

ร้านค้าสำหรับแอพพลิเคชั่นการประมวลผลภาพซึ่งจัดการงานบนระบบ

ประมวลผลแบบกลุ่ม

Market Place for Image Processing Application using Task

Management on Cluster Computing System

พศิน จันทร์ทัน

สุชี สาระพันธ์

ปริญญาอนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2565

ปริญญาบัณฑีปการศึกษา 2565

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ร้านค้าสำหรับแอพพลิเคชั่นการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผลแบบกลุ่ม

Market Place for Image Processing Application using Task Management on Cluster Computing System

ผู้จัดทำ

1. นายพศิน จันทร์ทัน รหัสนักศึกษา 63015121

2. นายสุรชี สาระพันธ์ รหัสนักศึกษา 63015190

อาจารย์ที่ปรึกษา

(รศ. ดร.อรุณัต จิตต์ไสวักร)

ร้านค้าสำหรับแอพพลิเคชั่นการประมวลผลภาพซึ่งจัดการงานบน

ระบบประมวลผลแบบกลุ่ม

นายพศิน	จันทร์ทัน	63015121
นายสุนี	สาระพันธ์	63015190
รศ.ดร.อรุณัตร จิตต์โสภกัคร์	อาจารย์ที่ปรึกษา	
ปีการศึกษา 2565		

บกคดย่อ

Market Place for Image Processing Application using Task Management on Cluster Computing System

Mr. Pasin Chanharathan 63015121

Mr. Sutee Saraphan 63015190

Assoc.Prof.Dr. Orachat Chitsobhuk Advisor

Academic Year 2022

ABSTRACT

กิตติกรรมประกาศ

กิตติกรรมประกาศ

พศิน จันทร์ทัน

ฤทธิ์ สาระพันธ์

บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

ในปัจจุบันผู้ให้บริการแอพพลิเคชั่นการประมวลผลภาพนั้นมีข้อจำกัดเรื่องงานการประมวลผล ใช้ ทรัพยากรของเครื่องสูง ไม่มีประสิทธิภาพ หรือ ยังไม่มีการกำหนด จำกัดหน่วยประมวลผลของงานในแต่ละงานประมวลผล และ ไม่รองรับการประมวลผลภาพจำนวนมากพร้อม ๆ กัน พร้อมด้วย แอพพลิเคชั่นการประมวลผลภาพนั้นมีตัวเลือกค่อนข้างน้อยในแอพพลิเคชั่นเดียวให้ใช้งาน และ เนื่องจากงานวิจัยด้าน Image Processing ต่าง ๆ นั้นมีการ Train ข้อมูลของ Weight Model เข้ามาช่วยในงานประมวลผลด้วยตัวอย่างในวิชา Image Processing ภายในสถาบันที่มีการให้นักศึกษา ศึกษาโครงงาน เรื่องการ Train Weight Model ภายในวิชาขึ้นมาเพื่อมาทดลองภายในห้องทดลอง แต่ยังขาดพื้นที่สำหรับ การให้บริการด้าน Model ของ Image Processing ที่นักศึกษาได้ทำการ Train Weight ข้อมูลไว้เพื่อให้ได้ทดสอบภายในห้องทดลองเนื่องจากใช้ทรัพยากรของเครื่องสูงในการประมวลผลของ Model และ ให้บุคลากรภายนอกได้เข้าถึงด้วย

ดังนั้น โครงงาน ร้านค้าสำหรับแอพพลิเคชั่นการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผล แบบกลุ่ม มีจุดประสงค์เพื่อสร้างระบบการแบ่งการจัดลำดับงานประมวลผลของผู้ใช้บริการแอพพลิเคชั่น เพื่อเป็นอีกทาง เลือกหนึ่งในการใช้งานการประมวลผลภาพในรูปแบบต่าง ๆ อย่างมีประสิทธิภาพ และ รองรับการประมวลผลภาพจำนวนมาก พร้อมด้วยรองรับแอพพลิเคชั่นที่สามารถนำเข้ามาเพิ่มในอนาคต ได้อีกด้วย และ ในส่วนของ Weight Model ที่นักศึกษาได้ทำการ Train มาแล้วสามารถนำเข้าภายใน แอพพลิเคชั่นเพื่อเปิดให้บริการในแอพพลิเคชั่นของเราได้เพื่อทำการซื้อขาย หรือ เป็นที่แสดงผลงานของ นักศึกษา และ เป็นอีกทางเลือกให้นักศึกษามีรายได้

โดยสรุปได้ว่า การตกแต่งรูปภาพ หรือ การปรับแต่งภาพ นั้นมีการใช้อยู่เพร่หลาย และ คนส่วนใหญ่ มีการใช้งานเรื่องการประมวลผลภาพกันเป็นปกติ การนำ Weight Model มาใช้งาน หรือ จัดแสดงผลงาน และ การซื้อขาย ซึ่งทั้งหมดที่กล่าวมาคือส่วนมากใช้งานบนแอพพลิเคชั่นต่างๆ ผ่านอินเทอร์เน็ตเราจึง พัฒนาโครงงานบนพื้นฐานของเว็บแอพพลิเคชั่นซึ่งสอดคล้อง และ รองรับกับปัญหาดังกล่าว โครงงาน “แอพพลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย” นี้ถูกจัดทำขึ้นมาเพื่อ แก้ไขในส่วนของการนี้ โดยมีการแบ่งเป็น 3 ส่วนคือ

- 1) ส่วนการประมวลผลงานแบบการจัดลำดับงาน (Server) Task Scheduling การแบ่งจัดลำดับการทำงานประมวลผลแต่ละงานไปแต่ละเครื่อง โดยมีการกำหนดทรัพยากรที่ใช้ในแต่ละงาน ประมวลผลเพื่อให้ได้ประสิทธิภาพเพียงพอต่อความต้องการของงานประมวลผล และ ใช้ทรัพยากรของหน่วยประมวลผลให้คุ้มค่ากับทรัพยากรที่มีอยู่ของหน่วยประมวลผล
- 2) ส่วนแอ�플ิเคชันการประมวลผลภาพ (User) Image Processing Application ส่วนระบบสำหรับประมวลผลภาพที่ระบบได้มีการเตรียมรูปแบบประมวลผลภาพต่าง ๆ ที่ หลากหลายให้ผู้ใช้ได้เข้ามาใช้งานการประมวลผลภาพจำนวนมากที่ทางเราได้เตรียมจัดทำไว้
- 3) ส่วนแอ�플ิเคชันให้บริการซื้อขาย Weight โมเดล (User) Market Place ส่วนระบบสำหรับให้ผู้ใช้มีการซื้อ Weight Model และ นำเข้า Weight Model ที่ได้ทำการเทรนนิ่งไว้แล้วมาปิดให้ใช้บริการบนแอ�플ิเคชันเพื่อทำการ ขาย หรือ เพื่อทดลองงานต่าง ๆ โดยเป็นจุดโซเชียลผลงานของตัว Weight Model ที่ได้ทำการเทรนนิ่งไว้ให้ออกตัวขึ้น

1.2 วัตถุประสงค์ของโครงงาน

- 1) เพื่อจัดสร้างใช้งานประมวลผลของแต่ละงานประมวลผลให้มีประสิทธิภาพเพียงพอ 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) ได้รับประสบการณ์ในการพัฒนาแอ�플ิเคชันการประมวลผลภาพที่นิยมในปัจจุบัน

1.4 ขอบเขตการวิจัย

- 1) ระบบ Cluster Computing เชื่อมต่อผ่านระบบเครือข่ายท้องถิ่น (LAN)
- 2) ระบบ Cluster Computing ในแอ��พลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่ายนี้นั้นเป็นการแบ่งงานในแต่ละเครื่องเพื่อให้ใช้ทรัพยากรที่จำกัด และ จำกัด ทรัพยากรในแต่ละงานประมวลผล
- 3) การจัดการลำดับงานบนระบบขึ้นอยู่กับประสิทธิภาพฮาร์ดแวร์ของ Node ที่พร้อมใช้งานในช่วงเวลาหนึ่น
- 4) แอ��พลิเคชั่นในส่วนของฝากรายละเอียด Structures ของ Directory แต่ละ User นั้นการแทรก ระดับขั้นของ Directory ได้ 1 ชั้น
- 5) แอ��พลิเคชั่นในส่วนของการประมวลผลนั้นสามารถทำการประมวลผลภาพจำนวนมาก จึงปรับค่า Parameter ได้ 1 ค่าในงานประมวลผลภาพจำนวนมาก
- 6) ส่วนของ Market Place ในการนำเข้า Weight นั้นจะต้องรองรับกับตัว Model ที่มีให้บริการภายใต้แอ��พลิเคชั่นเท่านั้น

บทที่ 2

เอกสารและทฤษฎีเกี่ยวข้อง

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

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 Digital Image Processing

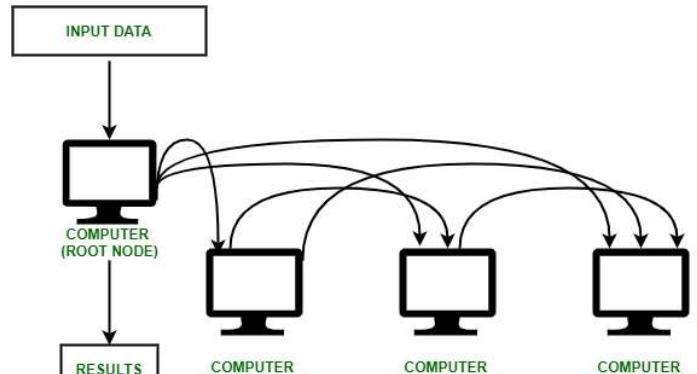
การประมวลผลภาพดิจิทัล หรือ Digital Image Processing คือ การใช้คอมพิวเตอร์ดิจิทัลในการประมวลผลภาพดิจิทัลผ่านอัลกอริธึม ภาพในที่นี่รวมความหมายถึงสัญญาณในระบบดิจิทัล 2 มิติ ภาพนิ่ง และ ภาพเคลื่อนไหว หรือเป็นชุดของภาพนิ่ง ที่เรียกว่า เฟรม (Frame) ซึ่งนับเป็นภาพดิจิตัล 3 มิติ ได้ เช่น ภาพ ทางการแพทย์ หรือ 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) ทำให้จุดที่น่าสนใจของภาพให้เด่นชัด
- 4) การวัดรูปแบบของภาพ (Measurement of pattern) วัดคุณภาพทั้งหมดนั้นจะถูกวัดค่ามาเพื่อหารูปแบบของวัตถุภายในภาพ
- 5) การรับจดจำภาพ (Image Recognition) สามารถแยกแยะวัตถุในภาพได้

2.1.2 Cluster Computing

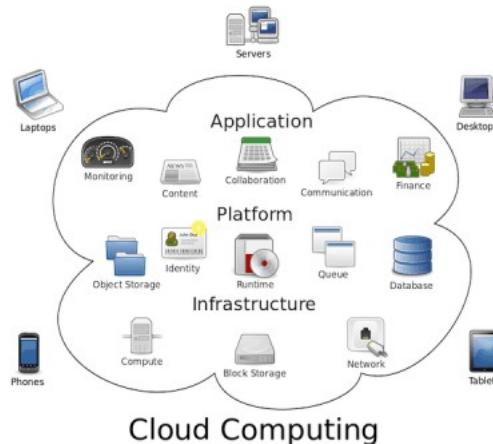
ระบบคลัสเตอร์ หรือ คลัสเตอร์ิng เป็นการเชื่อมต่อระบบการทำงานของกลุ่มคอมพิวเตอร์เข้าด้วยกันภายใต้ระบบเครือข่าย มีความสามารถในการกระจายงานที่ทำไปยังเครื่อง ภายในระบบเพื่อให้การประมวลผลมีประสิทธิภาพสูงขึ้น โดยอาจเทียบเท่าชูปเปอร์คอมพิวเตอร์ หรือ สูงกว่าสำหรับการประมวลผลงานที่มีความซับซ้อนโดยเฉพาะงานด้านวิทยาศาสตร์ เช่น การจำลองโครงสร้างของโมเลกุลทางเคมี, การวิเคราะห์เกี่ยวกับตำแหน่งการเกิดพายุฤดูร้อน, การประมวลผลภาพ เป็นต้น ดังรูป 2.1.2



