

ร้านค้าสำหรับแอพพลิเคชันการประมวลผลภาพซึ่งจัดการงานบนระบบ

ประมวลผลแบบกลุ่ม

Marketplace for Image Processing Application using Task

Management on Cluster Computing System

พศิน จันทร์ทัน

สุชี สาระพันธ์

ปริญญาอิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2565

ปริญญาบัณฑีปการศึกษา 2565

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ร้านค้าสำหรับแอพพลิเคชั่นการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผลแบบคลุ่ม

Marketplace for Image Processing Application using Task Management on Cluster Computing

System

ผู้จัดทำ

1. นายพศิน จันทร์หัน รหัสนักศึกษา 63015121

2. นายสุรี สาระพันธ์ รหัสนักศึกษา 63015190

อาจารย์ที่ปรึกษา

(รศ. ดร.อรุณัตร จิตต์ไสวภัตร)

ร้านค้าสำหรับแอพพลิเคชันการประมวลผลภาพซึ่งจัดการงานบน

ระบบประมวลผลแบบกลุ่ม

นายพศิน	จันทร์ทัน	63015121
นายสุธี	สาระพันธ์	63015190
รศ.ดร.อรนัตร์	จิตต์โสภกตร์	อาจารย์ที่ปรึกษา
ปีการศึกษา 2565		

บทคัดย่อ

ในปัจจุบันการประมวลผลภาพนั้นมีการใช้ทรัพยากรณ์ที่สูงจึงได้มีบริการ งานประมวลผลบน อินเทอร์เน็ตขึ้น (Cloud Processing) ซึ่งวิธีดังกล่าวมีค่าใช้จ่ายที่สูง และ ไม่ตอบโจทย์ในเรื่องของการ แสดงผลงาน งานประมวลผลภาพที่ได้ทำการจัดทำขึ้นมา

โครงการนี้ ร้านค้าสำหรับแอพพลิเคชันการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผล แบบกลุ่ม มีจุดประสงค์เพื่อสร้างระบบให้บริการใช้งานการประมวลผลภาพ และ แสดงผลงานการงาน ประมวลผลที่ได้จัดทำขึ้นมาเพื่อให้ผู้ใช้ได้เปิดให้ผู้อื่นใช้งาน และ ซื้อขาย

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

Nowadays, image processing is highly resource-intensive, so there is a service. processing work on the Internet (cloud processing), which has a high cost and does not answer the question of displaying the work. The image processing work that has been done

Marketplace for Image Processing Application using Task Management on Cluster Computing System. The purpose is to develop a system that offers image processing services and displays the processed images for customers to access, utilize, and purchase.

บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

ในปัจจุบันผู้ให้บริการแอพพลิเคชั่นการประมวลผลภาพนั้นมีข้อจำกัดเรื่องงานการประมวลผล ใช้ทรัพยากรของเครื่องสูง ไม่มีประสิทธิภาพ หรือ ยังไม่มีการกำหนด จำกัดหน่วยประมวลผลของงานในแต่ละงานประมวลผล และ ไม่รองรับการประมวลผลภาพจำนวนมากพร้อม ๆ กัน พร้อมด้วยแอพพลิเคชั่นการประมวลผลภาพนั้นมีตัวเลือกค่อนข้างน้อยในแอพพลิเคชั่นเดียวให้ใช้งาน และเนื่องจากงานวิจัยด้าน Image Processing ต่าง ๆ นั้นมีการ Train ข้อมูลของ Weight Model เข้ามาช่วยในงานประมวลผลดังตัวอย่างในวิชา Image Processing ภายในสถาบันที่มีการให้นักศึกษา ศึกษาโครงงานเรื่องการ Train Weight Model ภายในวิชาขึ้นมาเพื่อมาทดลองภายในห้องทดลอง แต่ยังขาดพื้นที่สำหรับการให้บริการด้าน Model ของ Image Processing ที่นักศึกษาได้ทำการ Train Weight ข้อมูลไว้เพื่อให้ได้ทดสอบภายในห้องทดลองเนื่องจากใช้ทรัพยากรของเครื่องสูงในการประมวลผลของ Model และ ให้บุคคลภายนอกได้เข้าถึงด้วย

ดังนั้น โครงงาน ร้านค้าสำหรับแอพพลิเคชั่นการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผลแบบกลุ่ม มีจุดประสงค์เพื่อสร้างระบบการแบ่งการจัดลำดับงานประมวลผลของผู้ใช้บริการแอพพลิเคชั่นเพื่อเป็นอีกทาง เลือกหนึ่งในการใช้งานการประมวลผลภาพในรูปแบบต่าง ๆ อย่างมีประสิทธิภาพ และรองรับการประมวลผลภาพจำนวนมาก พร้อมด้วยรองรับแอพพลิเคชั่นที่สามารถนำเข้ามาเพิ่มในอนาคตได้อีกด้วย และ ในส่วนของ Weight Model ที่นักศึกษาได้ทำการ Train มาแล้วสามารถนำเข้าภายในแอพพลิเคชั่นเพื่อเปิดให้บริการในแอพพลิเคชั่นของเราราได้เพื่อทำการซื้อขาย หรือ เป็นที่แสดงผลงานของนักศึกษา และ เป็นอีกทางเลือกให้นักศึกษามีรายได้

โดยสรุปได้ว่า การตัดแต่งรูปภาพ หรือ การปรับแต่งภาพ นั้นมีการใช้อุปกรณ์หลาย และ คนส่วนใหญ่มีการใช้งานเรื่องการประมวลผลภาพกันเป็นปกติ การนำ Weight Model มาใช้งาน หรือ จัดแสดงผลงาน และ การซื้อขาย ซึ่งทั้งหมดที่กล่าวมาคือส่วนมากใช้งานบนแอพพลิเคชั่นต่างๆ ผ่านอินเทอร์เน็ตเราจึงพัฒนาโครงงานบนพื้นฐานของเว็บแอพพลิเคชั่นซึ่งสอดคล้อง และ รองรับกับปัญหาดังกล่าว โครงงาน “ร้านค้าสำหรับแอพพลิเคชั่นการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผลแบบกลุ่ม” นี้ถูกจัดทำขึ้นมาเพื่อแก้ไขในส่วนของกระบวนการนี้ โดยมีการแบ่งเป็น 3 ส่วนคือ

- 1) ส่วนการประมวลผลงานแบบการจัดลำดับงาน (Server) Task Scheduling การแบ่งจัดลำดับการทำงานประมวลผลแต่ละงานไปแท่นเครื่อง โดยมีการกำหนดทรัพยากรที่ใช้ในแต่ละงาน

ประมวลผลเพื่อให้ได้ประสิทธิภาพเพียงพอต่อความต้องการของงานประมวลผล และ ใช้ทรัพยากรของหน่วยประมวลผลให้คุ้มค่ากับทรัพยากรที่มีอยู่ของหน่วยประมวลผล

- 2) ส่วนแอปพลิเคชันการประมวลผลภาพ (User) Image Processing Application ส่วนระบบสำหรับประมวลผลภาพที่ระบบได้มีการเตรียมรูปแบบประมวลผลภาพต่าง ๆ ที่ หลากหลายให้ผู้ใช้ได้เข้ามาใช้งานการประมวลผลภาพจำนวนมากที่ทางเราได้เตรียมจัดทำไว้
- 3) ส่วนแอปพลิเคชันให้บริการซื้อขาย Weight ไมเดล (User) Marketplace ส่วนระบบสำหรับให้ผู้ใช้มีการซื้อ Weight Model และ นำเข้า Weight Model ที่ได้ทำการเทรนนิ่งไว้แล้วมาปิดให้ใช้บริการบนแอปพลิเคชันเพื่อทำการขาย หรือ เพื่อทดลองงานต่าง ๆ โดยเป็นจุดแสดงผลงานของตัว Weight Model ที่ได้ทำการเทรนนิ่งไว้ได้อีกด้วย

1.2 วัตถุประสงค์ของโครงการ

- 1) เพื่อจัดสรรการใช้งานประมวลผลของแต่ละงานประมวลผลให้มีประสิทธิภาพเพียงพอ ต่อความต้องการของงาน
- 2) เพื่อให้งานในการประมวลผลภาพนั้นมีการจัดลำดับการประมวลผล และ มีประสิทธิภาพมากยิ่งขึ้นต่อทรัพยากรของเครื่องประมวลผล
- 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 ในแอ��พลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่ายนี้นั้นเป็นการแบ่งงานในแต่ละเครื่องเพื่อให้ใช้ทรัพยากรที่จำกัด และ จำกัด ทรัพยากรในแต่ละงานประมวลผล
- 3) การจัดการลำดับงานบนระบบขึ้นอยู่กับประสิทธิภาพชาร์คแวร์ของ Node ที่พร้อมใช้งานในช่วงเวลาหนึ่ง
- 4) แอ��พลิเคชั่นในส่วนของฝากรายลักษณะ Structures ของ Directory และ User นั้นการแตกระดับขั้นของ Directory ได้ 1 ขั้น
- 5) แอ��พลิเคชั่นในส่วนของการประมวลผลนั้นสามารถทำการประมวลผลภาพจำนวนมาก จึงปรับค่า Parameter ได้ 1 ค่าในงานประมวลผลภาพจำนวนมาก
- 6) ส่วนของ Marketplace ในการนำเข้า Weight นั้นจะต้องรองรับกับตัว Model ที่มีให้บริการภายใต้แอ��พลิเคชั่นเท่านั้น

บทที่ 2

เอกสารและทฤษฎีที่เกี่ยวข้อง

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

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 Digital Image Processing

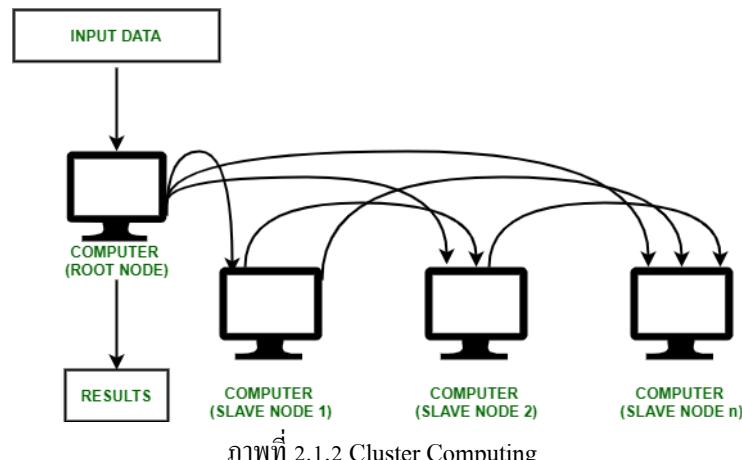
การประมวลผลภาพดิจิทัล หรือ Digital Image Processing คือ การใช้คอมพิวเตอร์ดิจิทัลในการประมวลผลภาพดิจิทัลผ่านอัลกอริทึม ภาพในที่นี่รวมความหมายถึงสัญญาณในระบบดิจิทัล 2 มิติ ภาพนิ่ง และ ภาพเคลื่อนไหว หรือเป็นชุดของภาพนิ่ง ที่เรียกว่า เฟรม (Frame) ซึ่งนับเป็นภาพดิจิทัล 3 มิติ ได้ เช่น ภาพทางการแพทย์ หรือ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

วัตถุประสงค์ของการประมวลผลภาพแบ่งได้ออกเป็น 5 กลุ่มหลักดังนี้

- 1) การสร้างภาพ (Visualization) จากวัตถุที่มองด้วยตาไม่เห็น สามารถทำให้มองเห็นภาพได้
- 2) การปรับความชัด และ การฟื้นฟูภาพ (Image sharpening and Restoration) ใช้สำหรับการปรับความละเอียดของภาพให้ดีขึ้น
- 3) การค้นภาพ (Image retrieval) ทำให้บุคคลที่น่าสนใจของภาพให้เด่นชัด
- 4) การวัดรูปแบบของภาพ (Measurement of pattern) วัดถูกในภาพทั้งหมดนั้นจะถูกวัดค่ามาเพื่อหารูปแบบของวัตถุภายในภาพ
- 5) การรับจำจำภาพ (Image Recognition) สามารถแยกแยะวัตถุในภาพได้

2.1.2 Cluster Computing

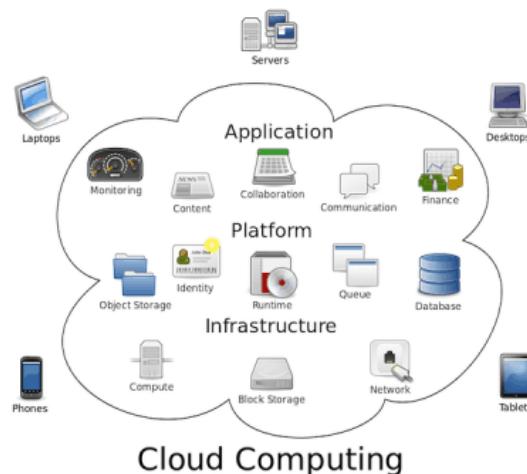
ระบบคลัสเตอร์ หรือ คลัสเตอร์ริง เป็นการเชื่อมต่อระบบการทำงานของกลุ่มคอมพิวเตอร์เข้าด้วยกันภายใต้ระบบเครือข่าย มีความสามารถในการกระจายงานที่ทำไปยังเครื่อง ภายในระบบเพื่อให้การประมวลผลมีประสิทธิภาพสูงขึ้น โดยอาจเทียบเท่าเซิร์ฟเวอร์คอมพิวเตอร์ หรือ สูงกว่าสำหรับการประมวลผลงานที่มีความซับซ้อนโดยเนพาะงานด้านวิทยาศาสตร์ เช่น การจำลองโครงสร้างของโมเลกุลทางเคมี, การวิเคราะห์เกี่ยวกับตำแหน่งการเกิดพายุสุริยะ, การประมวลผลภาพ เป็นต้น ดังรูป 2.1.



ภาพที่ 2.1.2 Cluster Computing

2.1.3 การประมวลผลแบบคลุ่มเมฆ (Cloud Computing)

การประมวลผลแบบคลุ่มเมฆ หรือ การให้บริการที่ตามความต้องการของผู้ใช้งานโดยที่ผู้ใช้งานระบุความต้องการ หรือ เลือกรูปแบบการใช้งานที่เหมาะสมกับความต้องการ ไปยังระบบการประมวลผลแบบคลุ่มเมฆ หลังจากนั้นระบบจะจัดสรรทรัพยากร และ บริการให้สอดคล้องกับความต้องการของผู้ขอใช้งาน และ ทรัพยากรของเครื่องประมวลผลในขณะที่ผู้ใช้งานไม่ต้องมีความรู้ ความเชี่ยวชาญ ไม่ต้องมีการติดตั้งตั้งค่าใด ๆ หรือ ไม่จำเป็นต้องทราบถึงการทำงานเบื้องหลังของระบบว่าจะเป็นอย่างไร และ ในขณะที่ใช้งานสามารถปรับเปลี่ยนทรัพยากรที่ใช้งานได้อย่างสะดวกมีประสิทธิภาพ และ รวดเร็ว สามารถเข้าใช้งานและเข้าถึงข้อมูลได้จากทุก ๆ ที่ทุกเวลา หรือ จากทุก ๆ อุปกรณ์ ดังแสดงในภาพที่ 2.1.3



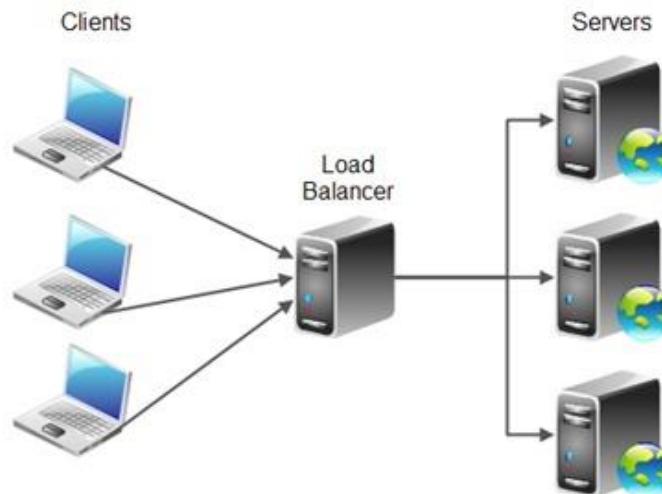
ภาพที่ 2.1.3 การประมวลผลแบบคลุ่มเมฆ (Cloud Computing)

2.1.4 การแบ่งภาระงาน (Load Balance)

การแบ่งภาระงาน หรือ Load Balance คือเทคนิควิธีการทางด้าน Network ที่ถูกคิดค้นขึ้นมา เพื่อช่วยลดการเกิดปัญหา Server ไม่สามารถรับงานประมวลผลจำนวนมากได้จน Overload และหยุดการทำงานไป โดยวิธีการคือการนำเครื่อง server ที่สามารถทำงานได้ในระดับเดียวกัน และ สามารถทำงานแบบเดียวกัน มาทำงานร่วมกันเพื่อช่วยกระจายปริมาณงานที่เข้ามา โดยจะแบ่งงานไปตามวิธีการที่ผู้ใช้ได้กำหนดเอาไว้ ซึ่งวิธีการแบ่งงานที่นิยมใช้กันมีต่อไป ดังนี้

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

ทั้งนี้การเลือกรูปแบบการแบ่งภาระงานนั้นขึ้นอยู่กับความต้องการของผู้ใช้ที่มีต่อระบบหรืองานนั้นๆ โดยรูปที่ 2.1.4 จะแสดงถึงแผนผังรูปแบบการใช้งานการแบ่งภาระงาน



ภาพที่ 2.1.3 แผนผังรูปแบบการใช้งานการแบ่งภาระงาน(Load Balance)

2.2 งานที่เกี่ยวข้อง

2.2.1 Image Processing Application

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

2.2.2 AnimeFilter

AnimeFilter หรือ AnimeFilter.com เป็นผู้ให้บริการ Application การประมวลผลภาพโดยการให้ผู้ใช้งานนำเข้าภาพของตนเองที่ต้องการเพื่อให้ Application นำภาพนั้นไปประมวลผลโดยมีการประมวลผลดังนี้ จับวัตถุของภาพที่ต้องการนำมาแปลงภาพให้อยู่ในรูปแบบ Model ที่ทางผู้ให้บริการได้เปิดให้ผู้ใช้งานเข้ามาใช้บริการแสดงดังรูปที่ 2.2.1



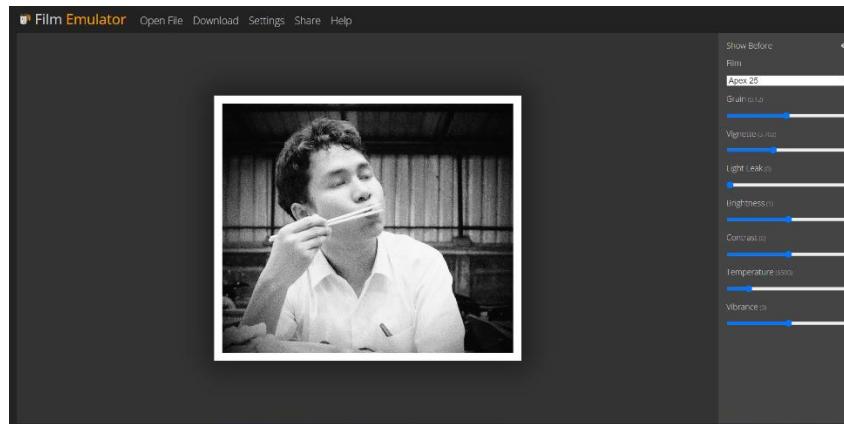
รูปที่ 2.2.1 ตัวอย่างของงานประมวลผลภาพบน AnimeFilter

สิ่งที่ AnimeFilter ยังไม่มี หรือ นำมาปรับปรุง มาพัฒนาต่อในแอพพลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย

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

2.2.2 Film Emulator

Film Emulator หรือ <https://29a.ch/film-emulator> เป็นผู้ให้บริการ Application การประมวลผลภาพโดยการให้ผู้ใช้งานนำเข้าภาพของตนเองที่ต้องการเพื่อให้ Application นำภาพนั้นไปประมวลผลโดยมีการประมวลผลดังนี้ ปรับแต่งสีของภาพตามที่ผู้ใช้ต้องการโดยมี parameter ที่กำหนดมาให้โดยมี Preset จากกล้องฟิล์ม ให้ผู้ใช้งานเข้ามาใช้บริการ ตัวอย่างแสดงดังรูปที่ 2.2.2



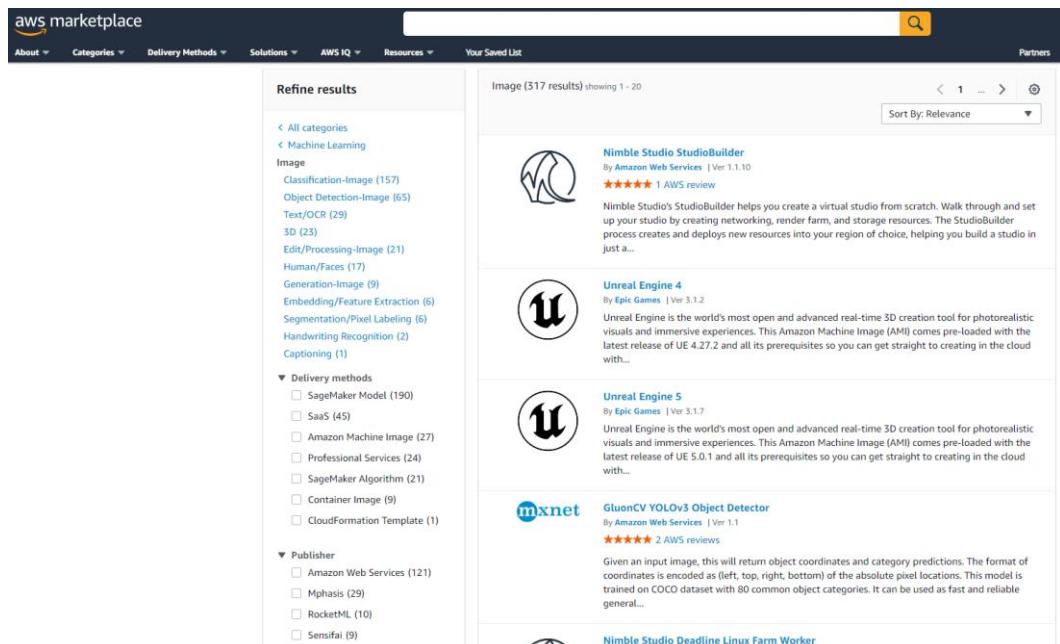
รูปที่ 2.2.2 ตัวอย่างของ Film Emulator

สิ่งที่ Film Emulator ยังไม่มี หรือ นำมาปรับปรุง มาพัฒนาต่อในแอพพลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครื่อข่าย

