาชิก จะแสดงดังรูปที่ 3.16

รูป 3.16 การออกแบบส่วนลงทะเบียนสมัครสมาชิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.3 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนหน้าข่าวสาร (Feed page)

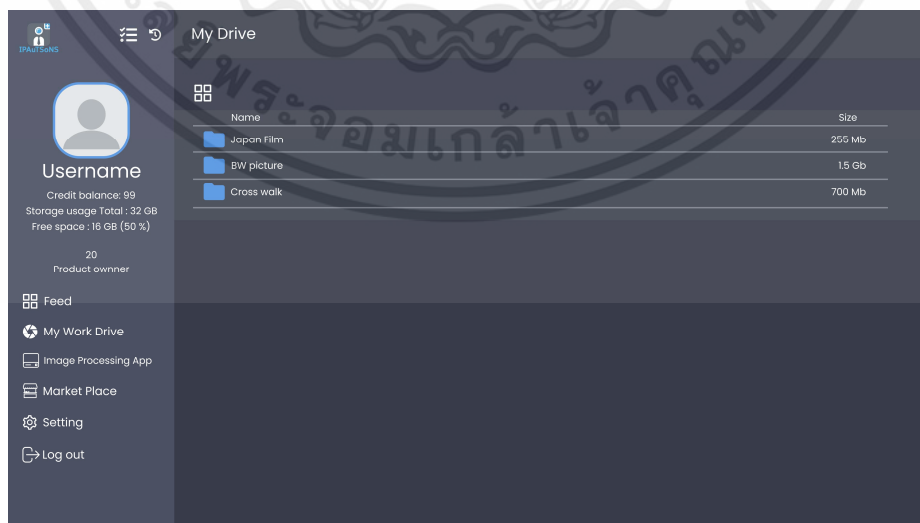
จะแสดงกล่องข้อความที่ประกอบด้วยสินค้า Weight model ที่มีการ Update ล่าสุดแต่จะแสดงเพียงจำนวนหนึ่งเท่านั้น และจะมีการแนะนำให้ไปสามารถไปดูสินค้าอื่นๆเพิ่มเติมได้ที่หน้าตลาดซื้อขาย โดยการออกแบบในส่วนหน้าข่าวสาร จะแสดงดังรูปที่ 3.17



รูป 3.17 การออกแบบส่วนหน้าข่าวสาร

3.8.4 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนหน้าพื้นที่เก็บบันทึกรูปภาพ (Drive page)

จะแสดงลิสต์ของไฟล์เตอร์ที่เก็บรูปภาพเพื่อให้ผู้ใช้เข้าถึงรูปภาพที่เก็บไว้ได้ โดยการออกแบบในส่วนพื้นที่เก็บบันทึกรูปภาพ จะแสดงดังรูปที่ 3.18



รูป 3.18 การออกแบบส่วนพื้นที่เก็บบันทึกรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.5 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนพื้นที่แสดงรูปภาพที่เก็บบันทึกไว้ (Show

Drive page)

จะแสดงรูปภาพพร้อมรายละเอียดของรูปที่อยู่ในโฟลเดอร์ที่เก็บรูปภาพนั้นๆ โดยการออกแบบในส่วนพื้นที่แสดงรูปภาพที่เก็บบันทึกไว้ จะแสดงดังรูปที่ 3.19

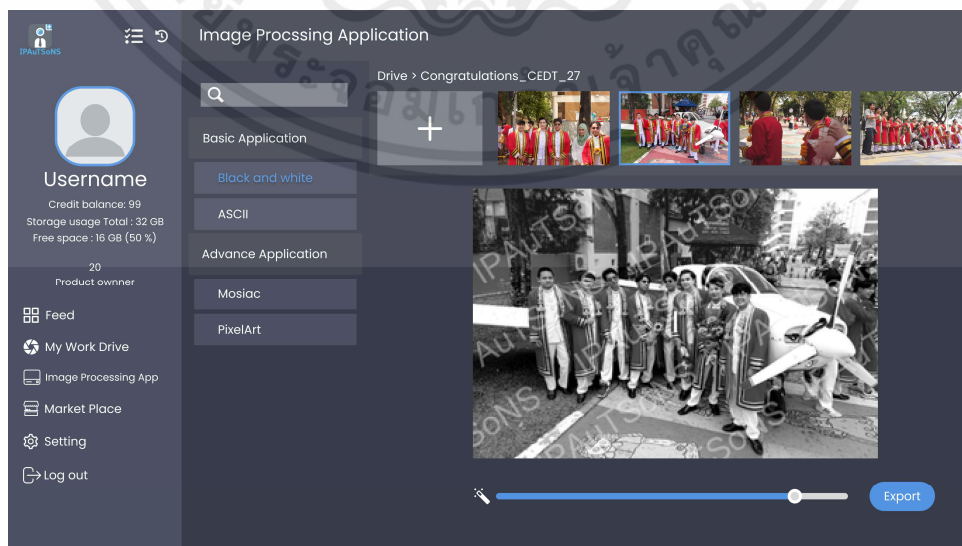


รูป 3.19 การออกแบบส่วนพื้นที่แสดงรูปภาพที่เก็บบันทึกไว้

3.8.6 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนแอปพลิเคชันประมวลผลภาพ (Image

application page)

จะแสดงแถบเลือก Application การปรับแต่งรูปที่ด้านซ้าย และแสดงผลการปรับแต่งที่ทางด้านพื้นที่ตรงกลาง โดยที่ด้านบนจะเป็นพื้นที่เลือกรูปภาพที่ต้องการ โดยการออกแบบในส่วนแอปพลิเคชันประมวลผลภาพ จะแสดงดังรูปที่ 3.20

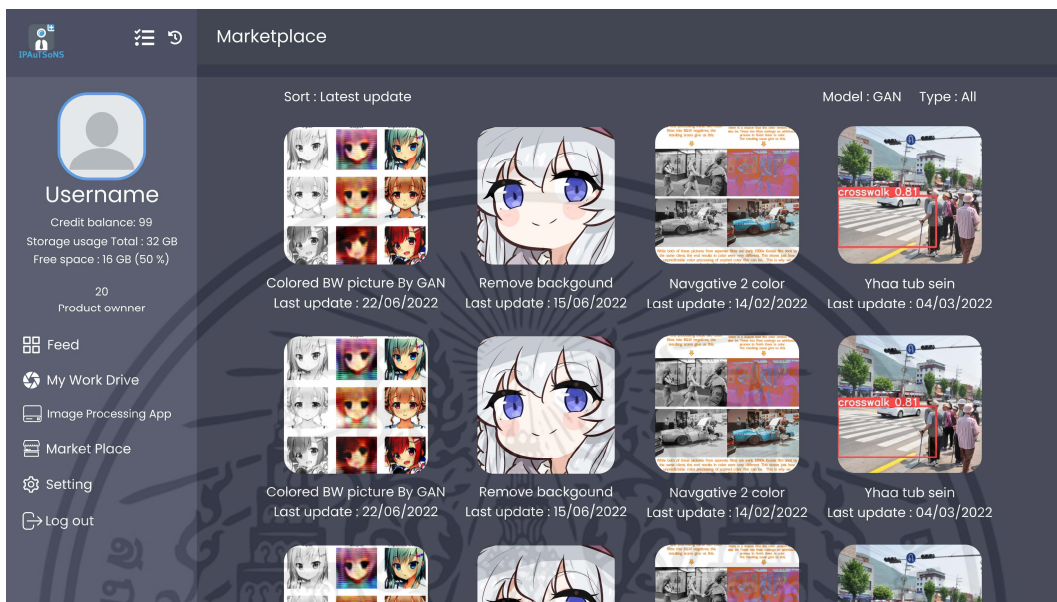


รูป 3.20 การออกแบบส่วนแอปพลิเคชันประมวลผลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.7 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนพื้นที่ซื้อขาย (Marketplace page)

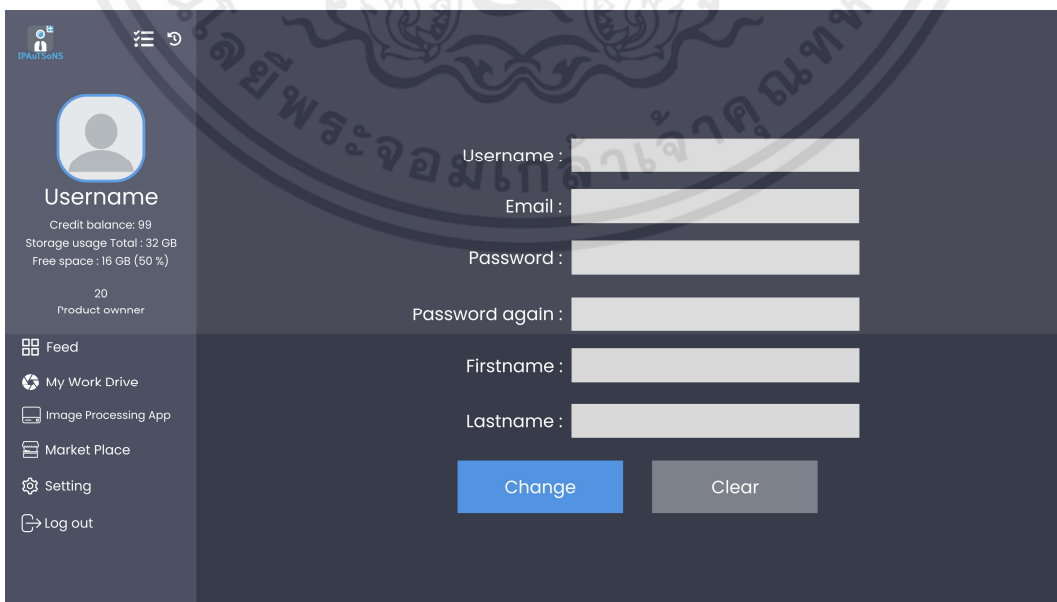
จะแสดงแถบเลือก Application การปรับแต่งรูปที่ด้านซ้าย โดยจะเป็นการกรองเลือก Application ที่สนใจ และ โดยที่ตรงกลางจะเป็นสินค้าที่นำมาขายเช่น Weight ที่ผ่านการ train มาแล้วเป็นต้น โดยการออกแบบในส่วนพื้นที่ซื้อขาย จะแสดงดังรูปที่ 3.21



รูป 3.21 การออกแบบส่วนพื้นที่ซื้อขาย

3.8.8 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนตั้งค่าบัญชี (Setting page)

จะแสดงกล่องข้อความที่มีข้อมูลของผู้ใช้อยู่ โดยสามารถแก้ไขได้รวมไปถึงรูปโปรไฟล์เองที่สามารถเปลี่ยนได้ โดยการออกแบบในส่วนตั้งค่าบัญชี จะแสดงดังรูปที่ 3.22



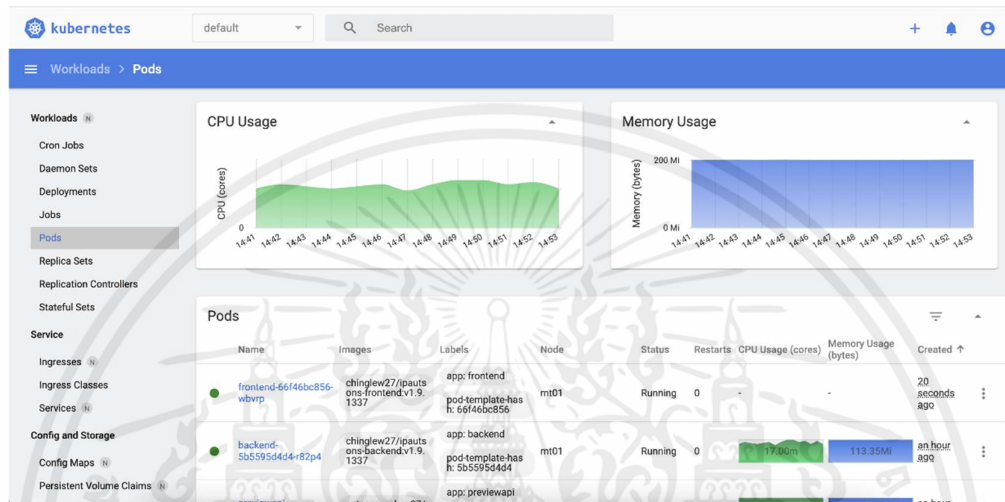
รูป 3.22 การออกแบบส่วนตั้งค่าบัญชี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.9 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนตั้งค่าและดูบันทึกการใช้ทรัพยากร

(Monitoring and Config page)

จะแสดงกราฟสถิติการใช้งานฮาร์ดแวร์ต่างๆ เช่น CPU, GPU, RAM เป็นต้น โดยสามารถปรับแต่งการสั่งงานได้ภายในหน้านี้ โดยการออกแบบในส่วนตั้งค่าและดูบันทึกการใช้ทรัพยากร จะแสดงดังรูปที่ 3.23



รูป 3.23 การออกแบบส่วนตั้งค่าและดูบันทึกการใช้ทรัพยากร

3.8.10 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนดูบันทึกการใช้งานของผู้ใช้ (User history

page)

จะแสดงบันทึกรายการการใช้งานสินค้าที่ใช้งานและเครดิตที่ใช้หรือได้รับของผู้ใช้ ซึ่งจะมีรูปแบบการเรียงลำดับและค้นหาตามวันที่ทำรายการให้ผู้ใช้เลือกเพื่อความสะดวกในการเลือกข้อมูล โดยการออกแบบในส่วนดูบันทึกการใช้งานของผู้ใช้ จะแสดงดังรูปที่ 3.24

The screenshot shows a 'Payment history' page. On the left, there is a user profile card with the following information:

- Username
- Credit balance: 99
- Storage usage Total: 32 GB
- Free space: 16 GB (50 %)
- 20 Product owner
- Feed
- My Work Drive
- Image Processing App
- Market Place
- Setting
- Log out

The main part of the page is a table with the following data:

payment_id	product_id	type	credit	pay_time
8l1pavlyfebva7l5aajyvocu		Receive	1000	2023-04-01T15:25:14.094000+07:00
slrcbniz4zqmq8mu3i24hajt		Receive	100000	2023-04-16T17:22:14.21000+07:00
o4lvq7n8rehple5b3bn3k59jg	0	Pay	100	2023-04-17T15:53:01.604000+07:00
ugxbt0n3d2hyep62ponyijoi	0	Pay	100	2023-04-17T16:52:34.670000+07:00
a5ptyqddixi3pxl95nn2v4fn	0	Pay	100	2023-04-17T15:55:33.766000+07:00
4lq1snon7esgt87kp7v73b2o	0	Pay	100	2023-04-17T15:59:59.403000+07:00
lfsadupljuc2urtalxygkele	0	Pay	100	2023-04-17T16:02:14.042000+07:00
s0ajhq4u9lftw32v5hkl32	0	Pay	100	2023-04-17T16:06:40.390000+07:00
pabqazve5xmq4ir7a29tsjh	0	Pay	100	2023-04-17T16:08:17.846000+07:00
aril6cojpa6zy8lpr5xwtigjm	0	Pay	100	2023-04-17T16:10:44.765000+07:00

รูป 3.24 การออกแบบส่วนดูบันทึกการใช้งานของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.11 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนดูบันทึกการขายของสินค้า (Product history page)

จะแสดงบันทึกการขายที่สินค้าถูกนำไปสั่งงานประมวลผลโดยผู้ใช้อื่นๆ ซึ่งจะมีรูปแบบการเรียงลำดับและค้นหาตามวันที่ทำการขายให้ผู้ใช้เลือกเพื่อความสะดวกในการเลือกดูข้อมูล โดยการออกแบบในส่วนดูบันทึกการขายของสินค้า จะแสดงดังรูปที่ 3.25

payment_id	product_id	type	credit	pay_time
koucpn6mdv8clddwcb6p2aya	0	Pay	100	2023-04-17T16:23:08.556000+07:00
whs4t27apj8lqcdockerm	0	Pay	100	2023-04-17T16:20:53.593000+07:00
430dahnwmpggsydm02253	0	Pay	100	2023-04-17T16:18:2581000+07:00
onl8cojn6s2y8p58id1gdm	0	Pay	100	2023-04-17T16:10:44.750000+07:00
paodqazvzwmq4nkuasgn	0	Pay	100	2023-04-17T16:08:17.940000+07:00
s00jng4ul8fth2vz6n332	0	Pay	100	2023-04-17T16:06:40.390000+07:00
ifsodupjuc2urtawlyklee	0	Pay	100	2023-04-17T16:02:14.042000+07:00
4iqjncn7aag8t87kpv7zb2o	0	Pay	100	2023-04-17T15:59:59.403000+07:00
alptyqddiw3xw85nk244n	0	Pay	100	2023-04-17T15:55:33.766000+07:00
ugqbt9n32hrypb7peryyca	0	Pay	100	2023-04-17T15:53:34.570000+07:00

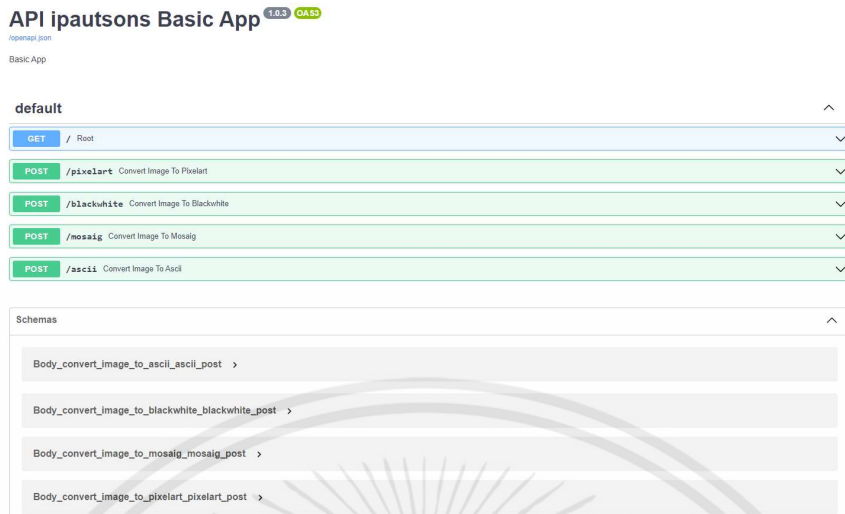
รูป 3.25 การออกแบบส่วนดูบันทึกการขายของสินค้า

3.8.12 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของการแสดงผลตัวอย่างงานประมวลผลแอปพลิเคชันเบื้องต้น (API Previews Basic Application)

API สำหรับการประมวลผลแสดงตัวอย่างงานประมวลผลเบื้องต้น โดยจะมีแอปพลิเคชันดังต่อไปนี้ ASCII, BlackWhite, Pixelart, Mosaic ให้ผู้ส่งทำการส่งไฟล์รูปภาพเข้ามาในระบบ และจะส่งกลับคืนไปเป็นรูปภาพตามงานประมวลผลที่ได้เลือก เป็นการเปิดรับคำสั่งจากฝั่งของ Back-End ระบบเพื่อแสดงผลภาพจากงานประมวลผลที่จัดทำไว้โดยจะมีการเตรียม Model หรือ Library ต่าง ๆ เปิดใช้งานพร้อมรองรับคำสั่งประมวลผลโดยจะมี API Routing ดังนี้

1. /ascii ประมวลผลภาพเป็นรูปแบบ ASCII
2. /blackwhite ประมวลผลภาพเป็นรูปแบบ BlackWhite หรือ ขาวดำ
3. /pixelart ประมวลผลภาพเป็นรูปแบบ PixelArt โดยสามารถปรับแต่งขนาดของ Pixel ในภาพได้
4. /mosiag ประมวลผลภาพเป็นรูปแบบ Mosaic

โดย API Documents จะแสดงดังรูปที่ 3.26



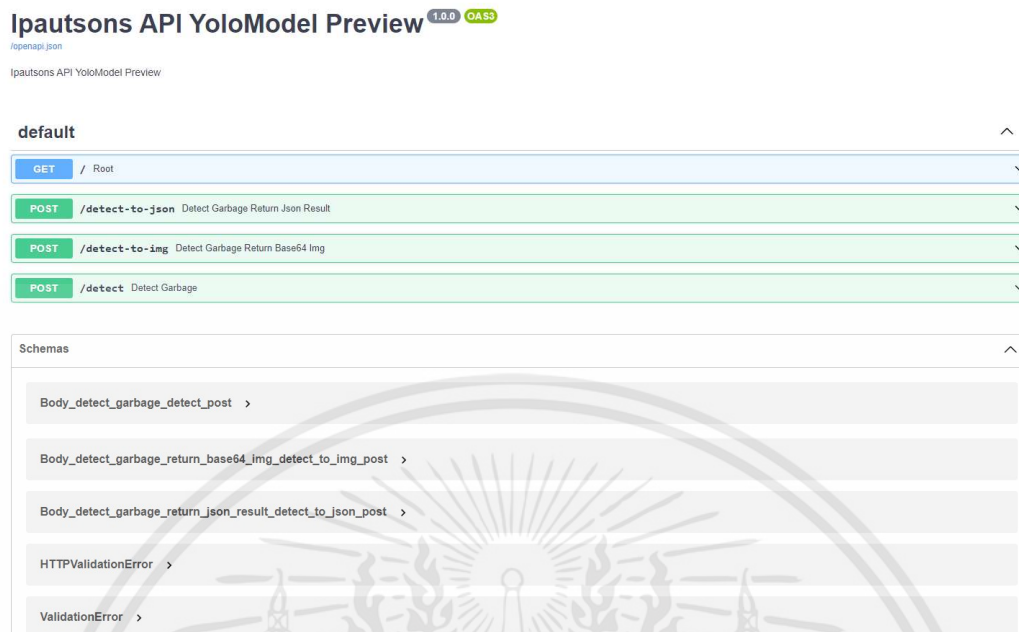
รูป 3.26 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของการแสดงผลตัวอย่างงานประมวลผล Basic App