ภาพที่ 2.1.2 Cluster Computing

2.1.3 การประมวลผลแบบคลุ่มเมฆ (Cloud Computing)

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



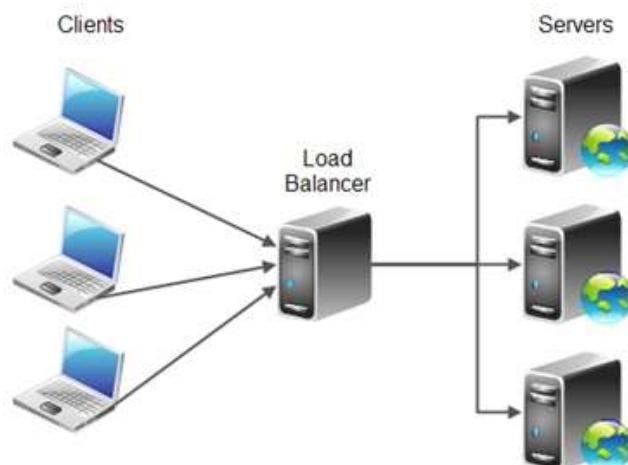
ภาพที่ 2.1.3 การประมวลผลแบบคลุ่มเมฆ (Cloud Computing)

2.1.4 การแบ่งภาระงาน (Load Balance)

การแบ่งภาระงาน หรือ Load Balance คือเทคนิควิธีการทางด้าน Network ที่ถูกคิดกันขึ้นมาเพื่อช่วยลดการเกิดปัญหา Server ไม่สามารถรับงานประมวลผลจำนวนมากได้จน Overload และหยุดการทำงานไป โดยวิธีการคือการนำเครื่อง server ที่สามารถทำงานได้ในระดับเดียวกัน และ สามารถทำงานแบบเดียวกัน มาทำงานร่วมกันเพื่อช่วยกระจายปริมาณงานที่เข้ามา โดยจะแบ่งงานไปตามวิธีการที่ผู้ใช้ได้กำหนดเอาไว้ ซึ่งวิธีการแบ่งงานที่นิยมใช้กันมีดังนี้

1. Round Robin เป็นการแบ่งงานให้กับ Server ในกลุ่มแบบเรียงลำดับ เช่น 1,2,3,1,2,...
2. Sticky เป็นการแบ่งงานให้กับ Server โดยอิงจาก Sessions ที่ผู้ใช้เคยเข้าไปใช้ก่อนหน้า
3. Work load 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 ไม่ว่าจะเป็นการตัดแต่งรูปภาพ 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 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> เป็นผู้ให้บริการ Application การประมวลผลภาพโดยการให้ผู้ใช้งานนำเข้าภาพของตนเองที่ต้องการเพื่อให้ Application นำภาพนั้นไปประมวลผลโดยมีการประมวลผลดังนี้ ปรับแต่งสีของภาพตามที่ผู้ใช้ต้องการ โดยมี parameter ที่กำหนดมาให้โดยมี Preset จากกล้องฟิล์มให้ผู้ใช้งานเข้ามาใช้บริการ ตัวอย่างแสดงดังรูปที่ 2.2.2



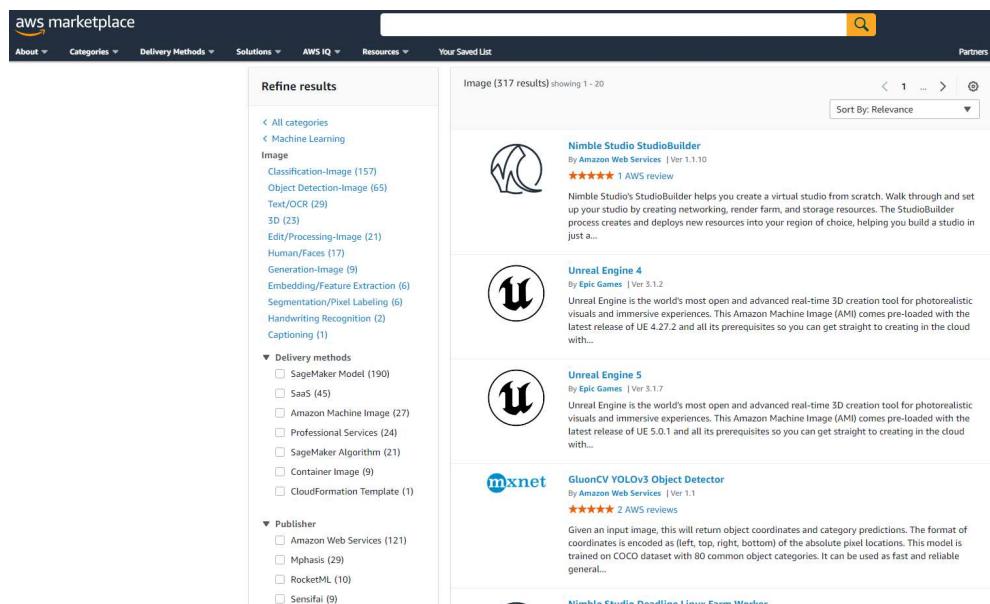
รูปที่ 2.2.2 ตัวอย่างของ Film Emulator

สิ่งที่ Film Emulator ยังไม่มี หรือ นำมาปรับปรุง มาพัฒนาต่อในแอปพลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย

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 รองรับ โดยให้ผู้ใช้ซื้อค่าบริการเป็นรายชั่วโมงเพื่อเข้าใช้บริการงานต่าง 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 มาพัฒนาต่อในแอพพลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย

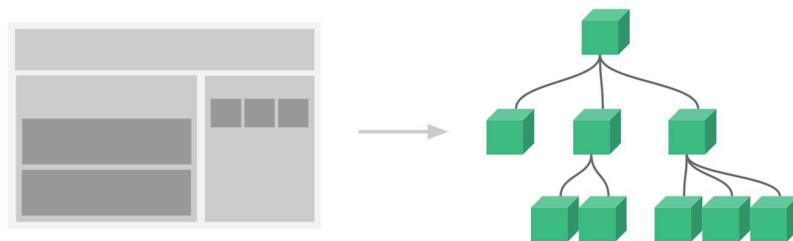
1. การให้บริการเป็นจำนวนนภาพที่ประมวลผลแทนการใช้บริการรายชั่วโมง
2. ให้ผู้ใช้รายย่อยอื่น ๆ นำเข้า Weight เพื่อมาซื้อขายได้จาก Model ที่ Application มีให้เหมือนกัน

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 และ สามารถนำ component เหล่านั้นมาใช้งานช้าได้อีกด้วย โดยตัวอย่างการทำงานของ Component stem ของ Vue.js แสดงดังรูปที่ 2.3.1



รูปที่ 2.3.1 Component system ของ Vue.js

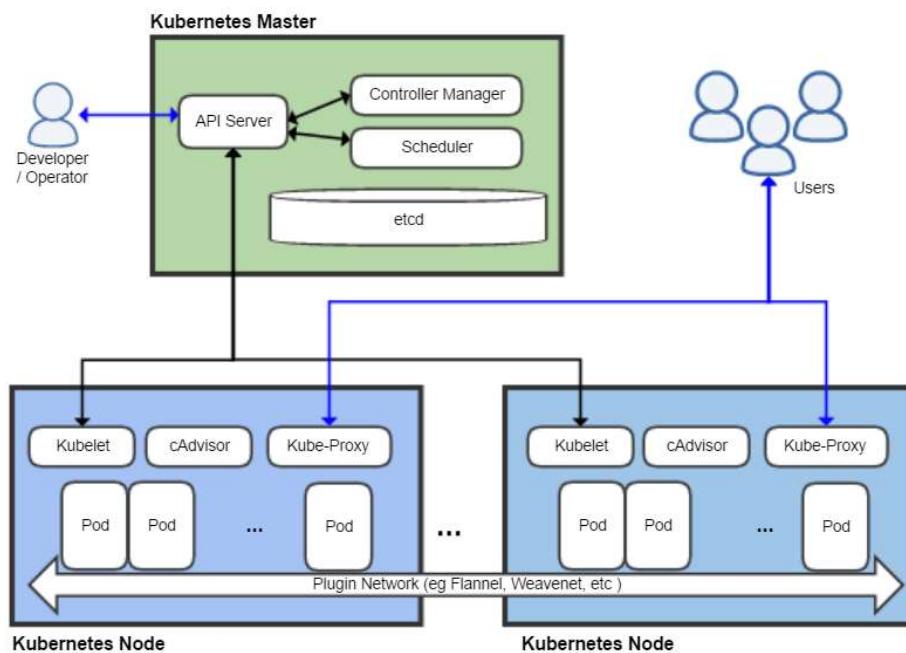
2.3.2 Docker

Docker คือชุด Platform ที่ใช้การจำลองเสมือนระดับ OS เพื่อคำนึงการทำงานส่วนแพ็คเกจที่เรียกว่าคอนเทนเนอร์ (Container) ซึ่งในแต่ละคอนเทนเนอร์แยกกันภายในแต่ละ คอนเทนเนอร์จะประกอบด้วยซอฟต์แวร์ ไลบรารี และ ไฟล์กำหนดค่าของตนเอง (Environment) อิกทึ้งในแต่ละ Container ก็สามารถสร้างช่องทางสื่อสารกันภายใน Container เพื่อให้ช่วยในงานประมวลผล

โดยมีการใช้ทรัพยากรที่น้อย เมื่อเทียบเท่ากับการใช้งานบน OS ปกติทั่วไป และ ลดปัญหาในการเกิดข้อผิดพลาดของ Environment ในการใช้งาน

2.3.3 Kubernetes

คือ กลุ่มพัฒนาโดย Google เพื่อนำมา จัดการระบบคลัสเตอร์ (Cluster management software) สำหรับ Docker Container โดยตัว Kubernetes นี้ใช้สำหรับการ จัดการปล่อยซอฟต์แวร์อัตโนมัติ (automating software deployment) และ ทำการ Scaling ตรวจสอบความถูกต้องของ Container บนระบบ Cluster ได้ตลอดเวลา ตัว Kubernetes นี้เนื่องจากเป็น Open-Source จึงมีการนำไปดัดแปลงแก้ไขเป็น Kubernetes เวอร์ชันอื่น ๆ มาอีกมากมาย ไม่ว่าจะเป็น MicroK8S, K3D, MiniKube โดยทั้งหมดที่กล่าวมานี้เงื่อนไข และ การทำงาน การใช้งานอยู่ในรูปแบบเดียวกัน โดยมีหลักการทำงาน พื้นฐานเหมือนกันดังรูป 2.3.3



รูปที่ 2.3.2 Component system ของ Vue.js

2.3.4 Python

Python เป็นภาษาในการเขียนโปรแกรมที่ใช้อย่างแพร่หลายในการพัฒนาซอฟต์แวร์ เว็บแอปพลิเคชัน วิทยาศาสตร์ข้อมูล การประมวลผลภาพ และ แมชชีนเลิร์นนิ่ง (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 เหมาะกับการพัฒนาเว็บที่มีการใช้งานบ่อย เช่น การตรวจสอบการดึงข้อมูลจากฐานข้อมูล การจัดการคุกคัก และ การทำงานควบคู่ไปกับ Python ซอฟต์แวร์อื่น ๆ มาปรับให้ใช้งานภายในเว็บแอปพลิเคชันได้ ตัว Django เป็น open-source ที่สามารถนำมาใช้งานได้ฟรี และ พัฒนาได้ฟรีโดยมี Community ที่ช่วยเหลือรักษาซอฟต์แวร์ตัวนี้ และมีองค์กรไม่แสวงหาผลกำไรซึ่งอ้างว่า Django Software Foundation ที่ทำการพัฒนา ปรับปรุง และบำรุงรักษา ตัว Django ให้มีคุณภาพสูง และ ฟีเจอร์อีกมากมายให้ใช้งานโดยไม่มีค่าใช้จ่ายตลอดเวลา