1. การรองรับการสั่งงานประมวลผลภาพจำนวนนัก
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 2.2.3



รูปที่ 2.2.3 ตัวอย่างของ aws marketplace

สิ่งที่ 3 aws marketplace ยังไม่มี หรือ นำมาปรับปรุง มาพัฒนาต่อในแอปพลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย

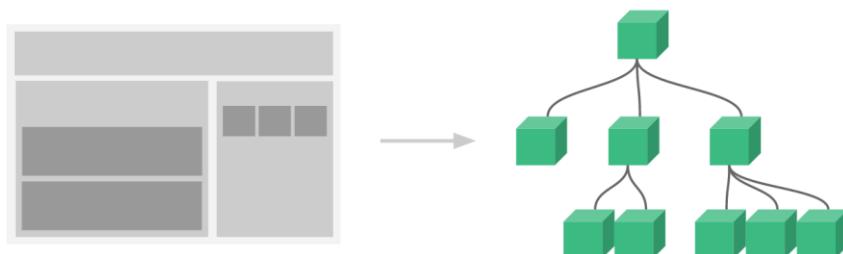
1. การให้บริการเป็นจำนวนภาพที่ประมวลผลแทนการใช้บริการรายชั่วโมง
2. ให้ผู้ใช้รายอยื่น ๆ นำเข้า Weight เพื่อมาซื้อขายได้จาก Model ที่ Application มีให้เหมือนกัน

2.3 เครื่องมือที่ใช้งานในการพัฒนาระบบ

2.3.1 Vue.js

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

Vue.js มีส่วนสำคัญอย่างหนึ่งนั้นคือ component system ที่ช่วยให้เราสามารถสร้าง Web application ขนาดใหญ่ที่ประกอบไปด้วย components ขนาดเล็ก และสามารถนำ component เหล่านั้นมาใช้งานซ้ำได้อีกด้วย โดยตัวอย่างการทำงานของ Component stem ของ Vue.js แสดงดังรูปที่ 2.3.1



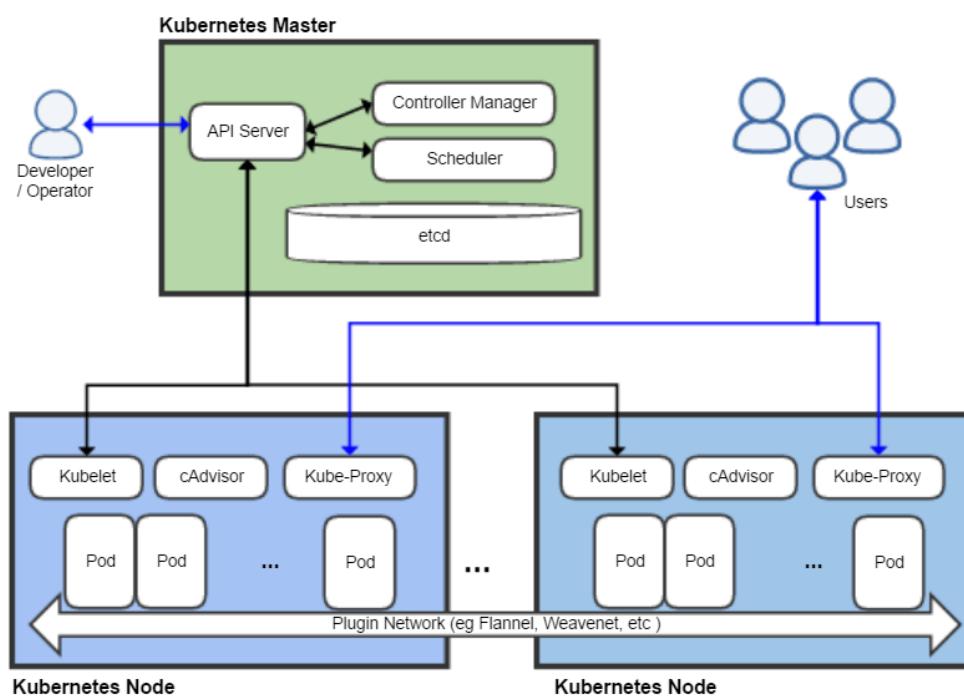
รูปที่ 2.3.1 Component system ของ Vue.js

2.3.2 Docker

Docker คือชุด Platform ที่ใช้การจำลองเสมือนระดับ OS เพื่อดำเนินการทำางานส่วนแพ็คเกจที่เรียกว่าคอนเทนเนอร์ (Container) ซึ่งในแต่ละคอนเทนเนอร์แยกจากกันภายใต้แต่ละ คอนเทนเนอร์จะประกอบด้วยซอฟต์แวร์ ไลบรารี และ ไฟล์กำหนดค่าของตนเอง (Environment) อิกทึ้งในแต่ละ Container ก็สามารถสร้างช่องทางสื่อสารกันภายใน Container เพื่อให้ช่วยในการประมวลผลโดยมีการใช้ทรัพยากริ่น้อย เมื่อเทียบเท่ากับการใช้งานบน OS ปกติทั่วไป และ ลดปัญหาในการเกิดข้อผิดพลาดของ Environment ใน การใช้งาน

2.3.3 Kubernetes

คือ ภูกพัฒนาโดย 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 มาอีกมากmany ไม่ว่าจะเป็น MicroK8S, K3D, MiniKube โดยทั้งหมดที่กล่าวมานี้มีเงื่อนไข และ การทำงาน การใช้งานอยู่ในรูปแบบเดียวกัน โดยมีหลักการทำงานพื้นฐานเหมือนกันดังรูป 2.3.3



รูปที่ 2.3.2 Component system ของ Vue.js

2.3.4 Python

Python เป็นภาษาในการเขียนโปรแกรมที่ใช้อย่างแพร่หลายในการพัฒนาซอฟต์แวร์ เว็บ แอ��พลิเคชั่น วิทยาศาสตร์ข้อมูล การประมวลผลภาพ และ แมชชีนเลร์นนิง (ML) มีการเรียนรู้ง่าย และ สามารถทำงานบนแพลตฟอร์มได้มากmany ไม่ว่าจะเป็นบน Window, MacOS, Linux ทั้งนี้ Python เป็นซอฟต์แวร์ที่สามารถดาวน์โหลด และ ใช้งานได้ฟรี

2.3.5 Django

Django เป็น Framework ที่ถูกเขียนด้วยภาษา Python สำหรับพัฒนาเว็บแอปพลิเคชัน โดยที่สามารถพัฒนาได้อย่างรวดเร็ว และ มีประสิทธิภาพ เหมาะกับการพัฒนาเว็บที่มีการใช้งานบอย เช่น การตรวจสอบการดึงข้อมูลจากฐานข้อมูล การจัดการคุกคัก และ การทำงานควบคู่ไปกับ Python ซอฟต์แวร์อื่น ๆ มาปรับให้ใช้งานภายใต้เว็บแอปพลิเคชันได้ ตัว Django เป็น open-source ที่สามารถนำมาใช้งานได้ฟรี และ พัฒนาได้ฟรี โดยมี Community ที่ช่วยดูแลรักษาซอฟต์แวร์ตัวนี้ และมีองค์กรไม่แสวงหาผลกำไรซึ่งเป็น Django Software Foundation ที่ทำการพัฒนา ปรับปรุง และบำรุงรักษา ตัว Django ให้มีคุณภาพสูง และ ฟีเจอร์อีกมากมายให้ใช้งานโดยไม่มีค่าใช้จ่ายตลอดมา

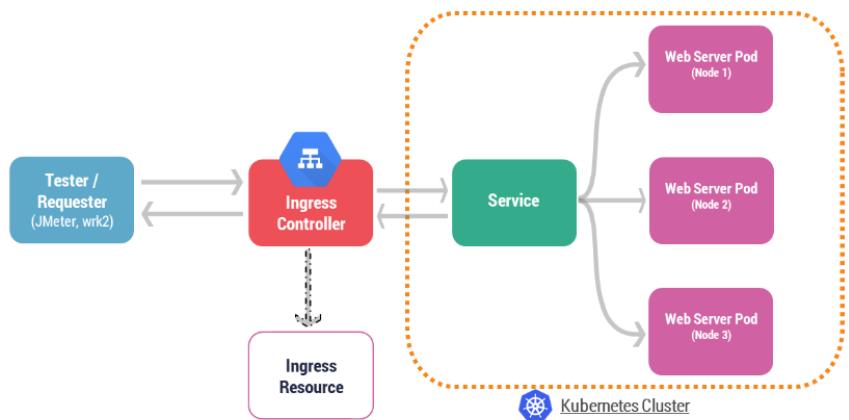
2.3.6 MongoDB

MongoDB เป็น NoSQL Database ที่สามารถใช้งานข้าม Platform ต่าง ๆ ได้โดยตัว MongoDB ได้ใช้ NoSQL เป็นการเข้าถึงข้อมูลโดยใช้เทคนิคของการดึงข้อมูลผ่าน Key Pair Value โดยหมายความว่า กับงานที่ต้องการเข้าถึงข้อมูลได้อย่างรวดเร็ว และ เหมาะการทำงานงาน Big Data ตัวของ MongoDB สามารถที่จะสร้างเป็น Cluster เพื่อตอบสนองความต้องการเป็น High Availability (HA) ได้ และ สามารถทำการ 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 คุณอาทิตย์ พวงคำไวย และ คุณชัยพร เบนนาภาตะพันธ์ กล่าวถึงเรื่องการทดลองเพื่อประเมินสมรรถนะ และเปรียบเทียบประสิทธิภาพของ ระบบ Load balance แบบ Layer – 7 ที่ถูกใช้งานในสภาพแวดล้อมการทำงานที่อยู่ในรูปแบบของ Container cluster ด้วย Kubernetes โดยแผนผังการวางแผนในการวิจัยจะเป็นดังรูป 2.4.1



รูปที่ 2.4.4 แผนผังแสดงระบบที่ใช้ในการทดสอบIngress controller

โดยโครงงานนี้ได้นำเอกสารนี้มาศึกษาเพื่อใช้ในการพัฒนาส่วน Cluster Computing เพื่อให้ การจัดลำดับงานและการประมวลผลใน Cluster มีประสิทธิภาพมากขึ้น

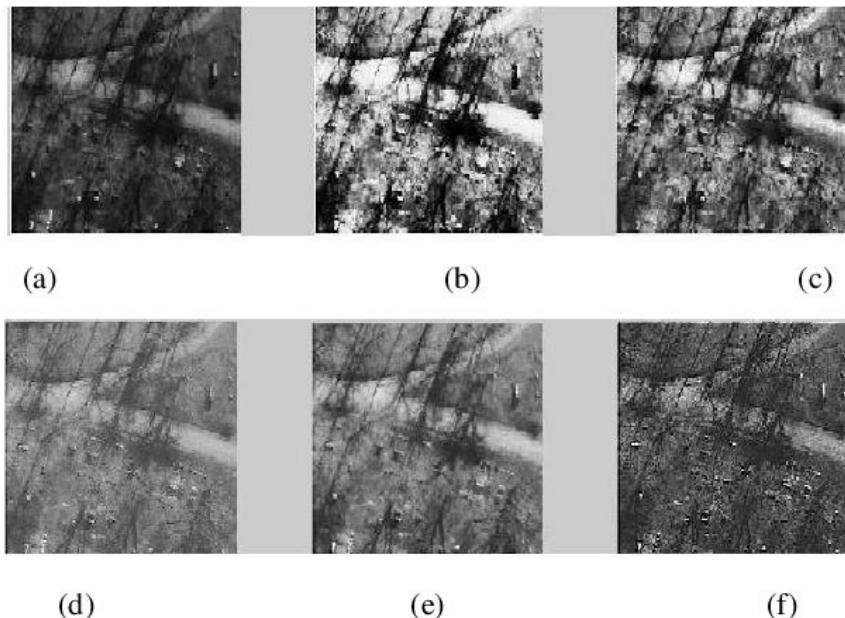
2.4.2 A modified gray-level difference algorithm for analyzing Gaussian Blurred texture images

งานวิจัยนี้จัดทำขึ้นในปี 2011 โดย Rui Zhang, Xiang Qian และ Datian Ye กล่าวถึงเรื่องการกระบวนการแยกแยะพื้นผิวของรูปภาพโดยการใช้อัลกอริทึม gray-level difference (GLD) โดยจะมี 4 ขั้นตอนคือ 1. ประเมินความแตกต่างของพื้นผิวภาพโดยเปรียบเทียบกับข้อมูลที่เตรียมเอาไว้ ก่อนหน้า 2. ใช้ตัวกรองแบบ Wiener เพื่อลบ Gaussian blur noise ที่อาจเกิดขึ้นหากค่าความแตกต่าง มีน้อย 3. ทำขั้ววิธีการที่ 2. ไปจนกระทั่งค่าความแตกต่างสูงกว่าข้อมูลที่เตรียมเอาไว้ 4. ใช้อัลกอริทึม gray-level difference แบบดั้งเดิม เพื่อเก็บข้อมูลพื้นผิวจากรูปภาพที่ผ่านการประมวลผลแล้ว เพื่อนำมาวัดประสิทธิภาพเทียบกับอัลกอริทึม gray-level difference แบบใหม่

โดยโครงงานนี้ได้นำเอกสารนี้มาเพื่อศึกษาใช้ในการพัฒนา 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 กล่าวถึงเรื่องการปรับแต่งรูปภาพที่ถูกถ่ายมาจากการถ่ายในส่วนของความคมชัดของภาพด้วยเทคนิค Local contrast stretching (LCS) โดยจะทำการ sliding window ผ่านรูปภาพแล้วปรับความคมชัดของภาพไปทีละส่วน โดยสามารถปรับให้ความคมชัดของตัวรูปภาพนั้นเพิ่มขึ้นและลดลงได้ตามความต้องการ และนอกจากนี้ก็ยังนำเสนอวิธีการอื่นที่ใช้ในการปรับแต่งความคมชัด เช่น the bi-histogram equalization (BHE) และ Global histogram equalization เป็นต้น โดยรูปที่ 2.4.3 จะเป็นรูปผลลัพธ์จากการปรับต่างด้วยวิธีที่แตกต่างกัน



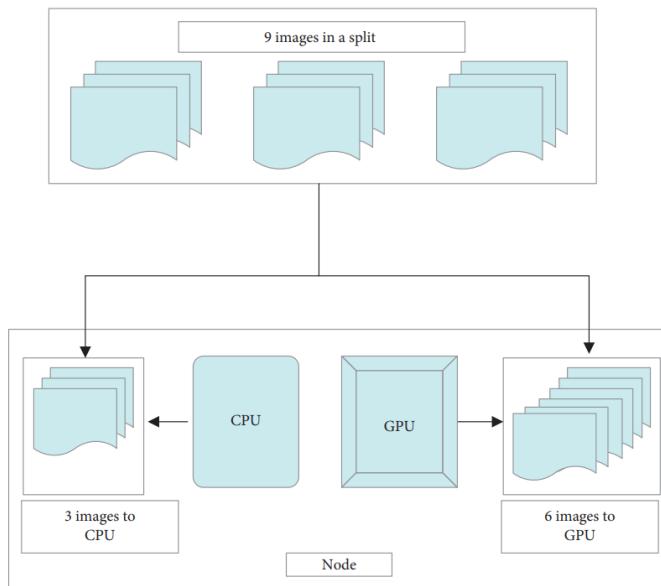
รูปที่ 2.4.3 รูปผลลัพธ์จากการปรับต่างความคมชัดด้วยวิธีที่แตกต่างกัน โดย (a) Original Image Contrast , (b) Histogram Equalization Contrast Enhancement , (c) Adaptive Histogram Equalization , (d) LPF Homomorphic Contrast Enhancement , (e) HPF Homomorphic Contrast Enhancement และ (f) Unsharp Mask Contrast Enhancement

โดยโครงงานนี้ได้นำเอกสารนี้มาศึกษาเพื่อใช้ในการพัฒนาส่วนของ 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 กล่าวถึงเรื่องการกระบวนการแปลงงานที่มีประสิทธิภาพสำหรับ Image processing application จากปกติการที่ประมวลผลงาน

ด้าน image processing สิ่งที่สามารถพูดเจ้อได้เป็นปกติคือ การกระจายภาระงานที่ไม่สมดุล ซึ่งทำให้ประสิทธิภาพของการทำงานนั้นลดลง โดยการแบ่งนี้จะแบ่งออกเป็น 2 ช่วง ช่วงแรกจะเป็นการแบ่งและแยกแยะข้อมูลของรูปภาพเป็นส่วนๆ ที่เหมาะสมต่อการกระจายไปยังหน่วยประมวลผลต่างๆ จากนั้นในช่วงที่สอง ทำการกระจายเพิ่มเติมไปยัง CPU และ GPU ขึ้นอยู่กับความเร็วในการประมวลผลดังรูป 2.4.3

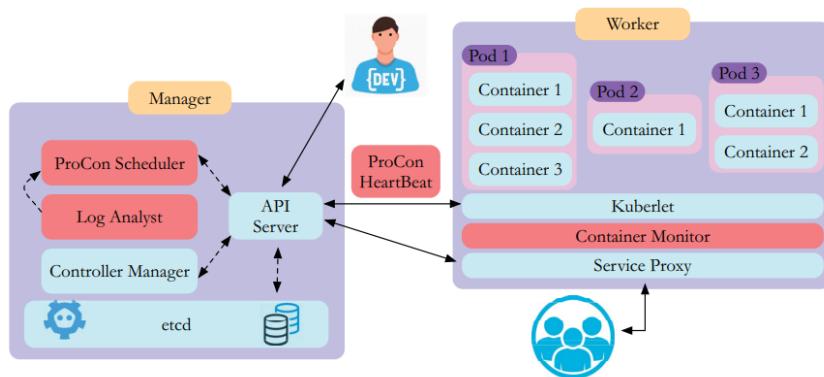


รูปที่ 2.4.3 แผนผังรูปแบบการกระจายงานในช่วงที่สองที่มีการกระจายภาระงานระหว่าง CPU และ GPU

โดยโปรแกรมนี้ได้นำเอกสารนี้มาศึกษาเพื่อใช้ในการพัฒนาส่วน 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 ที่ตั้งชื่อ ไว้ว่า ProCon โดยปกติแล้วจะมี Manager node เป็นตัวที่เลือกใช้งาน Worker node แต่ละตัวตาม อัลกอริทึมที่ตั้งค่าเอาไว้ โดยที่ ProCon จะทำหน้าที่ในการกำหนด input คอนเทนเนอร์แล้ว พิจารณา การใช้ทรัพยากรหันที่ รวมไปถึงการประเมินทรัพยากรในอนาคตด้วย และขณะที่ประมวลผลนั้น ProCon ก็จะทำการจัดสรรของทรัพยากรภายใน Cluster ไปด้วย



รูปที่ 2.4.4 แผนผัง System Architecture ของ ProCon

โดยโครงงานนี้ได้นำงานวิจัยมาศึกษาเพื่อสร้างรูปแบบการจัดการคอนเทนเนอร์ที่มีการจัด ทรัพยากรให้แต่ละ Worker อย่างมีประสิทธิภาพ

บทที่ 3

การออกแบบและการพัฒนา

การออกแบบและการพัฒนาระบบ แอปพลิเคชันการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย แบ่งออกเป็น 3 ส่วน คือ Web Application ส่วนของ Image Processing Application และส่วนของ Server โดยมีรายละเอียดส่วนต่าง ๆ ดังนี้

3.1 รายละเอียดของระบบ

1) Web Application

- 1) ผู้ใช้สามารถสมัครสมาชิก เข้าสู่ระบบได้
- 2) สามารถถ่ายรูป และ เปลี่ยนรหัสผ่านได้
- 3) สามารถจัดเก็บไฟล์รูปภาพได้
- 4) สามารถสั่งงานประมวลผลภาพได้
- 5) สามารถทดลองดูตัวอย่างงานประมวลผลภาพได้
- 6) สามารถจัดเก็บ Weight Model ได้
- 7) สามารถซื้อขาย Weight Model ได้
- 8) สามารถสั่งงานประมวลผลภาพบน Marker Place ได้
- 9) สามารถดูสถิติการประมวลผลภาพได้