3.8.13 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของการแสดงผลตัวอย่างงานประมวลผล และ งานประมวลผล ด้วย Yolo Model (API Yolo Model Preview)

API ในการประมวลผลตรวจจับวัตถุในภาพ โดยใช้ YoloV5 Model ที่จะมี Weight พื้นฐานที่รองรับวัตถุ 80 ชนิด และสามารถให้ผู้ใช้ทำการอัปโหลด และ เลือก Weight ที่มีอยู่ในระบบ เพื่อนำมาประมวลผลแทน Weight พื้นฐานได้ โดยสามารถตอบกลับเป็นในรูปแบบ Json และ รูปภาพ เป็นการเปิดรับคำสั่งจากฝั่งของ Back-End ระบบเพื่อแสดงผลภาพจากงานประมวลผลที่จัดทำไว้โดยจะมีการเตรียม Model หรือ Library ต่าง ๆ เปิดใช้งานพร้อมรองรับคำสั่งประมวลผลโดยจะมี API Routing

1. /detect เป็นการ detect ภาพด้วย Weight Model ตัวเริ่มต้นที่ตั้งค่าไว้
2. /detect-to-json เป็นการ detect ภาพจาก Weight Model ที่เลือก และ คืนค่ามาด้วยจำนวนของวัตถุ และ ชื่อวัตถุ ที่จับได้ในรูปแบบ Json
3. /detect-to-image เป็นการ detect ภาพจาก Weight Model ที่เลือก และ ส่งรูปภาพกลับมาพร้อมกรอบวัตถุที่จับได้พร้อมด้วยความแม่นยำของวัตถุนั้น ๆ

โดย API Documents จะแสดงดังรูปที่ 3.27



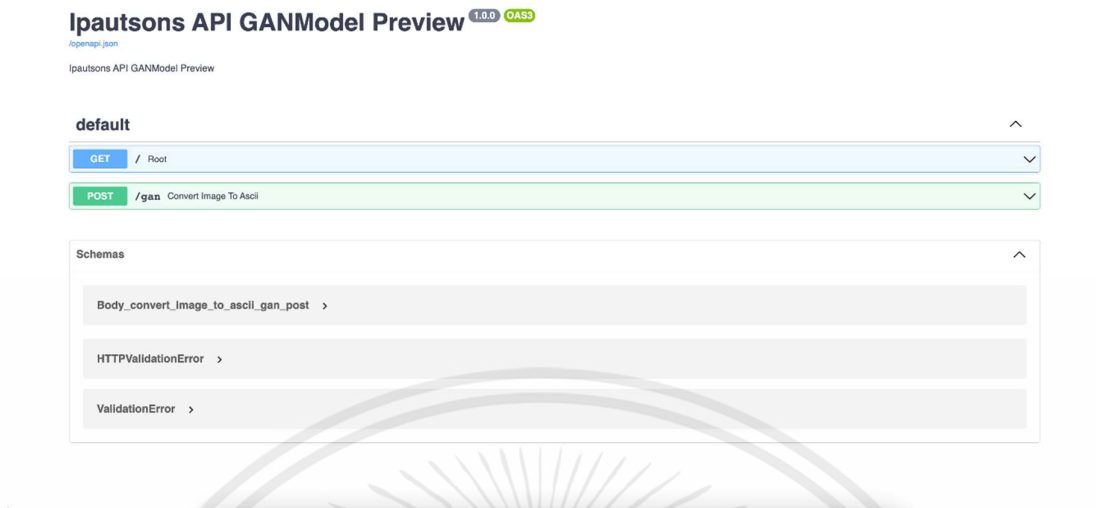
รูป 3.27 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของการแสดงผลตัวอย่างงานประมวลผล Yolo Model

3.8.14 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของการแสดงผลตัวอย่างงานประมวลผล และ งานประมวลผล ด้วย GAN Model (API GAN Model Preview)

API ในการประมวลผลตรวจจับวัตถุในภาพโดยใช้ GAN Model ที่จะมี Weight พื้นฐานที่เปลี่ยนชนิดสี หรือ ลายเส้นของภาพ และสามารถให้ผู้ใช้ทำการอัปโหลด และ เลือก Weight ที่มีอยู่ในระบบเพื่อนำมาประมวลผลแทน Weight พื้นฐานได้ โดยสามารถตอบกลับเป็นในรูปแบบภาพ เป็นการเปิดรับคำสั่งจากฝั่งของ Back-End ระบบเพื่อแสดงผลภาพจากงานประมวลผลที่จัดทำไว้โดยจะมีการเตรียม Model หรือ Library ต่าง ๆ เปิดใช้งานพร้อมรองรับคำสั่งประมวลผลเปิดใช้งานพร้อมรองรับคำสั่งประมวลผลโดยจะมี API Routing

1. /gan เป็นการวาดภาพขึ้นมาใหม่ด้วยภาพเดิม พร้อม Weight ที่ได้ทำการเลือกไว้ หรือ ตัวตั้งต้นที่ได้ตั้งค่าไว้

โดย API Documents จะแสดงดังรูปที่ 3.28



รูป 3.28 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของการแสดงผลตัวอย่างงานประมวลผล GAN Model

3.8.15 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของการบีบอัดไฟล์ของผู้ใช้ (API Zip file)

API ในการบีบอัดไฟล์ในโพลเดอร์ของผู้ใช้ที่ให้บริการอยู่ในระบบผ่าน Persistent Volume โดยจะให้ผู้ใช้ทำการเลือกโพลเดอร์ที่จะนำเข้ามา และ ส่งคืนกลับไปเป็นไฟล์ Zip เพื่อให้ผู้ใช้สามารถนำไฟล์ในโพลเดอร์ไปใช้งานในงานอื่น ๆ ต่อได้ โดยจะมี API Routing

1. /downloadzip เป็นการใส่ค่าตำแหน่งของโพลเดอร์ที่ต้องการจะทำการบีบอัดไฟล์ในรูปแบบ Zip เพื่อประมวลผลบีบอัดส่งกลับมาเป็นไฟล์ Zip

จะแสดงดังรูปที่ 3.29



รูป 3.29 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของการบีบอัดไฟล์ Zip API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง / ความก้าวหน้า

4.1 ส่วนของหน้าลงทะเบียนสมัครเข้าใช้งาน

4.1.1 การทดลองระบบการสมัครเข้าใช้งานของผู้ใช้งานระดับ Client

4.1.1.1 จุดประสงค์การทดลอง

เพื่อทดสอบการสมัครเข้าใช้งานจากส่วนหน้า Register page หรือ หน้าลงทะเบียนสมัครเข้าใช้งาน ว่าสามารถทำงานส่งข้อมูลผู้ใช้ที่กรอกเข้ามา แล้วส่งต่อไปยังส่วน Backend เพื่อทำการตรวจสอบข้อมูลผู้ใช้ และสถานะการมีอยู่ของผู้ใช้ว่าเคยสมัครมาก่อนหน้านี้หรือไม่ รวมไปถึงการตรวจสอบรหัสผ่านเบื้องต้น หากไม่มีข้อผิดพลาดใดๆ ให้ทำการเพิ่มผู้ใช้ไปยังฐานข้อมูล

4.1.1.2 วิธีการทดลอง

ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยการนำข้อมูลผู้ใช้งานหนึ่งไปทดลองกรอกลงในหน้าลงทะเบียนสมัครเข้าใช้งาน และ กดสมัครกรใช้งาน หากการทดลองสำเร็จ ข้อมูลจากหน้า Web Application จะถูกส่งไปตรวจสอบที่ส่วน Backend ได้ และ ถ้าข้อมูลถูกต้องสามารถลงทะเบียนสมัครเข้าใช้งานได้ และมีข้อมูลของผู้ใช้ปรากฏขึ้นในฐานข้อมูล แต่ถ้าหากข้อมูลผู้ใช้ไม่ถูกต้อง เช่น รหัสผ่านที่กรอกมาเพื่อยืนยันอีกครั้งไม่เหมือนกัน หรือ หากชื่อผู้ใช้หรืออีเมลที่สมัครเข้ามา มีอยู่แล้วในระบบ ก็จะแสดงข้อความตอบกลับว่าเกิดข้อผิดพลาดขึ้น

4.1.1.3 ผลการทดลอง

ส่วนของการลงทะเบียนสมัครเข้าใช้งาน สามารถส่งข้อมูลจากหน้า Web Application ไปยังส่วน Backend ได้ถูกต้องและครบถ้วน และส่วน Backend สามารถตรวจสอบข้อมูลที่ทำการส่งเข้ามาได้และเพิ่มข้อมูลผู้ใช้เข้าในฐานข้อมูลได้หากข้อมูลที่กรอกเข้ามามีความสมบูรณ์ และ สามารถส่งข้อความแจ้งเตือนกลับไปในกรณีที่มีข้อมูลผิดพลาด โดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.1 - 4.3

Email :
63015121@kmitl.ac.th

Password :

Password confirm :

Firstname :
Pasin

Lastname :
Chantharathan

```
System check identified no issues (0 silenced).
December 08, 2022 - 14:42:24
Django version 4.0.6, using settings 'IPAutSoNsAPI.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
{'user_id': 'lcpcs12cbgd', 'email': '63015121@kmitl.ac.th', 'password': '123456789', 'is_vip': 'false', 'first_name': 'Pasin', 'last_name': 'Chantharathan'}
[08/Dec/2022 14:43:16] "POST /api/register HTTP/1.1" 200 120
```

รูป 4.1 ข้อมูลที่ส่งจาก Web application (บน) เทียบกับข้อมูลที่ส่วน Backend (ล่าง) ได้รับ

The screenshot shows the MongoDB Compass interface for the 'ipautsons.api_user' collection. The left sidebar lists various databases and collections, with 'api_user' selected. The main area displays two documents from the collection:

```
{
  "_id": ObjectId("633e73a16b6268563235a9f3"),
  "last_login": null,
  "user_id": "i6r6mxxzco",
  "password": "pbkdf2_sha256$32000$YAx8sCTPZYcwRYtLzdWxCR$Gxw39nf2ypQtvEL6IEa2pAUprj_",
  "email": "pasin98@gmail.com",
  "is_vip": false,
  "first_name": "Pasin",
  "last_name": "Chantharathan"
}
```

```
{
  "_id": ObjectId("63919594c3c8722b080801dc"),
  "last_login": null,
  "user_id": "lcpcs12cbgd",
  "password": "pbkdf2_sha256$32000$0ZXXKqP7FLxLgC4CTty34$V/NvMPuqTHCL43NsNr8B8Vy9q9_",
  "email": "63015121@kmitl.ac.th",
  "is_vip": false,
  "first_name": "Pasin",
  "last_name": "Chantharathan"
}
```

รูป 4.2 ข้อมูลผู้ใช้ทำการสมัครสำเร็จถูกเพิ่มในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 4.3 ข้อความที่แสดงหากอีเมลที่ใช้สมัครนั้นมียู่แล้วในฐานข้อมูล

4.2 ส่วนของหน้าลงทะเบียนเข้าสู่ระบบใช้งาน

4.2.1 การทดลองระบบลงทะเบียนเข้าสู่ระบบเพื่อผู้ใช้ระดับ Client ใช้งาน Web application

4.2.1.1 จุดประสงค์การทดลอง

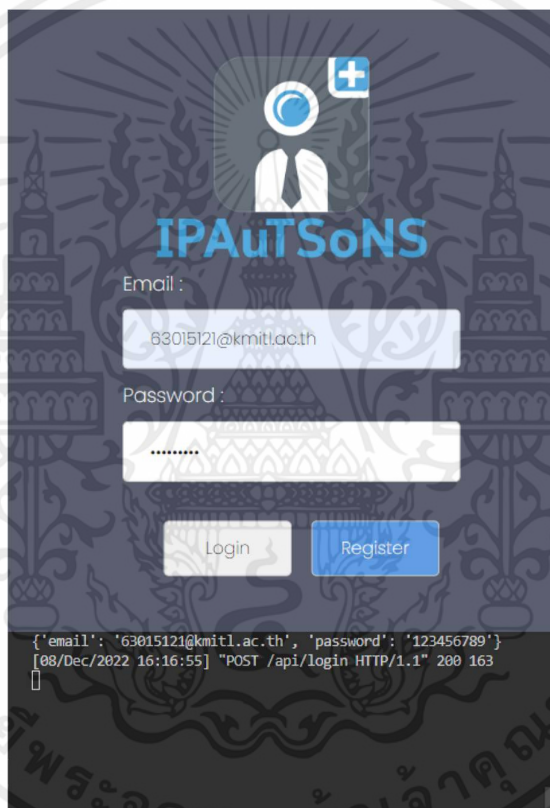
เพื่อทดสอบการทำงานของ Web application ส่วนหน้าลงทะเบียนเข้าสู่ระบบใช้งานว่าสามารถทำงานถูกต้อง สามารถส่งข้อมูลจากหน้าลงทะเบียนเข้าสู่ระบบไปยังส่วน Backend ได้อย่างครบถ้วน และนำข้อมูลที่ไปตรวจสอบเทียบกับข้อมูลผู้ใช้งานในฐานข้อมูลเพื่อทำการยืนยันตัวตนผู้ใช้ได้ และสามารถแจ้งเตือนให้ผู้ใช้ได้หากมีข้อมูลส่วนใดผิดพลาด และสามารถส่ง Cookie กลับมายัง Web application เพื่อใช้ในการยืนยันตัวตนได้

4.2.1.2 วิธีการทดลอง

ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยการนำข้อมูลผู้ใช้ที่มีและไม่มีในฐานข้อมูลจำนวนหนึ่งไปทดลองกรอกลงในหน้าลงทะเบียนเข้าสู่ระบบใช้งาน และ เข้าสู่ระบบ หากการทดลองสำเร็จ ข้อมูลผู้ใช้ที่ทำการลงทะเบียนเข้าสู่ระบบที่หน้า Web Application จะถูกส่งไปตรวจสอบที่ส่วน Backend ได้ และ ถ้าข้อมูลถูกต้องผู้ใช้จะเข้าใช้งาน Web applications ได้ แต่ถ้าหากข้อมูลผู้ใช้ไม่ถูกต้อง เช่น รหัสผ่านที่กรอกมาไม่เหมือนกับในฐานข้อมูล หรือ หากอีเมลที่กรอกเข้ามาเข้าสู่ระบบไม่มีภายในฐานข้อมูล ก็จะแสดงข้อความตอบกลับว่าเกิดข้อผิดพลาดขึ้น

4.2.1.3 ผลการทดลอง

ส่วนของการลงทะเบียนเข้าสู่ระบบสามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Back end ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถตรวจสอบข้อมูลที่ได้ทำการส่งเข้ามาและยืนยันตัวตนโดยเปรียบเทียบกับข้อมูลของผู้ใช้ในฐานข้อมูลได้หาข้อมูลของผู้ใช้ที่กรอกเข้ามานั้นมีความถูกต้องจะส่ง Cookie กลับไปยังเว็บแอปพลิเคชันเพื่อใช้ยืนยันตัวตนผู้ใช้ และ ในกรณีที่เกิดข้อผิดพลาดขึ้นเช่น ไม่พบ Email ของผู้ใช้ในระบบ หรือ รหัสผ่านผิดพลาดก็จะทำการส่งข้อความแจ้งเตือนกลับไปแสดงให้ผู้ใช้ทราบ โดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.4 - 4.6



รูป 4.4 ข้อมูลที่ส่งจาก Web application (บน) เทียบกับข้อมูลที่ส่วน Backend (ล่าง) ได้รับ

4.3 ส่วนของหน้าแก้ไขข้อมูลผู้ใช้งาน

4.3.1 การทดลองระบบแก้ไขข้อมูลผู้ใช้ระดับ Client

4.3.1.1 จุดประสงค์การทดลอง

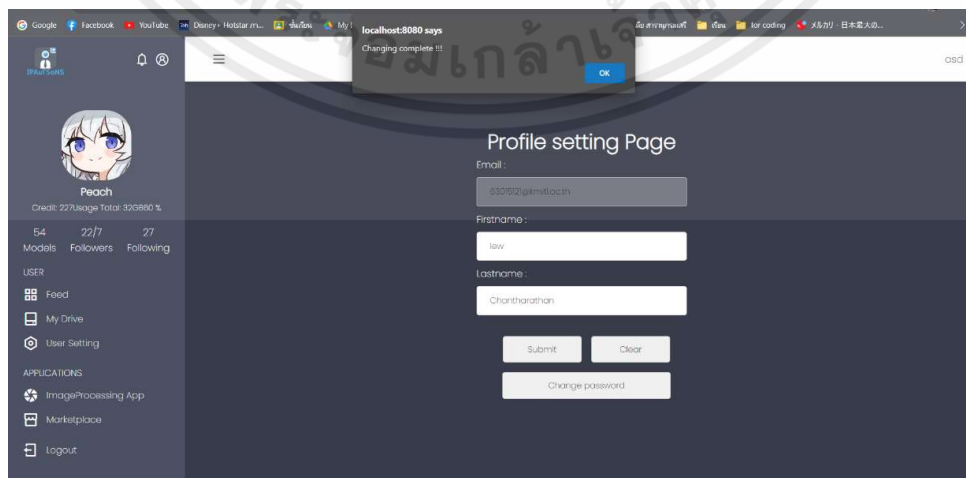
เพื่อทดสอบการทำงานของ Web applications ส่วนแก้ไขข้อมูลผู้ใช้งานว่าสามารถทำงานถูกต้อง สามารถส่งข้อมูลจากหน้าแก้ไขข้อมูลผู้ใช้งานและ Cookie ยืนยันตัวตนไปยังส่วน Backend ได้อย่างครบถ้วนและนำข้อมูลที่ทำกรแก้ไขไปเปลี่ยนแปลงข้อมูลในฐานข้อมูลได้ โดยสามารถแจ้งเตือนผู้ใช้งานได้หากมีข้อผิดพลาดเกิดขึ้น

4.3.1.2 วิธีการทดลอง

ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยการนำชุดข้อมูลที่คล้ายกับข้อมูลผู้ใช้งานไปทดลองกรอกลงในหน้าแก้ไขข้อมูลผู้ใช้งานและทำการกดเปลี่ยนแปลง หากสำเร็จข้อมูลที่ผู้ใช้งานกรอกเข้าไปเพื่อทำการเปลี่ยนแปลงและ Cookie ยืนยันตัวตนจะถูกส่งไปส่วนตรวจสอบที่ Backend ซึ่งส่วน Backend จะทำการยืนยันตัวตนของผู้ใช้หากสำเร็จจะทำการเปลี่ยนแปลงข้อมูลของผู้ใช้ในฐานข้อมูล แต่ถ้าหากยืนยันตัวตนไม่สำเร็จหรือเกิดข้อผิดพลาดขึ้นก็จะแสดงข้อความตอบกลับ และทดลองปิดการทำงานของฐานข้อมูลเพื่อทดสอบการแจ้งเตือนหากมีข้อผิดพลาดเกิดขึ้น

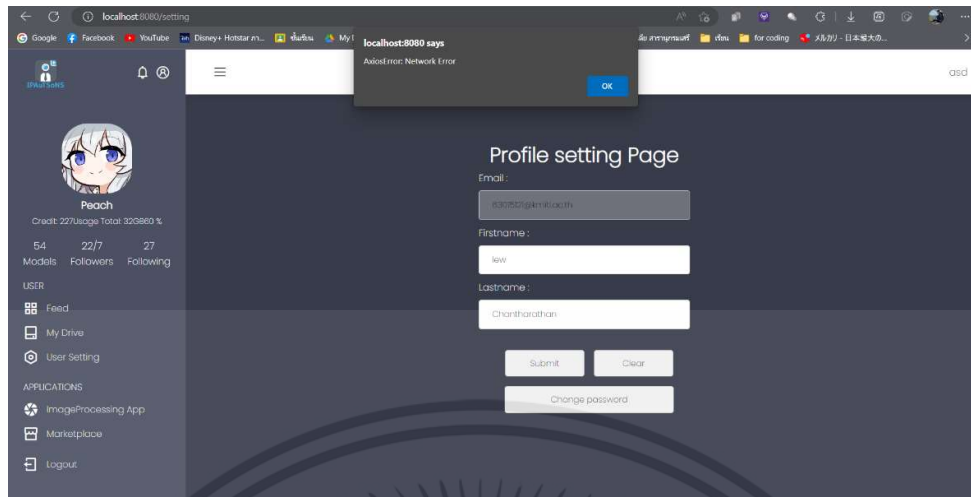
4.3.1.3 ผลการทดลอง

ส่วนของการแก้ไขข้อมูลผู้ใช้งานสามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Backend ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถตรวจสอบข้อมูลที่ได้ทำการส่งเข้ามา และยืนยันตัวตนโดยเปรียบเทียบ Cookie กับข้อมูลของผู้ใช้ในฐานข้อมูล และ เมื่อทำการยืนยันตัวตนเสร็จก็ทำการเปลี่ยนแปลงข้อมูลของผู้ใช้ในฐานข้อมูล และ เมื่อมีข้อผิดพลาดเกิดขึ้นก็จะทำการแจ้งเตือนให้ผู้ใช้ทราบโดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.7 - 4.8



รูป 4.7 ข้อความที่แสดงหากสามารถเปลี่ยนข้อมูลผู้ใช้งานได้สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.8 ข้อความที่แสดงหากเกิดข้อผิดพลาดขึ้นขณะเปลี่ยนแปลงข้อมูลผู้ใช้