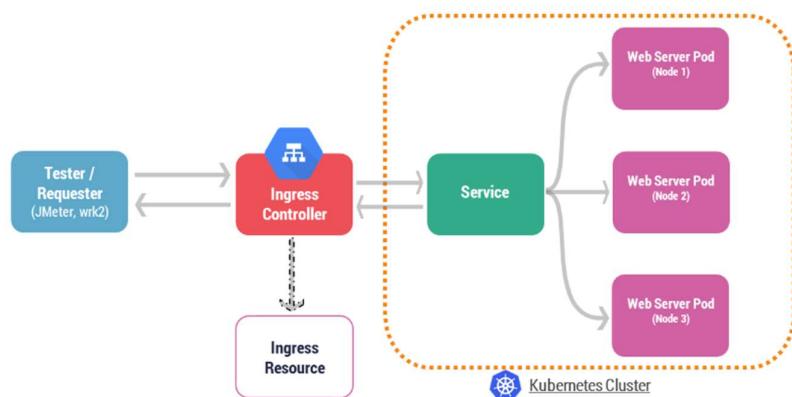
2.3.6 MongoDB

MongoDB เป็น NoSQL Database ที่สามารถใช้งานข้าม Platform ต่าง ๆ ได้ โดยตัว MongoDB ได้ใช้ NoSQL เป็นการเข้าถึงข้อมูลโดยใช้เทคนิคของการดึงข้อมูลผ่าน Key Pair Value โดยเหมาะสมกับงานที่ต้องการเข้าถึงข้อมูลได้อย่างรวดเร็ว 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 คุณอาชิป พวงคำไบ และ คุณชัยพร เนมະภาตะพันธ์ กล่าวถึงเรื่องการทดลองเพื่อประเมินสมรรถนะ และเปรียบเทียบประสิทธิภาพของ ระบบ Load balance แบบ Layer – 7 ที่ถูกใช้งานในสภาพแวดล้อมการทำงานที่อยู่ในรูปแบบของ Container cluster ด้วย Kubernetes โดยแผนผังการวางแผนระบบในการวิจัยจะเป็นดังรูป 2.4.1



รูปที่ 2.4.4 แผนผังแสดงระบบที่ใช้ในการทดสอบIngress controller

โดยโครงงานนี้ได้นำเอกสารนี้มาศึกษาเพื่อใช้ในการพัฒนาส่วน Cluster Computing เพื่อให้ การจัดลำดับงานและการประมวลผลใน Cluster มีประสิทธิภาพมากขึ้น

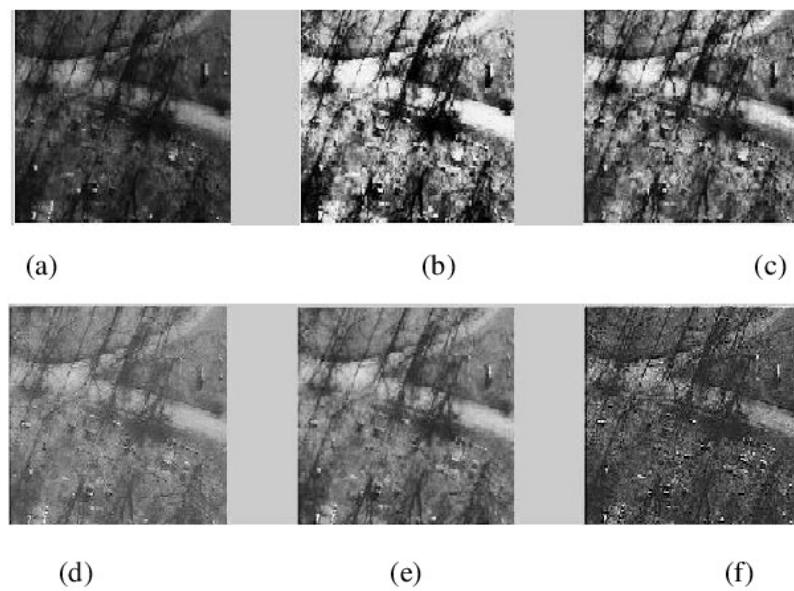
2.4.2 A modified gray-level difference algorithm for analyzing Gaussian Blurred texture images

งานวิจัยนี้จัดทำขึ้นในปี 2011 โดย Rui Zhang, Xiang Qian และ Datian Ye กล่าวถึงเรื่องการกระบวนการแยกແພັນພື້ນພົວອງຮູບປາກ โดยการใช้อัลกอริතົມ gray-level difference (GLD) โดยจะมี 4 ขั้นตอนคือ 1. ประเมินความแตกต่างของພື້ນພົວອງປາກ โดยเปรียบเทียบກັບຂໍ້ມູນທີ່ເຕີຍເອາໄວ້ ກ່ອນນັ້ນ 2. ໃຫ້ຕົວຮອງແບນ Wiener ເພື່ອລົບ Gaussian blur noise ທີ່ທີ່ອາຈເກີດຂຶ້ນຫາກຄ່າຄວາມແຕກຕ່າງ ມືນ້ອຍ 3. ທຳຫ້າວິທີກາຣີທີ່ 2. ໄປຈົນກະທັງຄ່າຄວາມແຕກຕ່າງສູງກວ່າຂໍ້ມູນທີ່ເຕີຍເອາໄວ້ 4. ໃຫ້ອັນດີກົດກັບ gray-level difference ແບບດັ່ງເດີມ ເພື່ອເກີບຂໍ້ມູນພື້ນພົວອງຮູບປາກທີ່ຜ່ານການປະມາລພຸດແລ້ວ ເພື່ອ นำມາວັດປະສິບປັບປຸງກາພທີ່ເຕີຍກັບອັນດີກົດກັບ gray-level difference ແບບໃໝ່

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 ผ่านรูปภาพแล้วปรับความคมชัดของภาพไปทีละส่วน โดยสามารถปรับให้ความคมชัดของตัวรูปภาพนั้นเพิ่มขึ้นและลดลงได้ตามความต้องการ และนอกจากนี้ยังนำเสนอวิธีการอื่นที่ใช้ในการปรับแต่งความคมชัด เช่น the bi-histogram equalization (BHE) และ Global histogram equalization เป็นต้น โดยรูปที่ 2.4.3 จะเป็นรูปผลลัพธ์จากการปรับต่างด้วยวิธีที่แตกต่างกัน

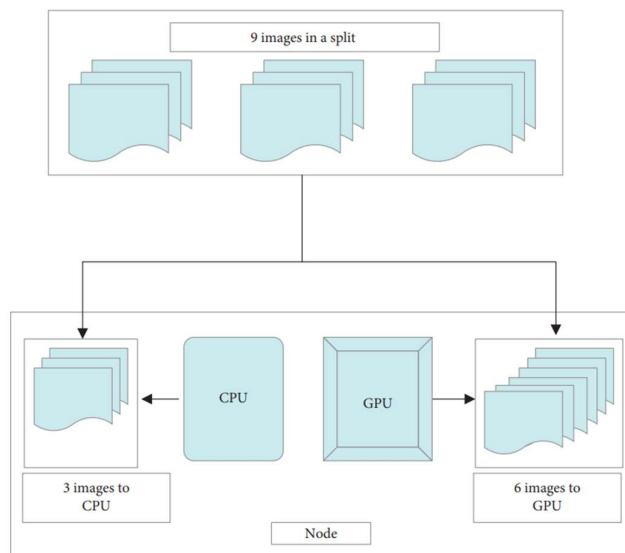


รูปที่ 2.4.3 รูปผลลัพธ์จากการปรับต่างความคมชัดด้วยวิธีที่แตกต่างกัน โดย (a) Original Image Contrast , (b) Histogram Equalization Contrast Enhancement , (c) Adaptive Histogram Equalization , (d) LPF Homomorphic Contrast Enhancement , (e) HPF Homomorphic Contrast Enhancement และ (f) Unsharp Mask Contrast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

โดยโครงการนี้ได้นำเอกสารนี้มาศึกษาเพื่อใช้ในการพัฒนาส่วนของการพัฒนา 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 จากปกติการที่ประมาณผลงาน ด้าน image processing สิ่งที่สามารถพบเจ้อได้เป็นปกติคือ การกระจายภาระงานที่ไม่สมดุล ซึ่งทำให้ ประสิทธิภาพของการทำงานนั้นลดลง โดยการแบ่งนั้นจะแบ่งออกเป็น 2 ช่วง ช่วงแรกจะเป็นการ แบ่งและแยกแยะข้อมูลของรูปภาพเป็นส่วนๆ ที่เหมาะสมต่อการกระจายไปยังหน่วยประมวลผล ต่าง จากนั้นในช่วงที่สอง ทำการกระจายเพิ่มเติมไปยัง CPU และ GPU ขึ้นอยู่กับความเร็วในการ ประมวลผลดังรูป 2.4.3

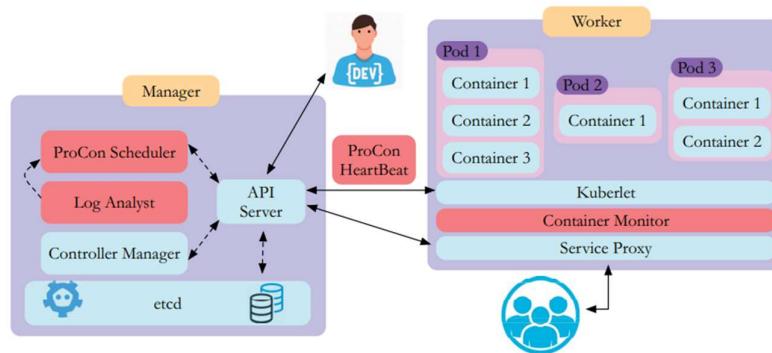


รูปที่ 2.4.3 แผนผังรูปแบบการกระจายงานในช่วงที่สองที่มีการกระจายภาระงานระหว่าง CPU และ GPU

โดยโครงงานนี้ได้นำเอกสารนี้มาศึกษาเพื่อใช้ในการพัฒนาส่วน Load balance ที่จะทำหน้าที่ แบ่งงานภาระงาน 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 กล่าวถึงร่องการจัดวางรูปแบบ คอนเทนเนอร์ ที่ตั้งชื่อ “ProCon” โดยปกติแล้วจะมี Manager node เป็นตัวที่เลือกใช้งาน Worker node แต่ละตัวตาม อัลกอริทึมที่ตั้งค่าเอาไว้ โดยที่ ProCon จะทำหน้าที่ในการกำหนด input คอนเทนเนอร์แล้ว พิจารณา การใช้ทรัพยากรทันที รวมไปถึงการประเมินทรัพยากรในอนาคตด้วย และขณะที่ประมวลผลนั้น ProCon ที่จะทำการจัดสรรคลุกของทรัพยากรภายใน Cluster ไปด้วย



รูปที่ 2.4.4 แผนผัง System Architecture ของ ProCon

โดยโครงงานนี้ได้นำงานวิจัยมาศึกษาเพื่อสร้างรูปแบบการจัดการคอนเทนเนอร์ที่มีการจัด ทรัพยากรให้แต่ละ Worker อย่างมีประสิทธิภาพ

บทที่ 3

การออกแบบและการพัฒนา

การออกแบบและการพัฒนาระบบ แอ��เพลิเคชันการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย แบ่งออกเป็น 3 ส่วน ก็อ Web Application ส่วนของ Image Processing Application และส่วนของ Server โดยมีรายละเอียดส่วนต่าง ๆ ดังนี้

3.1 รายละเอียดของระบบ

1) Web Application

- 1) ผู้ใช้สามารถสมัครสมาชิก เข้าสู่ระบบ ได้
- 2) สามารถถูกคืน และ เปลี่ยนรหัสผ่าน ได้
- 3) สามารถจัดเก็บไฟล์รูปภาพ ได้
- 4) สามารถสั่งงานประมวลผลภาพ ได้
- 5) สามารถทดลองดูตัวอย่างงานประมวลผลภาพ ได้
- 6) สามารถจัดเก็บ Weight Model ได้
- 7) สามารถซื้อขาย Weight Model ได้
- 8) สามารถสั่งงานประมวลผลภาพบน Marker Place ได้
- 9) สามารถดูสถิติการประมวลผลภาพ ได้