2) Image Processing Application

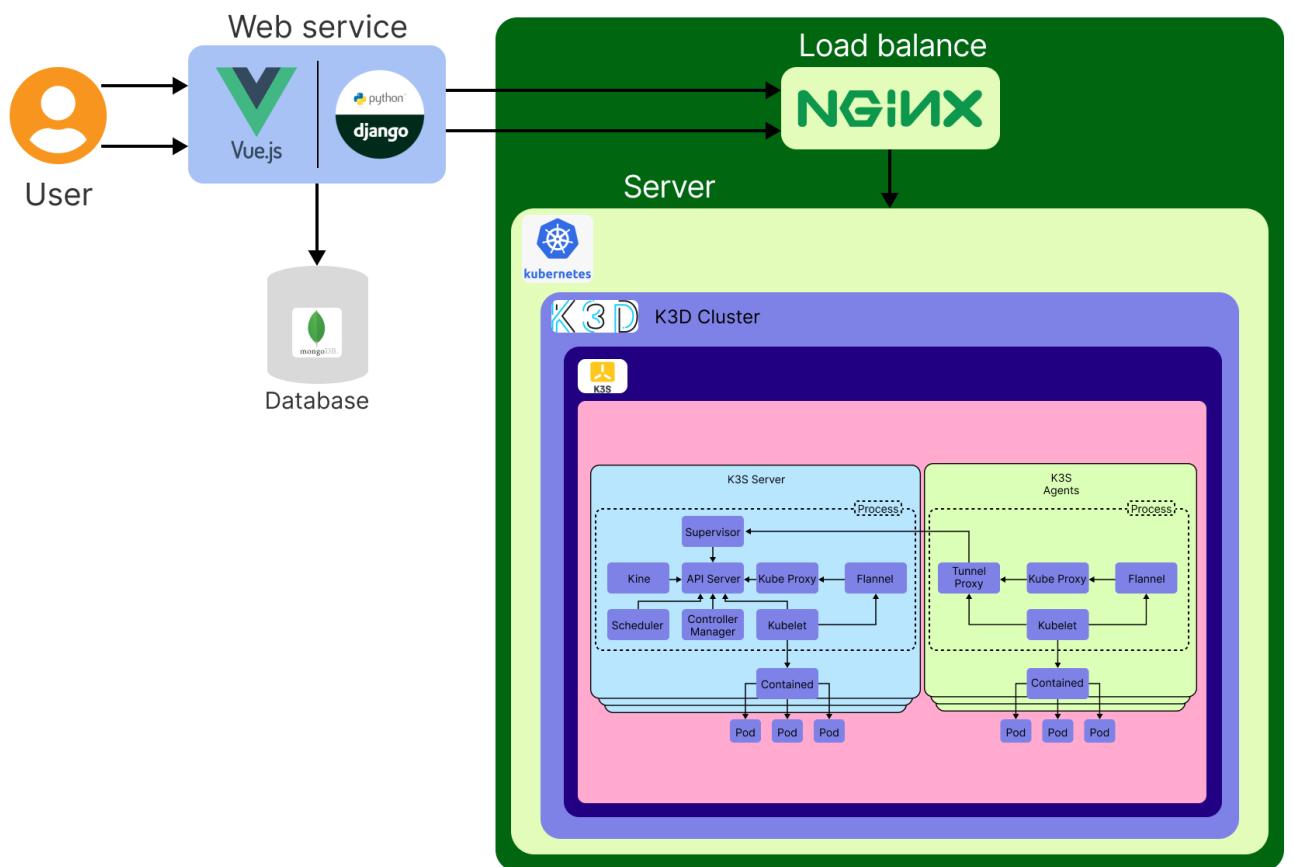
- 1) สามารถประมวลผลภาพเป็นภาพ ASCII ได้
- 2) สามารถประมวลผลภาพเป็นภาพ PixelArt ได้
- 3) สามารถประมวลผลภาพเป็นภาพ Mosaic ได้
- 4) สามารถประมวลผลภาพเป็นภาพขาวดำได้
- 5) สามารถประมวลผลภาพเป็นภาพเพื่อลบพื้นหลังภาพ ได้
- 6) สามารถประมวลผลภาพเป็นภาพจาก Model และ Weight ได้

3) Server

- 1) สามารถ Task Management งานประมวลผล ได้
- 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

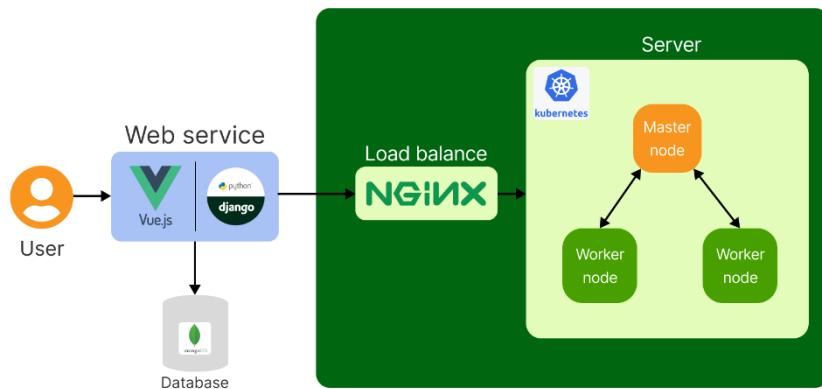
3.2 ภาพรวมของระบบ

ภาพรวมในการพัฒนาระบบที่แสดงให้เห็นส่วนต่าง ๆ ที่อยู่ในแผนการพัฒนาโครงงานที่ประกอบไปด้วยส่วนประกอบหลัก ๆ คือ Web Service (ส่วนให้บริการตัวเว็บไซต์ และ ฐานข้อมูล), ส่วน LoadBalance (ส่วนให้บริการในการจัดการผู้ใช้ที่เข้ามาขอใช้บริการเพื่อแบ่งงานประมวลผลไปยังเครื่องอื่น ๆ), ส่วน Server (ส่วนให้บริการในการสั่งงานประมวลผล และ หน่วยประมวลผลแบบ Cluster), ภาพรวมของระบบ จะแสดงดังรูปที่ 3.1



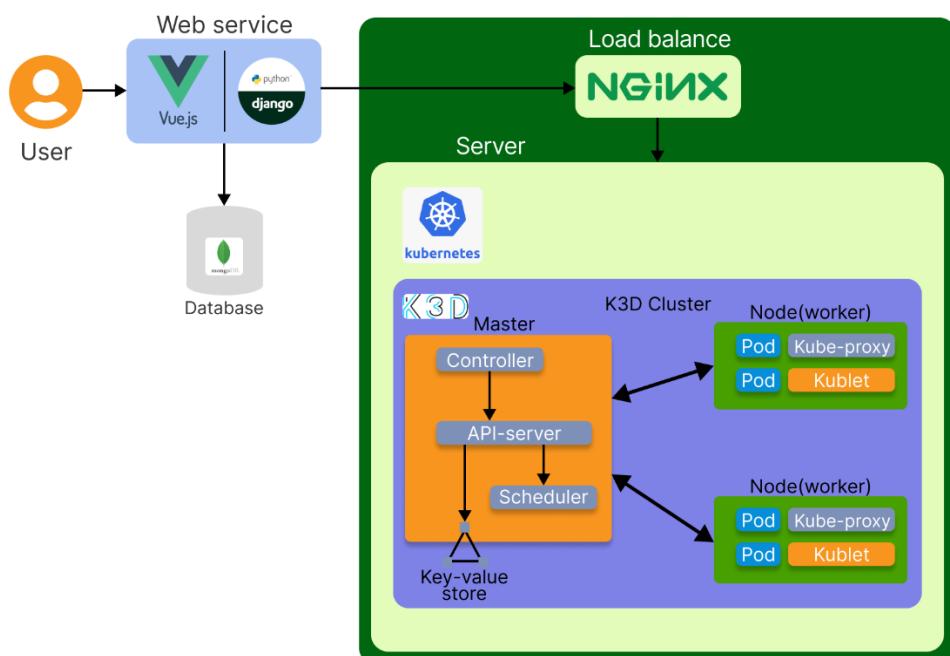
รูปที่ 3.1 ภาพรวมของระบบ Overview

โดยภาพรวมของระบบใน Level 0 นั้นจะกล่าวถึงการเชื่อมต่อของ Master Node และ Worker Node ที่มีการติดต่อสื่อสารกันภายในระบบ Kubernetes Cluster เพื่อให้ User เข้ามาขอการใช้งานเข้าถึง ดังรูปที่ 3.2



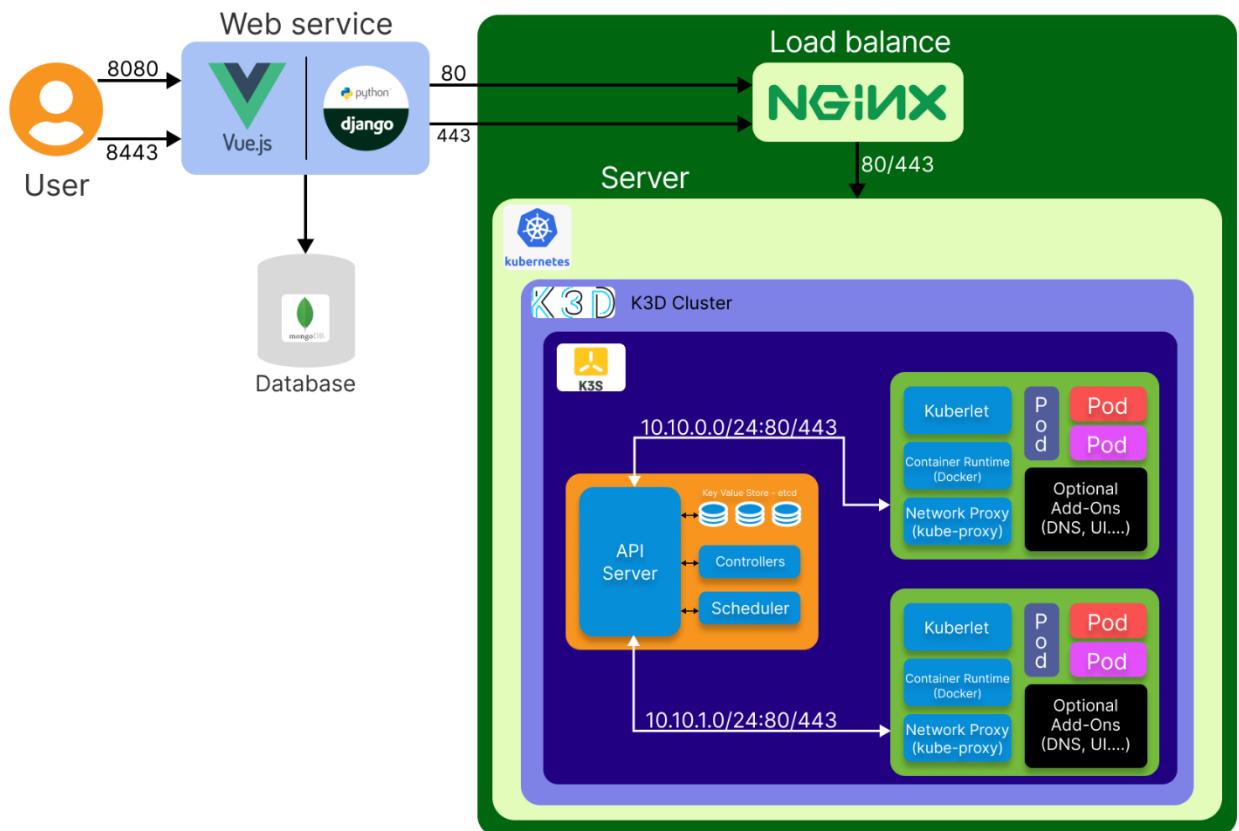
รูปที่ 3.2 ภาพรวมของระบบ Level 0

โดยภาพรวมของระบบใน Level 1 นั้นจะกล่าวถึงการเชื่อมต่อของ Master Node และ Worker Node ที่มีการ Scheduler ผ่าน API-server เพื่อทำการ Load Balance งานที่ Master ได้รับไปยัง Worker แต่ละเครื่องตามที่ได้ตั้งค่าไว้ใน Load Balancer ของ Kubernetes ดังรูปที่ 3.3



รูปที่ 3.3 ภาพรวมของระบบ Level 1

โดยภาพรวมของระบบใน Level 2 นั้นจะกล่าวถึงการเชื่อมต่อของ User ไปยัง Web Service นั้นโดยมี Port การเชื่อมต่อไป ในส่วนของ Web Service มี Vue.js สำหรับการทำ Front-End และ Python Django สำหรับ Back-End โดยที่มี Database เป็น MongoDB ส่วนของฝั่ง Server จะมีการเตรียม Load Balance Nginx คู่กับ Ingress สำหรับการ Load Balance การขอเข้าใช้งาน Web Service ไปยัง Kubernetes Cluster ใน Server โดย Master Node จะมีการเชื่อมต่อกับ Worker Node อื่น ๆ ดังรูปที่ 3.4

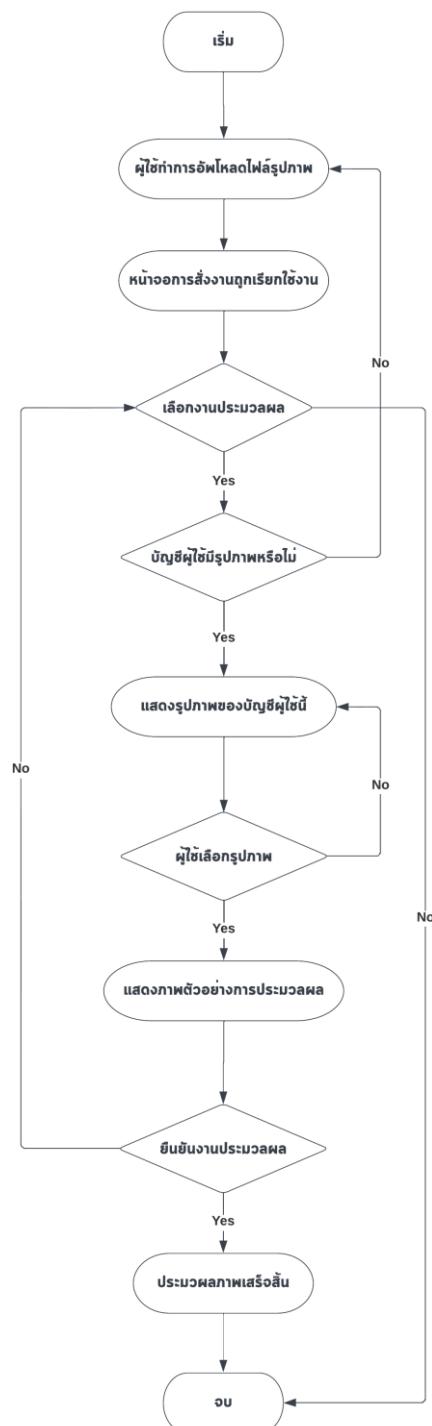


รูปที่ 3.4 ภาพรวมของระบบ Level 2

3.3 แผนภาพผังงาน (Flowchart)

3.3.1 แผนภาพผังงาน (Flowchart) ของการสั่งงานประมวลผลภาพ (Web Application)

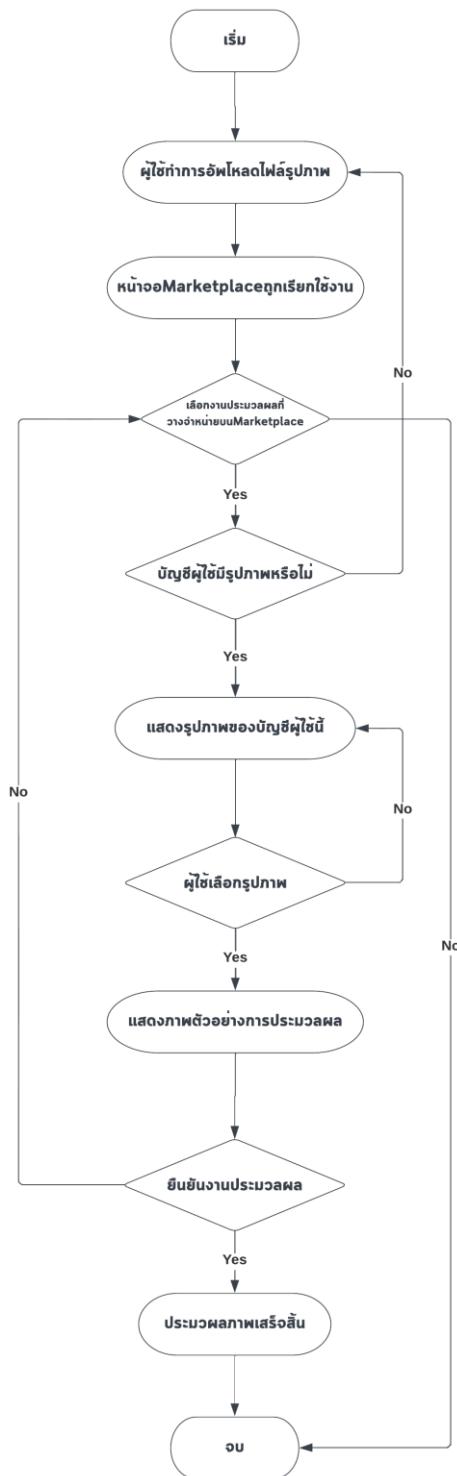
แผนภาพผังงานในส่วนของ Web Application บนเว็บที่ผู้ใช้งานเริ่มทำการใช้งานการสั่งงานประมวลผลภาพถึงการประมวลผลภาพเสร็จสิ้น โดยแผนภาพผังงานในส่วนของการสั่งงานประมวลผลภาพ จะแสดงดังรูปภาพที่ 3.5



รูปที่ 3.5 แผนภาพผังงานของการสั่งงานประมวลผลภาพ (Web Application)

3.3.2 แผนภาพผังงาน (Flowchart) การสั่งงานประมวลผลภาพบน Marketplace (Web Application)

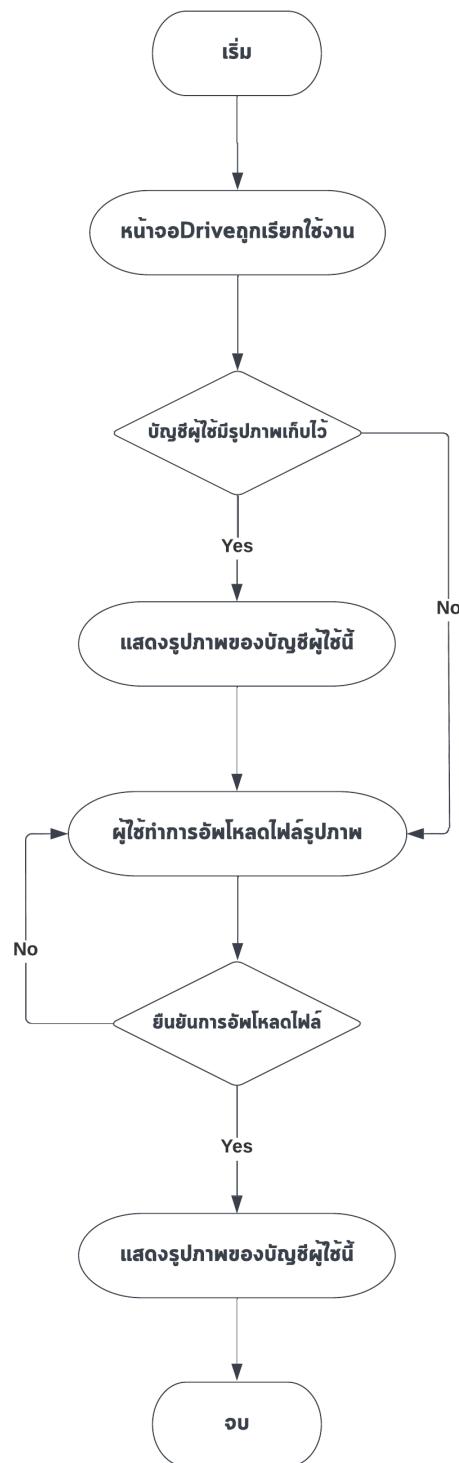
แผนภาพผังงานในส่วนของ Web Application บนเว็บที่ผู้ใช้งานเริ่มทำการใช้งานการสั่งงานประมวลผลภาพบน Marketplace จนถึงการประมวลผลภาพเสร็จสิ้น โดยแผนภาพผังงานในส่วนของการสั่งงานประมวลผลภาพ จะแสดงดังรูปภาพที่ 3.6



รูปที่ 3.6 แผนภาพผังงานของการสั่งงานประมวลผลภาพบน Marketplace (Web Application)

3.3.3 แผนภาพผังงาน (Flowchart) การจัดเก็บไฟล์ภาพไว้ใน Drive (Web Application)

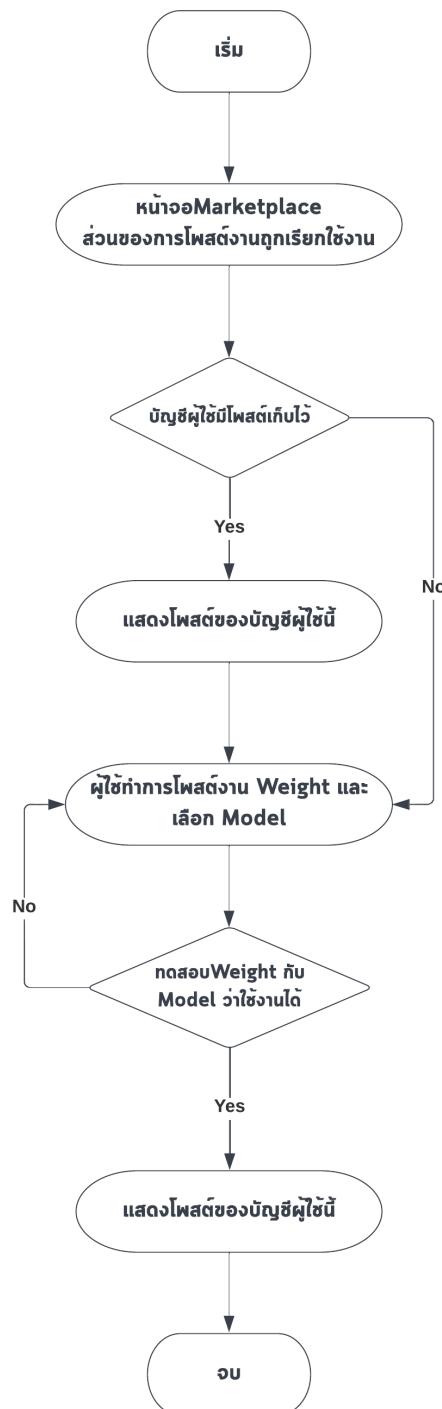
แผนภาพผังงานในส่วนของ Web Application บนเว็บที่ผู้ใช้งานเริ่มทำการจัดเก็บไฟล์ภาพไว้ใน Drive จนถึงการการจัดเก็บไฟล์ภาพเสร็จสิ้น โดยแผนภาพผังงานในส่วนของการจัดเก็บไฟล์ภาพไว้ใน Drive จะแสดงดังรูปภาพที่ 3.7