4.3.2 การทดลองระบบแก้ไขรหัสผ่านผู้ใช้ระดับ Client

4.3.2.1 จุดประสงค์การทดลอง

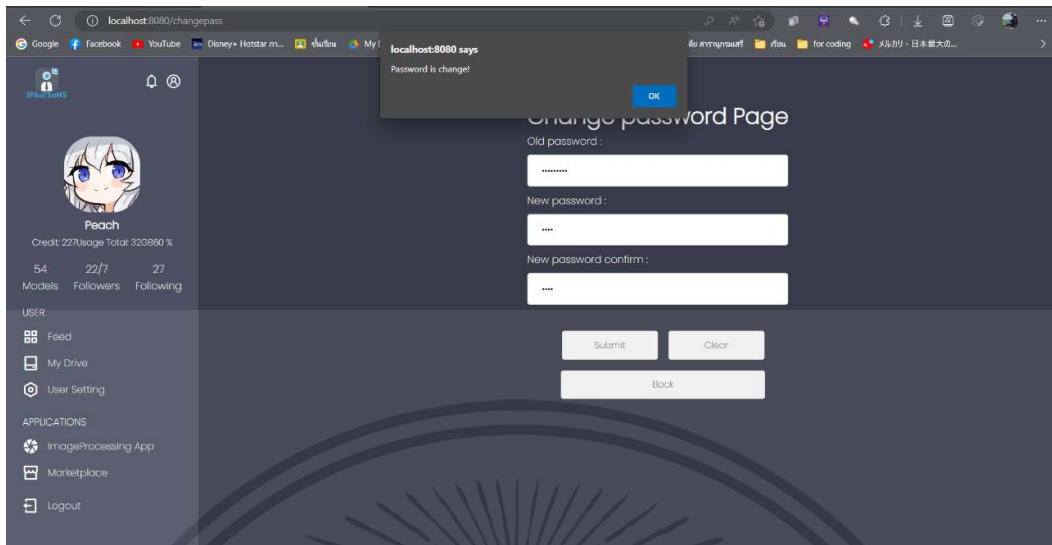
เพื่อทดสอบการทำงานของ Web applications ส่วนแก้ไขรหัสผ่านผู้ใช้ว่าสามารถทำงาน ถูกต้องสามารถส่งข้อมูลจากหน้าแก้ไขรหัสผ่านผู้ใช้และ Cookie ยืนยันตัวตนไปยังส่วน Backend ได้ อย่างครบถ้วนและนำรหัสผ่านที่ทำการแก้ไขไปเปลี่ยนแปลงข้อมูลในฐานข้อมูลได้ โดยสามารถแจ้ง เตือนผู้ใช้ได้หากมีข้อผิดพลาดเกิดขึ้น

4.3.2.2 วิธีการทดลอง

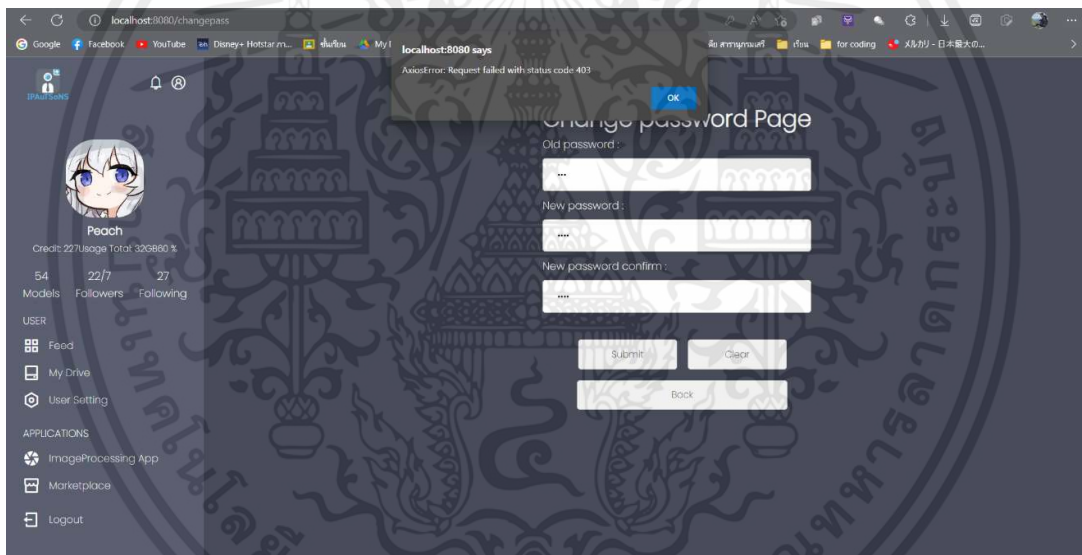
ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยการนำชุดข้อมูลรหัสผ่านผู้ใช้นำไปทดลอง กรอกลงในหน้าแก้ไขรหัสผ่านผู้ใช้และทำการกดเปลี่ยนแปลงแปลง หากสำเร็จรหัสผ่านที่ผู้ใช้กรอกเข้าไป เพื่อทำการเปลี่ยนแปลง และ Cookie ยืนยันตัวตนจะถูกส่งไปส่วนตรวจสอบที่ Backend ซึ่งส่วน Backend จะทำการยืนยันตัวตนของผู้ใช้หากสำเร็จจะทำการเปลี่ยนแปลงรหัสผ่านของผู้ใช้ในฐานข้อมูล แต่ถ้าหากยืนยันตัวตนไม่สำเร็จหรือเกิดข้อผิดพลาดขึ้นก็จะแสดงข้อความตอบกลับ และทดลองปิดการ ทำงานของฐานข้อมูลเพื่อทดสอบการแจ้งเตือนหากมีข้อผิดพลาดเกิดขึ้น

4.3.2.3 ผลการทดลอง

ส่วนของการแก้ไขรหัสผ่านผู้ใช้สามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Backend ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถตรวจสอบรหัสผ่านที่ได้ทำการส่งเข้ามา และ ยืนยันตัวตนโดยเปรียบเทียบ Cookie กับรหัสผ่านของผู้ใช้ในฐานข้อมูล และ เมื่อทำการยืนยันตัวตน เสร็จก็ทำการเปลี่ยนแปลงรหัสผ่านของผู้ใช้ในฐานข้อมูล และ เมื่อมีข้อผิดพลาดเกิดขึ้นก็จะทำการแจ้ง เตือนให้ผู้ใช้ทราบโดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.9 - 4.10



รูป 4.9 ข้อความที่แสดงหากสามารถเปลี่ยนข้อมูลผู้ใช้ได้สำเร็จ



รูป 4.10 ข้อความที่แสดงหากเกิดข้อผิดพลาดขึ้นขณะเปลี่ยนแปลงข้อมูลผู้ใช้

4.4 ส่วนของหน้าจัดการไฟล์

4.4.1 การทดลองระบบจัดการ Folder สำหรับเก็บไฟล์รูปภาพของผู้ใช้

4.4.1.1 จุดประสงค์การทดลอง

เพื่อทดสอบการทำงานของ Web applications หน้าจัดการไฟล์ในส่วนจัดการ Folder สำหรับเก็บไฟล์รูปภาพของผู้ใช้ว่าสามารถทำงานถูกต้อง สามารถเพิ่ม Folder และ ลบ Folder ไปได้โดยมีการยืนยันตัวตนจาก Cookie ที่เก็บเอาไว้ โดยจะเพิ่ม-ลบข้อมูลของ Folder ที่ฐานข้อมูลด้วย และ มีการตรวจเช็คชื่อ Folder โดยถ้าหากมีข้อผิดพลาดก็จะแสดงข้อความแจ้งเตือนข้อผิดพลาดขึ้นมา

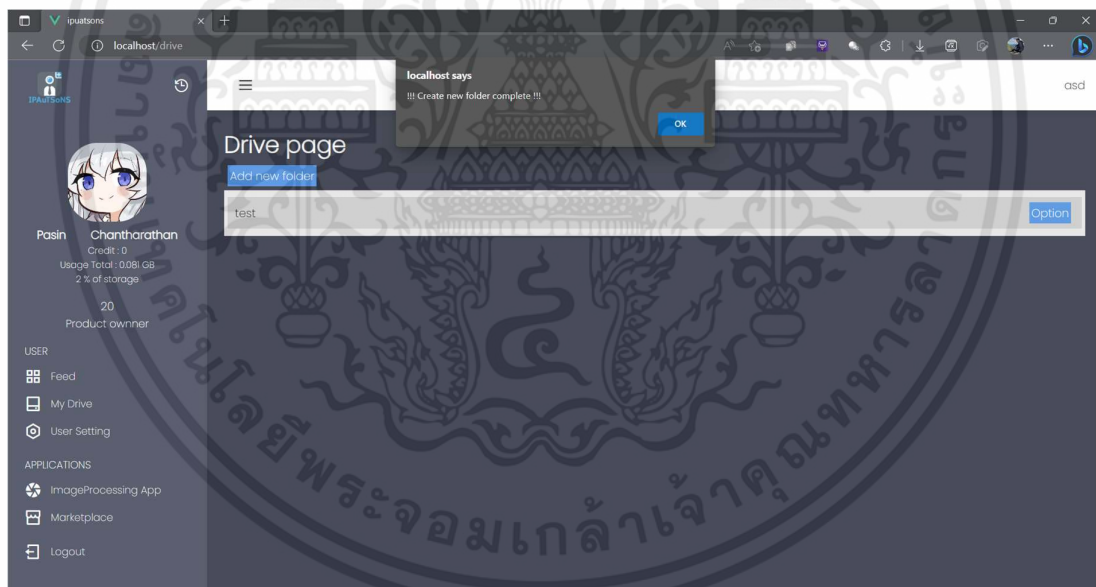
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.1.2 วิธีการทดลอง

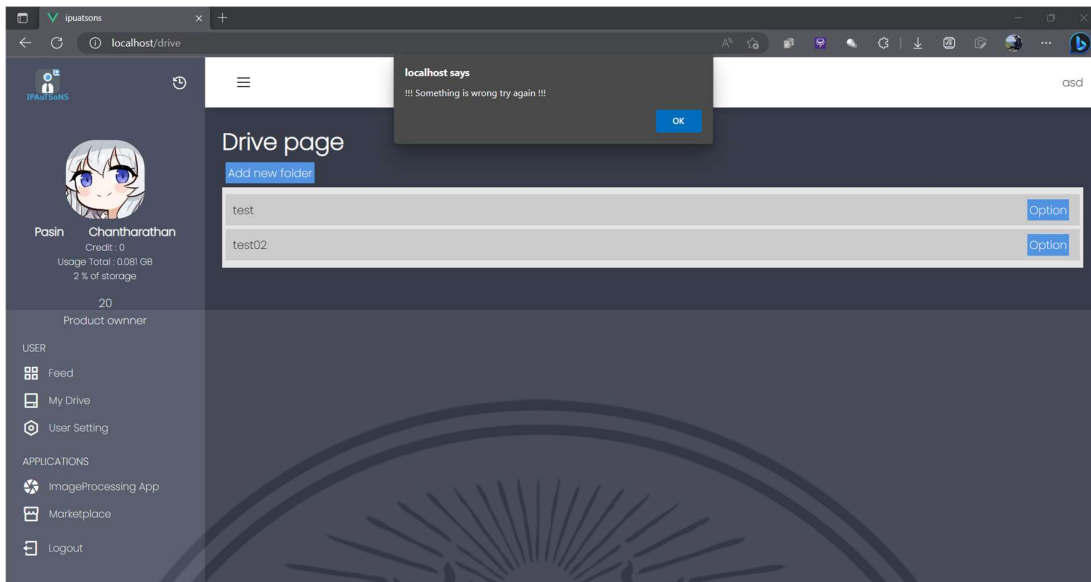
ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยทำการเพิ่ม-ลบ Folder เก็บไฟล์ในหลายๆ รูปแบบแบบที่ต่างกันและซ้ำกัน หากทำงานได้สำเร็จจะมีข้อความแสดงผลบอกว่าทำการเพิ่ม-ลบ Folder สำเร็จ และ ทำการเพิ่ม-ลบข้อมูลของ Folder ในฐานข้อมูล โดยที่การทำงานหากมีข้อผิดพลาดก็จะส่งแสดงผลข้อความว่าการทำงานนั้นไม่สำเร็จ

4.4.1.3 ผลการทดลอง

ส่วนของระบบจัดการ Folder สำหรับเก็บไฟล์รูปภาพสามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Backend ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถยืนยันตัวตนผู้ใช้ที่ได้ทำการส่งเข้ามา และ ทำการ เพิ่ม-ลบ Folder เก็บรูปภาพได้ และสามารถตรวจสอบชื่อ Folder ที่ซ้ำกันได้ รวมไปถึงทำการเพิ่ม-ลบข้อมูลของ Folder ในฐานข้อมูลได้ และสามารถส่งข้อความแจ้งเตือนได้หากเกิดการดำเนินงานที่ผิดพลาดขึ้น โดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.11 - 4.12



รูป 4.11 ข้อความที่แสดงหากสามารถเพิ่ม Folder สำหรับเก็บรูปภาพได้



รูป 4.12 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะเพิ่ม Folder สำหรับเก็บรูปภาพ

4.4.2 การทดลองระบบจัดการไฟล์รูปภาพของผู้ใช้

4.4.2.1 จุดประสงค์การทดลอง

เพื่อทดสอบการทำงานของ Web applications หน้าจัดการไฟล์ในส่วนจัดการไฟล์รูปภาพของผู้ใช้ว่าสามารถทำงานถูกต้อง สามารถเพิ่ม รูปภาพ และ ลบ รูปภาพ ไปได้โดยมีการส่ง Cookie ไปยัง Backend เพื่อยืนยันตัวตน โดยจะเพิ่ม-ลบข้อมูลของ รูปภาพ ที่ฐานข้อมูลด้วย โดยถ้าหากมีข้อผิดพลาดก็จะแสดงข้อความแจ้งเตือนข้อผิดพลาดขึ้นมา

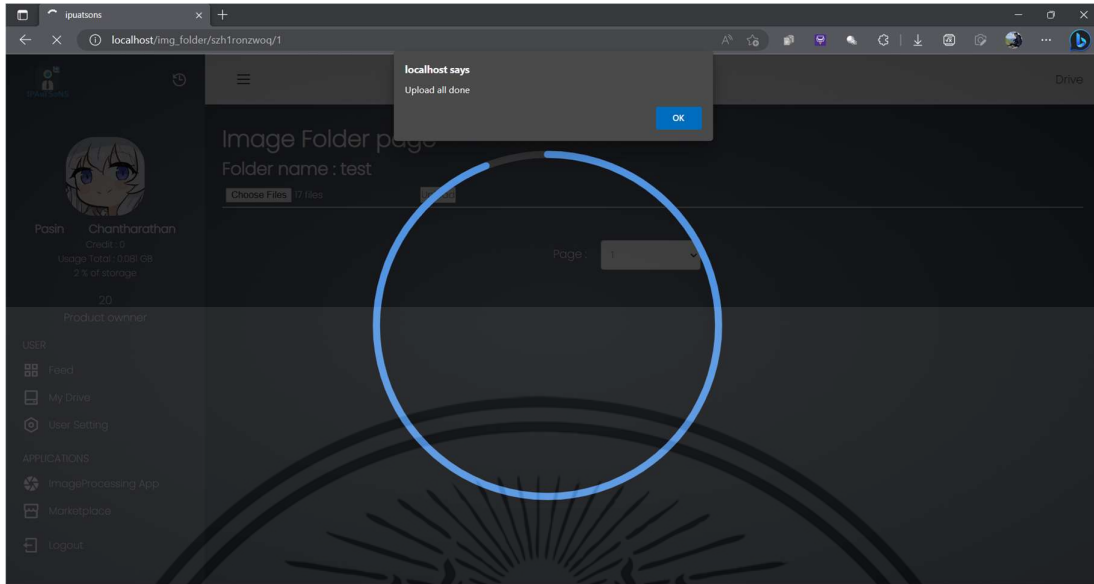
4.4.2.2 วิธีการทดลอง

ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยทำการเพิ่ม-ลบ Folder เก็บไฟล์ในหลายๆ รูปแบบแบบที่ต่างกันและซ้ำกัน หากทำงานได้สำเร็จจะมีข้อความแสดงผลบอกว่าทำการเพิ่ม-ลบ Folder สำเร็จ และ ทำการเพิ่ม-ลบข้อมูลของ Folder ในฐานข้อมูล โดยที่การทำงานหากมีข้อผิดพลาดก็จะส่งแสดงผลข้อความว่าการทำงานนั้นไม่สำเร็จ

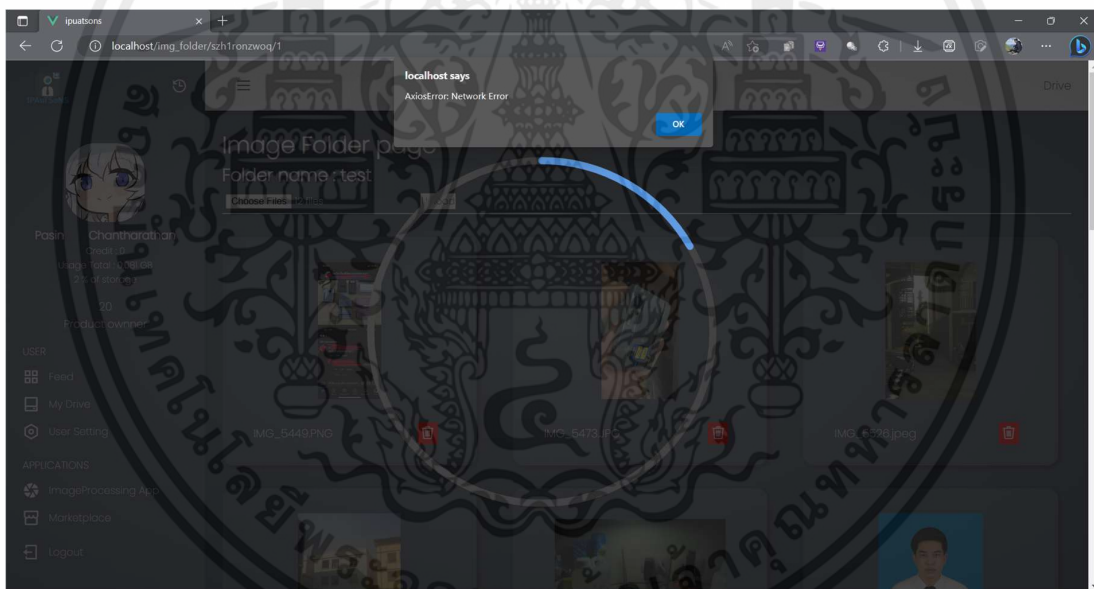
4.4.2.3 ผลการทดลอง

ส่วนของระบบจัดการไฟล์รูปภาพสามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Backend ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถยืนยันตัวตนผู้ใช้ที่ได้ทำการส่งเข้ามา และ ทำการ เพิ่ม-ลบรูปภาพได้ รวมไปถึงทำการเพิ่ม-ลบข้อมูลของรูปภาพในฐานข้อมูลได้ และสามารถส่งข้อความแจ้งเตือนได้หากเกิดการดำเนินงานที่ผิดพลาดขึ้น โดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.13

- 4.14

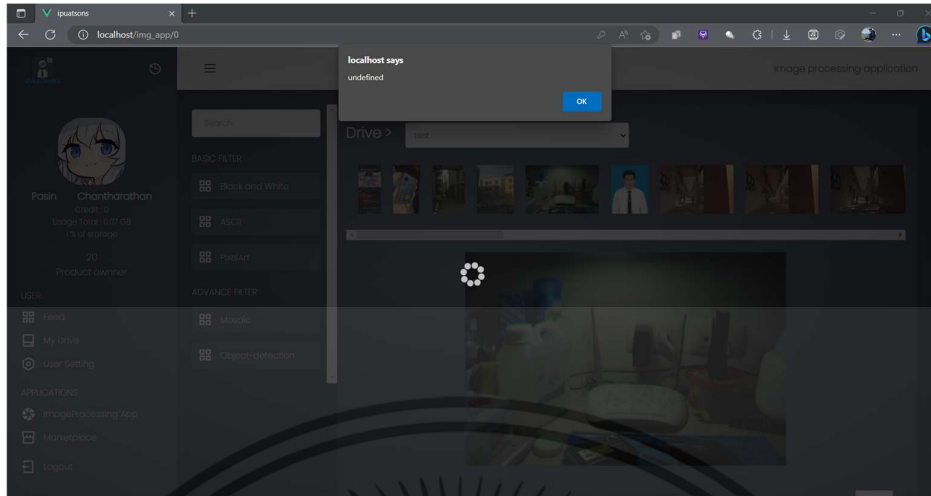


รูป 4.13 ข้อความที่แสดงหากสามารถเพิ่มรูปภาพสำเร็จ



รูป 4.14 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะเพิ่มรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.16 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะสร้างไฟล์งานประมวลผล

4.5.2 การทดลองระบบการสร้างไฟล์การสั่งงานประมวลผลภาพ

4.5.2.1 จุดประสงค์การทดลอง

เพื่อทดสอบการทำงานของ Web applications หน้าสั่งงานประมวลผลในส่วนระบบการสร้างไฟล์ YAML เพื่อสั่งงานประมวลผลภาพของผู้ใช้ว่าสามารถทำงานถูกต้อง สามารถสั่งงานประมวลผลภาพได้โดยที่ Web application ส่งข้อมูลไปยังส่วน Backend และมีการส่ง Cookie ไปยังส่วน Backend เพื่อยืนยันตัวตน โดยจะเพิ่มข้อมูลของ งานประมวลผล ที่ฐานข้อมูลด้วย โดยถ้าหากมีข้อผิดพลาดก็จะแสดงข้อความแจ้งเตือนข้อผิดพลาดขึ้นมา