2) Image Processing Application

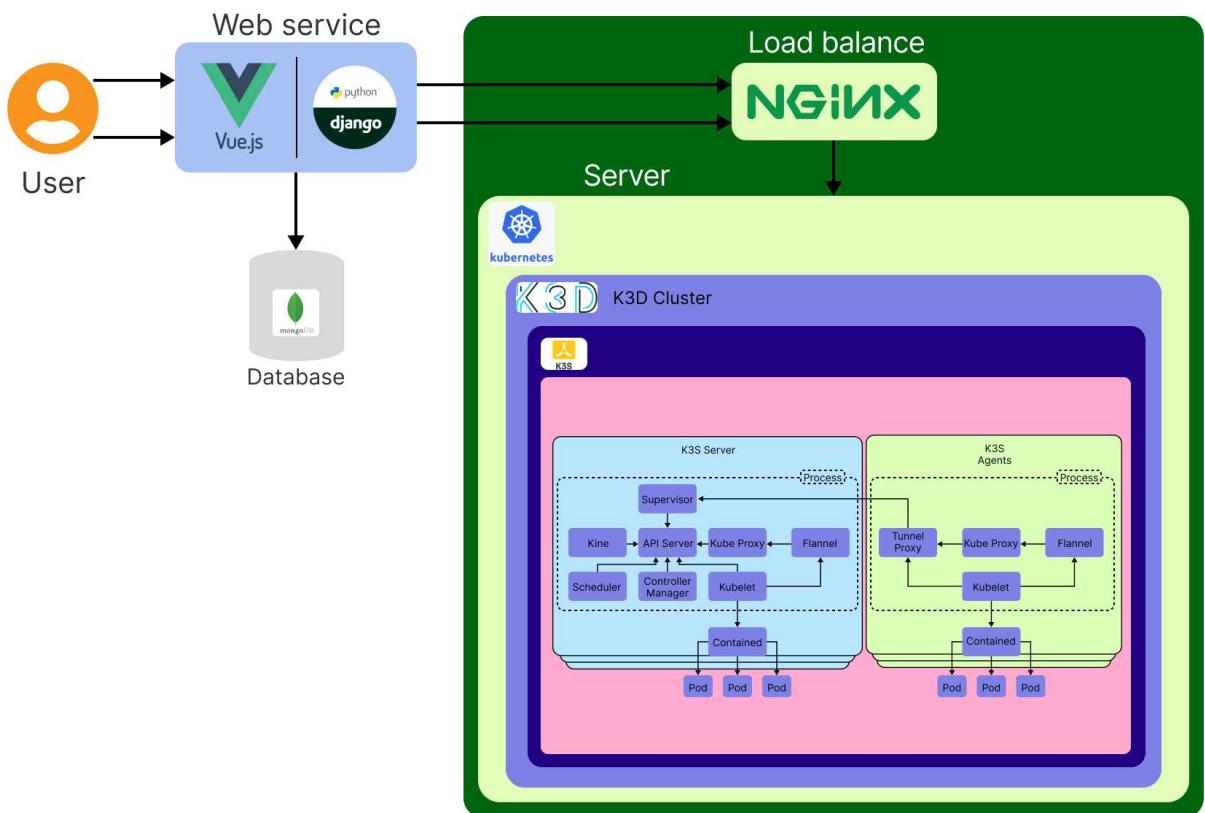
- 1) สามารถประมวลผลภาพเป็นภาพ ASCII ได้
- 2) สามารถประมวลผลภาพเป็นภาพ PixelArt ได้
- 3) สามารถประมวลผลภาพเป็นภาพ Mosaic ได้
- 4) สามารถประมวลผลภาพเป็นภาพขาวดำ ได้
- 5) สามารถประมวลผลภาพเป็นภาพเพื่อลบพื้นหลังภาพ ได้
- 6) สามารถประมวลผลภาพเป็นภาพจาก Model และ Weight ได้

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

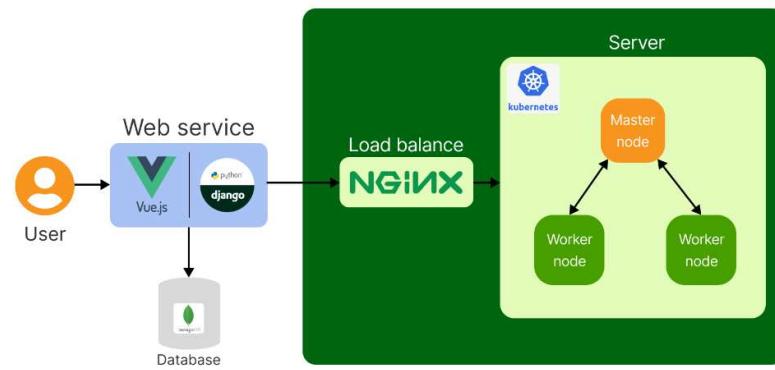
3.2 ภาพรวมของระบบ

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



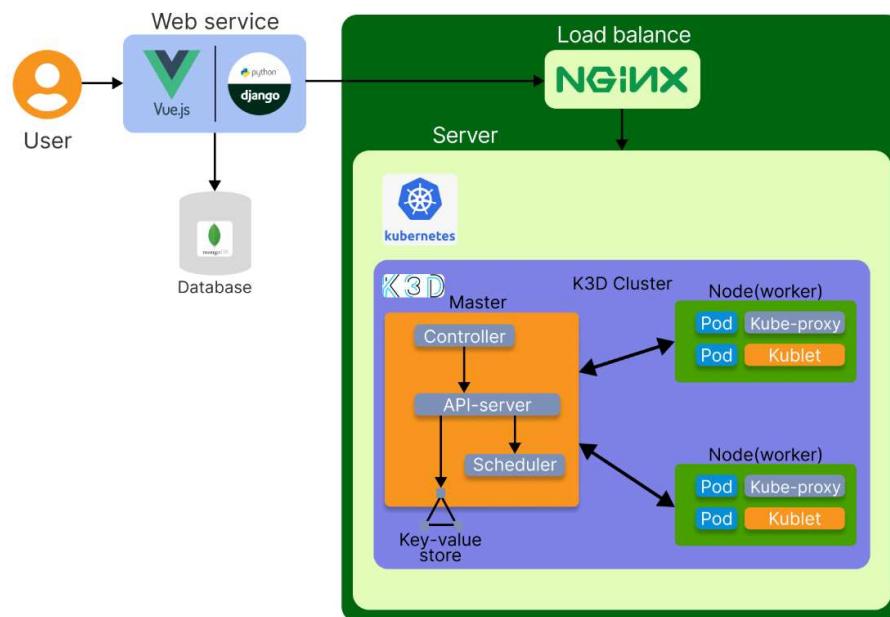
รูปที่ 3.1 ภาพรวมของระบบ Overview

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



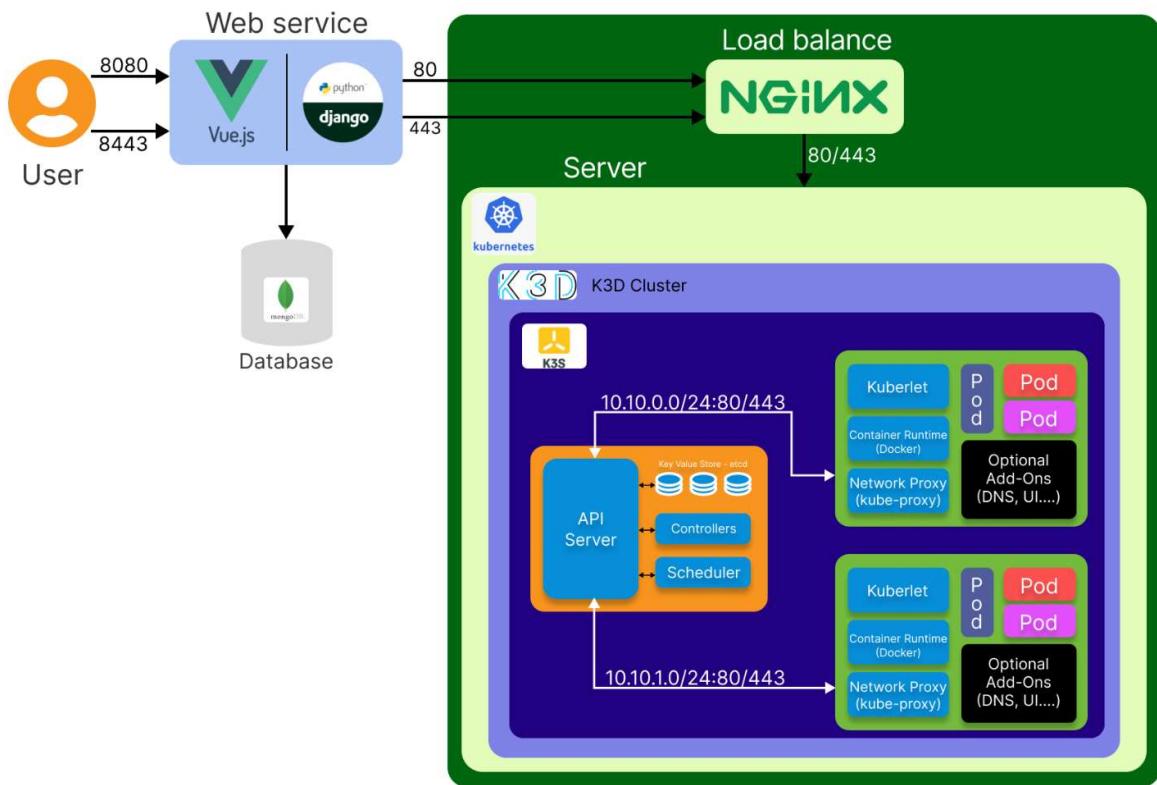
รูปที่ 3.2 ภาพรวมของระบบ Level 0

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 นี้จะกล่าวถึงการเชื่อมต่อของ User ไปยัง Web Service นี้โดยมี Port การเชื่อมต่อไป ในส่วนของ Web Service มี Vue.js สำหรับการทำ Front-End และ Python Django สำหรับ Back-End โดยที่มี Database เป็น MongoDB นส่วนของผู้ให้บริการมีการติดตั้ง Load Balance Nginx คู่กับ Ingress สำหรับการ Load Balance การขอเข้าใช้งาน Web Service ไปยัง Kubernetes Cluster ใน Server โดย Master Node จะมีการเชื่อมต่อกับ Worker Node อีก ๑ ดังรูปที่ 3.4

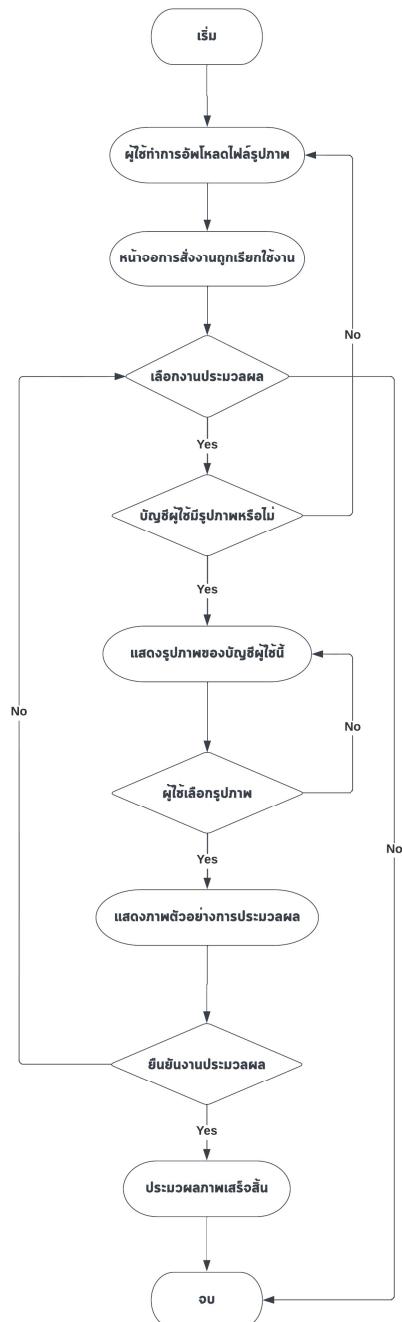


รูปที่ 3.4 ภาพรวมของระบบ Level 2

3.3 แผนภาพผังงาน (Flowchart)

3.3.1 แผนภาพผังงาน (Flowchart) ของการสั่งงานประมวลผลภาพ (Web Application)