ຮູບທີ 3.7 แผนภาพผังงานของการการจัดเก็บไฟล์ภาพไว้ใน Drive (Web Application)

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

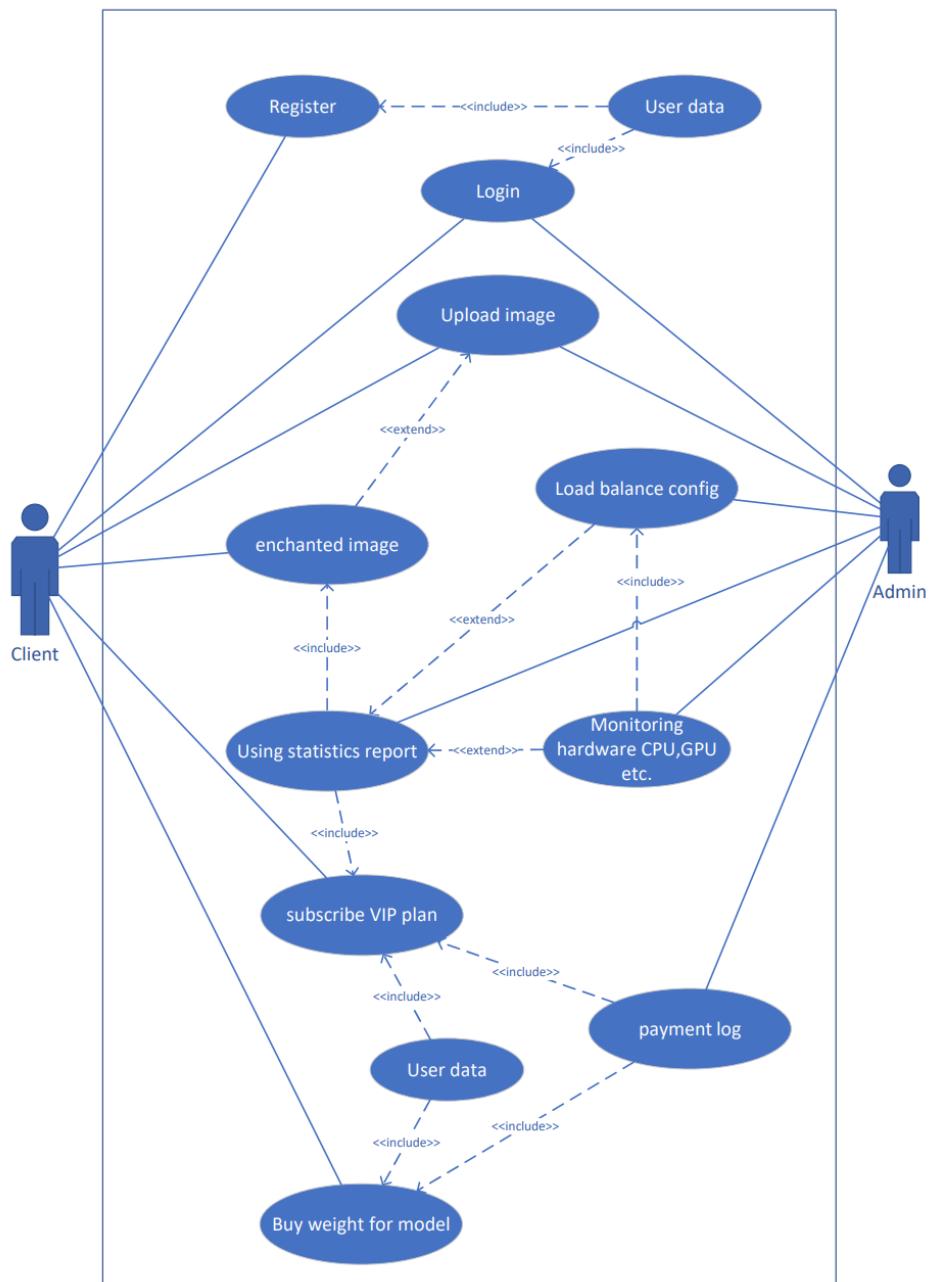
แผนภาพผังงานในส่วนของ Web Application บนเว็บที่ผู้ใช้งานเริ่มทำการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train จนถึงการโพสต์งานเสร็จสิ้น โดยแผนภาพผังงานในส่วนของการขาย Weight ที่ผู้ใช้ได้ทำการ Train จะแสดงดังรูปภาพที่ 3.8



รูปที่ 3.8 แผนภาพผังงานของการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train

3.4 แผนภาพยูสเคส (Use Case Diagram)

แผนภาพยูสเคสจะแสดงให้เห็นถึงผู้ใช้ที่ลูกค้าเป็น 2 ประเภท คือ Client คือ ผู้ใช้ที่เข้ามาในรูปแบบลูกค้าที่ต้องการใช้บริการงานประมวลผลภาพต่างๆ และ Administer คือผู้ใช้ที่เป็นผู้ดูแลในส่วนของข้อมูลในระบบ และ ตั้งค่า ในส่วนของการประมวลผลภาพ รวมไปถึงการตรวจสอบการ payment ต่างๆ โดยแผนภาพ ยูสเคส จะแสดงดังรูปภาพที่ 3.9



รูปที่ 3.9 แผนภาพยูสเคส

โดย ส่วนรายละเอียดของ Use case จะแสดงดังตารางที่ 3.1 ถึง 3.10

ตารางที่ 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	

ตารางที่ 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	

ตารางที่ 3.3 รายละเอียดของ Use case Upload image

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

ตารางที่ 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 “การปรับแต่งรูปภาพล้มเหลว เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
Trigger : เลือกรูปภาพที่จะปรับแต่ง เลือกฟังก์ชันการปรับแต่ง และกดปุ่มปรับแต่ง	

ตารางที่ 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 “การปรับแต่งรูปภาพล้มเหลว เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
Trigger : เลือกฟังก์ชันการปรับแต่ง และกดปุ่มปรับแต่ง	

ตารางที่ 3.6 รายละเอียดของ Monitoring hardware

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

ตารางที่ 3.7 รายละเอียดของ Using statistics report

Use Case Title : Using statistics report	Use case ID: 7
Description : ดูสถิติการใช้งานของผู้ใช้ที่อยู่ในระดับ Client	
Actor : Administer	
Precondition : ยืนยันตัวตนลงชื่อเข้าใช้	
Postcondition : ได้สถิติการใช้งาน	
Fail End Condition : แจ้งเตือนข้อความ “ไม่สามารถเข้าถึงสถิติผู้ใช้ได้ เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
Trigger : เลือกฟังก์ชันดูสถิติการใช้งาน	

ตารางที่ 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	

ตารางที่ 3.9 รายละเอียดของ Buy weight for model

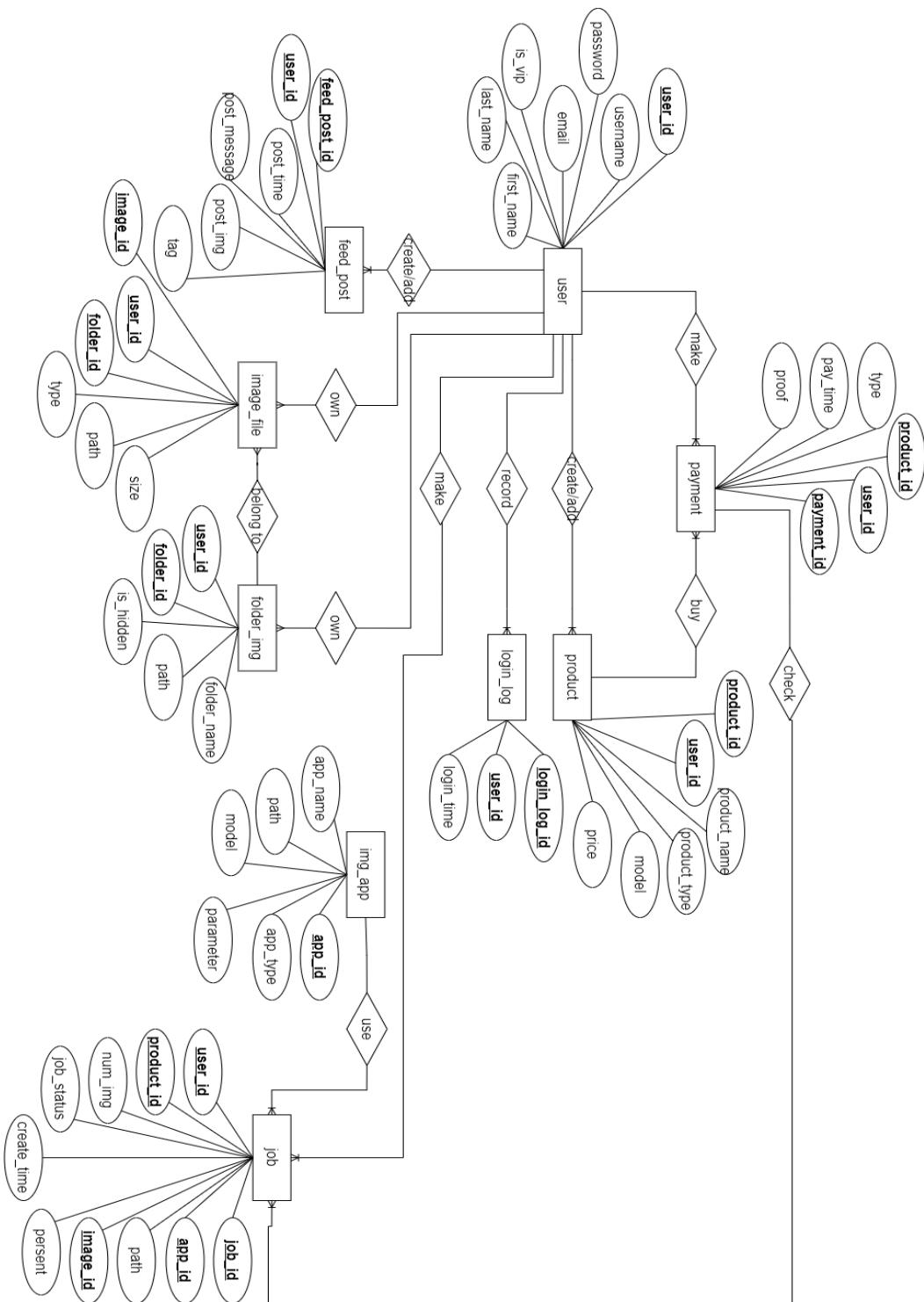
Use Case Title : Buy weight for model	Use case ID: 9
Description : ซื้อ weight สำหรับ Model เพื่อนำไปทำการประมวลผลภาพ	
Actor : Client	
Precondition : ยืนยันตัวตนลงชื่อเข้าใช้	
Postcondition : ได้รับ weight สำหรับ Model ไว้ในฐานข้อมูล	
Fail End Condition : แจ้งเตือนข้อความ “การซื้อขายล้มเหลว เนื่องจาก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

ตารางที่ 3.10 รายละเอียดของ Payment log

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

3.5 แผนภาพแสดงความสัมพันธ์ของข้อมูล (E-R Diagram)

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



รูปที่ 3.10 แผนภาพแสดงความสัมพันธ์ของข้อมูล

โดยรายละเอียดของแผนภาพแสดงความสัมพันธ์ของข้อมูลแสดงดังตารางที่ 3.12 ถึง 3.20

ตารางที่ 3.12 แสดงรายละเอียดตารางข้อมูลผู้ใช้ (user)

คีย์	ชื่อแอ็อดทริบิวต์	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

ตารางที่ 3.13 แสดงรายละเอียดตารางข้อมูลไฟล์รูปภาพ (Image_file)

คีย์	ชื่อแอ็อดทริบิวต์	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

ตารางที่ 3.14 แสดงรายละเอียดตารางข้อมูลของโฟลเดอร์เก็บรูปภาพ (folder_img)

คีย์	ชื่อแอ็อดทริบิวต์	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

ตารางที่ 3.15 แสดงรายละเอียดตารางเก็บข้อมูลการเข้าสู่ระบบ (login_log)

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

ตารางที่ 3.16 แสดงรายละเอียดตารางข้อมูล Image processing application (img_app)

คีย์	ชื่อแอ็อดทริบิวต์	ชนิดตัวแปร	คำอธิบาย
PK	app_id	Varchar(50)	รหัสประจำ Image processing application
-	app_name	Varchar(50)	ชื่อของ Image processing application
-	app_type	Varchar(50)	ชนิดของ Image processing application
คีย์	ชื่อแอ็อดทริบิวต์	ชนิดตัวแปร	คำอธิบาย
-	path	Varchar(50)	เส้นทางที่เก็บไฟล์ Image processing application
-	parameter	Varchar(50)	พารามิเตอร์ที่ Image processing application ต้องการ
-	model	Varchar(50)	โมเดลที่ Image processing application ใช้งาน

ตารางที่ 3.17 แสดงรายละเอียดตารางข้อมูลการชำระเงิน (payment)

คีย์	ชื่อแอ็อดทริบิวต์	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
-	proof	Varchar(50)	หลักฐานการชำระเงิน

ตารางที่ 3.18 แสดงรายละเอียดตารางข้อมูลของสินค้า (product)

คีย์	ชื่อแอ็อดทริบิวต์	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

ตารางที่ 3.19 แสดงรายละเอียดตารางข้อมูลของโพสต์ในฟีดข่าว (feed_post)

คีย์	ชื่อแอ็อดทริบิวต์	ชนิดตัวแปร	คำอธิบาย
PK	feed_post_id	Varchar(50)	รหัสประจำของโพสต์
FK	user_id	Varchar(50)	รหัสประจำตัวผู้ใช้ที่เป็นเจ้าของโพสต์
-	post_time	Datetime	เวลาที่โพสต์
-	post_message	Varchar(200)	ข้อความในโพสต์
-	post_img	Varchar(50)	รูปภาพในโพสต์
-	tag	Varchar(50)	แท็กของโพสต์

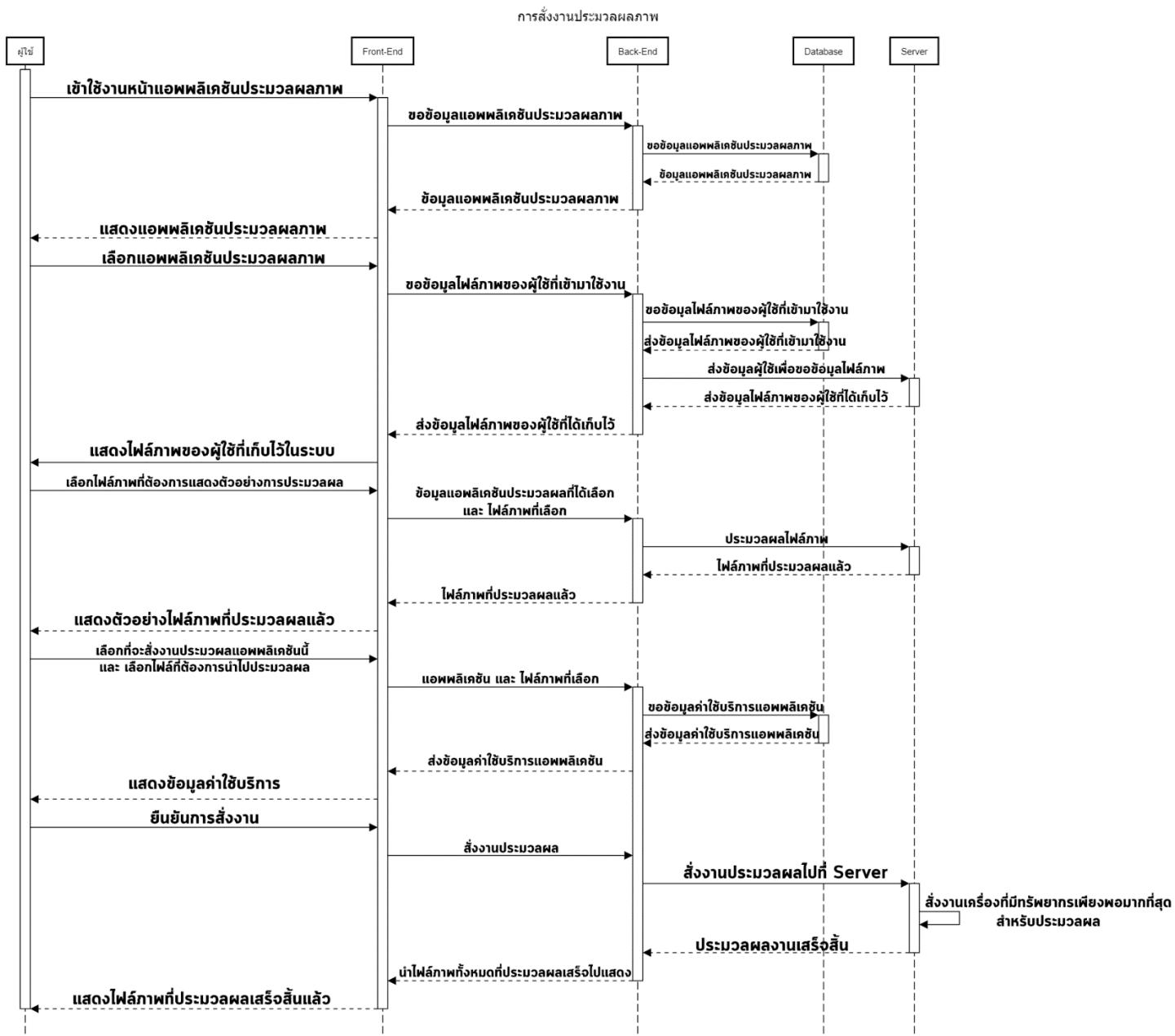
ตารางที่ 3.20 แสดงรายละเอียดตารางข้อมูลการสั่งงานประมวลผล (job)

คีย์	ชื่อแอ็อดทริบิวต์	ชนิดตัวแปร	คำอธิบาย
PK	job_id	Varchar(50)	รหัสระบุงานที่สั่ง
FK	user_id	Varchar(50)	รหัสประจำตัวของผู้ใช้งานที่สั่งงาน
FK	app_id	Varchar(50)	รหัสประจำ Image processing application ที่ใช้งาน
FK	product_id	Varchar(50)	รหัสข้อมูล สินค้า ที่ต้องการนำมาใช้งานร่วม
FK	folder_id	Varchar(50)	รหัสข้อมูลโฟลเดอร์ที่ใช้รูปสั่งงาน
FK	image_id	Varchar(50)	รหัสข้อมูลรูปที่สั่งงาน
-	path	Varchar(100)	เส้นทางการเก็บไฟล์สั่งงาน
-	num_img	Integer	จำนวนรูปภาพที่สั่งงาน
-	persent	Integer	เปอร์เซ็นต์แสดงผลการประมวลผล
-	job_status	Integer	สถานะของงานที่สั่ง
-	create_time	Datetime	เวลาที่สั่งงาน

3.6 แผนภารกิจการทำงานแบบลำดับปฏิสัมพันธ์ (Sequence Diagram)

3.6.1 แผนภารกิจการทำงานในส่วนของการสั่งงานประมวลผลภาพ (Web Application)

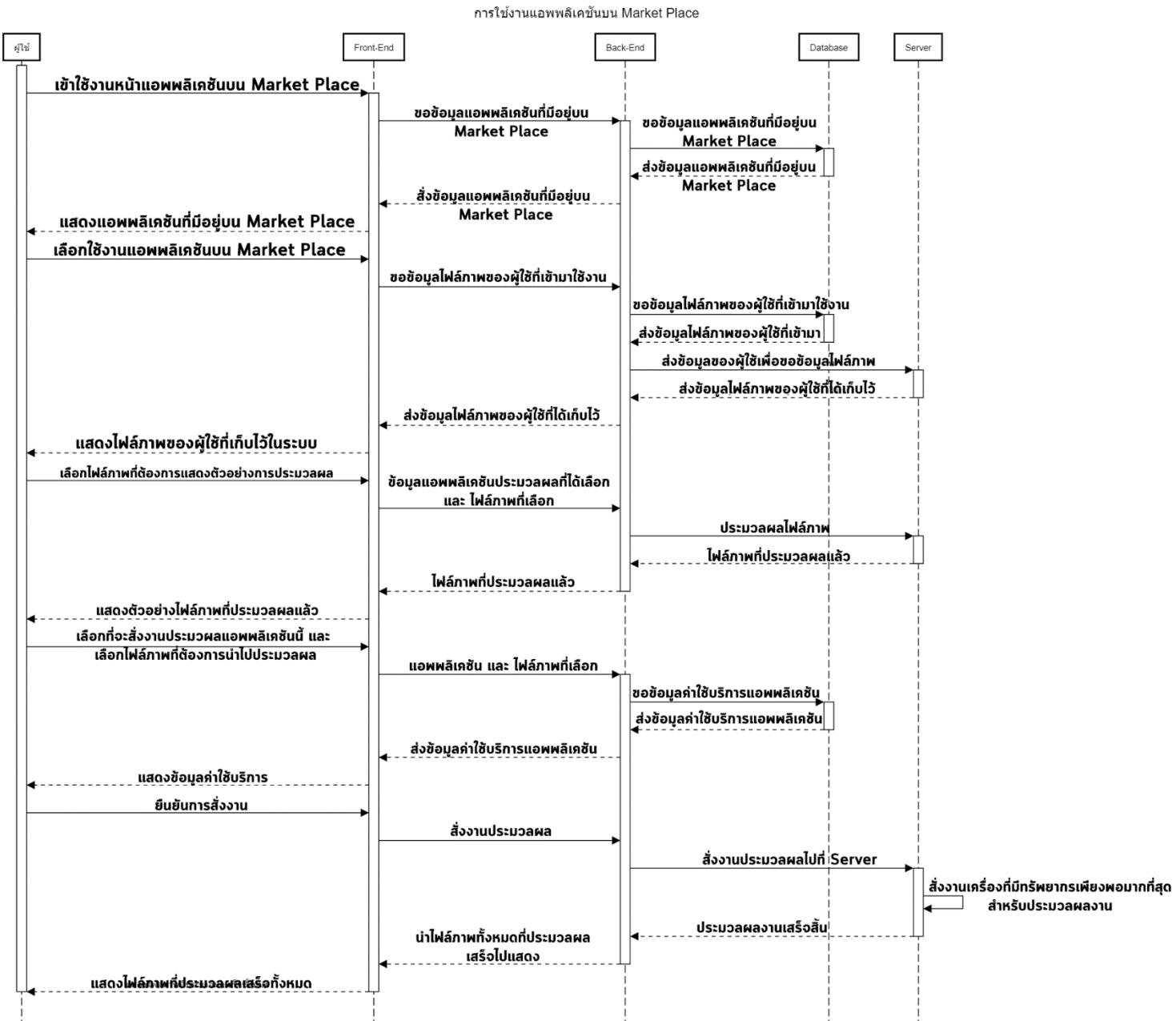
โดยรูปแบบลำดับปฏิสัมพันธ์นี้จะเริ่มตั้งแต่เหตุการณ์ที่ ผู้ใช้งานเริ่มทำการใช้งานการสั่งงาน ประมวลผลภาพจนถึงการประมวลผลภาพเสร็จสิ้น โดยรูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์ โดยรวมในส่วนของการสั่งงานประมวลผลภาพ จะแสดงดังรูปภาพที่ 3.11