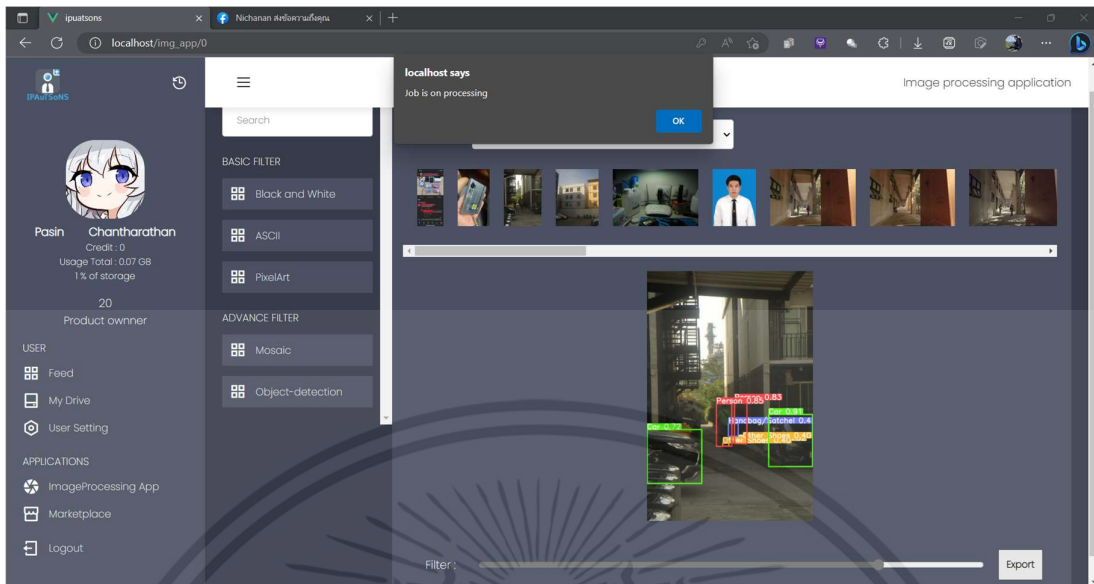
4.5.2.2 วิธีการทดลอง

ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยทำการเพิ่มงานประมวลผลภาพ ในหลายๆ รูปแบบแบบที่ต่างกัน หากทำงานได้สำเร็จจะมีข้อความแสดงผลบอกว่าทำการสร้างงานประมวลผลสำเร็จ และ ทำการเพิ่มข้อมูลของ งานประมวลผล ในฐานข้อมูล โดยที่การทำงานหากมีข้อผิดพลาดก็จะส่งแสดงผลข้อความว่าการทำงานนั้นไม่สำเร็จ

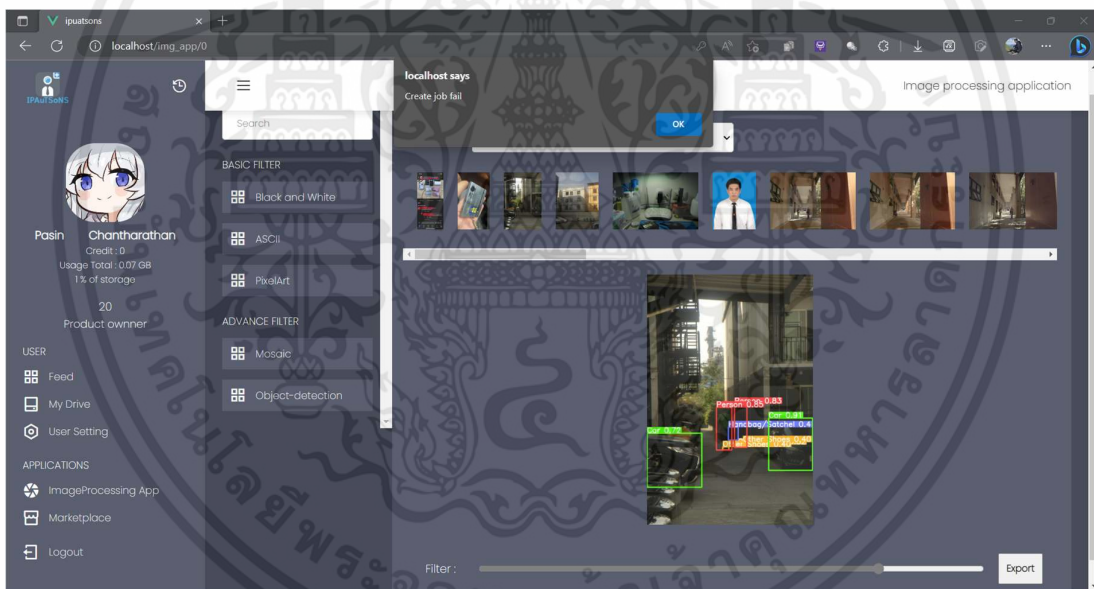
4.5.2.3 ผลการทดลอง

ส่วนของระบบจัดการไฟล์รูปภาพสามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Backend ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถยืนยันตัวตนผู้ใช้ที่ได้ทำการส่งเข้ามา และทำการ เพิ่ม-ลบรูปภาพได้ รวมไปถึงทำการเพิ่ม-ลบข้อมูลของรูปภาพในฐานข้อมูลได้ และสามารถส่งข้อความแจ้งเตือนได้หากเกิดการดำเนินงานที่ผิดพลาดขึ้น โดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.17

- 4.18



รูป 4.17 ข้อความที่แสดงหากสามารถสร้างไฟล์งานประมวลผลสำเร็จ



รูป 4.18 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะสร้างไฟล์งานประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ส่วนของหน้าตลาดซื้อขาย

4.6.1 การทดลองระบบการเพิ่มสินค้าในตลาดซื้อขาย

4.6.1.1 จุดประสงค์การทดลอง

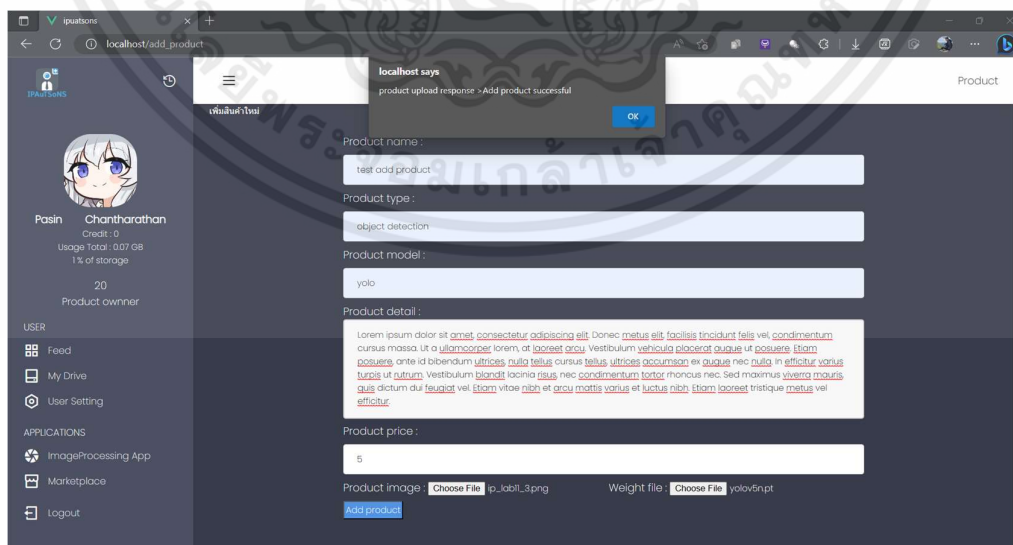
เพื่อทดสอบการใช้งานระบบการเพิ่มสินค้าว่ามีการทำงานที่ถูกต้อง โดยจะสามารถส่งข้อมูลสินค้า และ ตัว Weight ของ model และรูปภาพของการใช้สินค้า ที่ผู้ใช้กรอกเข้ามาจากหน้า Frontend ไปยัง Backend ได้และทดสอบการทำงานของตัว Weight ได้ โดยที่หากการทำงานสำเร็จ ข้อมูลของสินค้าถูกเพิ่มไปที่ฐานข้อมูลด้วย และ หากการทำงานไม่สำเร็จก็จะมีแจ้งเตือนความแจ้งเตือนผู้ใช้

4.6.1.2 วิธีการทดลอง

ทดสอบโดยการเพิ่มสินค้าด้วยข้อมูลที่แตกต่างกันหลายๆ ครั้ง โดยจะตรวจสอบข้อมูลที่ได้รับมาและถูกเพิ่มในฐานข้อมูลว่ามีความครบถ้วนและถูกต้องตรงตามรูปแบบที่ได้มีการออกแบบเอาไว้ และถ้าหากการทำงานเกิดข้อผิดพลาดขึ้น ก็จะมีการแจ้งเตือนความแจ้งเตือนผู้ใช้ว่าการเพิ่มสินค้านั้นไม่สำเร็จ

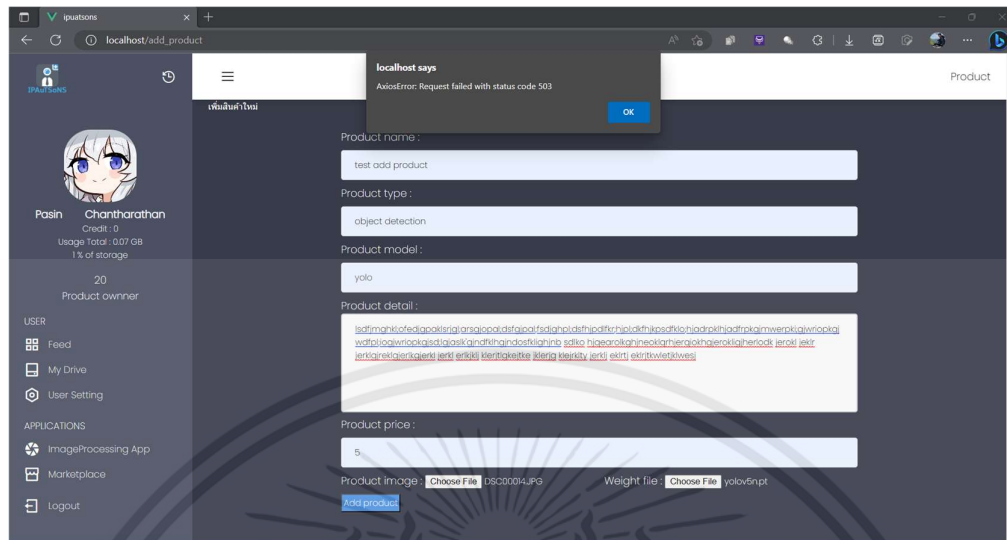
4.6.1.3 ผลการทดลอง

จากการการทดลองระบบการเพิ่มสินค้าในตลาดซื้อขายของ Web application สามารถเพิ่มสินค้า และสามารถตรวจสอบ Weight ของ Model ที่ Upload เข้ามาโดยหลังการทำงานเสร็จสิ้นจะแสดงข้อความว่าผู้ใช้นั้นเพิ่มสินค้าสำเร็จและแสดงหน้าต่างหน้า Marketplace ขึ้นมา และในการทำงานที่ไม่สำเร็จนั้นก็สามารถแสดงข้อความกลับมาให้ผู้ใช้ได้รับทราบ โดยการแสดงผลแต่ละส่วนสามารถดูได้จากรูปที่ 4.19 และ 4.20



รูป 4.19 ข้อความที่แสดงหากสามารถเพิ่มสินค้าได้สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.20 ข้อความที่แสดงหากไม่สามารถเพิ่มสินค้าได้

4.6.2 การทดลองระบบการแก้ไขข้อมูลสินค้าในตลาดซื้อขาย

4.6.2.1 จุดประสงค์การทดลอง

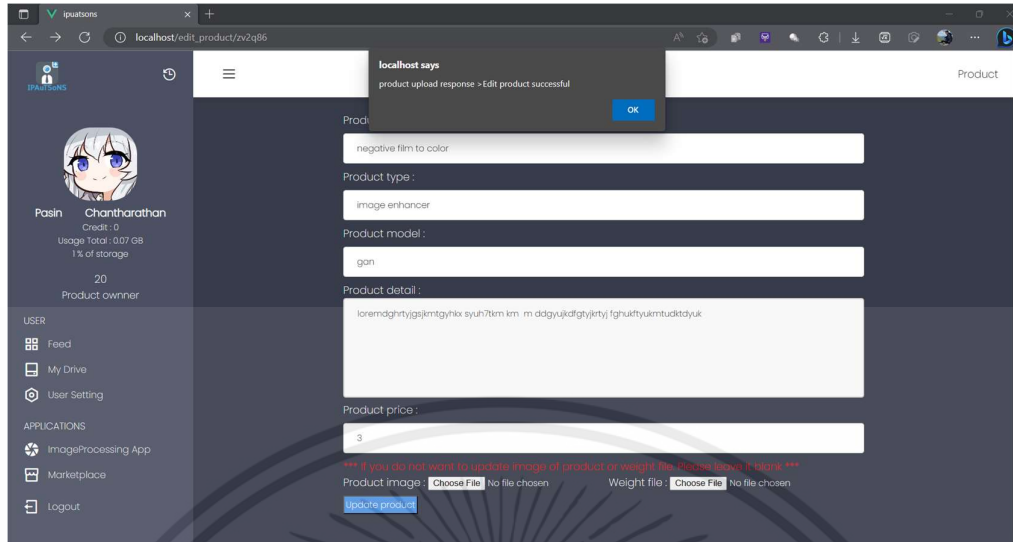
เพื่อทดสอบการใช้งานระบบการแก้ไขข้อมูลสินค้าในตลาดซื้อขายว่ามีการทำงานที่ถูกต้อง โดยจะสามารถส่งข้อมูลสินค้า และ ตัว Weight ของ model และรูปภาพของการใช้สินค้าที่ผู้ใช้ได้ทำการแก้ไข จากหน้า Frontend ไปยัง Backend ได้และทดสอบการทำงานของตัว Weight ได้ โดยที่หากการทำงานสำเร็จ ข้อมูลของสินค้าที่อยู่ในฐานข้อมูลจะมีการแก้ไข และ หากการทำงานไม่สำเร็จก็จะมีแสดงข้อความแจ้งเตือนผู้ใช้

4.6.2.2 วิธีการทดลอง

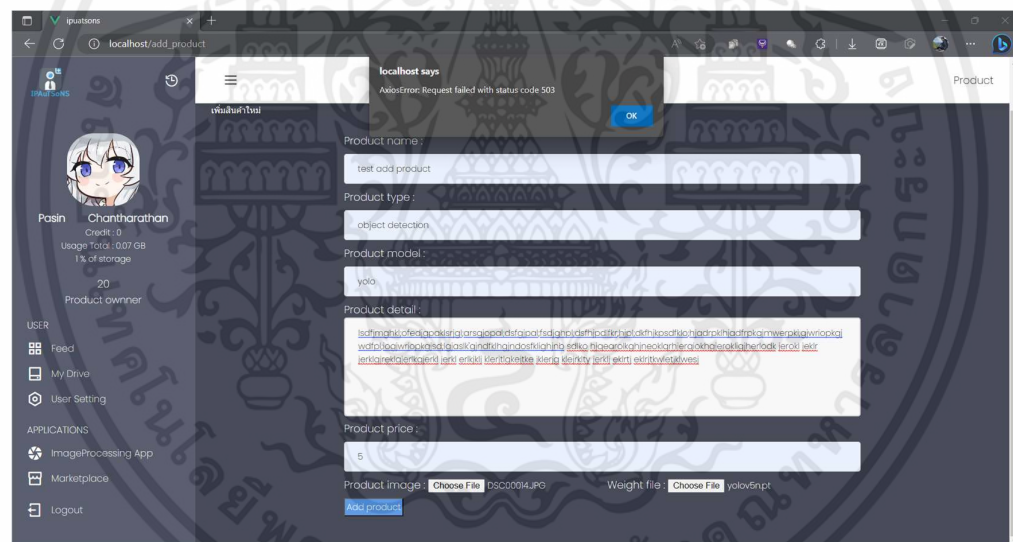
ทดสอบโดยการแก้ไขข้อมูลสินค้าด้วยข้อมูลที่แตกต่างกันหลายๆ ครั้ง โดยจะตรวจสอบข้อมูลที่รับมาและถูกแก้ไขในฐานข้อมูลว่ามีความครบถ้วนและถูกต้องตรงตามรูปแบบที่ได้มีการออกแบบเอาไว้และถ้าหากการทำงานเกิดข้อผิดพลาดขึ้น ก็จะมีการแสดงข้อความแจ้งเตือนผู้ใช้ว่าการแก้ไขข้อมูลสินค้านั้นไม่สำเร็จ

4.6.2.3 ผลการทดลอง

จากการการทดลองระบบการเพิ่มสินค้าในตลาดซื้อขายของ Web application สามารถแก้ไขข้อมูลสินค้าและสามารถตรวจสอบ Weight ของ Model ที่ Upload เข้ามาโดยหลังการทำงานเสร็จสิ้นจะแสดงข้อความว่าผู้ใช้นั้นแก้ไขข้อมูลสินค้าสำเร็จและแสดงหน้าต่างหน้า Marketplace ขึ้นมา และในการทำงานที่ไม่สำเร็จนั้นก็สามารถแสดงข้อความกลับมาให้ผู้ใช้ได้รับทราบ โดยการแสดงผลแต่ละส่วนสามารถดูได้จากรูปที่ 4.21 และ 4.22



รูป 4.21 ข้อความที่แสดงหากสามารถแก้ไขข้อมูลสินค้าได้สำเร็จ



รูป 4.22 ข้อความที่แสดงหากไม่สามารถแก้ไขสินค้าได้

4.6.3 การทดลองระบบการส่งข้อมูลสินค้าเพื่อทดลองใช้ในส่วนหน้าสำนักงานประมวลผล

4.6.3.1 จุดประสงค์การทดลอง

เพื่อทดสอบการใช้งานระบบการส่งข้อมูลสินค้าเพื่อทดลองใช้ในส่วนหน้าสำนักงานประมวลผลว่ามีการทำงานที่ถูกต้อง โดยจะสามารถส่งข้อมูลสินค้าจากหน้าส่วนตลาดซื้อขาย ไปยังหน้าสำนักงานประมวลผลได้ และ แสดงทดลองใช้งานตัวสินค้าในหน้าสำนักงานประมวลผลได้ โดยที่หากการทำงานสำเร็จ ข้อมูลของสินค้าที่อยู่ในฐานข้อมูลจะถูกส่งไปหน้าสำนักงานประมวลผลและสามารถทดลองใช้งานได้ และ หากการทำงานไม่สำเร็จก็จะมี การแสดงข้อความแจ้งเตือนผู้ใช้

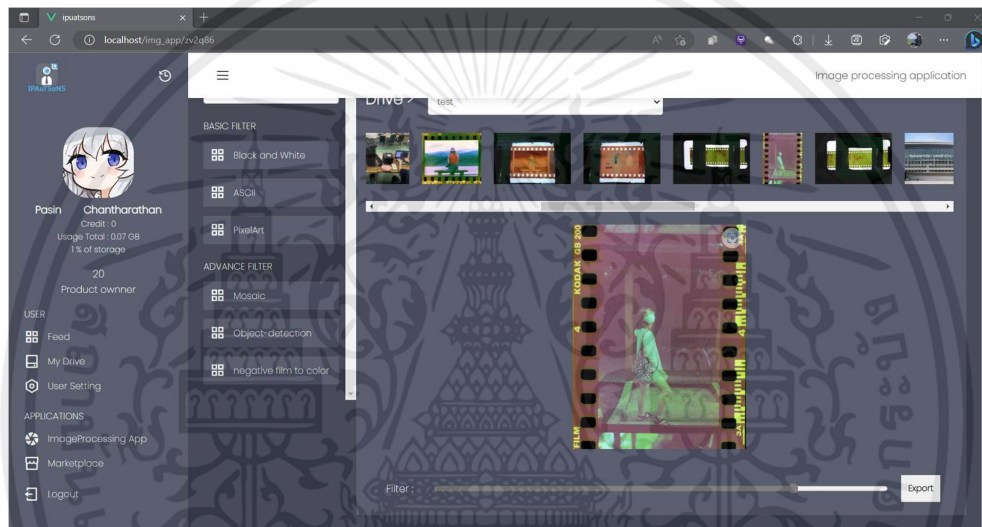
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.3.2 วิธีการทดลอง

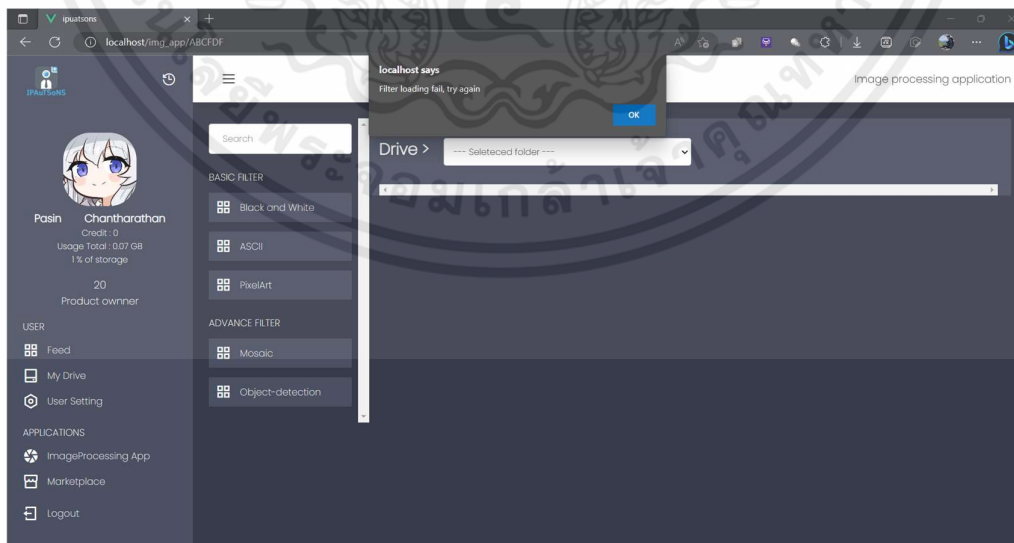
ทดสอบโดยการเพิ่มสินค้าจำนวนหนึ่งมาแล้วทำการเลือกทดลองใช้สินค้านั้นในหน้าสำนักงานประมวลผล โดยจะมีการส่งข้อมูลแบบที่ผิดพลาดไปด้วยเพื่อทดสอบการตรวจจับข้อผิดพลาด