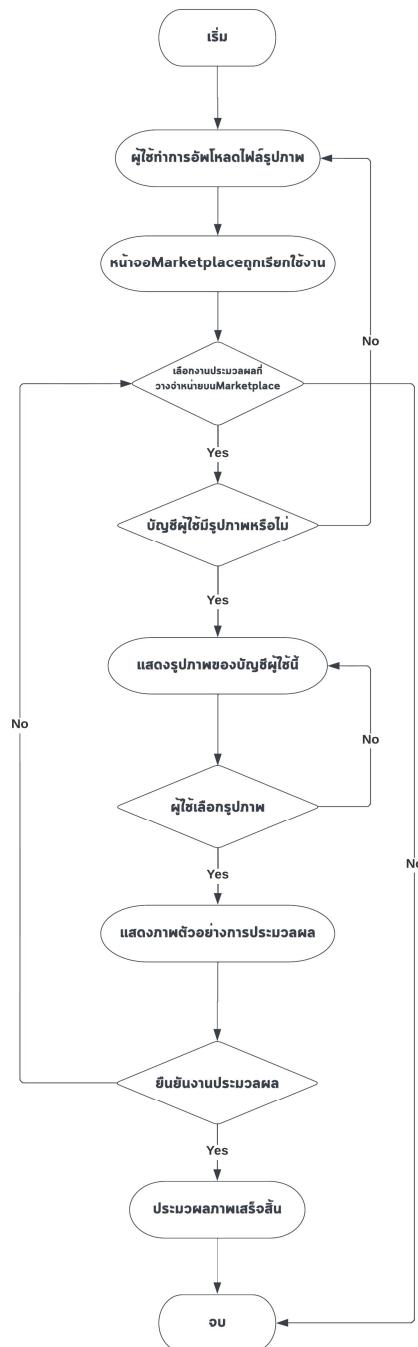
แผนภาพผังงานในส่วนของ Web Application บนเว็บที่ผู้ใช้งานเริ่มทำการใช้งานการสั่ง งาน ประมวลผลภาพจนถึงการประมวลผลภาพเสร็จสิ้น โดยแผนภาพผังงานในส่วนของการสั่งงานประมวล ภาพ จะแสดงดังรูปภาพที่ 3.5



รูปที่ 3.5 แผนภาพผังงานของการสั่งงานประมวลผลภาพ (Web Application)

3.3.2 แผนภาพผังงาน (Flowchart) การสั่งงานประมวลผลภาพบน Marketplace (Web Application)

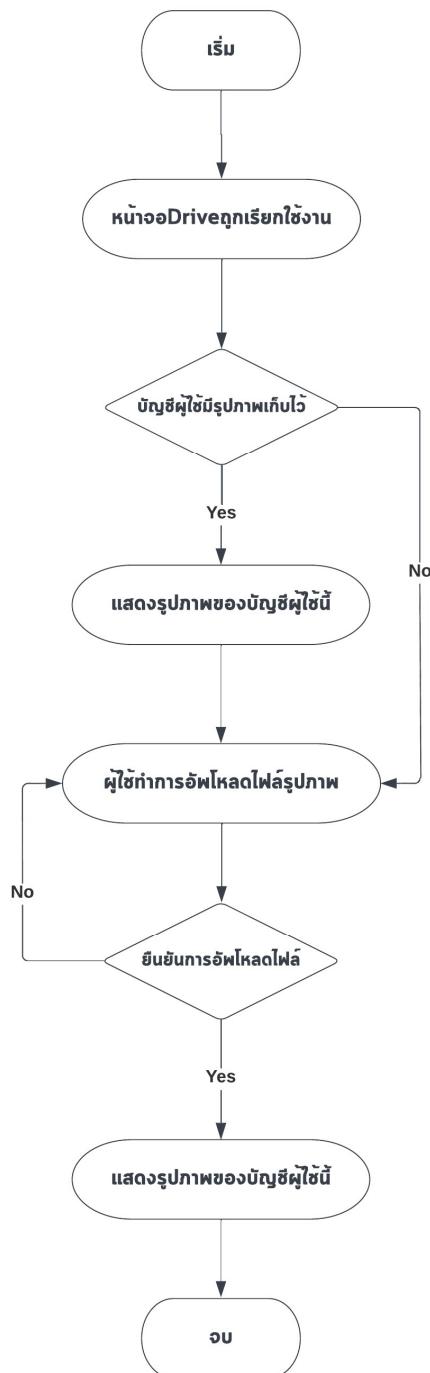
แผนภาพผังงานในส่วนของ Web Application บนเว็บที่ผู้ใช้งานเริ่มทำการใช้งานการสั่งงานประมวลผลภาพ Marketplace จนถึงการประมวลผลภาพเสร็จสิ้น โดยแผนภาพผังงานในส่วนของการสั่งงานประมวลผลภาพ จะแสดงดังรูปภาพที่ 3.6



รูปที่ 3.6 แผนภาพผังงานของการสั่งงานประมวลผลภาพบน Marketplace (Web Application)

3.3.3 แผนภาพผังงาน (Flowchart) การจัดเก็บไฟล์ภาพไว้ใน Drive (Web Application)

แผนภาพผังงานในส่วนของ Web Application บนเว็บที่ผู้ใช้งานเริ่มทำการจัดเก็บไฟล์ภาพไว้ใน Drive จะถูกจัดเก็บไว้ในรูปแบบที่ต้องการ โดยแผนภาพผังงานในส่วนของการจัดเก็บไฟล์ภาพไว้ใน Drive จะแสดงดังรูปภาพที่ 3.7

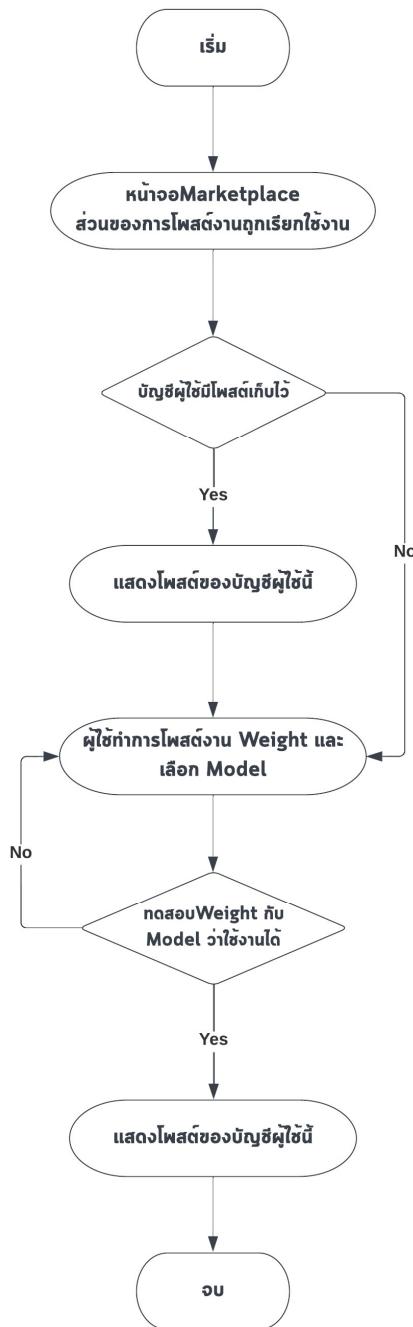


รูปที่ 3.7 แผนภาพผังงานของการจัดเก็บไฟล์ภาพไว้ใน Drive (Web Application)

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

Train (Web Application)

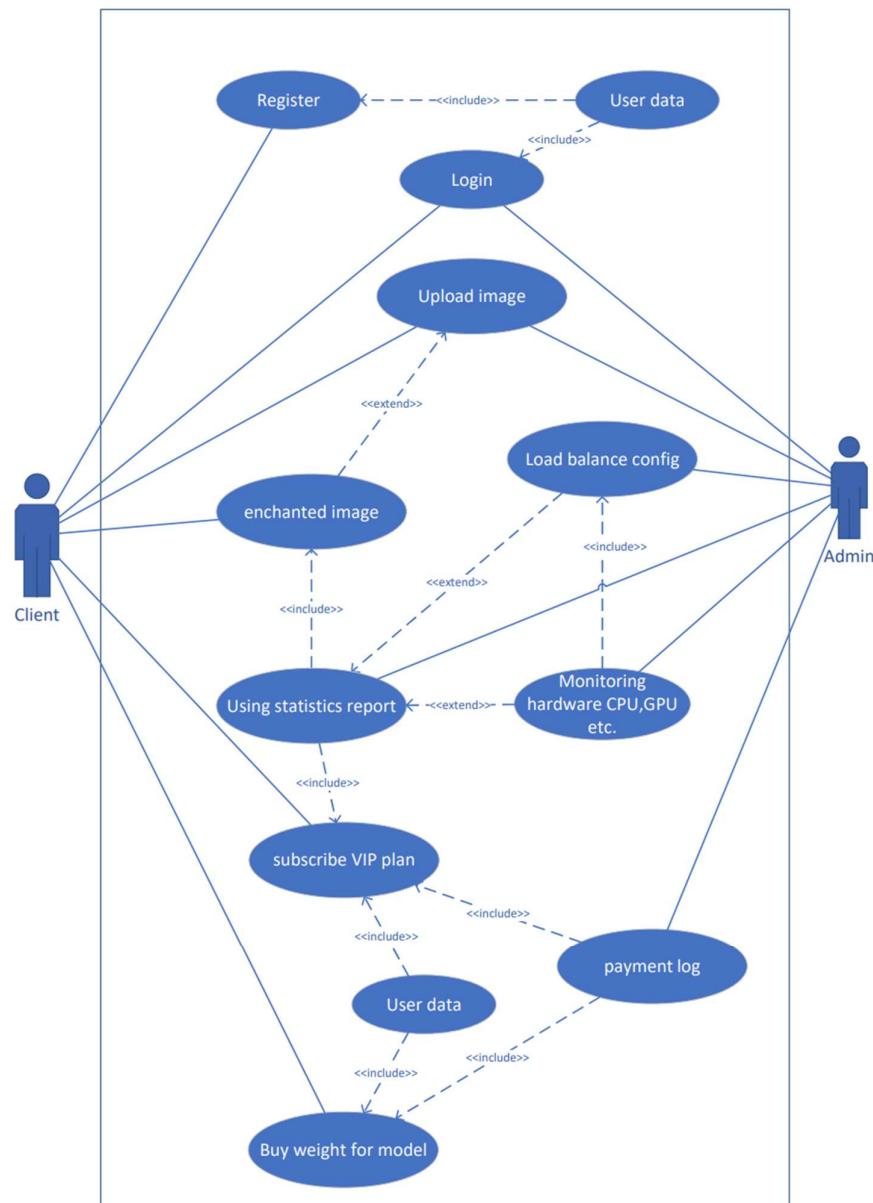
แผนภาพผังงานในส่วนของ Web Application บนเว็บที่ผู้ใช้งานเริ่มทำการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train จะถูกเรียกว่า Train จนถึงการโพสต์งานเสร็จสิ้น โดยแผนภาพผังงานในส่วนของ การขาย Weight ที่ผู้ใช้ได้ทำการ Train จะแสดงดังรูปภาพที่ 3.8



รูปที่ 3.8 แผนภาพผังงานของการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train

3.4 แผนภาพยูสเคส (Use Case Diagram)

แผนภาพยูสเคสจะแสดงให้เห็นถึงผู้ใช้ที่ถูกแบ่งเป็น 2 ประเภท คือ Client คือ ผู้ใช้ที่เข้ามายังรูปแบบลูกค้าที่ต้องการใช้บริการงานประมวลผลภาพต่างๆ 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 : มีนั่นตัวตนลงชื่อเข้าใช้	
Postcondition : ได้รับผลอัปโหลดรูปภาพและเก็บรูปภาพไว้ในฐานข้อมูล	
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 : มีนั่นตัวตนลงชื่อเข้าใช้	
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 : เลือกฟังค์ชันการปรับแต่ง และกดปุ่มปรับแต่ง	