รูปที่ 3.11 รูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการสั่งงานประมวลผลภาพ

3.6.2 แผนภาพการทำงานในส่วนของการสั่งงานประมวลผลภารบນ Marketplace (Web Application)

โดยรูปแบบลำดับปฏิสัมพันธ์นี้จะเริ่มตั้งแต่เหตุการณ์ที่ ผู้ใช้งานเริ่มทำการใช้งานการสั่งงาน ประมวลผลภารบນ Marketplace จนถึงการประมวลผลภาพเสร็จสิ้น โดยรูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการสั่งงานประมวลผลภาพ จะแสดงดังรูปภาพที่ 3.12



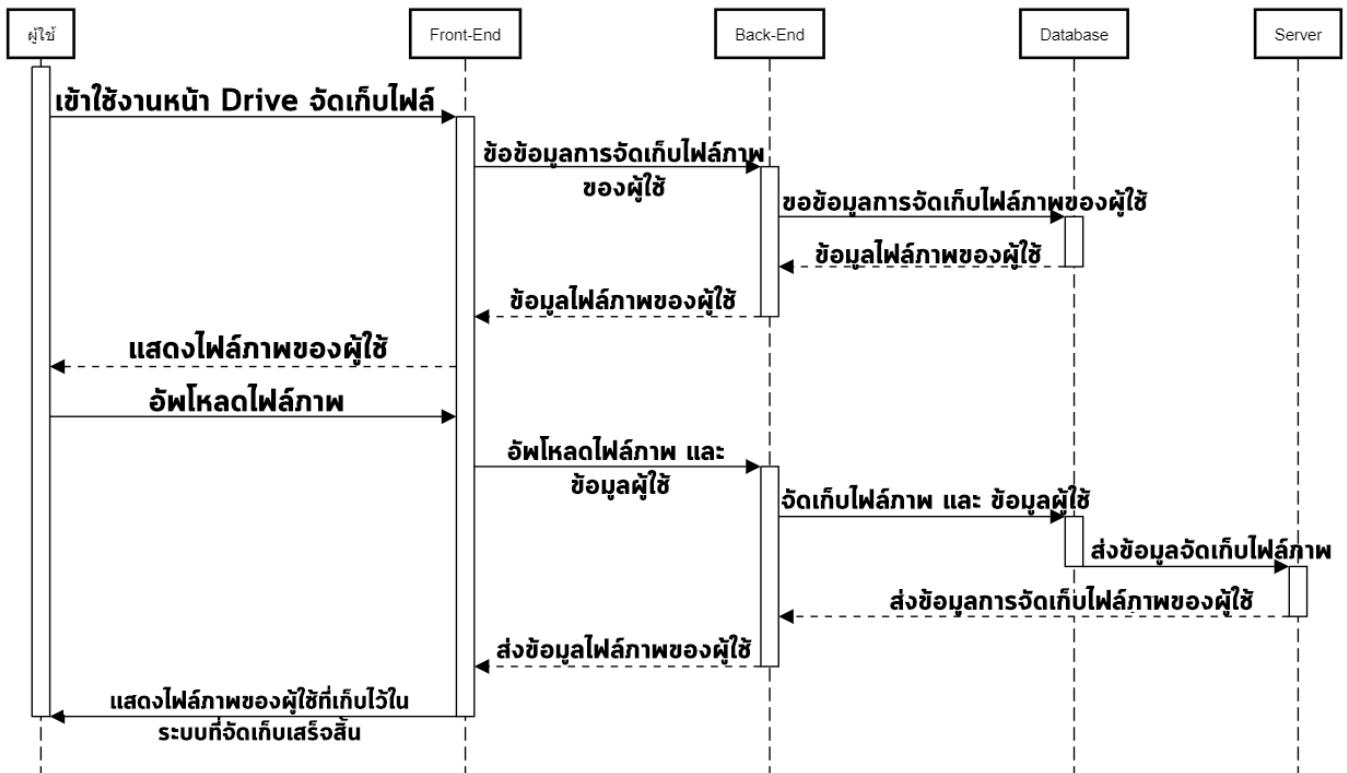
รูปที่ 3.12 รูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการสั่งงานประมวลผลภารบນ Marketplace

Marketplace

3.6.3 แผนภาพการทำงานในส่วนของการจัดเก็บไฟล์ภายใน Drive (Web Application)

โดยรูปแบบลำดับปฏิสัมพันธ์นี้จะเริ่มตั้งแต่เหตุการณ์ที่ ผู้ใช้งานเริ่มทำการจัดเก็บไฟล์ภายใน Drive จนถึงการจัดเก็บไฟล์เสร็จสิ้น โดยรูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการจัดเก็บไฟล์ภายใน Drive จะแสดงดังรูปภาพที่ 3.13

การจัดเก็บไฟล์ภายใน Drive

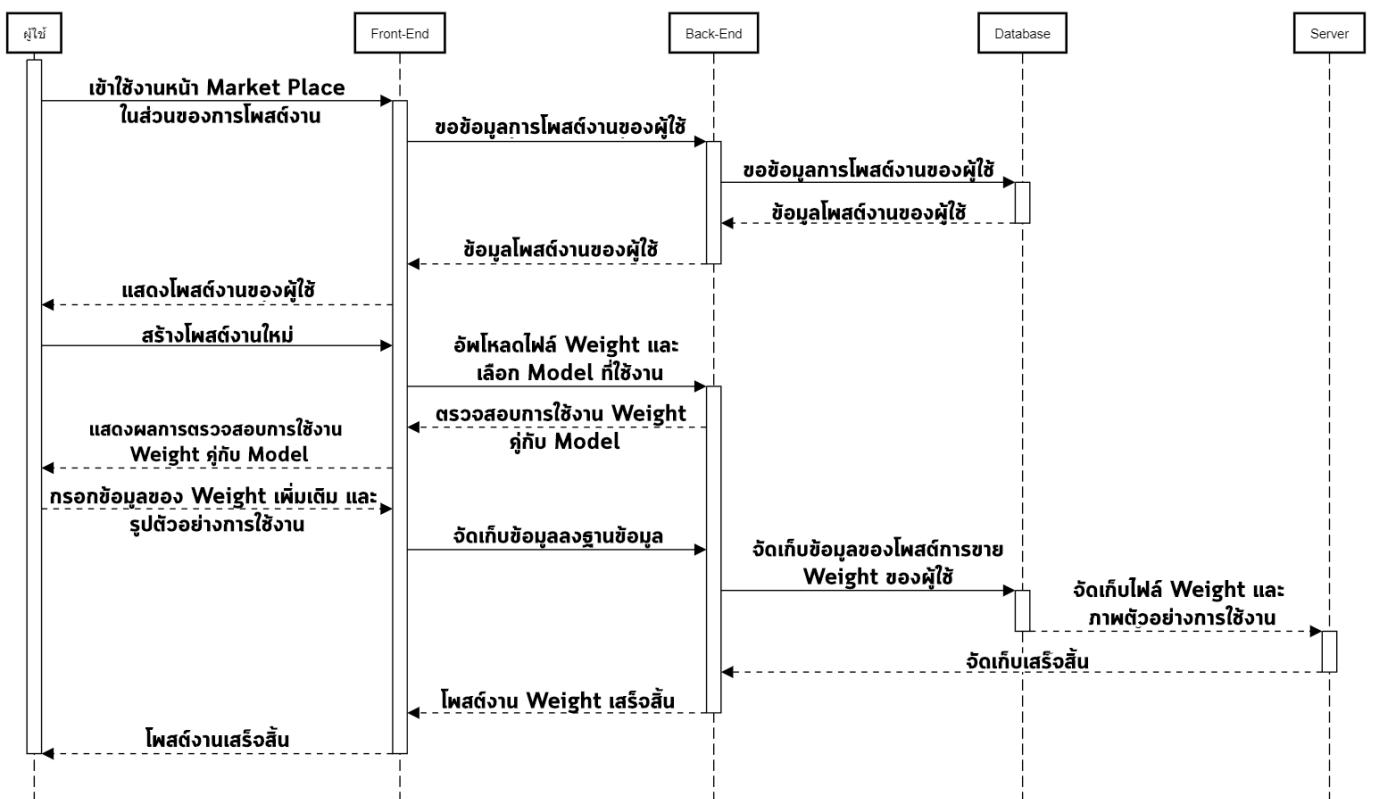


รูปที่ 3.13 รูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการจัดเก็บไฟล์ภายใน Drive

3.6.4 แผนภาพการทำงานในส่วนของการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train (Web Application)

โดยรูปแบบลำดับปฏิสัมพันธ์นี้จะเริ่มตั้งแต่เหตุการณ์ที่ การใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train จนถึงการโพสต์งานเสร็จสิ้น โดยรูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการ Weight ที่ผู้ใช้ได้ทำการ Train จะแสดงดังรูปภาพที่ 3.14

การใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train



รูปที่ 3.14 รูปภาพแสดงการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train

3.7 การออกแบบส่วนต่อประสานกับผู้ใช้ (User Interface)

3.7.1 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนลงชื่อเข้าใช้ (Login)

จะแสดงช่องข้อความสำหรับกรอก Email และรหัสผ่านเพื่อที่จะนำไปยืนยันตัวตนสำหรับเข้าใช้งาน web application โดยการออกแบบในส่วนลงชื่อเข้าใช้ จะแสดงดังรูปภาพที่ 3.15

รูปภาพที่ 3.15 การออกแบบส่วนลงชื่อเข้าใช้

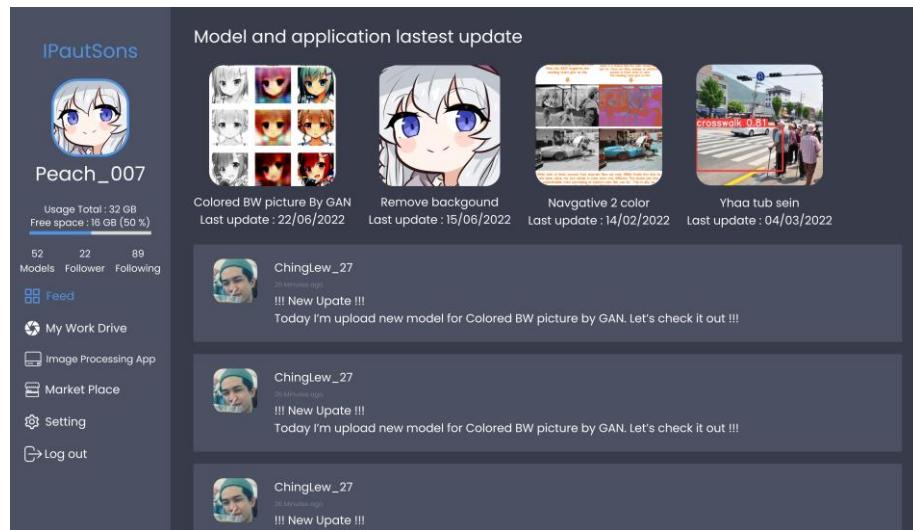
3.7.2 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนลงทะเบียนสมัครสมาชิก (Register)

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

รูปภาพที่ 3.16 การออกแบบส่วนลงทะเบียนสมัครสมาชิก

3.7.3 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนหน้าข่าวสาร (Feed page)

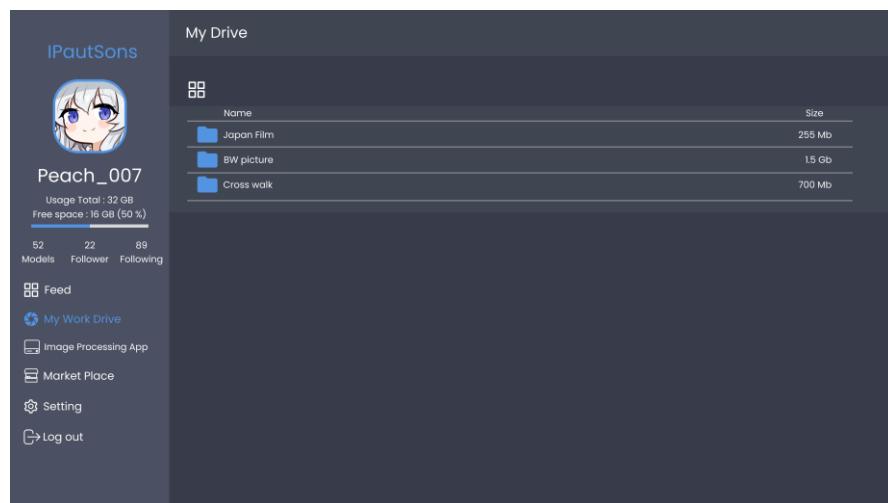
จะแสดงกล่องข้อความที่ประกอบด้วย รูปภาพและชื่อของผู้ตั้งหัวข้อข่าว เวลาที่ตั้งข่าวสาร เนื้อหาของข่าวสาร และ ส่วนที่เป็นหน้าข่าวสารของสินค้า Weight model และ Image processing application ที่มีการ Update ล่าสุด โดยการออกแบบในส่วนหน้าข่าวสาร จะแสดงดังรูปภาพที่ 3.17



รูปภาพที่ 3.17 การออกแบบส่วนหน้าข่าวสาร

3.7.4 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนหน้าพื้นที่เก็บบันทึกรูปภาพ (Drive page)

จะแสดงลิสต์ของโฟลเดอร์ที่เก็บรูปภาพเพื่อให้ผู้ใช้เข้าถึงรูปภาพที่เก็บไว้อยู่ได้ โดยการออกแบบในส่วนพื้นที่เก็บบันทึกรูปภาพ จะแสดงดังรูปภาพที่ 3.18



รูปภาพที่ 3.18 การออกแบบส่วนพื้นที่เก็บบันทึกรูปภาพ

3.7.5 การอุปแบบส่วนต่อประสานกับผู้ใช้ในส่วนพื้นที่แสดงรูปภาพที่เก็บบันทึกไว้ (Show Drive page)

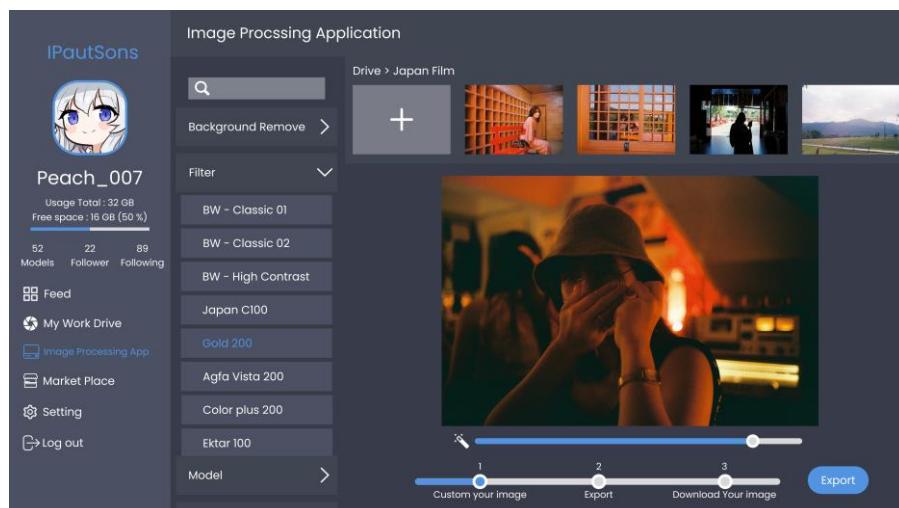
จะแสดงรูปภาพร่วมรายละเอียดของรูปที่อยู่ในโฟลเดอร์ที่เก็บรูปภาพนั้นๆ โดยการอุปแบบในส่วนพื้นที่แสดงรูปภาพที่เก็บบันทึกไว้ จะแสดงดังรูปภาพที่ 3.19



รูปภาพที่ 3.19 การอุปแบบส่วนพื้นที่แสดงรูปภาพที่เก็บบันทึกไว้

3.7.6 การอุปแบบส่วนต่อประสานกับผู้ใช้ในส่วนแอพพลิเคชันประมวลผลภาพ (Image application page)

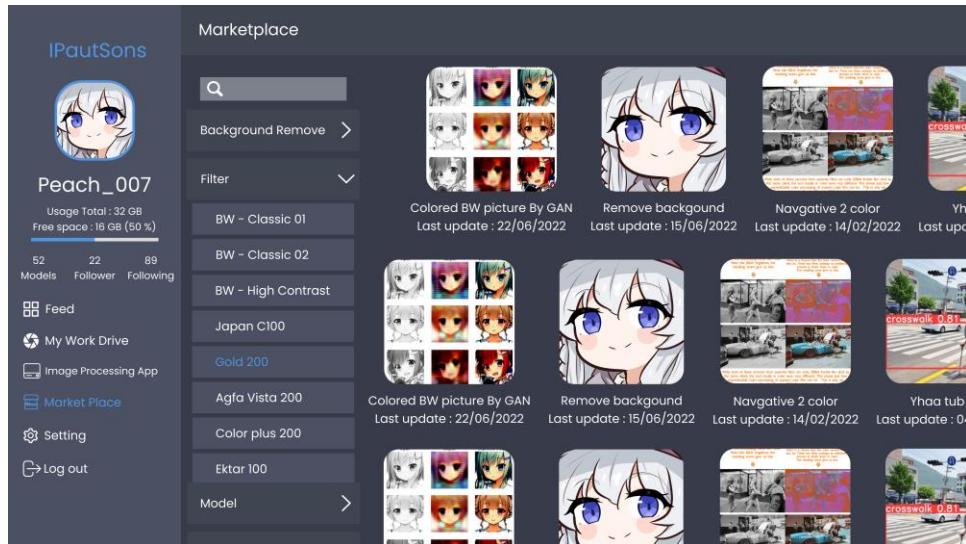
จะแสดงແຄນເລືອກ Application การປ່ຽບແຕ່ງຮູບທີ່ດ້ານຊ້າຍ ແລະ ແສດງຜົດການປ່ຽບແຕ່ງທີ່ທາງດ້ານພື້ນທີ່ຕຽງກາລົງ ໂດຍທີ່ດ້ານນັນຈະເປັນພື້ນທີ່ເລືອກຮູບປາພທີ່ຕ້ອງການ ໂດຍການອຸປະນະໃນສ່ວນເອພພລິເຄື່ນປະມວລຜົດກາພ ຈະແສດງດังຮູບປາພທີ່ 3.20



ຮູບປາພທີ່ 3.20 ການອຸປະນະສ່ວນເອພພລິເຄື່ນປະມວລຜົດກາພ

3.7.7 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนพื้นที่ชื่อขาย (Marketplace page)

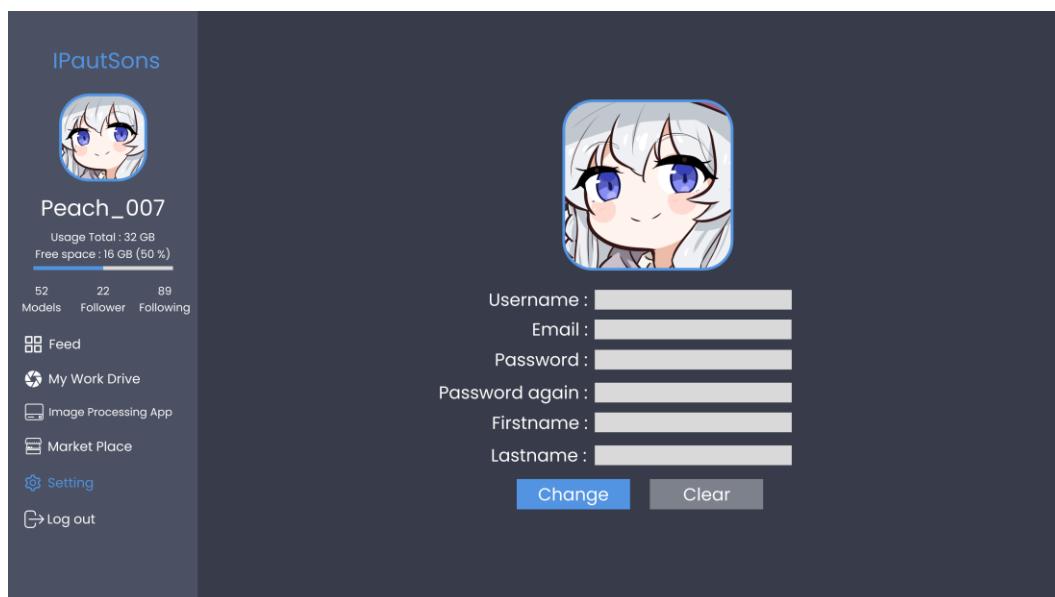
จะแสดงແຄນເລືອກ Application ການປ່ຽນແຕ່ງຮູບທີ່ດ້ານຊ້າຍ ໂດຍຈະເປັນການກວດເລືອກ Application ທີ່ສຳເນົາ ແລະ ໂດຍທີ່ຕຽດການຈະເປັນສິນຄ້າທີ່ນໍາມາຂາຍເຊັ່ນ Weight ທີ່ຝ່ານການ train ມາແລ້ວເປັນດັ່ງ ໂດຍການອອກແບນໃນສ່ວນພື້ນທີ່ເຊື້ອຂາຍ ຈະແສດງດັ່ງຮູບພາບທີ່ 3.21