4.6.3.3 ผลการทดลอง

จากการการทดลองระบบการส่งข้อมูลสินค้าเพื่อทดลองใช้ในส่วนหน้าสำนักงานประมวลผล ของ Web application สามารถส่งข้อมูลสินค้าเพื่อ ไปแสดงผลในส่วนหน้าสำนักงานประมวลผลได้โดยมีข้อมูลครบถ้วนและถูกต้องตามที่ได้ทำการออกแบบไว้ และเมื่อส่งข้อมูลในรูปแบบที่ผิดพลาดก็สามารถแสดงข้อความแจ้งเตือนผู้ใช้ได้ โดยแต่ละส่วนสามารถดูได้จากรูปที่ 4.23 และ 4.24



รูป 4.23 ข้อมูลสินค้าที่สามารถส่งมาทดลองใช้งานที่หน้าสำนักงานประมวลผล จะแสดงในแถบทางด้านซ้าย



รูป 4.24 ข้อความที่แสดงหากไม่สามารถส่งข้อมูลสินค้ามาที่หน้าสำนักงานประมวลผลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 ส่วนของหน้าประวัติการใช้งานของผู้ใช้

4.6.1 การทดลองระบบการเรียงลำดับและค้นหาการทำรายการด้วยวันที่

4.6.1.1 จุดประสงค์การทดลอง

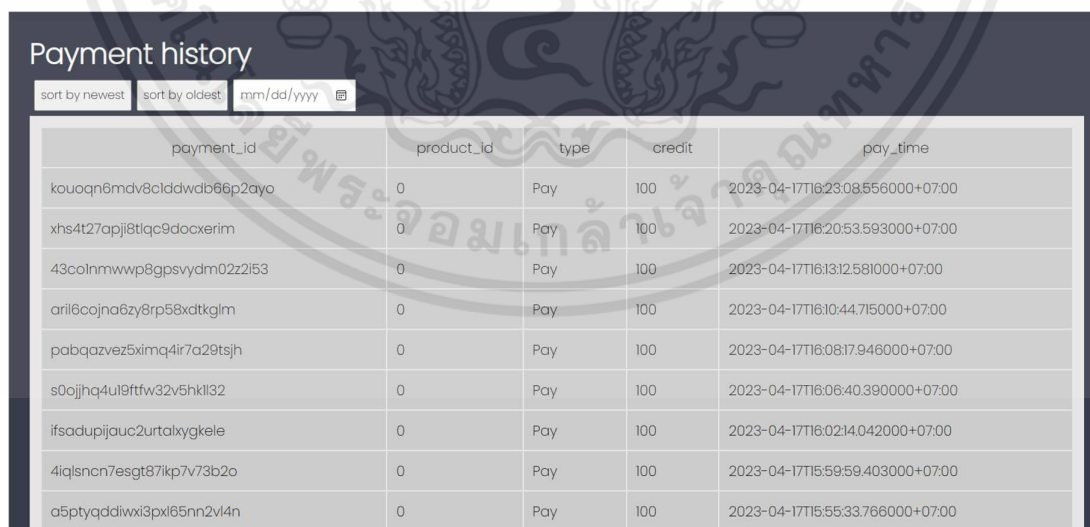
เพื่อทดสอบการแสดงผลและประมวลผลของการทำรายการต่าง ๆ ของผู้ใช้เช่นการจ่ายเครดิตเพื่อสั่งงานประมวลผล หรือ การได้รับเครดิตจากการที่มีผู้ใช้นำสินค้าเราไปสั่งงานประมวลผล ซึ่งจะมีการเรียงลำดับรายการเพื่อให้เกิดความสะดวกต่อผู้ใช้ในคูประวัติการใช้งานของตนเอง

4.6.1.2 วิธีการทดลอง

ทดลองโดยการเพิ่มรายการเป็นข้อมูลที่เตรียมเอาไว้โดยไม่เรียงลำดับการเพิ่มข้อมูลแต่วันที่ในข้อมูลนั้นเป็นเวลาใกล้เคียงกัน โดนห่างกันตั้งแต่ 1 นาที, 5 นาที จนไปถึง 2 วัน เมื่อมีการเรียกดูข้อมูลค่าเริ่มต้นจะเรียกดูข้อมูลที่มีวันที่ล่าสุดขึ้นมาก่อน และเรียงลำดับจนไปถึงต่ำสุด และเมื่อมีการเรียกดูข้อมูลเก่าสุด ก็จะแสดงผลเป็นข้อมูลที่มีวันที่เก่าที่สุด และเรียงลำดับไปจนถึงใหม่สุด และถ้าหากเลือกค้นหาเป็นวันที่ ก็จะแสดงผลรายการที่เก่าที่สุดนับตั้งแต่วันที่นั้นและเรียงลำดับมาจนถึงรายการที่มีวันที่ล่าสุด

4.6.1.3 ผลการทดลอง

จากการทดลองระบบการเรียงลำดับและค้นหาการทำรายการด้วยวันที่ของหน้าประวัติการใช้งานของผู้ใช้ ผลคือสามารถเรียงลำดับและแสดงผลรายการได้อย่างถูกต้องและครบถ้วน โดยสามารถดูผลการเรียงลำดับและค้นหา ได้รูปที่ 4.25 , 4.26 และ 4.27



The screenshot shows a 'Payment history' interface with a table of transactions. The table has columns for payment_id, product_id, type, credit, and pay_time. All transactions are of type 'Pay' with a credit of 100. The pay_time column shows dates in YYYY-MM-DD HH:MM:SS+07:00 format, ranging from 2023-04-17T16:23:08 to 2023-04-17T15:55:33.

payment_id	product_id	type	credit	pay_time
kouoqn6mdv8clddwb86p2ayo	0	Pay	100	2023-04-17T16:23:08.556000+07:00
xhs4t27apji8tlqc9docxerim	0	Pay	100	2023-04-17T16:20:53.593000+07:00
43coInmwpp8gpsvym02z2l53	0	Pay	100	2023-04-17T16:13:12.581000+07:00
aril6cojna6zy8rp58xdtkgim	0	Pay	100	2023-04-17T16:10:44.715000+07:00
pabqazvez5ximq4ir7a29tsjh	0	Pay	100	2023-04-17T16:08:17.946000+07:00
s0aijhnq4ul9ftfw32v5hkli32	0	Pay	100	2023-04-17T16:06:40.390000+07:00
ifsadupijauc2urtalxygkele	0	Pay	100	2023-04-17T16:02:14.042000+07:00
4iqlsncn7esgt87lqp7v73b2o	0	Pay	100	2023-04-17T15:59:59.403000+07:00
a5ptyqddiwxl3pxl65nn2vl4n	0	Pay	100	2023-04-17T15:55:33.766000+07:00

รูป 4.25 ประวัติรายการของผู้ใช้ที่มีการเรียงลำดับโดยการแสดงรายการล่าสุดก่อน

Payment history

sort by newest | sort by oldest | mm/dd/yyyy

payment_id	product_id	type	credit	pay_time
9llpovlyfebva7lsaojvvyocu		Receive	1000	2023-04-01T15:25:14.094000+07:00
slrcbniz4zqmq8mu3i24hajvt		Receive	100000	2023-04-16T17:22:21.421000+07:00
a4lvq7n8rehplesb3bn3k59j9	0	Pay	100	2023-04-17T15:53:01.604000+07:00
ugxbt9n32hrycp62penyjjoi	0	Pay	100	2023-04-17T15:53:34.570000+07:00
a5ptyqdaiwxi3pxi65nn2v4n	0	Pay	100	2023-04-17T15:55:33.766000+07:00
4iqslncn7esgt87kpv73b2o	0	Pay	100	2023-04-17T15:59:59.403000+07:00
ifsadupijauc2urtalxygkele	0	Pay	100	2023-04-17T16:02:14.042000+07:00
s0ajjhq4u19ftfw32v5hklI32	0	Pay	100	2023-04-17T16:06:40.390000+07:00
pabqazvez5ximq4ir7a29tsjh	0	Pay	100	2023-04-17T16:08:17.946000+07:00
aril6cojna6zy8rp58xdtkglm	0	Pay	100	2023-04-17T16:10:44.715000+07:00

รูป 4.26 ประวัติรายการของผู้ใช้ที่มีการเรียงลำดับโดยการแสดงรายการเก่าที่สุดก่อน

Payment history

sort by newest | sort by oldest | mm/dd/yyyy

payment_id	product_id	type	credit	pay_time
slrcbniz4zqmq8mu3i24hajvt		Receive	100000	2023-04-16T17:22:21.421000+07:00
a4lvq7n8rehplesb3bn3k59j9	0	Pay	100	2023-04-17T15:53:01.604000+07:00
ugxbt9n32hrycp62penyjjoi	0	Pay	100	2023-04-17T15:53:34.570000+07:00
a5ptyqdaiwxi3pxi65nn2v4n	0	Pay	100	2023-04-17T15:55:33.766000+07:00
4iqslncn7esgt87kpv73b2o	0	Pay	100	2023-04-17T15:59:59.403000+07:00
ifsadupijauc2urtalxygkele	0	Pay	100	2023-04-17T16:02:14.042000+07:00
s0ajjhq4u19ftfw32v5hklI32	0	Pay	100	2023-04-17T16:06:40.390000+07:00
pabqazvez5ximq4ir7a29tsjh	0	Pay	100	2023-04-17T16:08:17.946000+07:00
aril6cojna6zy8rp58xdtkglm	0	Pay	100	2023-04-17T16:10:44.715000+07:00

รูป 4.27 ประวัติรายการของผู้ใช้ที่มีค้นหาโดยใช้วันที่ ที่กำหนดเอาไว้

4.8 ส่วนของหน้าประวัติการใช้งานของสินค้า

4.6.1 การทดลองระบบการเรียงลำดับและค้นหาการทำรายการด้วยวันที่

4.6.1.1 จุดประสงค์การทดลอง

เพื่อทดสอบการแสดงผลและประมวลผลของการทำรายการการถูกใช้งานของตัวสินค้า ซึ่งจะ
สามารถดูได้เฉพาะเจ้าของของสินค้านั้นๆ เท่านั้น ซึ่งจะมีการเรียงลำดับรายการเพื่อให้เกิดความสะดวก
ต่อผู้ใช้ในดูประวัติการใช้งานของสินค้าตนเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.1.2 วิธีการทดลอง

ทดลองโดยการเพิ่มรายการเป็นข้อมูลที่เตรียมเอาไว้โดยไม่เรียงลำดับการเพิ่มข้อมูลแต่วันที่ในข้อมูลนั้นเป็นเวลาใกล้เคียงกัน โดนห่างกันตั้งแต่ 1 นาที, 5 นาที จนไปถึง 2 วัน เมื่อมีการเรียกดูข้อมูลค่าเริ่มต้นจะเรียกดูข้อมูลที่มีวันที่ล่าสุดขึ้นมาก่อน และเรียงลำดับจนไปถึงต่ำสุด และเมื่อมีการเรียกดูข้อมูลเก่าสุด ก็จะแสดงผลเป็นข้อมูลที่มีวันที่เก่าที่สุด และเรียงลำดับไปจนถึงใหม่สุด และถ้าหากเลือกค้นหาเป็นวันที่ ก็จะแสดงผลรายการที่เก่าที่สุดนับตั้งแต่วันที่นั้นและเรียงลำดับมาจนถึงรายการที่มีวันที่ล่าสุด

4.6.1.3 ผลการทดลอง

จากการทดลองระบบการเรียงลำดับและค้นหาการทำรายการด้วยวันที่ของหน้าประวัติการใช้งานของผู้ใช้ ผลคือสามารถเรียงลำดับและแสดงผลรายการได้อย่างถูกต้องและครบถ้วน โดยสามารถดูผลการเรียงลำดับและค้นหา ได้รูปที่ 4.28 , 4.29 และ 4.30

payment_id	user_id	credit	type	pay_time
r7pgtuphmoicwlnbup4cnuhb	p5mitrb9ar	5	1	2023-04-02T16:25:06.507000+07:00
sok07vruofm3y4dfr8ggrid3f	p5mitrb9ar	5	1	2023-04-01T16:25:38.0000+07:00
vn2bz4m8ze9d29mwbnbw42w	p5mitrb9ar	5	1	2023-04-01T16:25:30.540000+07:00
a902tlcdq5pb6nqt6qwhzy	p5mitrb9ar	5	1	2023-04-01T16:25:19.846000+07:00
9llpovlyfebva7saojwyocu	p5mitrb9ar	5	1	2023-04-01T16:25:14.094000+07:00

รูป 4.28 ประวัติรายการของสินค้าที่มีการเรียงลำดับ โดยการแสดงรายการล่าสุดก่อน

payment_id	user_id	credit	type	pay_time
9llpovlyfebva7saojwyocu	p5mitrb9ar	5	1	2023-04-01T16:25:14.094000+07:00
a902tlcdq5pb6nqt6qwhzy	p5mitrb9ar	5	1	2023-04-01T16:25:19.846000+07:00
vn2bz4m8ze9d29mwbnbw42w	p5mitrb9ar	5	1	2023-04-01T16:25:30.540000+07:00
sok07vruofm3y4dfr8ggrid3f	p5mitrb9ar	5	1	2023-04-01T16:25:38.0000+07:00
r7pgtuphmoicwlnbup4cnuhb	p5mitrb9ar	5	1	2023-04-02T16:25:06.507000+07:00

รูป 4.29 ประวัติรายการของสินค้าที่มีการเรียงลำดับ โดยการแสดงรายการเก่าที่สุดก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Product use history				
sort by newest		sort by oldest		mm/dd/yyyy
payment_id	user_id	credit	type	pay_time
rj7pgtuphmoicwlnbup4cnuhb	p5mitrb9ar	5	1	2023-04-02T16:25:06.507000+07:00

รูป 4.30 ประวัติรายการของสินค้าที่มีค้นหาโดยใช้วันที่ ที่กำหนดเอาไว้

4.9 ส่วนของหน้าร้านค้าแสดงสินค้า

4.9.1 การทดลองระบบการเรียงลำดับสินค้าด้วยวันที่ที่มีการแก้ไขข้อมูลสินค้า

4.9.1.1 จุดประสงค์การทดลอง

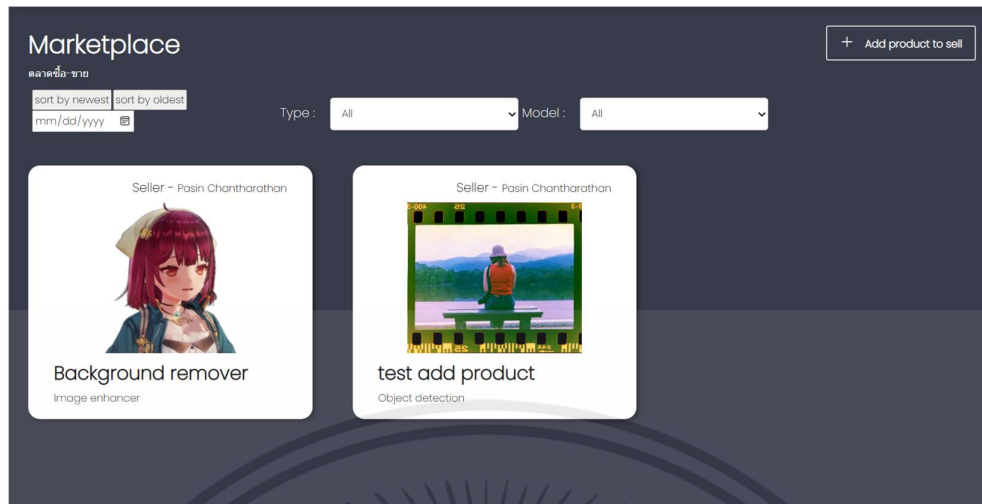
เพื่อทดสอบการแสดงผลและประมวลผลของสินค้าซึ่งจะมีการเรียงลำดับการแสดงผลสินค้าแต่ละรายการเพื่อให้เกิดความสะดวกต่อผู้ใช้ในการเลือกชมสินค้า

4.9.1.2 วิธีการทดลอง

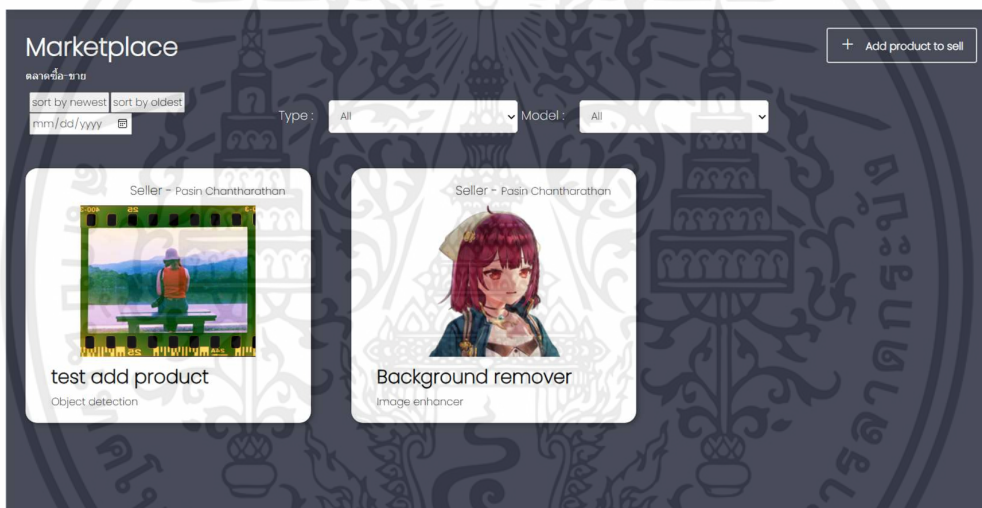
ทดลองโดยการเพิ่มสินค้าที่เป็นข้อมูลที่เตรียมเอาไว้โดยไม่เรียงลำดับการเพิ่มข้อมูลแต่วันที่ที่แก้ไขในข้อมูลนั้นเป็นเวลาใกล้เคียงกัน โดนห่างกันตั้งแต่ 1 นาที, 5 นาที จนไปถึง 2 วัน เมื่อมีการเรียกดูข้อมูลค่าเริ่มต้นจะเรียกดูสินค้าที่มีวันที่การแก้ไขล่าสุดขึ้นมาก่อน และเรียงลำดับจนไปถึงสินค้าวันที่การแก้ไขเก่าสุด และเมื่อมีการเรียกดูสินค้าที่มีวันที่แก้ไขเก่าสุด ก็จะแสดงผลเป็นข้อมูลที่มีวันที่ที่แก้ไขข้อมูลสินค้าเก่าที่สุด และเรียงลำดับไปจนถึงใหม่สุด

4.9.1.3 ผลการทดลอง

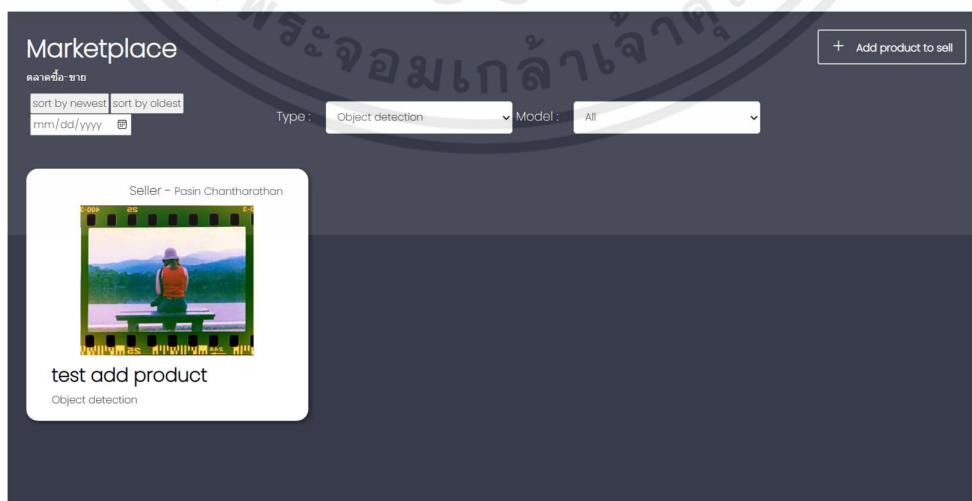
จากการทดลองระบบการเรียงลำดับและค้นหาการทำรายการด้วยวันที่ของหน้าประวัติการใช้งานของผู้ใช้ ผลคือสามารถเรียงลำดับและแสดงผลรายการได้อย่างถูกต้องและครบถ้วน โดยสามารถดูผลการเรียงลำดับและค้นหา ได้รูปที่ 4.31 , 4.32 และ 4.33



รูป 4.31 รายการของสินค้าที่มีการเรียงลำดับ โดยการแสดงสินค้าที่มีการแก้ไขล่าสุด



รูป 4.32 รายการของสินค้าที่มีการเรียงลำดับ โดยการแสดงสินค้าที่มีการแก้ไขล่าสุด



รูป 4.33 รายการของสินค้าที่มีการกรองเลือกเฉพาะประเภท Object detections

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.10 ส่วนของหน้าแสดงข้อมูลเฉพาะสินค้า