ตารางที่ 3.6 รายละเอียดของ Monitoring hardware

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

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

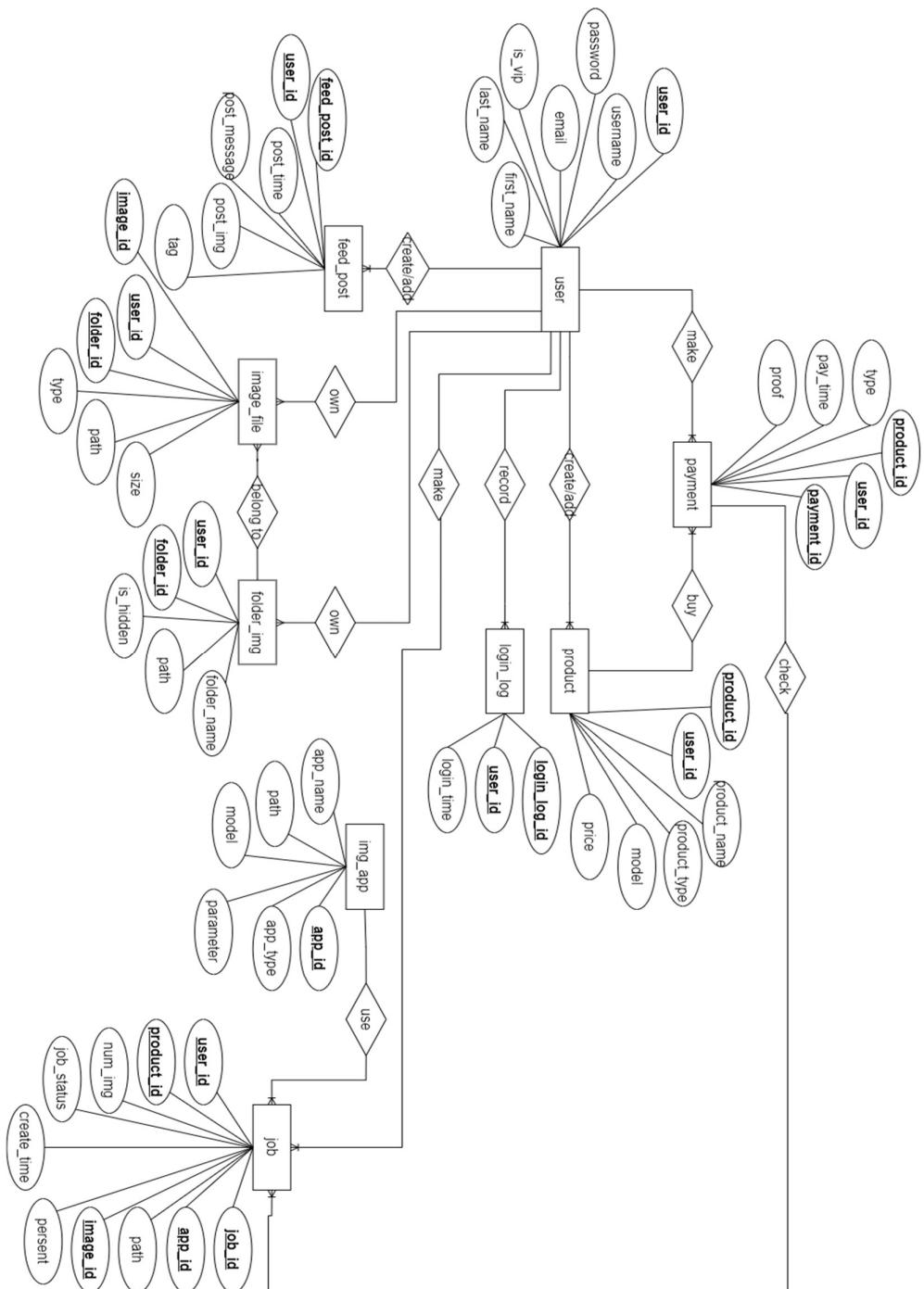
Use Case Title : Buy weight for model	Use case ID: 9
Description : ซื้อ weight สำหรับ Model เพื่อนำไปทำการประมวลผลภาพ	
Actor : Client	
Precondition : มีบัญชีตัวตนลงทะเบียนเข้าใช้งาน	
Postcondition : ได้รับ weight สำหรับ Model ไว้ในฐานข้อมูล	
Fail End Condition : แจ้งเตือนข้อความ “การซื้อขายล้มเหลว เนื่องจาก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

ตารางที่ 3.10 รายละเอียดของ Payment log

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

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

โดยรายละเอียดของแพนก้าฟแสดงความสัมพันธ์ของข้อมูลแสดงดังตารางที่ 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)	โมเดลที่สินค้าใช้ในการประมาณผล
-	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

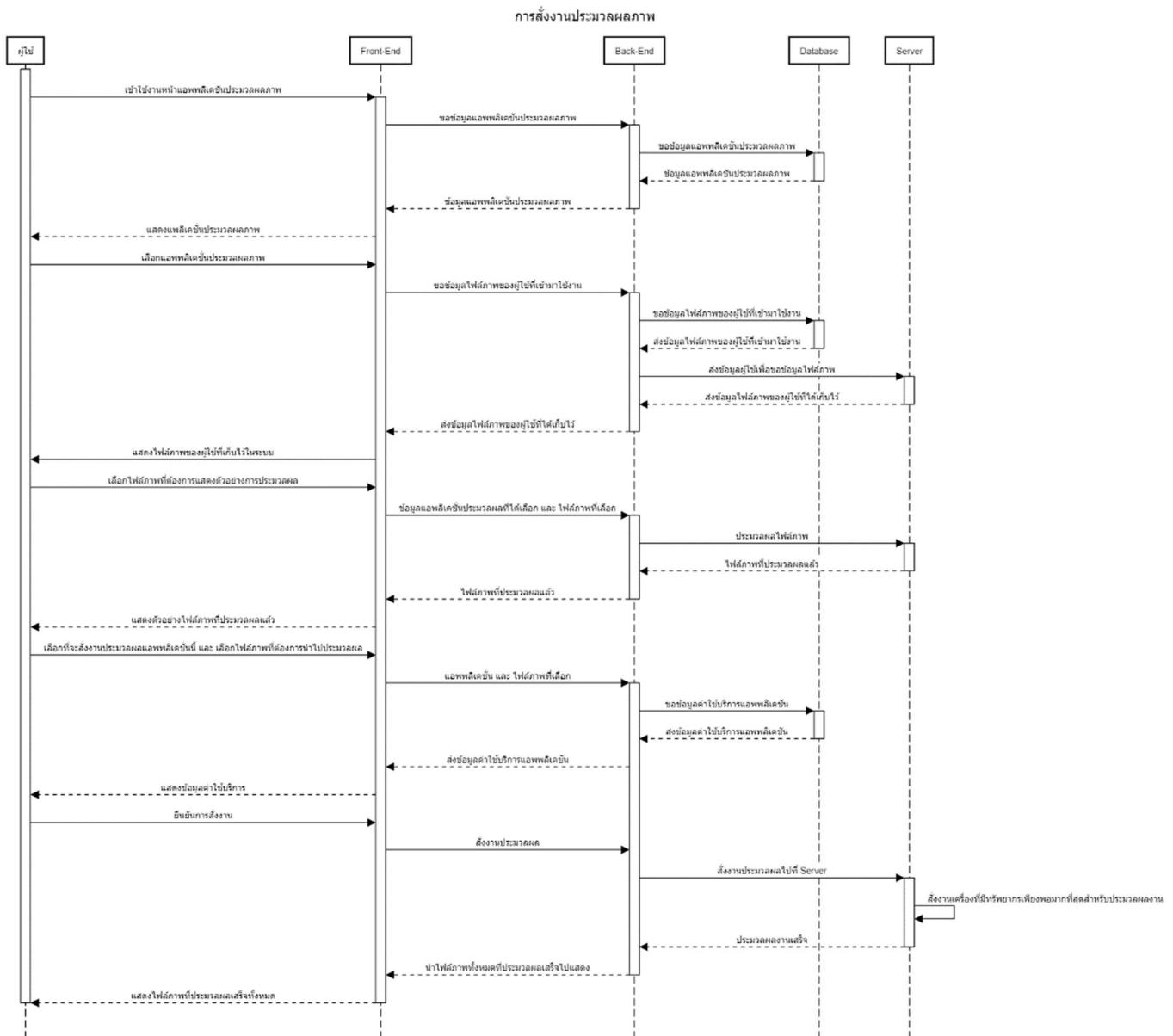
ตารางที่ 3.20 แสดงรายละเอียดตารางข้อมูลการสั่งงานประมวลผล (job)

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 ศินค้า 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 แผนภาพการทำงานแบบลำดับปฏิสัมพันธ์ (Sequence Diagram)

3.6.1 แผนภาพการทำงานในส่วนของการสั่งงานประมวลผลภาพ (Web Application)

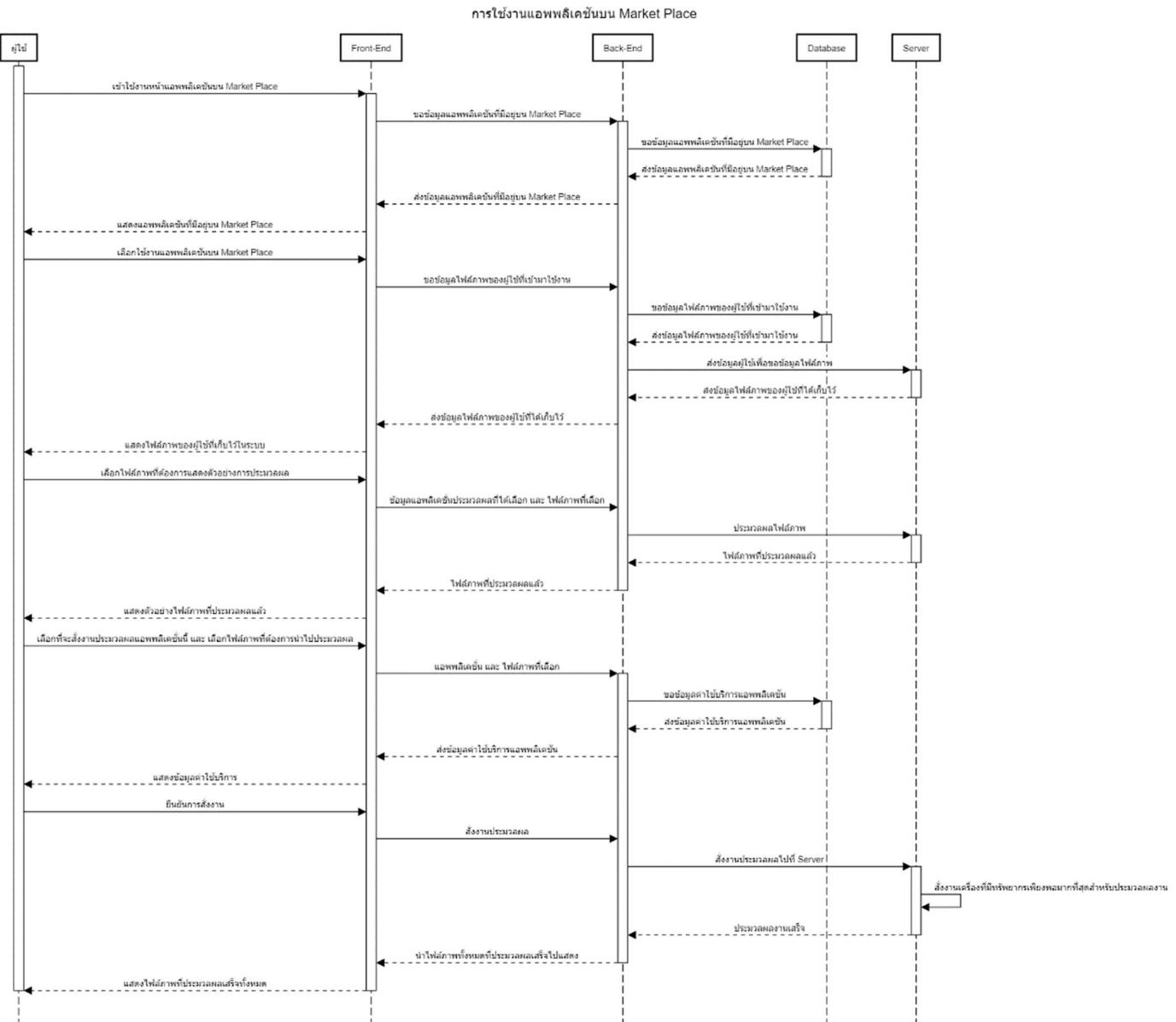
โดยรูปแบบลำดับปฏิสัมพันธ์นี้จะเริ่มต้นแต่เหตุการณ์ที่ ผู้ใช้งานเริ่มทำการใช้งานการสั่งงาน ประมวลผลภาพจนถึงการประมวลผลภาพเสร็จสิ้น โดยรูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์ โดยรวมในส่วนของการสั่งงานประมวลผลภาพ จะแสดงดังรูปภาพที่ 3.11