ຮູບພາບທີ່ 3.21 ການອອກແບນສ່ວນພື້ນທີ່ເຊື້ອຂາຍ

3.7.8 การออกแบบສ່ວນຕ່ອງປະກັນຜູ້ໃຫ້ໃນສ່ວນຕັ້ງຄ່າບໍ່ມີ (Setting page)

ຈະແສດງກລ່ອງຂໍ້ຄວາມທີ່ມີຂໍ້ອມຸລຂອງຜູ້ໃຫ້ຢູ່ ໂດຍສາມາຮອແກ້ໄຂໄດ້ຮັວມໄປຄຶງຮູບໂປຣໄຟລ໌ ເອງທີ່ສາມາຮເປີ່ຍິນໄດ້ ໂດຍການອອກແບນໃນສ່ວນຕັ້ງຄ່າບໍ່ມີ ຈະແສດງດັ່ງຮູບພາບທີ່ 3.22



ຮູບພາບທີ່ 3.22 ການອອກແບນສ່ວນຕັ້ງຄ່າບໍ່ມີ

3.7.9 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนตั้งค่าและคุณนทึกการใช้ทรัพยากร (Monitoring and Config page)

จะแสดงกราฟสถิติการใช้งานฮาร์ดแวร์ต่างๆ เช่น CPU, GPU, RAM เป็นต้น โดยสามารถปรับแต่งการสั่งงานได้ภายในหน้านี้ โดยการออกแบบในส่วนตั้งค่าและคุณนทึกการใช้ทรัพยากร จะแสดงดังรูปภาพที่ 3.23



รูปภาพที่ 3.23 การออกแบบส่วนตั้งค่าและคุณนทึกการใช้ทรัพยากร

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 และสถานะการมีอยู่ของผู้ใช้わ่ายสมัครมาก่อนหน้านี้หรือไม่ รวมไปถึงการตรวจสอบรหัสผ่านเบื้องต้น หากไม่มีข้อผิดพลาดใดๆ ให้ทำการเพิ่มผู้ใช้ไปยังฐานข้อมูล

4.1.1.2 วิธีการทดลอง

ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยการนำข้อมูลผู้ใช้จำนวนหนึ่งไปทดลองกรอกลงในหน้าลงทะเบียนสมัครเข้าใช้งาน และ กดสมัครการใช้งาน 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 สามารถตรวจสอบข้อมูลที่ทำการส่งเข้ามาได้และเพิ่มข้อมูลผู้ใช้เข้าในฐานข้อมูลได้หากข้อมูลที่กรอกเข้ามาไม่มีความสมบูรณ์ และ สามารถส่งข้อความแจ้งเตือนกลับไปในกรณีที่ข้อมูลมีข้อผิดพลาด โดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.1 - 4.3

Email :
63015121@kmit.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': 'lcpcls12cbqd', 'email': '63015121@kmit.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

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

Connect View Collection Help

localhost:27017 ... Documents ipautsons.api_user +

My Queries Databases +

Search

ipautsons.api_user 2 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

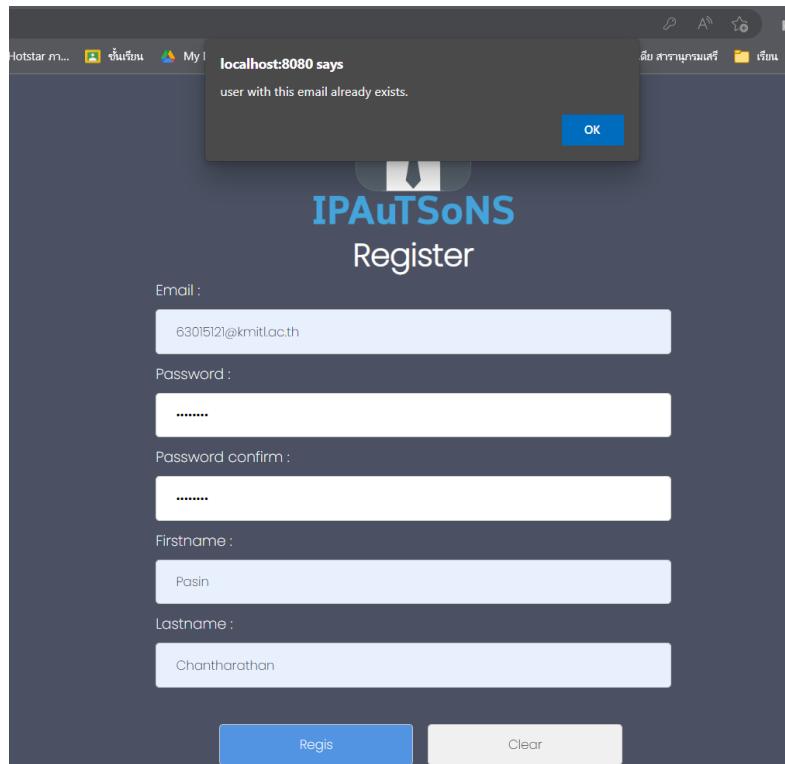
Filter Type a query: { field: 'value' } Reset Find More Options ▾

ADD DATA EXPORT COLLECTION 1-2 of 2 ⏪ ⏴ ⏵ ⏩ ⏷ ⏸ ⏹

```
_id: ObjectId('633e73a16b6268563235a9f3')
last_login: null
user_id: "i6r6mvxxzco"
password: "pbkdf2_sha256$32000$YAx8sCTPZYcwRYtLzdWxCR$Gxw39nf2ypQtvEL6IEa2pAUUpRj_"
email: "pasin98@gmail.com"
is_vip: false
first_name: "Pasin"
last_name: "Chantharathan"
```

```
_id: ObjectId('63919594c3c8722b008001dc')
last_login: null
user_id: "lcpcls12cbqd"
password: "pbkdf2_sha256$32000$0ZXKKqP7FlxLgC4CTtvy34$V/NvMPuqTHCl43NsNrBB8Yy0q9_"
email: "63015121@kmit.ac.th"
is_vip: false
first_name: "Pasin"
last_name: "Chantharathan"
```

รูปภาพที่ 4.2 ข้อมูลที่ทำการสมัครสำเร็จถูกเพิ่มในฐานข้อมูล



รูปภาพที่ 4.3 ข้อความที่แสดงหากอีเมลที่ใช้สมัครนั้นมีอยู่แล้วในฐานข้อมูล

4.2 ส่วนของหน้าลงลงทะเบียนเข้าสู่ระบบใช้งาน

4.2.1 การทดลองระบบลงทะเบียนเข้าสู่ระบบเพื่อผู้ใช้ระดับ Client เข้าใช้งาน Web application

4.2.1.1 จุดประสงค์การทดลอง

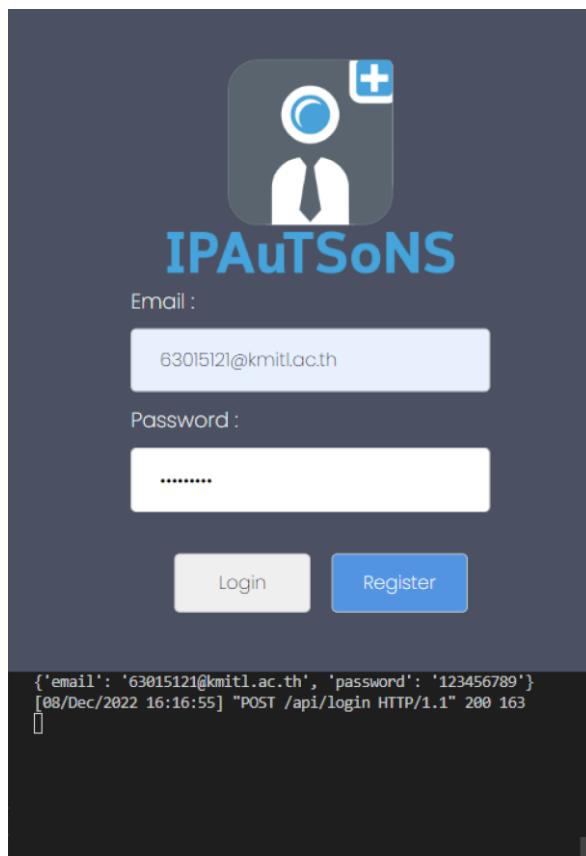
เพื่อทดสอบการทำงานของ Web application ส่วนหน้าลงลงทะเบียนเข้าสู่ระบบใช้งานว่าสามารถทำงานถูกต้อง สามารถส่งข้อมูลจากหน้าลงลงทะเบียนเข้าสู่ระบบไปยังส่วน Backend ได้อย่างครบถ้วน และนำข้อมูลที่ไปตรวจสอบเทียบกับข้อมูลผู้ใช้ในฐานข้อมูลเพื่อทำการยืนยันตัวตนผู้ใช้ได้ และสามารถแจ้งเตือนให้ผู้ใช้ได้หากมีข้อมูลส่วน个体พลด และ สามารถส่ง Cookie กลับมายัง Web application เพื่อใช้ในการยืนยันตัวตนได้

4.2.1.2 วิธีการทดลอง

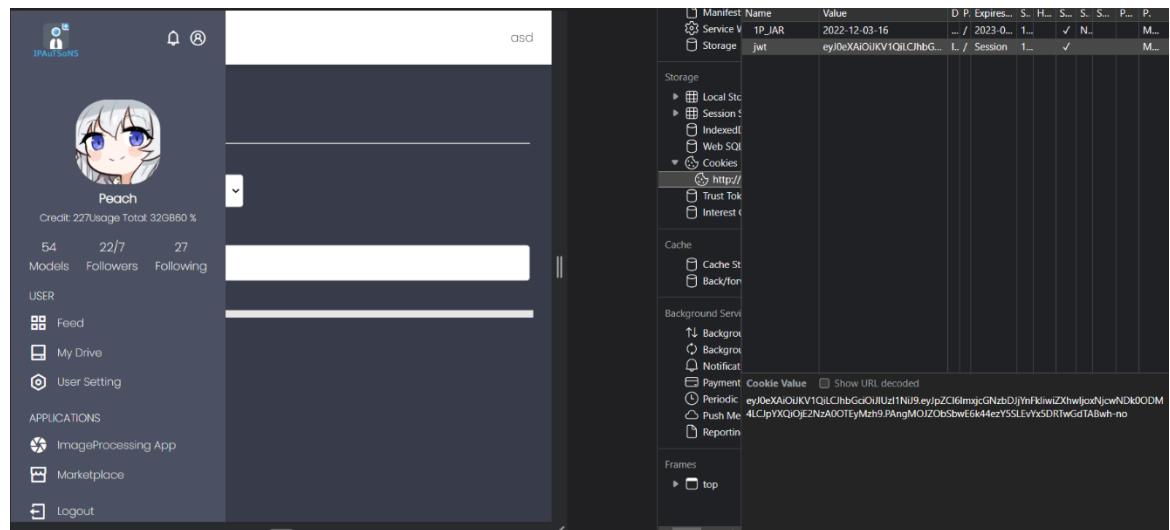
ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยการนำข้อมูลผู้ใช้ที่มีและไม่มีในฐานข้อมูลจำนวนหนึ่งไปทดลองกรอกลงในหน้าลงลงทะเบียนเข้าสู่ระบบเข้าใช้งาน และ เข้าสู่ระบบ หากการทดลองสำเร็จ ข้อมูลผู้ใช้ที่ทำการลงทะเบียนเข้าสู่ระบบที่หน้า Web Application จะถูกส่งไปตรวจสอบที่ส่วน Backend ได้ และ ถ้าข้อมูลถูกต้องผู้ใช้จะเข้าใช้งาน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

4.2.1.3 ผลการทดสอบ

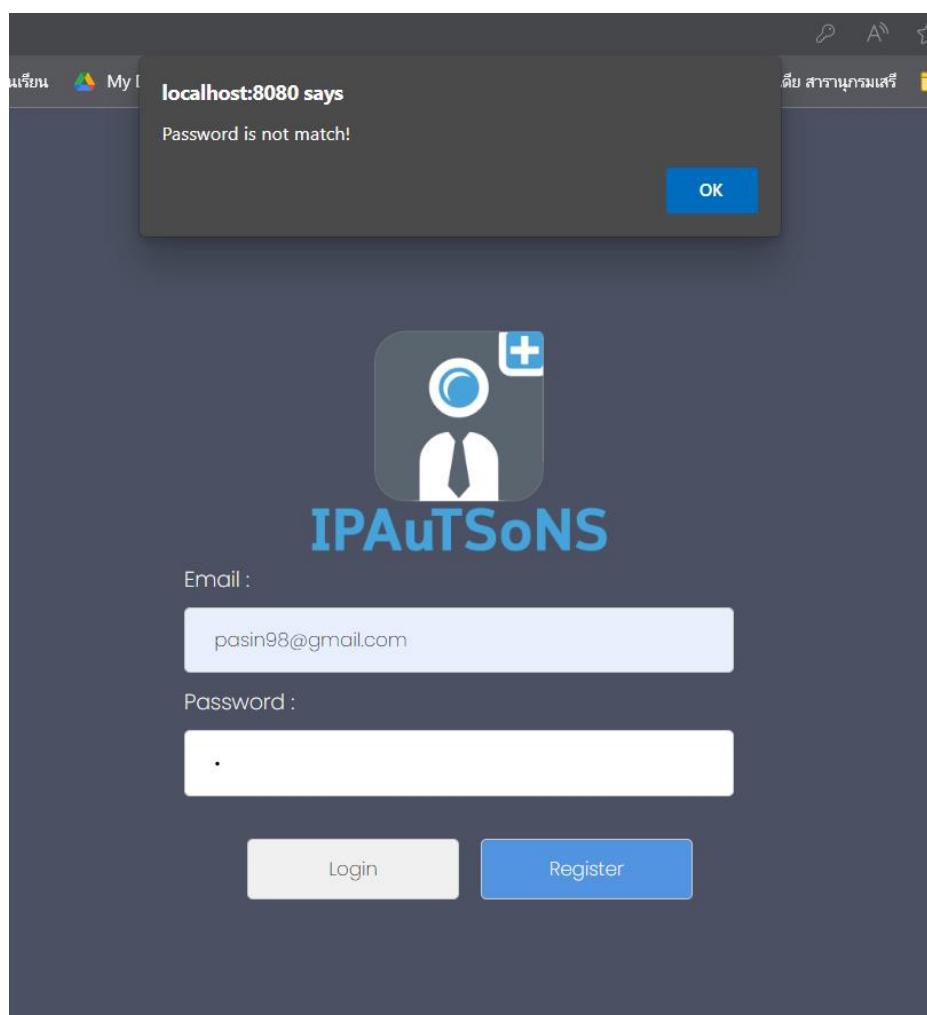
ส่วนของการลงทะเบียนเข้าสู่ระบบสามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Back end ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถตรวจสอบข้อมูลที่ได้ทำการส่งเข้ามาและยืนยันตัวตนโดยเบริญเทียบกับข้อมูลของผู้ใช้ในฐานข้อมูล ได้หากข้อมูลของผู้ใช้ที่กรอกเข้ามามีความถูกต้องจะส่ง Cookie กลับไปยังเว็บแอปพลิเคชันเพื่อใช้ยืนยันตัวตนผู้ใช้ และ ในการมีที่เกิดข้อผิดพลาดขึ้น เช่น ไม่พบ Email ของผู้ใช้ในระบบ หรือ รหัสผ่านผิดพลาดก็จะทำการส่งข้อความแจ้งเตือนกลับไปแสดงให้ผู้ใช้ทราบ โดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.4 - 4.6



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



รูปภาพที่ 4.5 ข้อมูล Cookie ของผู้ใช้หากทำการเข้าสู่ระบบสำเร็จ



รูปภาพที่ 4.6 ข้อความที่แสดงหากรหัสผ่านที่กรอกเข้าสู่ระบบนั้นไม่ถูกต้อง

4.3 ส่วนของหน้าแก้ไขข้อมูลผู้ใช้งาน

4.3.1 การทดลองระบบแก้ไขข้อมูลผู้ใช้ระดับ Client

4.3.1.1 จุดประสงค์การทดลอง

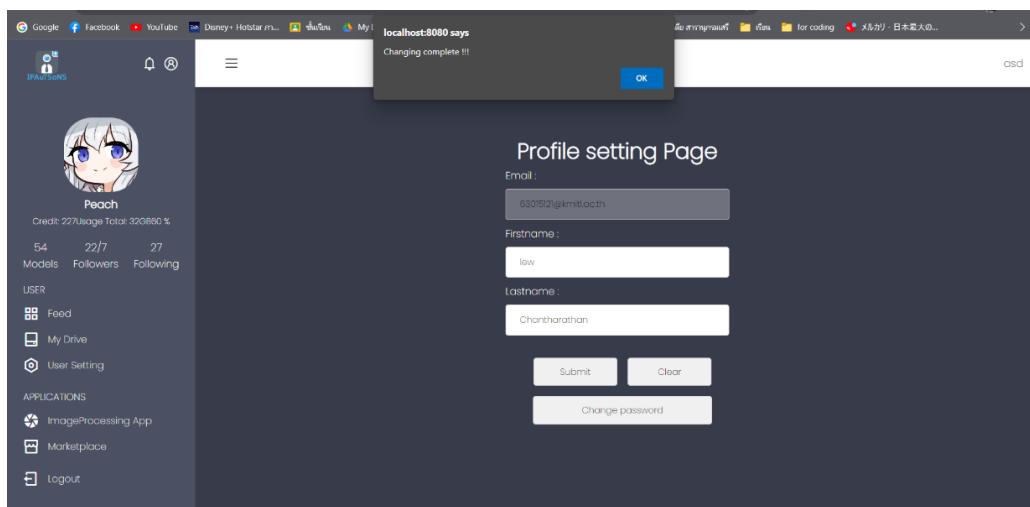
เพื่อทดสอบการทำงานของ Web applications ส่วนแก้ไขข้อมูลผู้ใช้ว่าสามารถทำงานถูกต้องสามารถส่งข้อมูลจากหน้าแก้ไขข้อมูลผู้ใช้และ Cookie ยืนยันตัวตนไปยังส่วน Backend ได้อย่างครบถ้วนและนำข้อมูลที่ทำการแก้ไขไปเปลี่ยนแปลงข้อมูลในฐานข้อมูลได้ โดยสามารถแจ้งเตือนผู้ใช้ได้หากมีข้อผิดพลาดเกิดขึ้น

4.3.1.2 วิธีการทดลอง

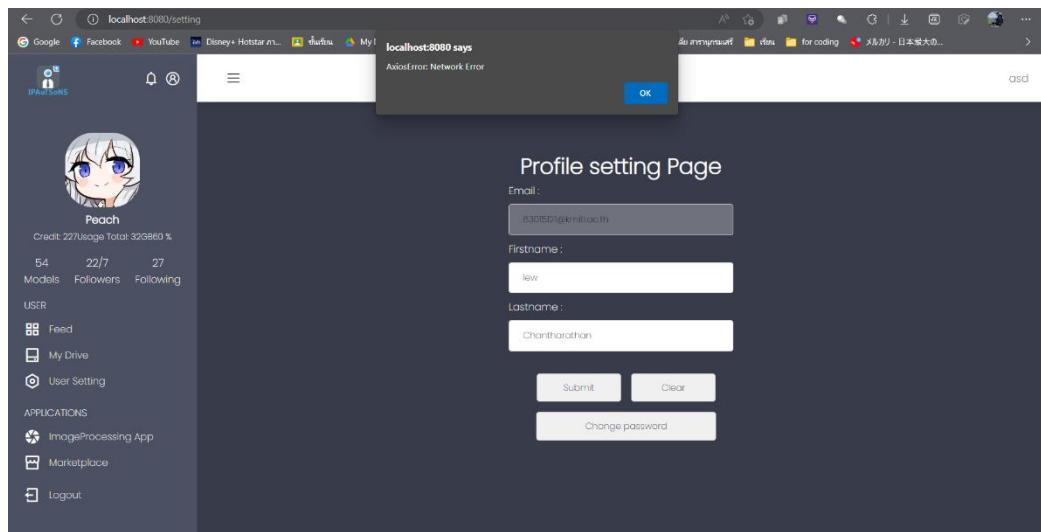
ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยการนำชุดข้อมูลที่คล้ายกับข้อมูลผู้ใช้นำไปทดลองกรอกลงในหน้าแก้ไขข้อมูลผู้ใช้และทำการกดเปลี่ยนแปลงแปลง หากสำเร็จข้อมูลที่ผู้ใช้กรอกเข้าไปเพื่อทำการเปลี่ยนแปลงและ Cookie ยืนยันตัวตนจะถูกส่งไปส่วนตรวจสอบที่ Backend ซึ่งส่วน Backend จะทำการยืนยันตัวตนของผู้ใช้หากสำเร็จจะทำการเปลี่ยนแปลงข้อมูลของผู้ใช้ในฐานข้อมูลแต่ถ้าหากยืนยันตัวตนไม่สำเร็จหรือเกิดข้อผิดพลาดขึ้นก็จะแสดงข้อความตอบกลับ และทดลองปิดการทำงานของฐานข้อมูลเพื่อทดสอบการแจ้งเตือนหากมีข้อผิดพลาดเกิดขึ้น