4.10.1 การทดลองระบบแสดงผลข้อมูลสินค้า

4.9.1.1 จุดประสงค์การทดลอง

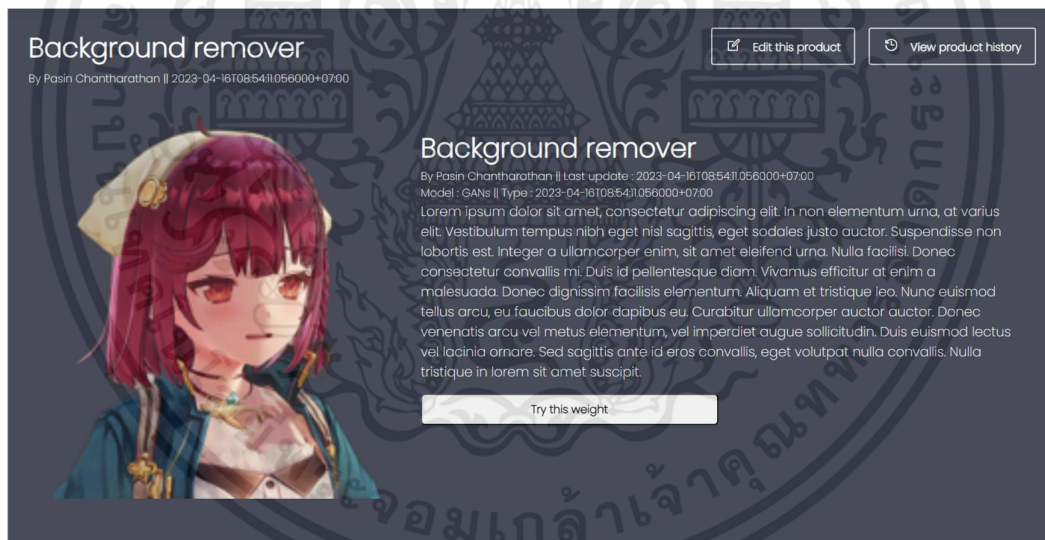
เพื่อทดสอบการแสดงผลข้อมูลของสินค้าซึ่งจะมีรายละเอียดของตัวสินค้าเพื่อให้เกิดความสะดวกต่อผู้ใช้ในการเลือกชมสินค้า

4.10.1.2 วิธีการทดลอง

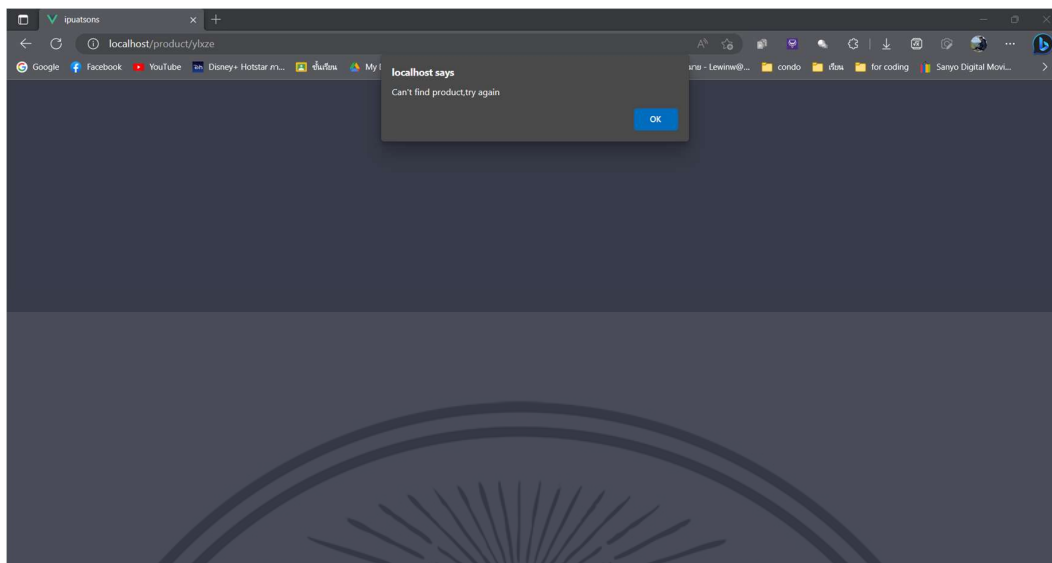
ทดลองโดยการเพิ่มสินค้าที่เป็นข้อมูลที่เตรียมเอาไว้ หลากๆรูปแบบที่สามารถถูกเพิ่มเข้ามาใน Web applications ได้และ ทดลองทำการเข้าหาสินค้าที่ไม่มีอยู่จริงซึ่งถ้าหากหาสินค้าในฐานข้อมูลไม่พอก็จะแสดงข้อความแจ้งเตือนข้อผิดพลาด

4.10.1.3 ผลการทดลอง

จากการทดลองระบบแสดงผลข้อมูลสินค้า ผลคือแสดงผลรายละเอียดสินค้าได้อย่างถูกต้องและครบถ้วน และสามารถแจ้งเตือนผู้ใช้ได้หากไม่มีสินค้านั้นๆอยู่หรือเกิดข้อผิดพลาด ได้รูปที่ 4.34 , 4.35



รูป 4.34 แสดงรายละเอียดของสินค้าที่ถูกบันทึกเอาไว้ในฐานข้อมูล



รูป 4.35 แสดงข้อความแจ้งเตือนหากทำหาสินค้าที่ไม่มีอยู่ในฐานข้อมูล

4.11 ส่วนของการแบ่งภาระงาน

4.11.1 การทดลองระบบการแบ่งภาระงานของ Web Service

4.11.1.1 จุดประสงค์การทดลอง

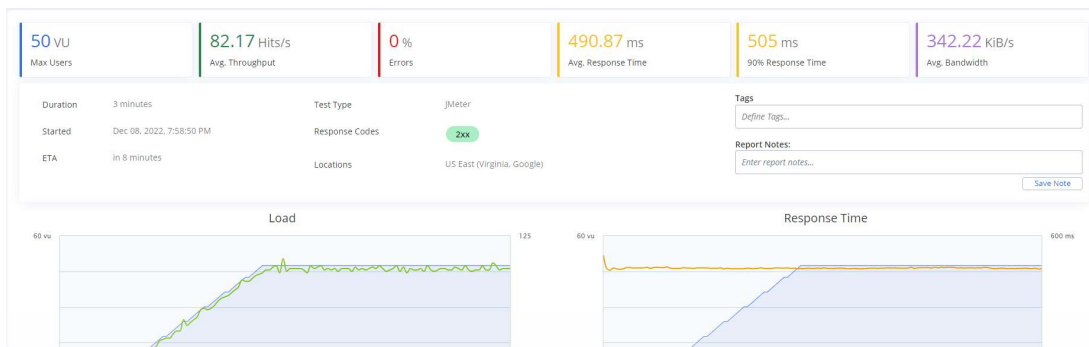
เพื่อทดสอบการแบ่งภาระงานของการเข้าใช้งาน Web Service โดยที่มีการกระจายผู้ใช้งานเข้าไปยังเครื่องอื่น ๆ บนระบบ Cluster เพื่อกระจายทรัพยากรในการเข้าใช้งานไปให้มีประสิทธิภาพ

4.11.1.2 วิธีการทดลอง

ทดสอบผ่าน Demo Web Application ที่พัฒนาขึ้นมาสำหรับการทดสอบโดยเฉพาะโดยการนำเครื่องมือในการส่ง Request เข้ามาจำนวน 50 คนพร้อม ๆ กันในเวลา 10 นาทีเพื่อตรวจสอบว่ายังสามารถเข้าใช้งานได้ สามารถแบ่งปันจัดสรรทรัพยากรได้คุ้มค่าเพียงพอกับแต่ละผู้ใช้ในการใช้งานในส่วน of Web Application ทั้ง Front-End และ Back-End ของระบบ Web Application ในการเข้ามาใช้งานในหน้าเว็บไซต์

4.11.1.3 ผลการทดลอง

จากการทดลองกระจายภาระงานของ Web Service สามารถรองรับการเข้ามาใช้งานในตัว of เว็บไซต์พร้อมกันได้เฉลี่ย 8 คนต่อ 1 เครื่องใน Cluster โดยที่หลังจาก 8 คนไปประสิทธิภาพจะลดลงเรื่อย ๆ และ มีการตอบสนองของ Server ต่อการเข้ามาใช้งานบริการในหน้าเว็บไซต์ที่ช้าลงจากรูป 4.36



รูป 4.36 ผลการทดลองส่ง Request เข้ามาใช้งาน

4.11.2 การทดลองระบบการแบ่งภาระงานของ Application

4.11.2.1 จุดประสงค์การทดลอง

เพื่อทดสอบการแบ่งภาระงานของการตั้งงานประมวลผล Application โดยที่มีการกระจายงานไปยังเครื่องอื่น ๆ บนระบบ Cluster เพื่อกระจายทรัพยากรในการเข้าใช้งานไปให้มีประสิทธิภาพมากขึ้นในการประมวลผล และเปรียบเทียบการแบ่งภาระงานระหว่างการประมวลผลด้วยเครื่องเดียวและการประมวลผลด้วยระบบคลัสเตอร์

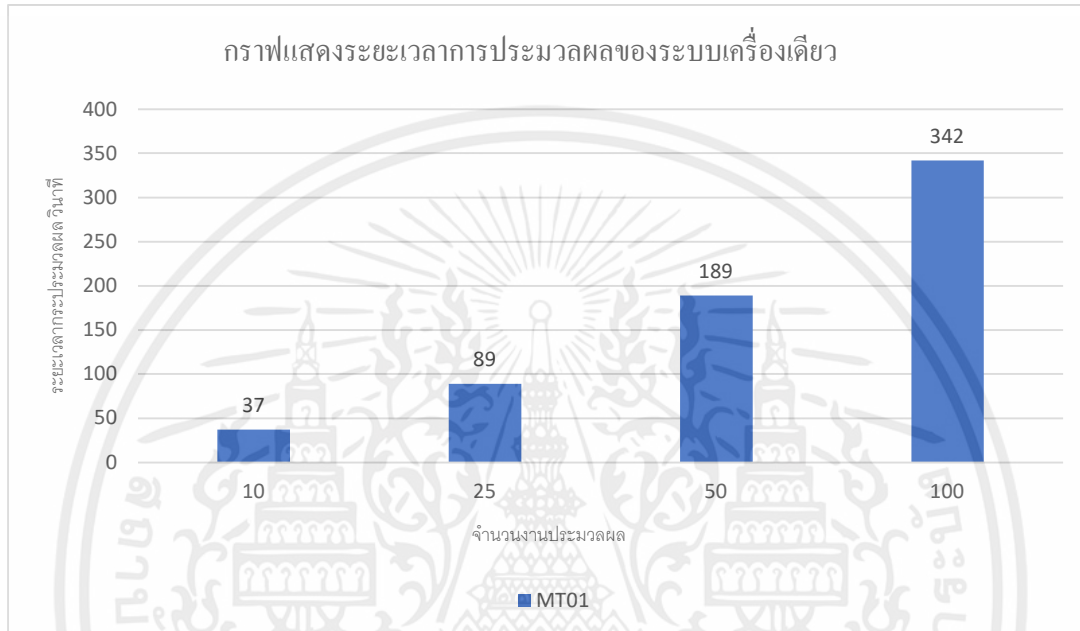
4.11.2.2 วิธีการทดลอง

ทดสอบผ่าน Demo Application ที่พัฒนาขึ้นมาสำหรับการทดสอบโดยเฉพาะ โดยให้ตัว Demo Application นั้นตั้งงานประมวลผลภาพโดยกำหนดให้ใช้รูปภาพชุดเดียวกันในการประมวลผลและประมวลผลด้วย Image processing application เดียวกัน เป็นจำนวนตั้งแต่ 10, 25, 50, และ 100 งาน ซึ่งจะทำกรจับเวลาในการประมวลผลแต่ละจำนวนในแต่ละรูปแบบการประมวลผล โดยจะเปรียบเทียบระหว่างการประมวลผลด้วยเครื่องเดียว ที่ใช้เครื่องคอมพิวเตอร์ชื่อ MT01 โดยมีสเปคคือ CPU i9-9900K, RAM 32 GB, GPU RTX 2060 6 GB และการประมวลผลด้วยระบบคลัสเตอร์ ที่ใช้เครื่องคอมพิวเตอร์ชื่อ MT01 โดยมีสเปคคือ CPU i9-9900K, RAM 32 GB, GPU RTX 2060 6 GB เป็น Node หลัก และ เครื่องคอมพิวเตอร์ชื่อ WK01 โดยมีสเปคคือ CPU i5-8300H, RAM 16 GB, GPU GTX 1050 4 GB เป็น Node ย่อย

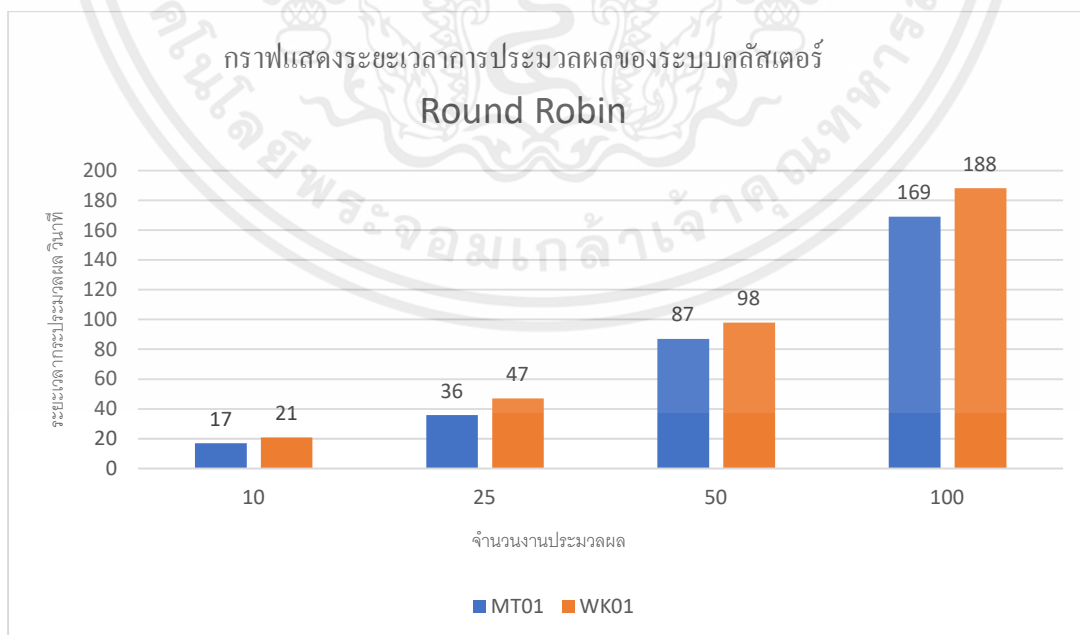
4.11.2.3 ผลการทดลอง

จากการทดลองเมื่อมีการตั้งงานประมวลผลแบบเครื่องเดียวนั้นยังมีปริมาณงานมากขึ้นระยะเวลาที่ใช้งานก็จะสูงขึ้นตาม ซึ่งถ้าเทียบกับการประมวลผลด้วยระบบคลัสเตอร์ ระยะเวลาที่ใช้งานการประมวลผลนั้นใช้ระยะเวลาที่ลดลงเฉลี่ยถึงร้อยละ 40 เมื่อเปรียบเทียบกับ การประมวลผลด้วยเครื่องเดียว แต่ทั้งนี้ระยะเวลาที่ได้มานั้นยังไม่แน่นอน เพราะเมื่อมีงานประมวลผลเข้ามาจะมีการจัดการกระจายงานประมวลผลไปที่ Node แต่ละเครื่อง โดยที่แต่ละเครื่องนั้นจะมีทรัพยากรไม่เท่ากัน และ อาจ

ใช้เวลาในการประมวลผลไม่เท่ากันแต่ยังคงความสามารถในการกระจายงานประมวลผล และ เนื่องจากยังมีระยะของการจัดการงาน และ ระยะเวลาในการตรวจสอบทรัพยากร จึงทำให้มีระยะเวลาที่ทำให้ไม่คุ้มค่ากับการประมวลผลงานในจำนวนที่น้อย แต่เมื่อมีงานจำนวนมากทำให้คุ้มค่ากับระยะเวลา ซึ่งผลการทดลองแบบ Round Robin และ Least Connection สามารถดูได้จากรูปภาพที่ 4.37, 4.38 และ 4.39

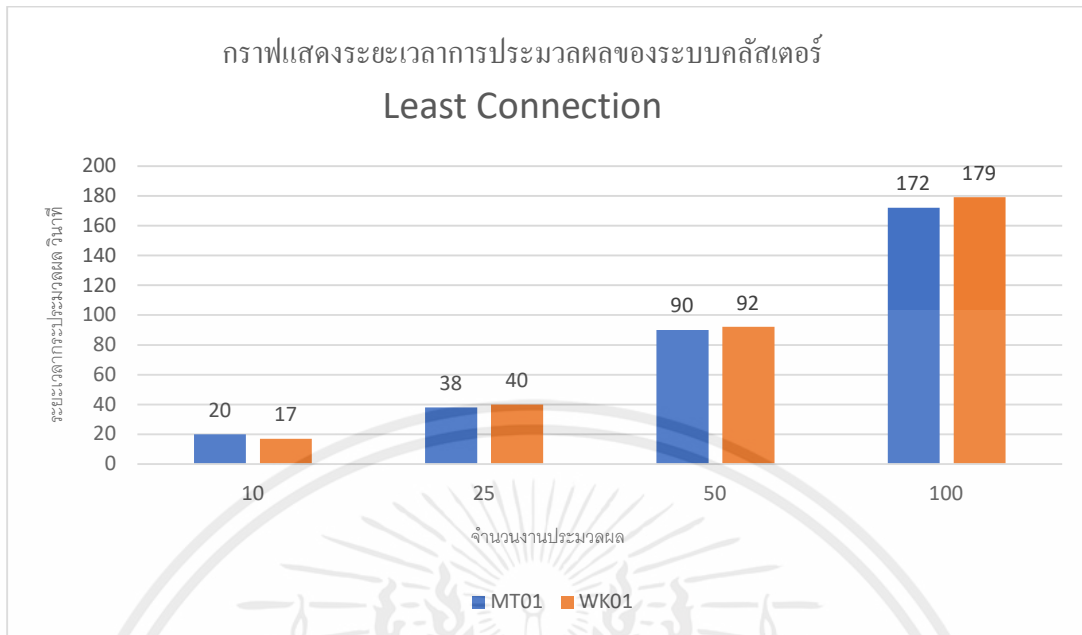


รูป 4.37 กราฟของผลการทดลองเมื่อสั่งงานประมวลผลด้วยการประมวลผลด้วยเครื่องเดียว



รูป 4.38 กราฟของผลการทดลองเมื่อสั่งงานประมวลผลด้วยการประมวลผลด้วยระบบคลัสเตอร์ด้วย Round Robin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.39 กราฟของผลการทดลองเมื่อสั่งงานประมวลผลด้วยการประมวลผลด้วยระบบคลัสเตอร์ด้วย Least Connection

4.11.2.4 เป้าหมายการทดลอง

จากการทดลองเพื่อทดสอบว่าระบบในการประมวลผลแบบ Cluter นั้นมีการประมวลผลที่ใช้ระยะเวลาโดยรวมน้อยกว่ารูปแบบ Single ในกรณีทำงานปริมาณเท่า ๆ กันเพื่อเป็นข้อสรุปของเป้าหมายตัวโครงการนี้ในการช่วยลดระยะเวลาในงานประมวลผล ซึ่งจากการเปรียบเทียบการระยะเวลาการประมวลผลของระบบคลัสเตอร์แบบ Round Robin และ แบบ Least Connection ทำให้ทราบได้ว่าการประมวลผลของระบบคลัสเตอร์แบบ Least Connection นั้นมีระยะเวลาที่ใช้ในการประมวลผลที่ได้ดีกว่าซึ่งสามารถวิเคราะห์ได้ว่าในกรณีเครื่องคอมพิวเตอร์ที่อยู่ในระบบคลัสเตอร์นั้นมีทรัพยากรที่แตกต่างกันใช้การประมวลผลของระบบคลัสเตอร์แบบ Least Connection นั้นน่าจะทำได้ดีกว่าเพราะด้วยการที่ทรัพยากรในการประมวลผลของแต่ละเครื่องนั้นมีไม่เท่ากันจึงทำให้การประมวลผลแต่ละงานในแต่ละเครื่องนั้นสำเร็จช้า-เร็วต่างกันออกไป ซึ่งการที่การประมวลผลของระบบคลัสเตอร์แบบ Least Connection จะพิจารณาการแจกจ่ายงานประมวลผลด้วยจำนวนของ Connection และจำนวน Queue ในการเชื่อมต่อเข้าใช้งาน ซึ่งทำให้งานประมวลผลส่วนใหญ่จะโดนแจกจ่ายไปที่เครื่องที่มีทรัพยากรสูง ซึ่งสามารถประมวลผลได้ไวและมีจำนวนของ Connection และจำนวน Queue ในการเชื่อมต่อเข้าใช้งานน้อยกว่า ทำการประมวลผลงานนั้นถูกจัดแจงการใช้ทรัพยากรได้ดีกว่า เทียบกับการประมวลผลของระบบคลัสเตอร์แบบ Round Robin ที่พิจารณาการแจกจ่ายงานประมวลผลด้วยลำดับของเครื่องโดยไม่สนจำนวนของ Connection และจำนวน Queue ในการเชื่อมต่อเข้าใช้งาน ซึ่งทำให้งานประมวลผลนั้นถูกแบ่งไปยังเครื่องที่มีทรัพยากรสูงและต่ำในปริมาณที่เท่ากัน จึงทำให้ระยะเวลาที่ใช้ในการประมวลผลนั้นเพิ่มขึ้นมาจากจำนวนงานประมวลผลที่ถูกแบ่งไปยังเครื่องที่มีทรัพยากรต่ำและประมวลผลได้ช้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.12 ส่วนของ Application Image Processing