รูปที่ 3.11 รูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการสั่งงานประมวลผลภาพ

3.6.2 แผนภาพการทำงานในส่วนของการสั่งงานประมวลผลภาพบน Marketplace (Web Application)

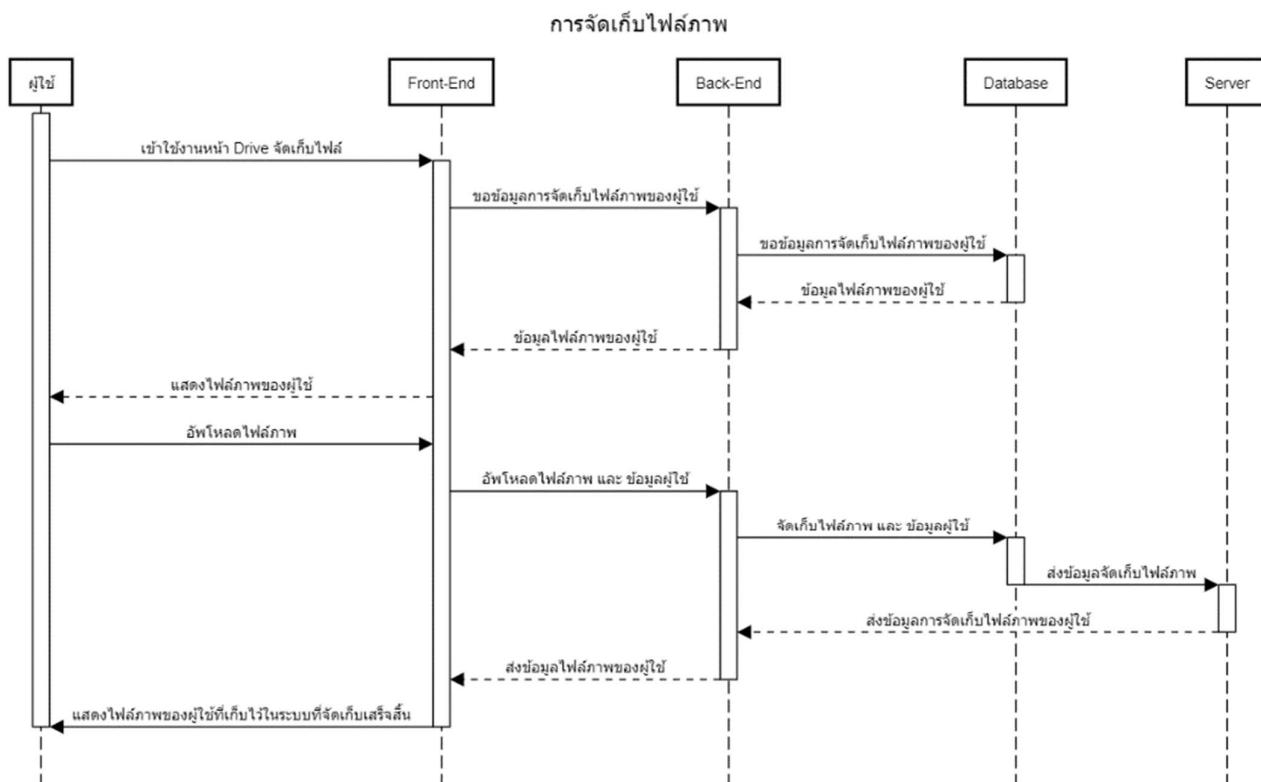
โดยรูปแบบคำอันปฏิสัมพันธ์นี้จะเริ่มตั้งแต่เหตุการณ์ที่ ผู้ใช้งานเริ่มทำการใช้งานการสั่ง งาน ประมวลผลภายนอก Marketplace จนถึงการประมวลผลภาพเสร็จสิ้น โดยรูปภาพแสดงการทำงานแบบ คำอันปฏิสัมพันธ์โดยรวมในส่วนของการสั่งงานประมวลผลภาพ จะแสดงดังรูปภาพที่ 3.12



รูปที่ 3.12 รูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการสั่งงานประมวลผลภาพบน

3.6.3 แผนภาพการทำงานในส่วนของการจัดเก็บไฟล์ภาพไว้ใน Drive (Web Application)

โดยรูปแบบลำดับปฏิสัมพันธ์นี้จะเริ่มตั้งแต่เหตุการณ์ที่ ผู้ใช้งานเริ่มทำการจัดเก็บไฟล์ภาพไว้ใน Drive จนถึงการจัดเก็บไฟล์ภาพเสร็จสิ้น โดยรูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการจัดเก็บไฟล์ภาพไว้ใน Drive จะแสดงดังรูปภาพที่ 3.13

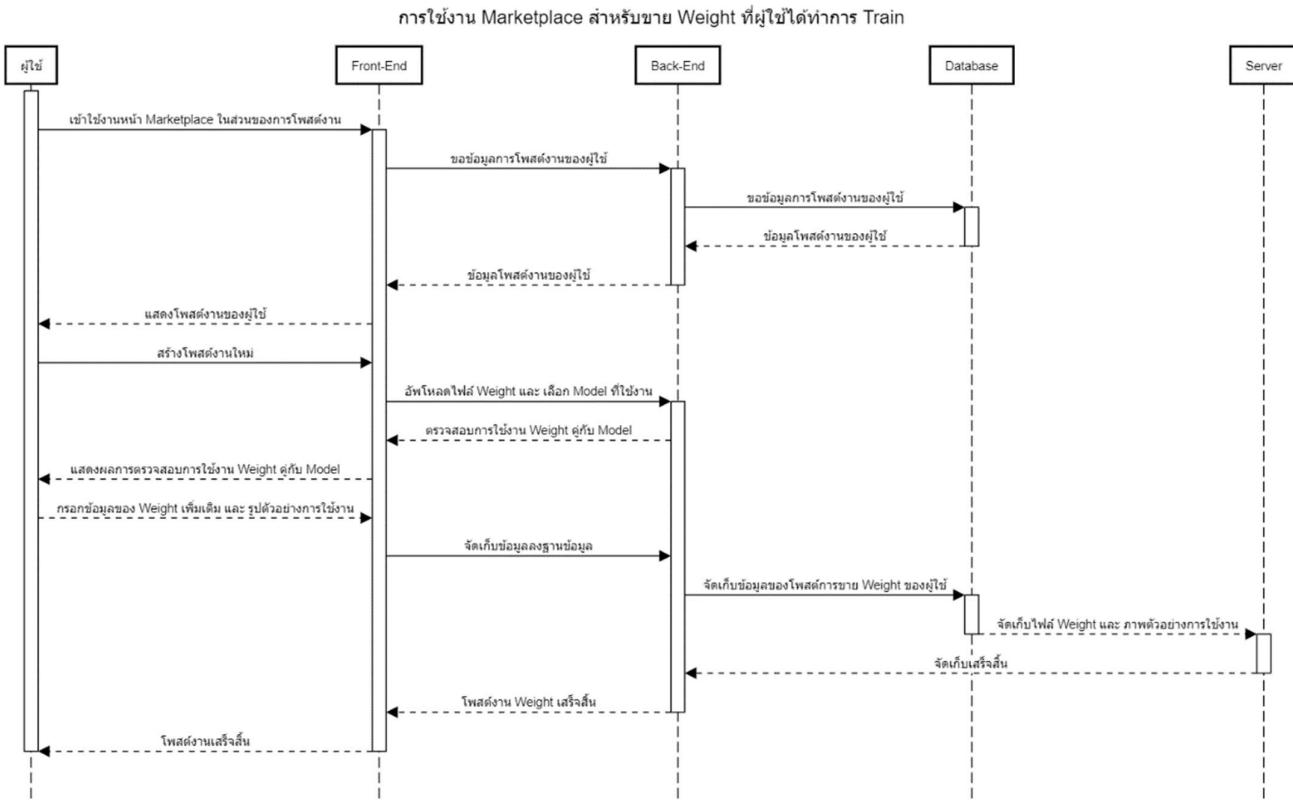


รูปที่ 3.13 รูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการจัดเก็บไฟล์ภาพไว้ใน Drive

3.6.4 แผนภาพการทำงานในส่วนของการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train (Web Application)

Train (Web Application)

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



รูปที่ 3.14 รูปภาพแสดงการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train

3.7 การออกแบบส่วนต่อประสานกับผู้ใช้ (User Interface)

3.7.1 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนลงชื่อเข้าใช้ (Login)

จะแสดงช่องข้อความสำหรับกรอก Email และ รหัสผ่านเพื่อที่จะนำไปยืนยันตัวตน สำหรับเข้าใช้งาน web application โดยการออกแบบในส่วนลงชื่อเข้าใช้ จะแสดงดังรูปภาพที่ 3.15

The screenshot shows a dark-themed login interface. On the left, there is a vertical sidebar with the project title 'IPautSons' at the top, followed by 'About This project' and 'This project Brief description'. The main area has the same 'IPautSons' title at the top. It contains fields for 'Username' and 'Password', each with a corresponding input field. Below these are two buttons: 'Login' (blue) and 'Clear' (grey). A link 'Register here!' is also present.

รูปภาพที่ 3.15 การออกแบบส่วนลงชื่อเข้าใช้

3.7.2 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนลงทะเบียนสมัครสมาชิก (Register)

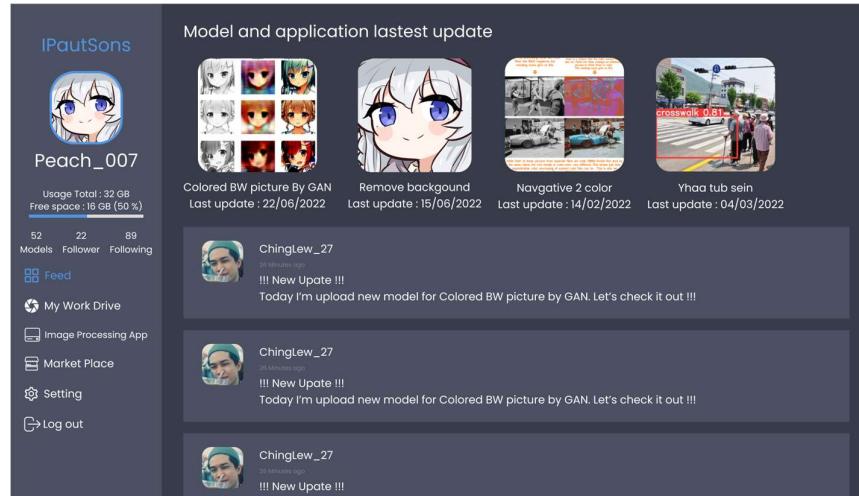
จะแสดงช่องข้อความสำหรับกรอก ชื่อ Username , Email , รหัสผ่าน , รหัสผ่านอีกครั้ง , ชื่อจริง และ นามสกุล เพื่อที่จะนำไปยืนยันตัวตนสำหรับเข้าใช้งาน web application โดยการออกแบบในส่วนลงทะเบียนสมัครสมาชิก จะแสดงดังรูปภาพที่ 3.16

The screenshot shows a dark-themed registration interface. On the left, there is a vertical sidebar with the project title 'IPautSons' at the top, followed by 'About This project' and 'This project Brief description'. The main area has the title 'Project name' at the top. It contains several input fields: 'Username', 'Email', 'Password', 'Password again', 'Firstname', and 'Lastname', each with a corresponding input field. Below these fields are two buttons: 'Regis' (blue) and 'Clear' (grey).