4.3.1.3 ผลการทดลอง

ส่วนของการแก้ไขข้อมูลผู้ใช้สามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Backend ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถตรวจสอบข้อมูลที่ได้ทำการส่งเข้ามา และ ยืนยันตัวตนโดยเบริญเทียบ Cookie กับข้อมูลของผู้ใช้ในฐานข้อมูล และ เมื่อทำการยืนตัวตนเสร็จก็ทำการเปลี่ยนแปลงข้อมูลของผู้ใช้ในฐานข้อมูล และ เมื่อมีข้อผิดพลาดเกิดขึ้นก็จะทำการแจ้งเตือนให้ผู้ใช้ทราบ โดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.7 - 4.8



รูปภาพที่ 4.7 ข้อความที่แสดงหากสามารถเปลี่ยนข้อมูลผู้ใช้ได้สำเร็จ



รูปภาพที่ 4.8 ข้อความที่แสดงหากเกิดข้อผิดพลาดขึ้นขณะเปลี่ยนแปลงข้อมูลผู้ใช้

4.3.2 การทดสอบระบบแก้ไขรหัสผ่านผู้ใช้ระดับ Client

4.3.2.1 จุดประสงค์การทดสอบ

เพื่อทดสอบการทำงานของ Web applications ส่วนแก้ไขรหัสผ่านผู้ใช้ว่าสามารถทำงาน

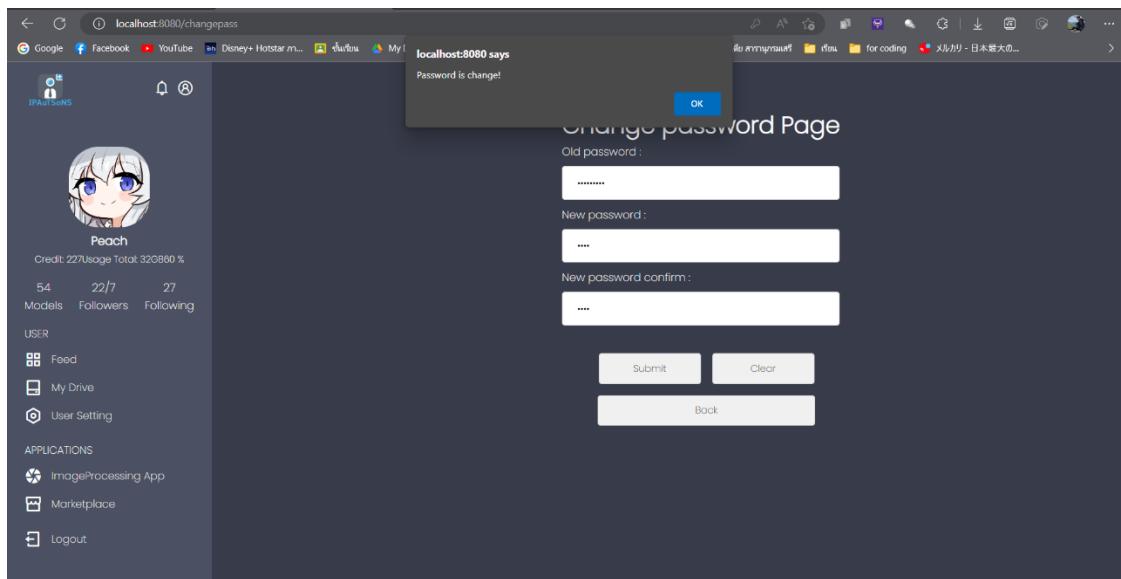
ถูกต้องสามารถส่งข้อมูลจากหน้าแก้ไขรหัสผ่านผู้ใช้และ Cookie ยืนยันตัวตนไปยังส่วน Backend ได้ อย่างครบถ้วนและนำรหัสผ่านที่ทำการแก้ไขไปเปลี่ยนแปลงข้อมูลในฐานข้อมูลได้ โดยสามารถแจ้งเตือนผู้ใช้ได้หากมีข้อผิดพลาดเกิดขึ้น

4.3.2.2 วิธีการทดสอบ

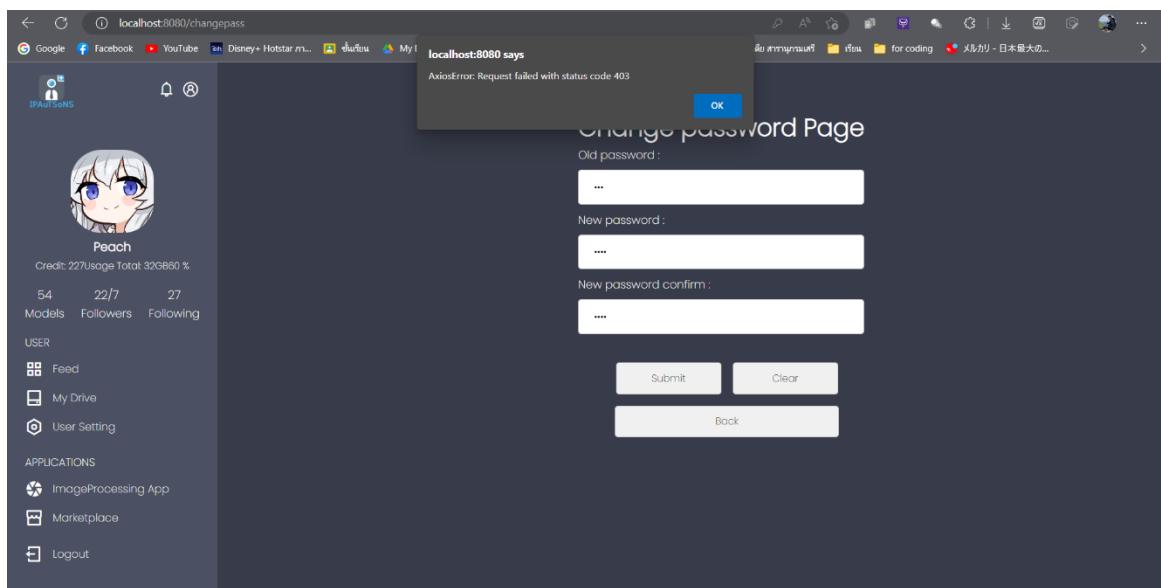
ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยการนำชุดข้อมูลรหัสผ่านผู้ใช้เข้าไปทดสอบ กรอกลงในหน้าแก้ไขรหัสผ่านผู้ใช้และทำการกดเปลี่ยนแปลงแปลง หากสำเร็จรหัสผ่านที่ผู้ใช้กรอกเข้าไปเพื่อทำการเปลี่ยนแปลง และ Cookie ยืนยันตัวตนจะถูกส่งไปส่วนตรวจสอบที่ Backend ซึ่งส่วน Backend จะทำการยืนยันตัวตนของผู้ใช้หากสำเร็จจะทำการเปลี่ยนแปลงรหัสผ่านของผู้ใช้ในฐานข้อมูล แต่ถ้าหากยืนยันตัวตนไม่สำเร็จหรือเกิดข้อผิดพลาดขึ้นก็จะแสดงข้อความตอบกลับ และทดสอบปิดการทำงานของฐานข้อมูลเพื่อทดสอบการแจ้งเตือนหากมีข้อผิดพลาดเกิดขึ้น

4.3.2.3 ผลการทดสอบ

ส่วนของการแก้ไขรหัสผ่านผู้ใช้สามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Backend ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถตรวจสอบรหัสผ่านที่ได้ทำการส่งเข้ามา และ ยืนยันตัวตนโดยเปรียบเทียบ Cookie กับรหัสผ่านของผู้ใช้ในฐานข้อมูล และ เมื่อทำการยืนยัน เสร็จก็ทำการเปลี่ยนแปลงรหัสผ่านของผู้ใช้ในฐานข้อมูล และ เมื่อมีข้อผิดพลาดเกิดขึ้นก็จะทำการแจ้งเตือนให้ผู้ใช้ทราบโดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.9 - 4.10



รูปภาพที่ 4.9 ข้อความที่แสดงหากสามารถเปลี่ยนข้อมูลผู้ใช้ได้สำเร็จ



รูปภาพที่ 4.10 ข้อความที่แสดงหากเกิดข้อผิดพลาดขึ้นขณะเปลี่ยนแปลงข้อมูลผู้ใช้

4.4 ส่วนของหน้าจัดการไฟล์

4.4.1 การทดลองระบบจัดการ Folder สำหรับเก็บไฟล์รูปภาพของผู้ใช้

4.4.1.1 จุดประสงค์การทดลอง

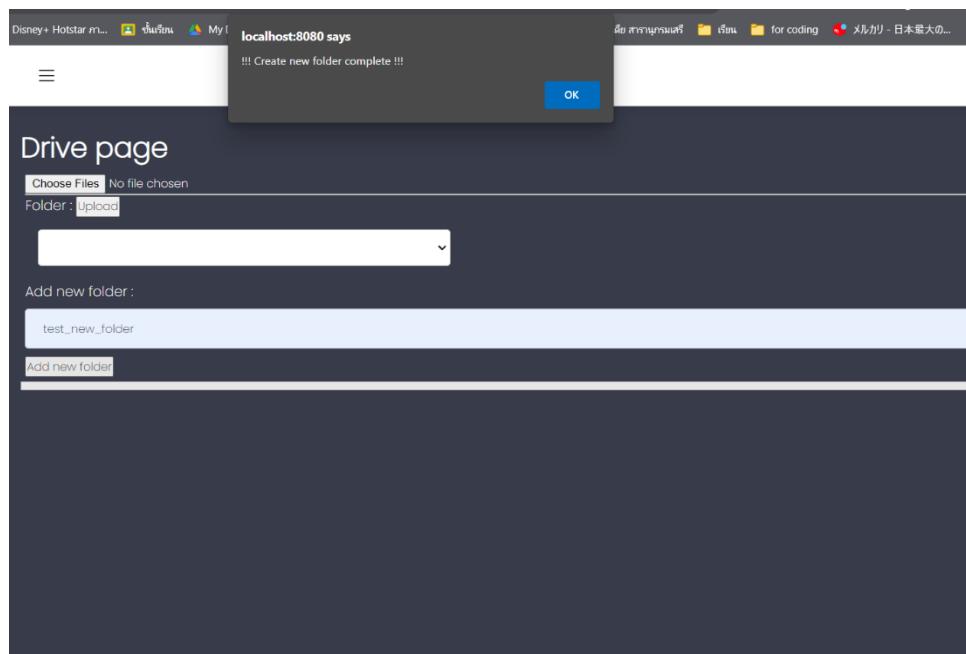
เพื่อทดสอบการทำงานของ Web applications หน้าจัดการไฟล์ในส่วนจัดการ Folder สำหรับเก็บไฟล์รูปภาพของผู้ใช้ว่าสามารถทำงานอย่างต้อง สามารถเพิ่ม Folder และ ลบ Folder ไปได้โดยมีการยืนยันตัวตนจาก Cookie ที่เก็บเอาไว้ โดยจะเพิ่ม-ลบข้อมูลของ Folder ที่ฐานข้อมูลด้วย และ มีการตรวจสอบชื่อ Folder โดยถ้าหากมีข้อผิดพลาดก็จะแสดงข้อความแจ้งเตือนข้อผิดพลาดขึ้นมา

4.4.1.2 วิธีการทดลอง

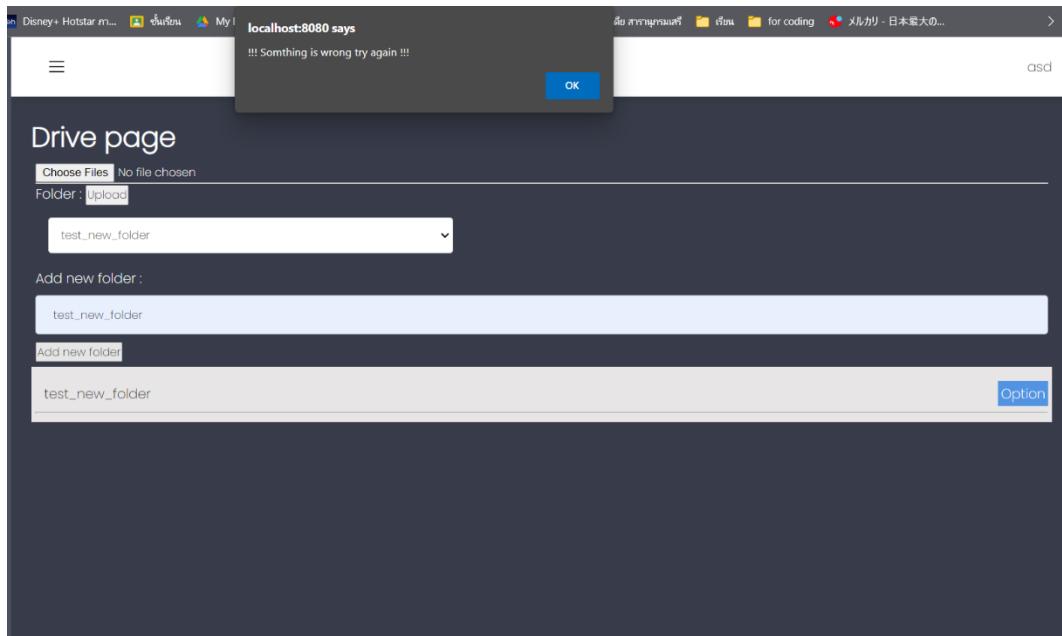
ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยทำการเพิ่ม-ลบ Folder เก็บไฟล์ในหลายๆ รูปแบบแบบที่ต่างกันและซ้ำกัน หากทำงานได้สำเร็จจะมีข้อความแสดงผลบอกว่าทำการเพิ่ม-ลบ Folder สำเร็จ และทำการเพิ่ม-ลบข้อมูลของ Folder ในฐานข้อมูล โดยที่การทำงานหากมีข้อผิดพลาดก็จะส่งแสดงผลข้อความว่าการทำงานนั้นไม่สำเร็จ

4.4.1.3 ผลการทดลอง

ส่วนของระบบจัดการ Folder สำหรับเก็บไฟล์รูปภาพสามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Backend ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถยืนยันตัวตนผู้ใช้ที่ได้ทำการล็อกอิน และทำการ เพิ่ม-ลบ Folder เก็บรูปภาพได้ และสามารถตรวจสอบชื่อ Folder ที่ซ้ำกันได้รวมไปถึงทำการเพิ่ม-ลบข้อมูลของ Folder ในฐานข้อมูลได้ และสามารถส่งข้อความแจ้งเตือนได้หากเกิดการทำงานที่ผิดพลาดขึ้น โดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.11 - 4.12



รูปภาพที่ 4.11 ข้อความที่แสดงหากสามารถเพิ่ม Folder สำหรับเก็บรูปภาพได้



รูปภาพที่ 4.12 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะเพิ่ม Folder สำหรับเก็บรูปภาพ

4.4.2 การทดลองระบบจัดการไฟล์รูปภาพของผู้ใช้

4.4.2.1 จุดประสงค์การทดลอง

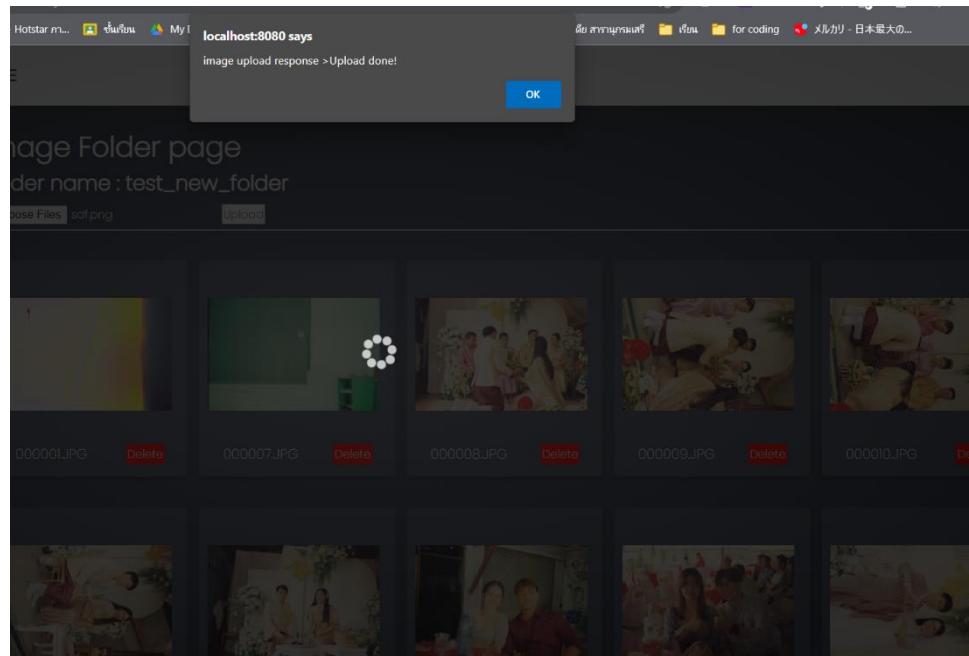
เพื่อทดสอบการทำงานของ Web applications หน้าจัดการไฟล์ในส่วนจัดการไฟล์รูปภาพ ของผู้ใช้สามารถทำงานถูกต้อง สามารถเพิ่ม รูปภาพ และ ลบ รูปภาพ ไปได้โดยมีการส่ง Cookie ไปยัง Backend เพื่อยืนยันตัวตน โดยจะเพิ่ม-ลบข้อมูลของ รูปภาพ ที่ฐานข้อมูลด้วย โดยถ้าหากมีข้อผิดพลาดก็จะแสดงข้อความแจ้งเตือนข้อผิดพลาดขึ้นมา

4.4.2.2 วิธีการทดลอง

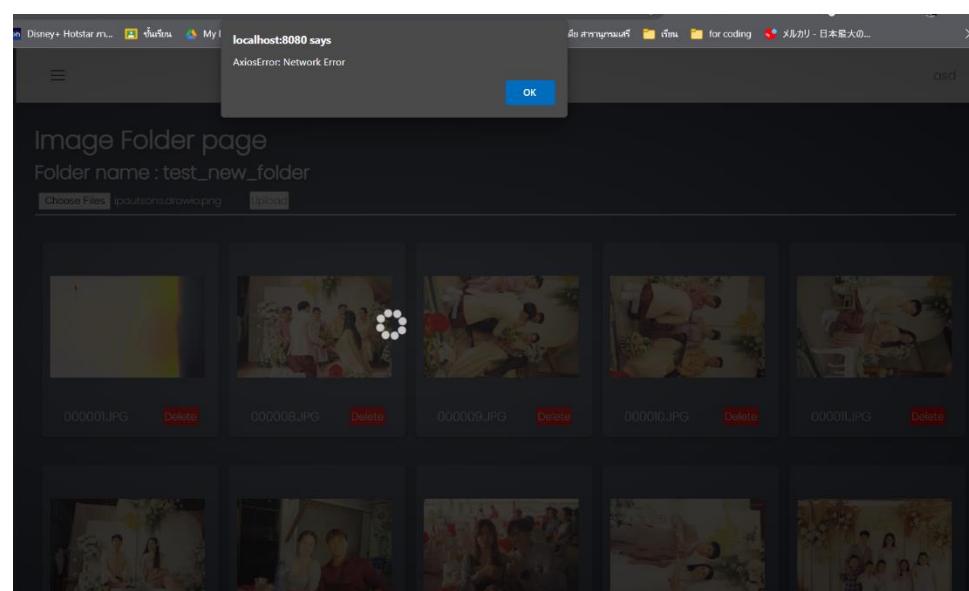
ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยทำการเพิ่ม-ลบ Folder เก็บไฟล์ในหลายรูปแบบที่ต่างกันและซ้ำกัน หากทำงานได้สำเร็จจะมีข้อความแสดงผลบอกร่วมกับการทำการเพิ่ม-ลบ Folder สำเร็จ และ ทำการเพิ่ม-ลบข้อมูลของ Folder ในฐานข้อมูล โดยที่การทำงานหากมีข้อผิดพลาดก็จะส่งแสดงผลข้อความว่าการทำงานนั้นไม่สำเร็จ

4.4.2.3 ผลการทดลอง

ส่วนของระบบจัดการไฟล์รูปภาพสามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Backend ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถยืนยันตัวตนผู้ใช้ที่ได้ทำการส่งเข้ามา และทำการ เพิ่ม-ลบรูปภาพได้ รวมไปถึงทำการเพิ่ม-ลบข้อมูลของรูปภาพในฐานข้อมูลได้ และสามารถส่งข้อความแจ้งเตือนได้หากเกิดการทำงานที่ผิดพลาดขึ้น โดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.13