4.12.1 การทดลองระบบการทำงานของ Application Image Processing

4.12.1.1 จุดประสงค์การทดลอง

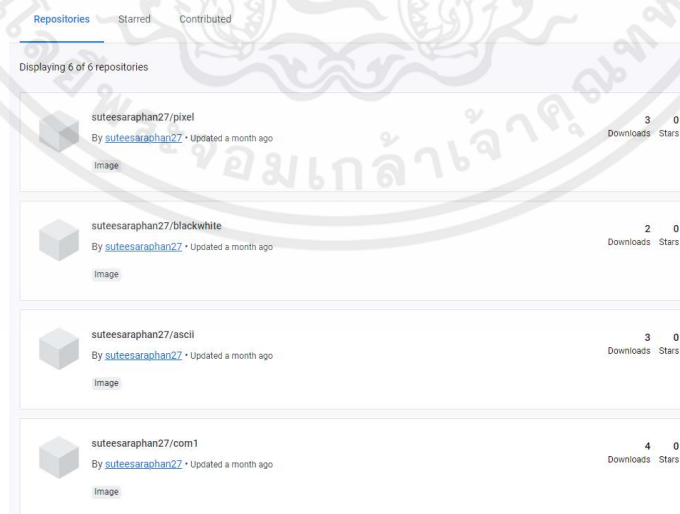
เพื่อทดสอบว่าระบบในการสั่งงานประมวลผลบนรูปแบบ Container ที่อยู่บน DockerHub ที่เราได้ทำการ Build และ Push ขึ้นไว้สำหรับเรียกใช้งานเพื่อนำมาประมวลผลงานตามคำสั่งเรียกใช้งานที่เราได้เขียนไว้บน Kubernetes Cluster และสามารถประมวลผลได้อย่างถูกต้อง

4.12.1.2 วิธีการทดลอง

ทดสอบการสั่งงานด้วย Kubernetes ในการจัดเก็บ Application ไว้บน DockerHub และทดสอบด้วยรูปภาพจำนวนหนึ่งที่เก็บไว้ภายในเครื่อง Cluster สามารถถูกประมวลผลได้อย่างถูกต้องและครบถ้วน โดยทำการเปรียบเทียบรูปภาพที่ผ่านการประมวลผลโดย Application ที่จัดเก็บไว้บน DockerHub และ Application ที่ทำการพัฒนาบนเครื่อง Local ซึ่งรูปผลลัพธ์ที่ได้รับจากทั้ง 2 แห่งต้องมีความเหมือนกัน

4.12.1.3 ผลการทดลอง

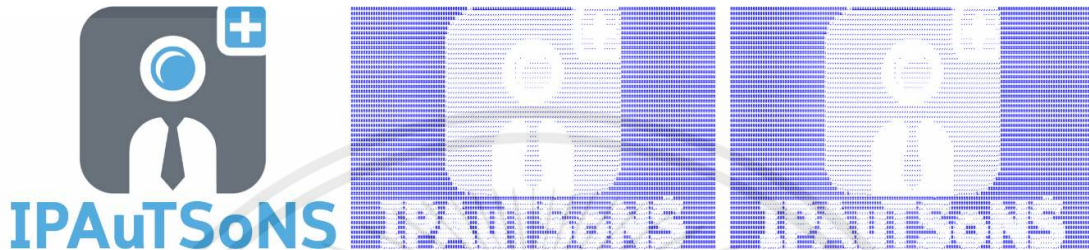
จากการทดลองสามารถประมวลผลภาพได้ถูกต้องบนระบบ Cluster ด้วย Kubernetes ซึ่งผลลัพธ์รูปภาพที่ได้ผ่านการประมวลผลด้วย Application ที่เก็บไว้บน DockerHub นั้น เหมือนกับ รูปภาพที่ได้รับจากการประมวลผลด้วย Application ที่พัฒนาบนเครื่อง Local โดยไม่มีการทำงานผิดพลาดใดๆ แต่มีเวลาในการโหลดครั้งแรกของการสั่งงานที่ค่อนข้างนานเนื่องจากต้องไปทำการโหลด Image ของ Application มาจาก DockerHub ก่อนที่จะเริ่มทำงานประมวลผลได้ โดยรูปภาพของ Repositories ของ DockerHub ที่ทำการเก็บ Application จะอยู่ในรูปภาพที่ 4.40



รูป 4.40 Repositories ของ DockerHub ที่ทำการเก็บ Application ไว้

4.12.1.3.1 ผลการทดลอง Application ASCII

จากการทดลองด้วย Application ASCII โดยการทำงานอยู่บน Docker Image ด้วยการประมวลผลภาพเป็นรูปแบบ ASCII สามารถทำงานได้ปกติกับการทำงานอยู่บนระบบ Local โดยจะมีการประมวลผลสำเร็จแล้วดังรูปภาพที่ 4.41



รูป 4.41 (ซ้าย)รูปภาพต้นฉบับ (กลาง)รูปภาพที่ผ่านการประมวลผลด้วย Application ASCII บน Local (ขวา)รูปภาพที่ผ่านการประมวลผลด้วย Application ASCII ใ้บน DockerHub

4.12.1.3.2 ผลการทดลอง Application BlackWhite

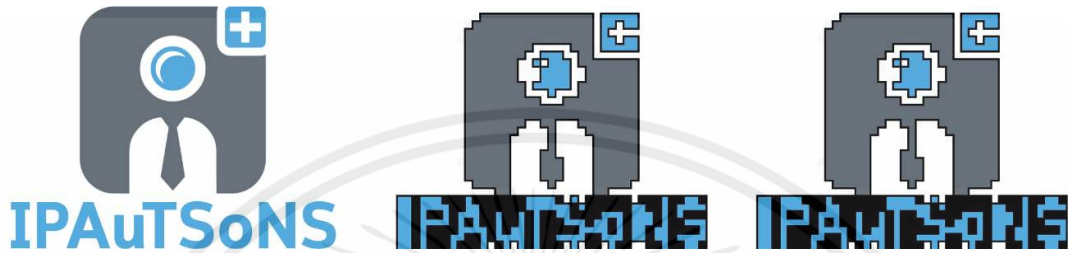
จากการทดลองด้วย Application BlackWhite โดยการทำงานอยู่บน Docker Image ด้วยการประมวลผลภาพเป็นรูปแบบ BlackWhite สามารถทำงานได้ปกติกับการทำงานอยู่บนระบบ Local โดยจะมีการประมวลผลสำเร็จแล้วดังรูปภาพที่ 4.42



รูป 4.42 (ซ้าย)รูปภาพต้นฉบับ (กลาง)รูปภาพที่ผ่านการประมวลผลด้วย Application BlackWhite บน Local (ขวา)รูปภาพที่ผ่านการประมวลผลด้วย Application BlackWhite ใ้บน DockerHub

4.12.1.3.3 ผลการทดลอง Application PixelArt

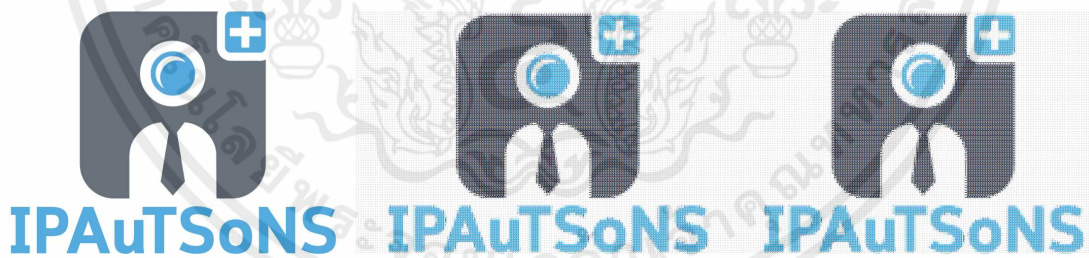
จากการทดลองด้วย Application PixelArt โดยการทำงานอยู่บน Docker Image ด้วยการประมวลผลภาพเป็นรูปแบบ PixelArt สามารถทำงานได้ปกติกับการทำงานอยู่บนระบบ Local โดยจะมีการประมวลผลสำเร็จแล้วดังรูปภาพที่ 4.43



รูป 4.43 (ซ้าย)รูปภาพต้นฉบับ (กลาง)รูปภาพที่ผ่านการประมวลผลด้วย Application PixelArt บน Local (ขวา)รูปภาพที่ผ่านการประมวลผลด้วย Application PixelArt ไว้บน DockerHub

4.12.1.3.4 ผลการทดลอง Application Mosaic

จากการทดลองด้วย Application Mosaic โดยการทำงานอยู่บน Docker Image ด้วยการประมวลผลภาพเป็นรูปแบบ Mosaic สามารถทำงานได้ปกติกับการทำงานอยู่บนระบบ Local โดยจะมีการประมวลผลสำเร็จแล้วดังรูปภาพที่ 4.44



รูป 4.44 (ซ้าย)รูปภาพต้นฉบับ (กลาง)รูปภาพที่ผ่านการประมวลผลด้วย Application Mosaic บน Local (ขวา)รูปภาพที่ผ่านการประมวลผลด้วย Application Mosaic ไว้บน DockerHub

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผลที่จากโครงการ

โครงการร้านค้าสำหรับแอปพลิเคชันการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผลแบบกลุ่ม มีส่วนการทำงานหลักๆอยู่ 3 ส่วนคือ ส่วนเว็บแอปพลิเคชัน (Web Service) ส่วนของแอปพลิเคชัน (Application Image Processing) และ ส่วนของเซิร์ฟเวอร์คลัสเตอร์คอมพิวเตอร์ (Server Cluster) มีฟังก์ชันที่พัฒนาแล้วเสร็จตามแผนการพัฒนา ซึ่งได้มีการพัฒนาและ Deploy โพรเจก ที่สามารถให้ทดลองใช้งานได้แล้วดังนี้

5.1.1 ส่วนของเว็บแอปพลิเคชัน (Web Service)

5.1.1.1 ส่วนของหน้าลงทะเบียนสมัครเข้าใช้งาน

เป็นส่วนต่อประสานกับผู้ใช้ระดับ Client สำหรับทำการลงทะเบียนสมัครเข้าใช้งาน Web application โดยต้องกรอก Email, รหัสผ่าน, ชื่อ และ นามสกุล เพื่อเป็นข้อมูลในการสมัครใช้งาน

5.1.1.2 ส่วนของหน้าลงทะเบียนเข้าสู่ระบบใช้งาน

เป็นส่วนต่อประสานกับผู้ใช้ระดับ Client สำหรับทำการลงทะเบียนเข้าสู่ระบบเพื่อใช้งาน Web application โดยต้องกรอก Email, รหัสผ่าน เพื่อเป็นข้อมูลในการเข้าสู่ระบบเพื่อใช้งาน

5.1.1.3 ส่วนของหน้าแก้ไขข้อมูลผู้ใช้งาน

เป็นส่วนต่อประสานกับผู้ใช้สำหรับการแก้ไขข้อมูลของผู้ใช้เอง รวมไปถึงการแก้ไขรหัสผ่านด้วย

5.1.1.4 ส่วนของหน้าจัดการไฟล์

เป็นส่วนต่อประสานกับผู้ใช้สำหรับการจัดการ Folder สำหรับเก็บไฟล์รูป และ ตัวไฟล์รูปด้วย เพื่อให้สามารถนำรูปภาพไปใช้งานต่อในการสั่งงานประมวลผลภาพ

5.1.1.5 ส่วนของแอปพลิเคชันประมวลผลภาพ

เป็นส่วนต่อประสานกับผู้ใช้สำหรับการสั่งงานประมวลผลภาพ โดยจะต้องกรอกข้อมูลรายละเอียดของงานเข้าไปเพื่อทำการสร้างไฟล์ YAML ไปสั่งงานประมวลผลภาพ

5.1.1.6 ส่วนของหน้าข่าวสาร

เป็นส่วนต่อประสานกับผู้ใช้สำหรับแสดงผลสินค้าที่มีการเปลี่ยนแปลงข้อมูลล่าสุดจำนวนหนึ่งเพื่อให้ผู้ใช้ได้ทราบถึงข่าวสารล่าสุดของสินค้า

5.1.1.7 ส่วนของหน้าคูปันที่การใช้งานของผู้ใช้

เป็นส่วนต่อประสานกับผู้ใช้สำหรับแสดงบันทึกรายการการใช้หรือได้รับเครดิตของผู้ใช้ว่าถูกใช้ไปกับสินค้าใดหรือบริการใด

5.1.1.8 ส่วนของหน้าคูปันที่การขายของสินค้า

เป็นส่วนต่อประสานกับผู้ใช้สำหรับแสดงบันทึกรายการการนำสินค้าไปใช้งานสั่งงานประมวลผลเมื่อใดและจำนวนเท่าไร เป็นเครดิตให้กับเจ้าของสินค้าเท่าไร

5.1.1.9 ส่วนของหน้าร้านค้าแสดงสินค้า

เป็นส่วนต่อประสานกับผู้ใช้สำหรับจัดแสดงสินค้าให้ผู้ใช้ได้เข้ามาเลือกดูและรับชม รวมไปถึงจัดแสดงสินค้าของตนเองได้

5.1.1.10 ส่วนของหน้าแสดงข้อมูลเฉพาะสินค้า

เป็นส่วนต่อประสานกับผู้ใช้สำหรับทำแสดงข้อมูลเฉพาะสินค้านั้นๆ โดยในแสดงข้อมูลจะมีรูปภาพตัวอย่างของสินค้า รายละเอียดการใช้งาน ความสามารถ ประเภทของสินค้า และ ราคา ระบุอยู่ในภายใน

5.1.2 ส่วนแอปพลิเคชัน (Application Image Processing)

5.1.2.1 แอปพลิเคชันประมวลผลภาพเป็นขาวดำ (Black White Application)

การแปลงภาพสี RGB ที่มีอยู่ 3 Channel สีให้ออกมาเป็น 1 Channel สีจึงเกิดเป็น ภาพสีขาว ดำ ขึ้นมาโดยใช้ตัวของ Python PIL ในการทำงาน

<https://hub.docker.com/repository/docker/suteesaraphan27/blackwhite>

5.1.2.2 แอปพลิเคชันประมวลผลภาพเป็นอักษรแอสกี (ASCII Application)

ในส่วนของ ASCII Image คือ การแปลงรูปภาพให้เป็น Pixel อยู่ในรูปของ Array และ นำตัวอักษร ASCII เช่น “.:irs?@9B&#” มารวมแทนที่แต่ละ Pixel นั้นตามค่าตัวอักษรที่เราได้เรียงมาเป็นในรูปแบบของ Array List โดยสีของรูปภาพ ASCII นั้นจะขึ้นอยู่กับความเข้มของสีในภาพเดิมก่อนนำมาทำเป็น ASCII Image ยิ่งภาพมีสีเข้มตัวของ ASCII นั้นจะมีสีเข้มตามไล่ระดับจาก สีขาว สีเทา สีน้ำเงิน และ สีดำ

<https://hub.docker.com/repository/docker/suteesaraphan27/ascii>

5.1.2.3 แอปพลิเคชันประมวลผลภาพเป็นโมเสก (Mosaic Application)

การนำรูปภาพ 1 ภาพมาตัดแบ่งเป็น Tiles โดยกำหนดขนาดของ Tiles ออกมาได้ และ ยิ่งขนาดเล็กเท่าไรความละเอียดของ Tiles ก็จะมีสูงขึ้น โดยเมื่อแบ่ง Tiles เสร็จเราจึงนำรูปภาพอื่น ๆ จำนวนมากที่เราเตรียมไว้มาทำการ Feature Map กับตัว Tiles เพื่อทำให้ Tiles นั้น ๆ เกิดออกมาเป็นภาพอื่น ๆ ที่

เราได้เตรียมไว้และออกมาเหมือนกับภาพแบบ Mosaic

<https://hub.docker.com/repository/docker/suteesaraphan27/mosaic>

5.1.2.4 แอปพลิเคชันประมวลผลภาพเป็นพิกเซลศิลปะ (PixelArt Application)

การแปลงภาพปกติให้ออกมาเป็น Pixel ให้สวยงามโดยใช้การกำหนดค่า Kenel_size, Pixel_size และ Edge_thresh โดยเมื่อกำหนดค่าทั้ง 3 แล้วจึงนำภาพมาประมวลผลให้อยู่ในรูปแบบ tensor ก่อนแล้วจึงนำมาคำนวณด้วยการเติมค่าทั้ง 3 ลงไปใน tensor ที่ได้แปลงจากภาพมาแล้วจึงนำมาแปลงค่า tensor กลับมาเป็นภาพ

<https://hub.docker.com/repository/docker/suteesaraphan27/pixel>

5.1.3 ส่วนของเซิร์ฟเวอร์คลัสเตอร์คอมพิวเตอร์ (Server Cluster)

5.1.3.1 ในส่วนของ K3S

ได้จัดการติดตั้ง K3S เพื่อทำการจำลองระบบ Cluster ขึ้นมาในรูปแบบของ Container เพื่อทดลองใช้งาน และ พัฒนาตัวโครงการนี้ขึ้น

5.1.3.2 ในส่วนของ K3D

ได้ทำการแก้ไขตัวของ K3D เพื่อให้รองรับการทำงานของ GPU และ ได้ปรับปรุงใหม่ให้ตัว Kubernetes มีการรองรับการทำงานของ CUDA ขึ้นมาด้วย

5.1.3.3 ในส่วนของ Monitor

ได้ทำการติดตั้ง LENS Kubernetes สำหรับการ Monitoring ตัว Cluster เพื่อตรวจสอบการทำงาน และการแจ้งเตือนต่าง ๆ ได้ง่ายขึ้น และได้ติดตั้ง Prometheus Grafana เพื่อทำการตรวจเช็คการทำงานของ CPU RAM GPU ในรูปแบบ Metric เพื่อเก็บสถิติ และ นำไปใช้งานต่อไปได้

5.1.3.4 ในส่วนของ Load Balancer

ได้ทำการตั้งค่า และ ติดตั้ง Nginx ด้วย Ingress สำหรับการ Exposing Services ออกไปสู่ภายนอก และ มีการ Load Balancing ในการเข้าใช้งาน Web Service ในส่วนของการ Load Balancer งานประมวลผลได้ใช้อัลกอริทึมเป็น Round Robin

5.1.3.5 ในส่วนของฐานข้อมูล (Database)

ได้ทำการติดตั้ง Database เป็น MongoDB และ ได้จัดการตั้งค่าต่าง ๆ ของ Server ให้ใช้งานจัดเก็บข้อมูลไว้นอก Cluster

5.1.3.6 ในส่วนของ Persistent Volume

ได้จัดการในส่วนของ Persistent Volume ขึ้นมาเพื่อให้ Cluster System มอง Volume จากภายนอก และ เข้าใช้งานได้ทุกเครื่องบน Cluster เพื่อกันการที่ข้อมูลเสียหายภายใน Cluster จึงได้ทำการเก็บข้อมูลไว้ในส่วนนอกของ Cluster System

5.2 ปัญหาและอุปสรรค

- 1) การทำระบบ Cluster นั้นจะต้องมีเครื่องคอมพิวเตอร์อย่างน้อย 2 เครื่องขึ้นไปจึงมีปัญหในการหาเครื่องคอมพิวเตอร์มาใช้ทำโครงการนี้
- 2) เนื่องจากการทดลองทดสอบระบบส่วนใหญ่ยังใช้การจำลองเพียงเครื่องเดียวจึงทำให้ได้ผลสรุปมาก่อนข้างไม่ตรงกับความต้องการที่จะทดลอง
- 3) การพัฒนา Web application ที่ต้องใช้ Library เสริมเข้ามาช่วยในการพัฒนานั้น เกิดปัญหาการ Conflict กันเองระหว่าง Library ที่เมื่อมี Library ใหม่ถูกติดตั้งเพิ่มเข้ามาก็อาจจะทำให้เกิดปัญหากับ Library เดิมที่มีอยู่
- 4) ระบบ Network ที่ใช้ในการพัฒนานั้นเป็น Network ของทางสถาบันที่มีการตั้งข้อจำกัดด้านความปลอดภัยเอาไว้ ทำให้ไม่สามารถใช้งานแบบ Public ได้ จึงจำเป็นต้องใช้การจำลอง Private Network ในการทดสอบแทนการใช้ Public Network

5.3 แนวทางการพัฒนาต่อไปในอนาคต

- 1) พัฒนาส่วนของการแบ่งภาระงานประมวลผลให้มีประสิทธิภาพมากยิ่งขึ้น
- 2) พัฒนา Web Application ให้ใช้งานได้ปลอดภัยในการส่งข้อมูลไปยัง Database และ Cluster โดยที่ยังคงหรือเพิ่มความรวดเร็ว และ ลดจำนวนครั้งที่ส่ง-รับข้อมูลลงไป
- 3) พัฒนาแก้ไขในส่วนของ User interface ให้มีความสวยงาม และ เหมาะสมกับการใช้งานมากขึ้น
- 4) พัฒนาตัวจัดการ Weight Model ในการใช้งานบน Application
- 5) พัฒนาระบบเติมเงินซื้อเครดิตเข้ามาใช้จริง
- 6) แก้ไข-ปรับปรุงในส่วนของระบบเครดิต เพื่อให้การใช้งานสั่งงานประมวลผลนั้นคุ้มค่ากับ Client และผู้ให้บริการ

บรรณานุกรม

Salem Saleh Al-amri. 2011. **Contrast Stretching Enhancement in Remote Sensing Image.**

Department of Computer Science

North Maharashtra University, Jalgaon, India.

Najia Naz, Abdul Haseeb Malik, Abu Bakar Khurshid, Furqan Aziz, Bader Alouffi, M. Irfan Uddin,

Ahmed AlGhamdi. 2020. **Efficient Processing of Image Processing Applications on CPU/GPU.** [Online].

Available : <https://www.hindawi.com/journals/mpe/2020/4839876> .

Yuqi Fu, Shaolun Zhang, Jose Terrero, Ying Mao, Guangya Liu, Sheng Li, Dingwen Tao. 2019.

Progress-based Container Scheduling for Short-lived Applications in a Kubernetes Cluster. [Online].

Available : <https://ieeexplore.ieee.org/abstract/document/9006427> .

Jonas Wagner. 2015. **Javascript Film Emulation.** [Online].

Available : <https://29a.ch/2015/06/07/javascript-film-emulation> .

Wongsathorn Insawasd. 2020. **Vue.js คืออะไร มีดีอย่างไร ทำไมถึงต้องใช้.** [Online].