รูปภาพที่ 3.16 การออกแบบส่วนลงทะเบียนสมัครสมาชิก

3.7.3 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนหน้าข่าวสาร (Feed page)

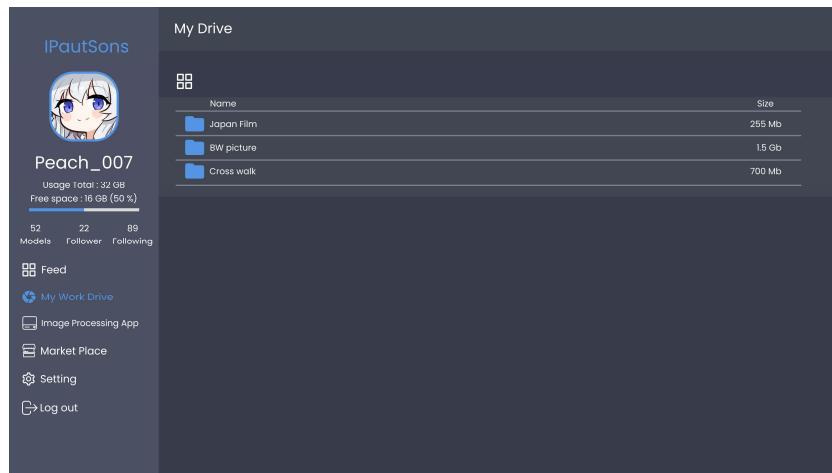
จะแสดงผลลั่งข้อความที่ประกอบด้วยรูปภาพและชื่อของผู้ตั้งหัวข้อข่าว เวลาที่ตั้งหัวสาร เนื้อหาของข่าวสาร และ ส่วนที่เป็นหน้าข่าวสารของสินค้า Weight model และ Image processing application ที่มีการ Update ล่าสุด โดยการออกแบบในส่วนหน้าข่าวสาร จะแสดงดังรูปภาพที่ 3.17



รูปภาพที่ 3.17 การออกแบบส่วนหน้าข่าวสาร

3.7.4 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนหน้าพื้นที่เก็บบันทึกรูปภาพ (Drive page)

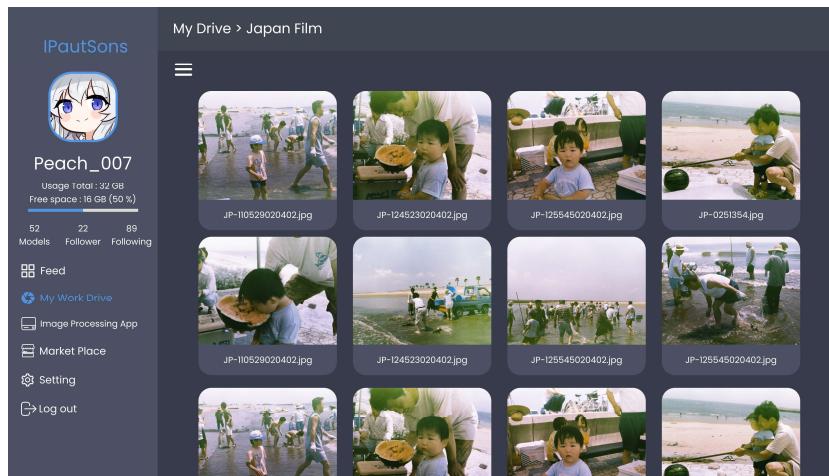
จะแสดงลิสต์ของไฟล์เดอร์ที่เก็บรูปภาพเพื่อให้ผู้ใช้เข้าถึงรูปภาพที่เก็บไว้อยู่ได้ โดยการออกแบบในส่วนพื้นที่เก็บบันทึกรูปภาพ จะแสดงดังรูปภาพที่ 3.18



รูปภาพที่ 3.18 การออกแบบส่วนพื้นที่เก็บบันทึกรูปภาพ

3.7.5 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนพื้นที่แสดงรูปภาพที่เก็บบันทึกไว้ (Show Drive page)

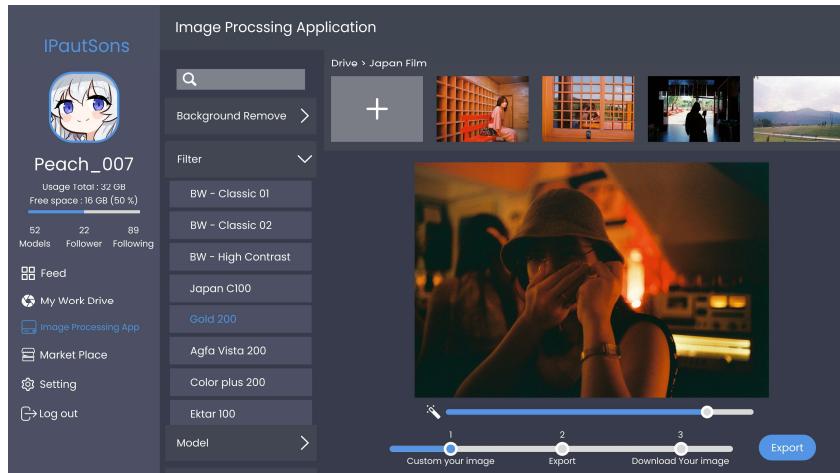
จะแสดงรูปภาพพร้อมรายละเอียดของรูปที่อยู่ในโฟลเดอร์ที่เก็บรูปภาพนั้นๆ โดยการออกแบบในส่วนพื้นที่แสดงรูปภาพที่เก็บบันทึกไว้ จะแสดงดังรูปภาพที่ 3.19



รูปภาพที่ 3.19 การออกแบบส่วนพื้นที่แสดงรูปภาพที่เก็บบันทึกไว้

3.7.6 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนแอปพลิเคชันประมวลผลภาพ (Image application page)

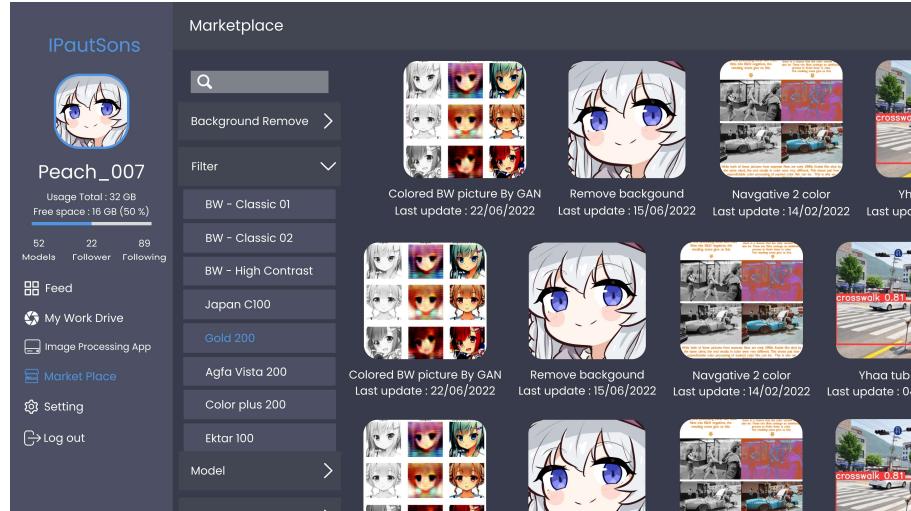
จะแสดงແນບເລືອກ Application ການປັບແຕ່ງຮູບທີ່ດ້ານຊ້າຍ ແລະ ແສດງຜລກການປັບແຕ່ງທີ່
ທາງດ້ານພື້ນທີ່ຕຽດກລາງ ໂດຍທີ່ດ້ານນີ້ຈະເປັນພື້ນທີ່ເລືອກຮູບທີ່ຕ້ອງການ ໂດຍການອອກແນບໃນສ່ວນແອປ
ພລິເຄີບປະມາລຸກກາພ ຈະ ແສດງດັງຮູບກາພທີ່ 3.20



ຮູບກາພທີ່ 3.20 ການອອກແນບສ່ວນແອປພລິເຄີບປະມາລຸກກາພ

3.7.7 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนพื้นที่ซื้อขาย (Marketplace page)

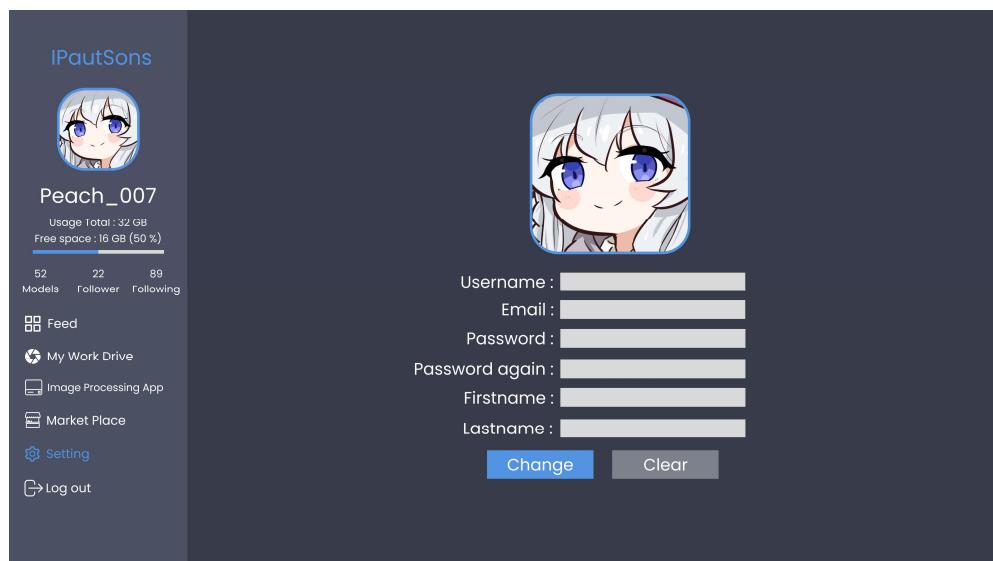
จะแสดงແນບເລືອກ Application ການປະບົບແຕ່ງຮູປທີ່ດ້ານຊ້າຍ ໂດຍຈະເປັນການກອງເລືອກ Application ທີ່ສັນໃຈ ແລະ ໂດຍທີ່ຕຽດກາງຈະເປັນສິນຄ້າທີ່ນໍາມາຫຍເຊັ່ນ Weight ທີ່ຝ່ານການ train ມາແບ່ວເປັນ ຕັ້ນ ໂດຍການອອກແນບໃນສ່ວນພື້ນທີ່ຂໍ້ອາຍ ຈະແສດງດັ່ງຮູປກາພທີ່ 3.21



ຮູປກາພທີ່ 3.21 ການອອກແນບສ່ວນພື້ນທີ່ຂໍ້ອາຍ

3.7.8 การออกแบบສ່ວນຕ່ອປະສານກັນຜູ້ໃຊ້ໃນສ່ວນຕັ້ງຄ່າບັນຫຼື (Setting page)

ຈະແສດງກລ່ອງຂໍ້ອານວັນທີມີຂໍ້ອຸນດອນຜູ້ໃຊ້ຂໍ້ອູ່ ໂດຍສາມາດແກ່ໄຂໄດ້ຮັວມໄປຢຶງຮູປໂປຣໄຟລ໌ ເອງທີ່ສາມາດປັບປຸງໄໝ ໂດຍການອອກແນບໃນສ່ວນຕັ້ງຄ່າບັນຫຼື ຈະແສດງດັ່ງຮູປກາພທີ່ 3.22



ຮູປກາພທີ່ 3.22 ການອອກແນບສ່ວນຕັ້ງຄ່າບັນຫຼື

3.7.9 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนตั้งค่าและคุณบันทึกการใช้ทรัพยากร (Monitoring and Config page)

จะแสดงกราฟสถิติการใช้งานhardt์ต่างๆ เช่น CPU , GPU , RAM เป็นต้น โดยสามารถปรับแต่งการสั่งงานได้ภายในหน้าจอ โดยการออกแบบในส่วนตั้งค่าและคุณบันทึกการใช้ทรัพยากร จะแสดงดังรูปภาพที่ 3.23



รูปภาพที่ 3.23 การออกแบบส่วนตั้งค่าและคุณบันทึกการใช้ทรัพยากร