รูปภาพที่ 4.13 ข้อความที่แสดงหากสามารถเพิ่มรูปภาพสำเร็จ



รูปภาพที่ 4.14 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะเพิ่มรูปภาพ

4.5 ส่วนของหน้าสั่งงานประมวลผล

4.5.1 การทดลองระบบการสร้างไฟล์การสั่งงานประมวลผลภาพ

4.5.1.1 จุดประสงค์การทดลอง

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

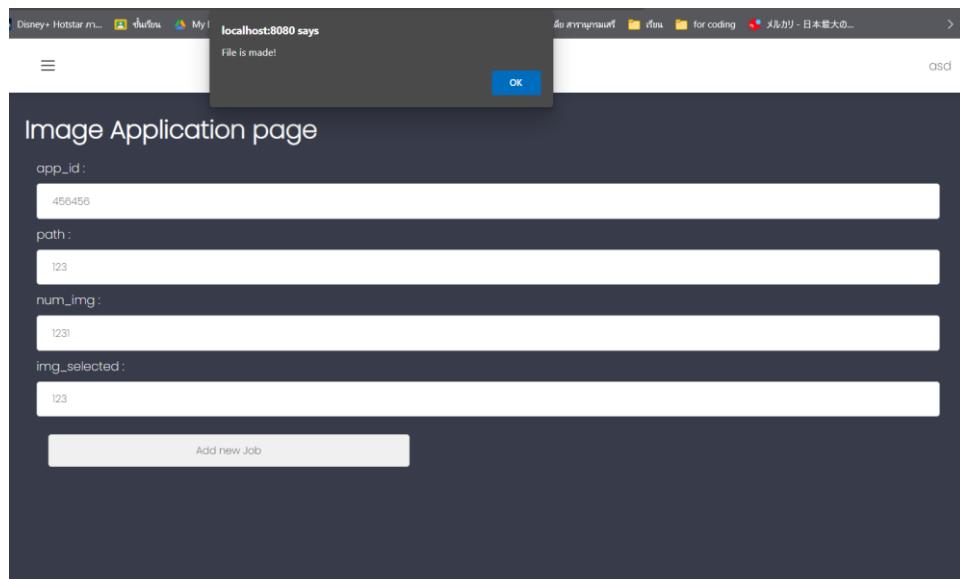
4.5.1.2 วิธีการทดลอง

ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยทำการเพิ่มงานประมวลผลภาพ ในหลายๆ รูปแบบที่ต่างกัน หากทำงานได้สำเร็จจะมีข้อความแสดงผลบอกว่าทำการสร้างงานประมวลผลสำเร็จ และทำการเพิ่มข้อมูลของงานประมวลผล ในฐานข้อมูล โดยที่การทำงานหากมีข้อผิดพลาดก็จะส่งแสดงผลข้อความว่าการทำงานนั้นไม่สำเร็จ

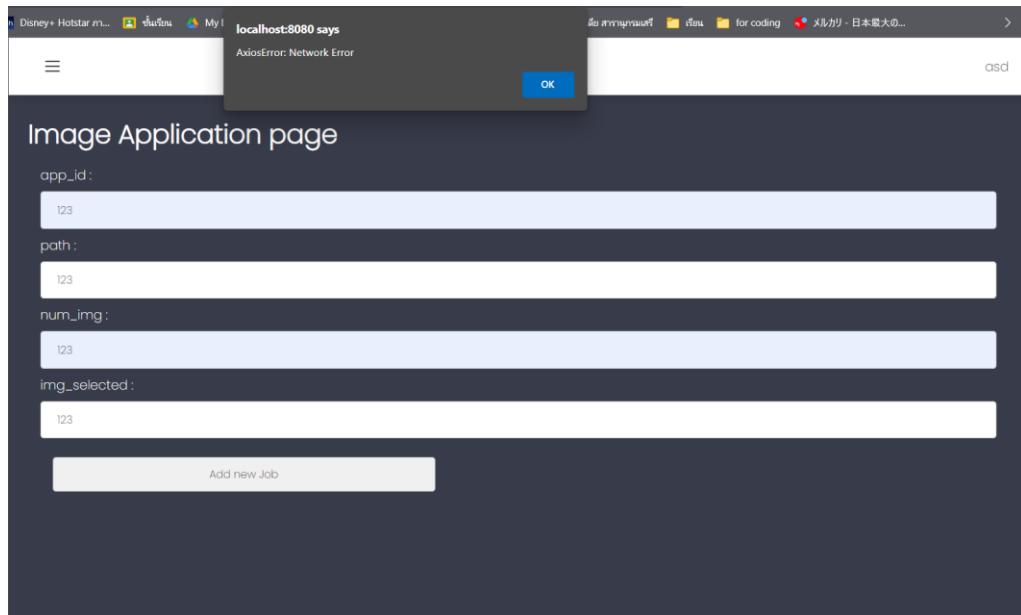
4.5.1.3 ผลการทดลอง

ส่วนของระบบจัดการไฟล์รูปภาพสามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Backend ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถยืนยันตัวตนผู้ใช้ที่ได้ทำการส่งเข้ามา และทำการเพิ่ม-ลบรูปภาพได้ รวมไปถึงทำการเพิ่ม-ลบข้อมูลของรูปภาพในฐานข้อมูลได้ และสามารถส่งข้อความแจ้งเตือนได้หากเกิดการทำงานที่ผิดพลาดขึ้น โดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.15

- 4.16



รูปภาพที่ 4.15 ข้อความที่แสดงหากสามารถสร้างไฟล์งานประมวลผลสำเร็จ



รูปภาพที่ 4.16 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะสร้างไฟล์งานประมวลผล

4.6 ส่วนของการแบ่งภาระงาน

4.6.1 การทดลองระบบการการแบ่งภาระงานของ Web Service

4.6.1.1 จุดประสงค์การทดลอง

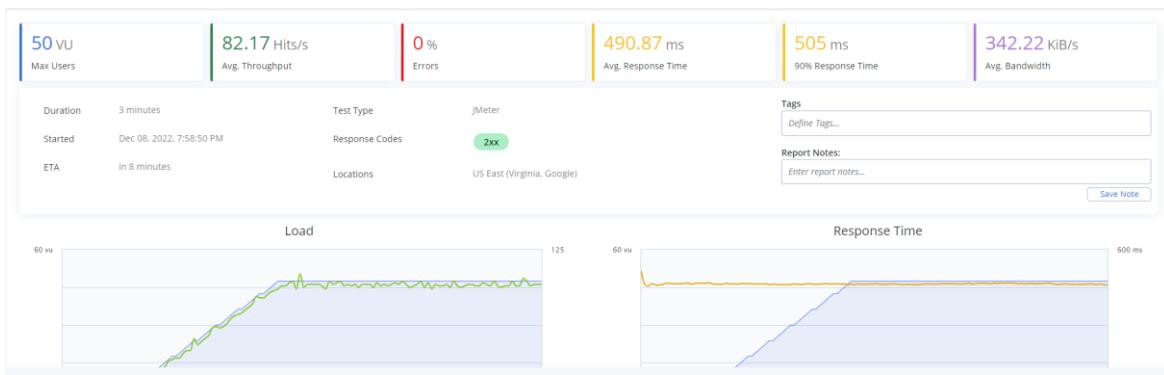
เพื่อทดสอบการแบ่งภาระงานของการเข้าใช้งาน Web Service โดยที่มีการกระจายผู้ใช้งานเข้าไปยังเครื่องอื่น ๆ บนระบบ Cluster เพื่อกระจายทรัพยากรในการเข้าใช้งานไปให้มีประสิทธิภาพ

4.6.1.2 วิธีการทดลอง

ทดสอบผ่าน Demo Web Application ที่พัฒนาขึ้นมาสำหรับการทดสอบโดยเนพาะ โดยการนำเครื่องมือในการส่ง Request เข้ามาจำนวน 50 คนพร้อม ๆ กันในเวลา 10 นาทีเพื่อตรวจสอบว่าสามารถเข้าใช้งานได้

4.6.1.3 ผลการทดลอง

จากการทดลองกระจายภาระงานของ Web Service สามารถรองรับการใช้งานพร้อมกันได้เฉลี่ย 8 คนต่อ 1 เครื่องใน Cluster โดยที่หลังจาก 8 คนไปประสิทธิภาพจะลดลงเรื่อย ๆ และ มีการตอบสนองของ Server ที่ช้าลงจากรูป 4.17



รูปภาพที่ 4.17 ผลการทดสอบสิ่ง Request เข้ามาใช้งาน

4.5.2 การทดสอบระบบการแบ่งภาระงานของ Application

4.5.2.1 จุดประสงค์การทดสอบ

เพื่อทดสอบการแบ่งภาระงานของการสั่งงานประมวลผล Application โดยที่มีการกระจายงานไปยังเครื่องอื่น ๆ บนระบบ Cluster เพื่อกระจายทรัพยากรในการเข้าใช้งานไปให้มีประสิทธิภาพ

4.5.2.2 วิธีการทดสอบ

ทดสอบผ่าน Demo Application ที่พัฒนาขึ้นมาสำหรับการทดสอบโดยเฉพาะ โดยการนำการจับเวลาในการประมวลผลเข้าไปเพื่อทดสอบว่าเมื่อมีภาระงานมากขึ้นงานประมวลผลจะช้าลงเท่าใด

4.5.2.3 ผลการทดสอบ

จากการทดสอบเมื่อมีภาระงานสั่งงานประมวลผล 5 งานขึ้นไปต่อเครื่องจะเริ่มมีการประมวลผลที่ช้าลงไปเรื่อย ๆ ตามจำนวนงานที่เพิ่มขึ้นมา

4.7 ส่วนของ Application Image Processing

4.6.1 การทดสอบระบบการทำงานของ Application Image Processing

4.6.1.1 จุดประสงค์การทดสอบ

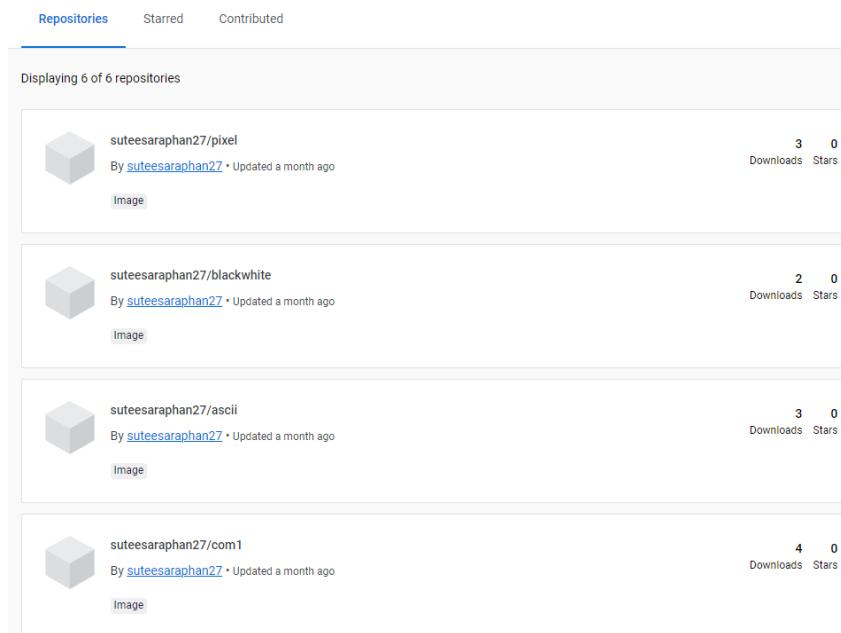
เพื่อทดสอบการทำงานของ Application บนระบบ Cluster สามารถทำงานได้อย่างถูกต้อง และ มีประสิทธิภาพในการทำงานได้อย่างดีเยี่ยม

4.6.1.2 วิธีการทดสอบ

ทดสอบการสั่งงานด้วย Kubernetes ในการจัดเก็บ Application ไว้บน DockerHub และ ทดสอบด้วยรูปภาพภายในเครื่อง Cluster สามารถถูกประมวลผลได้อย่างถูกต้อง

4.6.1.3 ผลการทดลอง

จากการทดลองสามารถประมวลผลภาพได้ถูกต้องบนระบบ Cluster ด้วย Kubernetes แต่มีเวลาในการโหลดครั้งแรกของการสั่งงานที่ค่อนข้างนานเนื่องจากต้องไปทำการโหลดมาจาก DockerHub ก่อนที่จะเริ่มทำงานประมวลผลได้ ตัวอย่างจากรูป 4.18 – 4.19



รูปภาพที่ 4.18 Repositories ของ DockerHub ที่ทำการเก็บ Application ไว้



รูปภาพที่ 4.19 ไฟล์ภาพที่ได้ทดสอบการทำงานของการประมวลผลภาพ

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผลที่จากโครงการ

โครงการร้านค้าสำหรับแอพพลิเคชั่นการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผลแบบกลุ่ม มีส่วนการทำงานหลักๆอยู่ 3 ส่วนคือ ส่วนเว็บแอพพลิเคชั่น (Web Service) ส่วนของแอพพลิเคชั่น (Application Image Processing) และ ส่วนของเซิฟเวอร์คลัสเตอร์คอมพิวเตอร์ (Server Cluster) มีฟังก์ชันที่พัฒนาแล้วเสร็จตามแผนการพัฒนา ซึ่งได้มีการพัฒนาและ Deploy ไปเรียบร้อย ที่สามารถใช้ทดลองใช้งานได้แล้วดังนี้

5.1.1 ส่วนของเว็บแอพพลิเคชั่น (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

เป็นส่วนต่อประสานกับผู้ใช้สำหรับการแก้ไขข้อมูลของผู้ใช้งาน รวมไปถึงการแก้ไขรหัสผ่านด้วย

5.1.1.4 ส่วนของหน้าจัดการไฟล์

เป็นส่วนต่อประสานกับผู้ใช้สำหรับทำการจัดการ Folder สำหรับเก็บไฟล์รูป และ ตัวไฟล์รูปด้วย เพื่อให้สามารถนำรูปภาพไปใช้งานต่อในการสั่งงานประมวลผลภาพ

5.1.1.5 ส่วนของหน้าสั่งงานประมวลผล

เป็นส่วนต่อประสานกับผู้ใช้สำหรับทำการสั่งงานประมวลผลภาพ โดยจะต้องกรอกข้อมูลรายละเอียดของงานเข้าไปเพื่อทำการสร้างไฟล์ YAML ไปสั่งงานประมวลผลภาพ

5.1.2 ส่วนแอพพลิเคชัน (Application Image Processing)

5.1.2.1 แอพพลิเคชันประมวลผลภาพเป็นขาวดำ (Black White Application)

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

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

5.1.2.2 แอพพลิเคชันประมวลผลภาพเป็นอักษรและสี (ASCII Application)

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

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

5.1.2.3 แอพพลิเคชันประมวลผลภาพเป็นโมเสก (Mosaic Application)

การนำรูปภาพ 1 ภาพมาตัดแบ่งเป็น Tiles โดยกำหนดขนาดของ Tiles ออกมายield และ ยิ่งขนาดเล็กเท่าไหร่ความละเอียดของ Tiles ก็จะยิ่งสูงขึ้น โดยเมื่อแบ่ง Tiles เสร็จเรียบร้อยแล้ว จำนวนมากที่เราเตรียมไว้มาทำการ Feature Map กับตัว Tiles เพื่อทำให้ Tiles นั้น ๆ เกิดออกมายieldภาพอื่น ๆ ที่เราได้เตรียมไว้และออกมายieldกับภาพแบบ Mosaic

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

5.1.2.4 แอพพลิเคชันประมวลผลภาพเป็นพิกเซลศิลปะ (PixelArt Application)

การแปลงภาพปกติให้ออกมาเป็น Pixel ให้สวยงามโดยใช้การกำหนดค่า Kernel_size, Pixel_size และ Edge_thresh โดยเมื่อกำหนดค่าทั้ง 3 แล้วจึงนำภาพมาประมวลผลให้อยู่ในรูปแบบ tensor ก่อนแล้วจึงนำมาคำนวณด้วยการเดิมค่าทั้ง 3 ลงไปใน tensor ที่ได้แปลงจากภาพมาแล้วจึงนำมาแปลงค่า tensor กลับมาเป็นภาพ

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

5.1.3 ส่วนของเซิฟเวอร์คลัสเตอร์คอมพิวเตอร์ (Server Cluster)

5.1.3.1 ในส่วนของ K3S

ได้จัดการติดตั้ง K3S เพื่อทำการจำลองระบบ Cluster ขึ้นมาในรูปแบบของ Container เพื่อทดลองใช้งาน และ พัฒนาตัวโครงงานนี้ขึ้น

5.1.3.2 ในส่วนของ K3D

ได้ทำการเก็บข้อมูลตัวของ K3D เพื่อให้รองรับการทำงานของ GPU และ ได้ปรับปรุงใหม่ให้ตัว Kubernetes มีการรองรับการทำงานของ CUDA ขึ้นมาด้วย

5.1.3.3 ในส่วนของ Monitor

ได้ทำการติดตั้ง LENS Kubernetes สำหรับการ Monitoring ตัว Cluster เพื่อตรวจสอบการทำงาน และ การแจ้งเตือนต่าง ๆ ได้ง่ายยิ่งขึ้น และ ได้ติดตั้ง Prometheus Grafana เพื่อทำการตรวจสอบการทำงานของ 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 งานประมวลผล ได้ใช้อัลกอริทึมเป็น Round Robin

5.1.3.5 ในส่วนของฐานข้อมูล (Database)

ได้ทำการติดตั้ง Database เป็น MongoDB และ ได้จัดการตั้งค่าต่าง ๆ ของ Server ให้ใช้งานจัดเก็บข้อมูลไว้ใน Cluster

5.1.3.6 ในส่วนของ Persistent Volume (PV)

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

5.2 ปัญหาและอุปสรรค

1) การทำระบบ Cluster นี้จะต้องมีเครื่องคอมพิวเตอร์อย่างน้อย 2 เครื่องขึ้นไปจึงมีปัญหาในการหาเครื่องคอมพิวเตอร์มาใช้ทำโครงงานนี้

2) เนื่องจากการทดลองทดสอบระบบส่วนใหญ่ใช้การจำลองเพียงเครื่องเดียวจึงได้ผลสรุปมาค่อนข้างไม่ตรงกับความต้องการที่จะทดลอง

3) การพัฒนา Web application ที่ต้องใช้ Library เสริมเข้ามาช่วยในการพัฒนานี้ เกิดปัญหาการ Conflict กันเองระหว่าง Library ที่เมื่อมี Library ใหม่ถูกติดตั้งเพิ่มเข้ามา ก็อาจจะทำให้เกิดปัญหากับ Library เดิมที่มีอยู่

5.3 แนวทางการพัฒนาต่อไปในอนาคต

- 1) พัฒนาส่วนของการแบ่งภาระงานประมวลผลให้มีประสิทธิภาพมากยิ่งขึ้น
- 2) พัฒนา Web Application ให้ใช้งานได้ปลอดภัยในการส่งข้อมูลไปยัง Database และ Cluster
- 3) พัฒนา Marketplace บน Web Application
- 4) พัฒนาตัวจัดการ Weight Model ในการใช้งานบน Application

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 มีดีย่างไร ทำไม่ถึงต้องใช้.** [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

การดำเนินงานตามปริมาณงานขั้นต่ำ

ตาราง ก.1 ปริมาณงานที่ได้ดำเนินการไปในช่วงโครงการ 1

รายการ	ปริมาณงาน	หมายเหตุ
ระบบการสมัครสมาชิก และ เข้าสู่ระบบ	82%	เพิ่มความปลอดภัย โดยใช้ JWT เข้ามาในการเข้ารหัสข้อมูล แต่ยังขาดการเพิ่มเติม ในส่วนของ Front End เพื่อความสวยงาม
ระบบการจัดการของสมาชิก	68%	เนื่องจากยังไม่ได้เพิ่มในส่วนของหน้า Admin และ การจัดการสมาชิกในแต่ละระดับสมาชิก
ระบบการจัดการไฟล์	74%	เนื่องจากยังไม่ได้เพิ่มในส่วนของการแก้ไขไฟล์ และ การจัดการไฟล์เพิ่มเติม
ระบบการสั่งงานประมวลผล	39%	ได้ทำการสร้างไฟล์ใบสั่งงานประมวลเรียบร้อยแล้ว แต่ยังไม่ได้เชื่อมต่อกับระบบ Server และ ยังไม่ได้ทดลองผลการทำงาน
ระบบ Marketplace	0%	
ระบบ Feed	0%	
ระบบ Weight Model	0%	
Image Processing Application	75%	ได้ทำการพัฒนาแอปพลิเคชันประมวลผลภาพเบื้องต้นที่จะมีให้ผู้ใช้ได้ใช้งานบนเว็บไซต์ และ ทดสอบอยู่ในรูปแบบของ Docker Image โดยที่สามารถใช้งานได้ตามวัตถุประสงค์แต่ยังมีเรื่องของความเร็วในการประมวลผลครั้งแรกของชนิดงานที่สั่ง
Marketplace Application	0%	
Cluster System	97%	ได้ทำการติดตั้ง OS และ ใช้ระบบ K3S ในการจำลองระบบ Cluster ให้อยู่ในรูปแบบของ Container และใช้ K3D ใน

		การสร้าง Kubernetes ที่ได้ปรับปรุง แก้ไขแล้วให้รองรับกับการทำงานของ GPU CUDA แต่เนื่องจากยังไม่ได้ ทดลองการทำงานบนระบบ Cluster โดย ใช้เครื่องแยกจริง จัดการ Load Balance การเข้าใช้งานใน ส่วนของ Web Service ด้วย Nginx และ Ingress ในส่วนของการส่งงานประมวลผลได้ใช้ อัลกอริทึม Round Robin ใน การจัดการ แบ่งภาระงานประมวลผล และ จัดเก็บ ไฟล์ข้อมูลต่าง ๆ แยกจากตัว Cluster โดยใช้ Persistent volume
ทดสอบการเชื่อมต่อ กันระหว่าง Web Application และ Server	25%	เนื่องจากยังไม่ได้ทดสอบส่วนงานผ่าน Web Application ที่ทำงานอยู่บน Cluster System
ทดสอบการประมวลผล	5%	ได้ทำการทดสอบการประมวลผล เบื้องต้นแล้ว แต่ยังไม่ได้ทดสอบทั้งหมด และ เก็บสถิติในการประมวลผล
ทดสอบการทำงานของ Load Balance	5%	ได้ทำการทดสอบการแบ่งภาระงาน เบื้องต้นในส่วนของ Nginx และ Ingress แล้ว แต่เนื่องจากยังไม่ได้ทำเป็น Cluster จริง ๆ บนเครื่องคอมพิวเตอร์จึงทำให้ยัง ไม่ได้ทดสอบการทำงานของการแบ่ง ภาระงานประมวลผล จึงได้ใช้เป็น อัลกอริทึมของ Round Robin ไปก่อน
รวม	48%	

จากการแสดงรายละเอียดงาน และ ปริมาณงานเฉพาะ ในส่วนของการพัฒนา Web Application, Image Processing Application และ Server จะเห็นได้ว่า มีความก้าวหน้ารวมอยู่ที่ 48%