Available : <https://insawasd.com/what-is-vuejs/> .

ไม่ปรากฏชื่อผู้เขียน. 2021. **Docker (software).** [Online].

Available : [https://en.wikipedia.org/wiki/Docker_\(software\)](https://en.wikipedia.org/wiki/Docker_(software)) .

ภาคผนวก ก

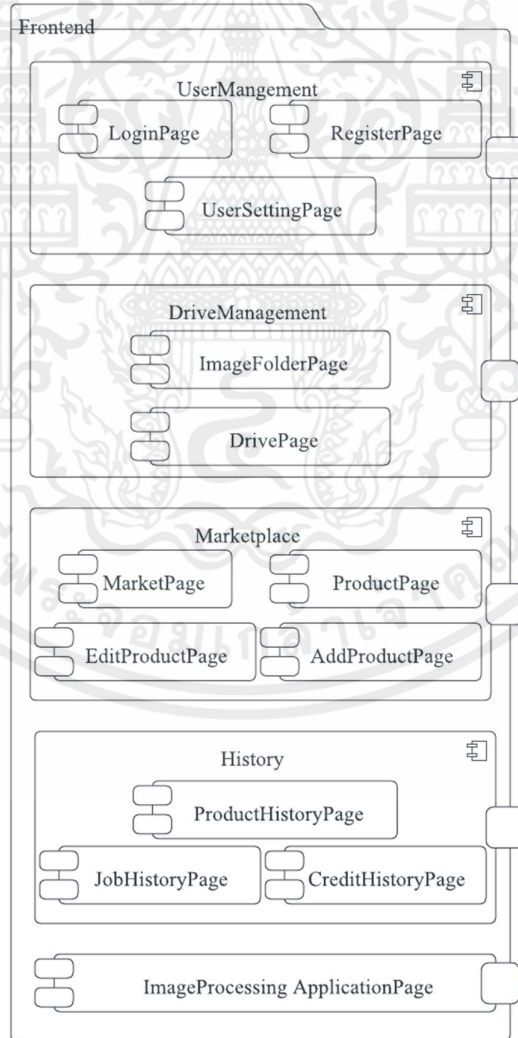
รายละเอียดของ diagram

ก.1 Component diagram(Detail)

แผนภาพองค์ประกอบหรือ Component diagram จะถูกแบ่งออกเป็น 4 ส่วนด้วยกันคือ

ก.1.1 ส่วน Frontend

รูปที่ ก.1 จะแสดงแผนภาพองค์ประกอบส่วน Frontend ซึ่งมีหน้าที่หลักในการต่อประสานกับผู้ใช้ด้วยหน้าจอ GUI ที่ได้ออกแบบเอาไว้ โดยให้ผู้ใช้สามารถป้อนค่าแล้วส่งผ่านไปให้ Backend เพื่อประมวลผลและส่งกลับมาเพื่อแสดงผล



รูป ก.1 แผนภาพองค์ประกอบส่วน Frontend

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสามารถแบ่งย่อยแต่ละส่วนได้ลงไปอีกตามตาราง ก.1 - ก.5

ตาราง ก.1 UserManagement component

Component name	UserManagement
Sub-component	LoginPage, RegisterPage, UserSettingPage
Role	แสดงผลในส่วนของหน้าการจัดการข้อมูลผู้ใช้เป็นหลัก
Required Interface	Backend(user_authentication), Backend(user_management)

ตาราง ก.2 DriveManagement component

Component name	DriveManagement
Sub-component	ImageFolderPage, DrivePage
Role	แสดงผลในส่วนของหน้าการจัดการรูปภาพที่ผู้ใช้ได้ทำการ Upload ขึ้นมาบน Web application
Required Interface	Backend(user_authentication), Backend(drive_management)

ตาราง ก.3 Marketplace component

Component name	Marketplace
Sub-component	MarketPage, ProductPage, EditProductPage, AddProductpage
Role	แสดงผลในส่วนของหน้าตลาดการซื้อขายและเปิดใช้งานสินค้า
Required Interface	Backend(user_authentication), Backend(product_management), Backend(payment_management)

ตาราง ก.4 History component

Component name	History
Sub-component	ProductHistoryPage, JobHistoryPage, CreditHistoryPage
Role	แสดงผลในส่วนของหน้าการดูประวัติการทำรายการ ต่างๆ
Required Interface	Backend(user_authentication), Backend(product_management), Backend(image_processing_application), Backend(payment_management)

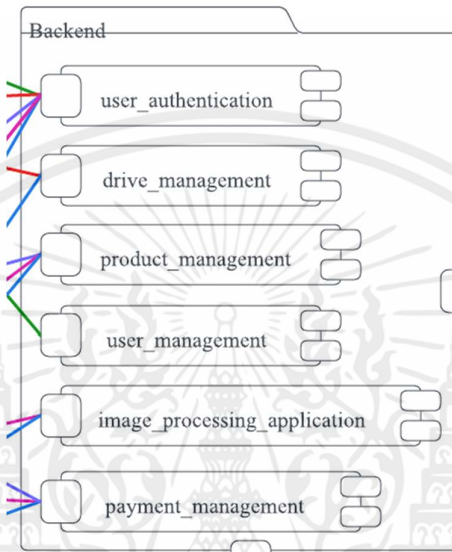
ตาราง ก.5 ImageProcessingApplicationPage component

Component name	ImageProcessingApplicationPage
Sub-component	-
Role	แสดงผลในส่วนของหน้าการใช้งาน Image processing application เช่น การสั่งงานประมวลผล เป็นต้น
Required Interface	Backend(user_authentication), Backend(drive_management), Backend(product_management), Backend(product_management), Backend(payment_management),

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.1.2 ส่วน Backend

รูปที่ ก.2 จะแสดงแผนภาพองค์ประกอบส่วน Frontend ซึ่งมีหน้าที่หลักในการต่อประสานกับผู้ใช้ด้วยหน้าจอ GUI ที่ได้ออกแบบเอาไว้ โดยให้ผู้ใช้สามารถป้อนค่าแล้วส่งผ่านไปให้ Backend เพื่อประมวลผลและส่งกลับมาเพื่อแสดงผล



รูป ก.2 แผนภาพองค์ประกอบส่วน Backend

โดยสามารถแบ่งย่อยแต่ละส่วนได้ลงไปอีกตามตาราง ก.6 - ก.11

ตาราง ก.6 user_authentication component

Component name	user_authentication
Sub-component	-
Role	ทำงานในส่วนของการยืนยันตัวผู้ใช้
Required Interface	Storage(Database)

ตาราง ก.7 drive_management component

Component name	drive_management
Sub-component	-
Role	ทำงานในส่วนของการทำงานจัดการ Drive เก็บรูปภาพ เช่น เพิ่ม, ลบ, คำนวณโหลต เป็นต้น
Required Interface	Storage(Database), Storage(File storage),

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง ก.8 product_management component

Component name	product_management
Sub-component	-
Role	ทำงานในส่วนของการทำงานจัดการตลาดซื้อขายและสินค้าเช่น เพิ่ม, ลบ, แก้ไข เป็นต้น
Required Interface	Storage(Database), Storage(File storage),

ตาราง ก.9 user_management component

Component name	user_management
Sub-component	-
Role	ทำงานในส่วนของการทำงานจัดการผู้ใช้เช่น เพิ่มผู้ใช้, แก้ไขข้อมูล เป็นต้น
Required Interface	Storage(Database)

ตาราง ก.10 image_processing_application component

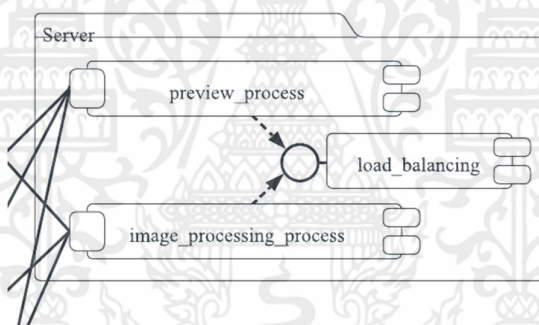
Component name	image_processing_application
Sub-component	-
Role	ทำงานในส่วนของการสั่งงานประมวลผลภาพและ preview รูปภาพก่อนประมวลผลจริง
Required Interface	Storage(Database), Storage(File storage), Server(preview_process), Server(image_processing_process), Server(load_balancing),

ตาราง ก.11 payment_management component

Component name	payment_management
Sub-component	-
Role	ทำงานในส่วนของการทำงานจัดการเครดิตของผู้ใช้ เช่น การบันทึกรายการการเปลี่ยนแปลงเครดิต เป็นต้น
Required Interface	Storage(Database)

ก.1.3 ส่วน Server

รูปที่ ก.3 จะแสดงแผนภาพองค์ประกอบส่วน Frontend ซึ่งมีหน้าที่หลักในการต่อประสานกับผู้ใช้ด้วยหน้าจอ GUI ที่ได้ออกแบบเอาไว้ โดยให้ผู้ใช้สามารถป้อนค่าแล้วส่งผ่านไปให้ Backend เพื่อประมวลผลและส่งกลับมาเพื่อแสดงผล



รูป ก.3 แผนภาพองค์ประกอบส่วน Server

โดยสามารถแบ่งย่อยแต่ละส่วนได้ลงไปอีกตามตาราง ก.12 - ก.14

ตาราง ก.12 preview_process component

Component name	preview_process
Sub-component	-
Role	ทำงานในส่วนของการ Preview รูปภาพของผู้ใช้ที่ส่งงานประมวลผลกับสินค้าต่างๆ เพื่อให้ได้รูปผลลัพธ์ที่เร็วที่สุด
Required Interface	Storage(Database), Storage(File storage)

ตาราง ก.13 image_processing_process component

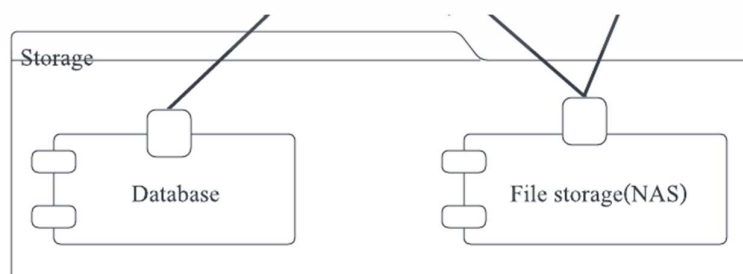
Component name	image_processing_process
Sub-component	-
Role	ทำงานในส่วนของการประมวลผลรูปจากงานที่ผู้ใช้ได้ ส่งประมวลผลเข้ามา โดยจะรองรับงานประมวลผล จำนวนมากๆ
Required Interface	Storage(Database), Storage(File storage)

ตาราง ก.14 load balancing component

Component name	load balancing
Sub-component	-
Role	ทำงานในส่วนของการแบ่งปันจัดสรรทรัพยากรในการ ประมวลผลให้กับส่วนอื่นๆ
Required Interface	-
Provided interfaces	Server(image_processing_process), Server(preview_process)

ก.1.4 ส่วน Storage

รูปที่ ก.4 จะแสดงแผนภาพองค์ประกอบส่วน Frontend ซึ่งมีหน้าที่หลักในการต่อ
ประสานกับผู้ใช้ด้วยหน้าจอ GUI ที่ได้ออกแบบเอาไว้ โดยให้ผู้ใช้สามารถป้อนค่าแล้วส่งผ่าน
ไปให้ Backend เพื่อประมวลผลและส่งกลับมาเพื่อแสดงผล



รูป ก.4 แผนภาพองค์ประกอบส่วน Storage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสามารถแบ่งย่อยแต่ละส่วนได้ลงไปอีกตามตาราง ก.15 และ ก.16

ตาราง ก.15 Database component

Component name	Database
Sub-component	-
Role	เป็นฐานข้อมูลในการเก็บข้อมูลต่างๆ เพื่อนำไปใช้ แสดงผลและประมวลผล
Required Interface	-

ตาราง ก.16 File storage(NAS)component

Component name	File storage(NAS)
Sub-component	-
Role	เป็นส่วนของพื้นที่การเก็บไฟล์รูปภาพและไฟล์ Weight ของผู้ใช้เพื่อนำไปประมวลผลได้
Required Interface	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

Test case

ข.1 Frontend test

Frontend test เป็นการทดสอบหน้า Web application ส่วนแสดงผลหลักว่าสามารถแสดงผลได้ถูกต้องตรงตามที่ได้ออกแบบเอาไว้และสามารถเชื่อมต่อและ ส่ง-รับค่าต่างๆ จากทาง API Backend ได้ โดยไม่เกิดข้อผิดพลาดขึ้น โดยมีทั้งหมด 103 case ที่ใช้ในการทดสอบดังตารางที่ ข.1

ข.1 Frontend testcase

Test ID	Test description
F-001	Register form cannot be null
F-002	Register form password and password confirm check
F-003	Register form call API backend correctly when press regis button
F-004	Register form clear all input values when press clear button
F-005	Register form deny special character in First name and Last name input
F-006	Login form cannot be null
F-007	Login form call API backend correctly when press Login button
F-008	Login page can enter home page when login process is success
F-009	Login page add user token to local storage in web browser when login process is success
F-010	Login page can show pop up window to alert when login process is unsuccess
F-011	Go to Register page when press register button
F-012	Sidebar can show pop up window to alert when user token is not exited and redirection to Login page
F-013	Sidebar display user data correctly
F-014	Sidebar can link to JobHistory page
F-015	Sidebar can link to History page
F-016	Sidebar can link to Home page
F-017	Sidebar can link to Drive page
F-018	Sidebar can link to Setting page

Test ID	Test description
F-019	Sidebar can link to Image processing applications page
F-020	Sidebar can link to Marketplace page
F-021	Sidebar delete user token from local storage in web browser when press logout button
F-022	Home page call API backend and get display data correctly
F-023	Home page show 9 latest product
F-024	Home page can enter Product page when click on product card
F-025	Home page can link to Marketplace page with see more button
F-026	JobHistory page call API backend and get display data correctly
F-027	JobHistory page can display user job order history correctly
F-028	JobHistory page can change display user job order history list to latest record first
F-029	JobHistory page can change display user job order history list to oldest record first
F-030	JobHistory page can change display user job order history list to specific date only
F-031	History page call API backend and get display data correctly
F-032	History page can display user credit earn and pay history correctly
F-033	History page can change display user credit earn and pay history list to latest record first
F-034	History page can change display user credit earn and pay history list to oldest record first
F-035	History page can change display user credit earn and pay history list to specific date only
F-036	Drive page call API backend and get display data correctly
F-037	Drive page can display all image folder that user owned
F-038	Drive page can send new image folder data to API backend
F-039	Drive page can sent delete image folder data to API backend
F-040	Drive page can download chosen image folder in zip file format
F-041	Drive page can enter chosen image folder
F-042	Image folder page can authenticate image folder owner user
F-043	Image folder page can show pop up window to alert when user that authenticate is not image folder owner user
F-044	Image folder page can send upload images data to backend by press upload button
F-045	Image folder page can sent delete image to API backend by press delete button on image

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test ID	Test description
F-046	Image folder page can make web browser download image data from API backend by press download button on image
F-047	Image folder page can select range of image to display by select options image folder page
F-048	Image folder page can show full quality image by press on image card
F-049	Marketplace page can call API backend and get display data correctly
F-050	Marketplace page can display products data correctly
F-051	Marketplace page can go to Add product page by press add product button
F-052	Marketplace page can change display products order to latest product update first
F-053	Marketplace page can change display products order to oldest product update first
F-054	Marketplace page can change display products order to specific date only
F-055	Marketplace page can change display products type filter
F-056	Marketplace page can change display products model filter
F-057	Marketplace page can go to Product page by press on product card
F-058	Product page can call API backend and get display data correctly
F-059	Product page can display product data correctly
F-060	Product page can show pop up window to alert when product is not exited and redirection to Marketplace page
F-061	Product page can show edit this product button and view product use history button when user is authenticating as product owner
F-062	Product page can add product to open product list by press try this product button
F-063	Add product page can call API backend
F-064	Add product page user can filled all form
F-065	Add product page user can upload product image when press choose product image upload button
F-066	Add product page user can upload product weight file when press choose product weight file upload button
F-067	Add product page can check all form are filled when press add product button
F-068	Add product page can show pop up window to alert when add product process is success
F-069	Add product page can show pop up window to alert when add product process is unsuccess

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test ID	Test description
F-070	Edit product data page can call API backend and display product data correctly
F-071	Edit product data page user can filled all form
F-072	Edit product data page user can upload product image when press choose product image upload button
F-073	Edit product data page user can upload product weight file when press choose product weight file upload button
F-074	Edit product data page can show pop up window to alert when edit product data process is success
F-075	Edit product data page can show pop up window to alert when edit product data process is unsuccess
F-076	User setting page can call API backend and get display data correctly
F-077	User setting page can display user data correctly
F-078	User setting page all form must be filled when press submit button
F-079	User setting page user can change only first name and last name of user data
F-080	User setting page can send changed user data to API backend when press submit button
F-081	User setting page can show pop up window to alert when change is not complete
F-082	User setting page can show pop up window to alert when change is complete
F-083	User setting page can go to Change password page when press change password button
F-084	User setting page can clear value in all input form by press clear button
F-085	Change password page display user data correctly
F-086	Change password page can check password and password confirm is match
F-087	Change password page all form must be filled when press submit button
F-088	Change password page can send user old password and new password to API backend when press
F-089	Change password page can show pop up window to alert when change is not complete
F-090	Change password page can show pop up window to alert when change is complete
F-091	Change password page can clear value in all input form by press clear button
F-092	Image processing application page can call API backend and get display data correctly
F-093	Image processing application page can display all image folders of user correctly

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test ID	Test description
F-094	Image processing application page can display list of all images from user chosen image folder when select image folder in image folder list
F-095	Image processing application page can display user chosen image from list of all images from chosen image folder when press image thumbnail
F-096	Image processing application page can display all product that user opened correctly
F-097	Image processing application page user can select product to preview product processing result with chosen image
F-098	Image processing application page user can change parameter of preview product processing result with chosen image
F-099	Image processing application page can display result image of preview product processing result
F-100	Image processing application page user can send order job to API backend when press export button
F-101	Image processing application page can pop up window to alert when credit payment is required
F-102	Image processing application page can pop up window to alert when job is on processing
F-103	Image processing application page can pop up window to alert when job is failed to process

ข.2 Backend test

Backend test เป็นการทดสอบการทำงานเบื้องหลังในส่วนของ Backend ส่วนประมวลผลหลักว่าสามารถประมวลผลและจัดเก็บข้อมูลที่ได้รับจากส่วน Frontend ได้ถูกต้องตรงตามที่ได้ออกแบบเอาไว้ โดยไม่เกิดข้อผิดพลาดขึ้น โดยมีทั้งหมด 51 case ที่ใช้ในการทดสอบดังตารางที่ ข.2

ข.2 Backend testcase

Test ID	Test description
B-001	Backend can start and using at configuration IP address
B-002	Backend can display backend develop version at URL /API/ for checking version
B-003	Backend can add new user data to database
B-004	Backend can send response message back to frontend if able to regis new user
B-005	Backend can send response error message back to frontend if unable to regis new user

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test ID	Test description
B-006	Backend can make and send response with user authenticate token and user data back to frontend if able to login
B-007	Backend can add user login log to database
B-008	Backend can send response error message back to frontend if unable to login
B-009	Backend must authenticate user token in any method when have call from frontend
B-010	Backend can edit user data in database
B-011	Backend can send response message back to frontend if able to edit user data
B-012	Backend can send response error message back to frontend if unable to edit user data
B-013	Backend can edit user password in database
B-014	Backend can send response message back to frontend if able to edit user password
B-015	Backend can send response error message back to frontend if unable to edit user password
B-016	Backend can send response message with user job order history query to frontend
B-017	Backend can send response message with user credit earn and pay history query to frontend
B-018	Backend can send response message with user image folders quarry to frontend
B-019	Backend can add new image folder data to database and create folder in NAS
B-020	Backend can send response message to frontend if able to add new image folder
B-021	Backend can send response error message to frontend if unable to add new image folder
B-022	Backend can add delete image folder data to database and delete folder in NAS
B-023	Backend can send response message to frontend if able to delete image folder
B-024	Backend can send response error message to frontend if unable to delete image folder
B-025	Backend can receive image data and image files to add to database and store in NAS
B-026	Backend can send response message to frontend if able to add new image
B-027	Backend can send response error message to frontend if unable to add new image
B-028	Backend can send response message with compress image data in selected image folder to frontend
B-029	Backend can send response message with full size image data of selected image to frontend
B-030	Backend can delete image data from database and delete image from NAS
B-031	Backend can send response message to frontend if able to delete new image
B-032	Backend can send response error message to frontend if unable to delete new image

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test ID	Test description
B-033	Backend can send response message to frontend if able to download new image
B-034	Backend can send response error message to frontend if unable to download new image
B-035	Backend can send response message with product list query to frontend
B-036	Backend can send response message with product data to frontend
B-037	Backend can send response error message to frontend if unable to query product list
B-038	Backend can send response error message to frontend if unable to find product
B-039	Backend can receive product image file ,weight file and product data to add to database and store in NAS
B-040	Backend can send response message to frontend if able to add new product data
B-041	Backend can send response error message to frontend if unable to add new product data
B-042	Backend can receive product image file ,weight file and product data to edit product data database and store in NAS
B-043	Backend can send response message to frontend if able to edit product data
B-044	Backend can send response error message to frontend if unable to edit product data
B-045	Backend can receive open product data to add to database
B-046	Backend can receive image data and product data for send to API preview and get result image
B-047	Backend can send preview result image to frontend
B-048	Backend can send response message with credit payment require to frontend
B-049	Backend can receive product data, parameter and image folder for order image processing job
B-050	Backend can send response message to frontend if able to order image processing job
B-051	Backend can send response error message to frontend if unable to order image processing job