**แอพพลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย  
IMAGE PROCESSING APPLICATION USING TASK SCHEDULING ON NETWORK SYSTEM**

**พศิน จันทรทัน  
สุธี สาระพันธ์**

**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญวิศวกรรมศาสตร์บัณฑิต**

**สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ปีการศึกษา 2565**

ปริญญานิพนธ์ปีการศึกษา 2565

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แอพพลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย

Image Processing Application using Task Scheduling on Network System

ผู้จัดทํา

1. นายพศิน จันทรทัน รหัสนักศึกษา 63015121

2. นายสุธี สาระพันธ์ รหัสนักศึกษา 63015190

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_อาจารย์ที่ปรึกษา

(รศ. ดร.อรฉัตร จิตต์โสภักตร์)

**แอพพลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย**

นายพศิน จันทรทัน 63015121

นายสุธี สาระพันธ์ 63015190

รศ.ดร.อรฉัตร จิตต์โสภักตร์ อาจารย์ที่ปรึกษา

ปี การศึกษา 2565

**Image Processing Application using Task Scheduling on Network System**

Mr. Pasin Chantharathan 63015121

Mr. Sutee Saraphan 63015190

Assoc.Prof.Dr. Orachat Chitsobhuk Advisor

Academic Year 2022

**บทที่ 1**

**บทนํา**

**1.1 ความเป็นมาของปัญหา**

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

ดังนั้นโครงงาน แอพพลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย มีจุด ประสงค์เพื่อสร้างระบบการแบ่งการจัดลำดับงานประมวลผลของผู้ใช้บริการแอพพลิเคชั่น เพื่อเป็นอีกทาง เลือกหนึ่งในการใช้งานการประมวลผลภาพในรูปแบบต่าง ๆ อย่างมีประสิทธิภาพ และ รองรับการประมวลผลภาพจำนวนมาก พร้อมด้วยรองรับแอพพลิเคชั่นที่สามารถนำเข้ามาเพิ่มในอนาคตได้อีกด้วย และ ในส่วนของ Weight Model ที่นักศึกษาได้ทำการ Train มานั้นก็สามารถนำเข้าภายในแอพพลิเคชั่นเพื่อเปิดให้บริการในแอพพลิเคชั่นของเราได้เพื่อทำการซื้อขาย หรือ เป็นที่แสดงผลงานของนักศึกษา และ เป็นอีกทางเลือกให้นักศึกษามีรายได้

โดยสรุปได้ว่า การตกแต่งรูปภาพ หรือ การปรับแต่งภาพ นั้นมีการใช้อยู่แพร่หลาย และ คนส่วนใหญ่มีการใช้งานเรื่องการประมวลผลภาพกันเป็นปกติ การนำ Weight Model มาใช้งาน หรือ จัดแสดงผลงาน และ การซื้อขาย ซึ่งทั้งหมดที่กล่าวมาคือส่วนมากใช้งานบนแอพพลิเคชั่นต่างๆผ่านอินเทอร์เน็ตเราจึงพัฒนาโครงงานบนพื้นฐานของเว็บแอพพลิเคชั่นซึ่งสอดคล้อง และ รองรับกับปัญหาดังกล่าว โครงงาน “แอพพลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย” นี้ถูกจัดทําขึ้นมาเพื่อแก้ไขในส่วนของกระบวนการนี้ โดยมีการแบ่งเป็น 3 ส่วนคือ

1) ส่วนการประมวลผลงานแบบการจัดลำดับงาน (Server) Task Scheduling การแบ่งจัดลำดับการทำงานประมวลผลแต่ละงานไปแต่ละเครื่อง โดยมีการกำหนดทรัพยากรที่ใช้ในแต่ละงานประมวลผลเพื่อให้ได้ประสิทธิภาพเพียงพอต่อความต้องการของงานประมวลผล และ ใช้ทรัพยากรของหน่วยประมวลผลให้คุ้มค่ากับทรัพยากรที่มีอยู่ของหน่วยประมวผล

2) ส่วนแอพพลิเคชั่นการประมวลผลภาพ (User) Image Processing Application ส่วนระบบ สําหรับประมวลผลภาพที่ระบบได้มีการเตรียมรูปแบบประมวลผลภาพต่าง ๆ ที่ หลากหลายให้ผู้ใช้ได้เข้ามาใช้งานการประมวลผลภาพจำนวนมากที่ทางเราได้เตรียมจัดทำไว้

3) ส่วนแอพพลิเคชั่นให้บริการซื้อขาย Weight โมเดล (User) Market Place ส่วนระบบสำหรับให้ผู้ใช้มีการซื้อ Weight Model และ นำเข้า Weight Model ที่ได้ทำการเทรนนิ่งไว้แล้วมาเปิดให้ใช้บริการบนแอพพลิเคชั่นเพื่อทำการ ขาย หรือ เพื่อทดลองงานต่าง ๆ โดยเป็นจุดโชว์ผลงานของตัว Weight Model ที่ได้ทำการเทรนนิ่งไว้ได้อีกด้วย

**1.2 วัตถุประสงค์ของโครงงาน**

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) ได้รับประสบการณ์ในการพัฒนาแอพพลิเคชั่นการประมวลผลภาพที่นิยมในปัจจุบัน

**1.4 ขอบเขตการวิจัย**

1) ระบบ Cluster Computing เชื่อมต่อผ่านระบบเครือข่ายท้องถิ่น (LAN)

2) ระบบ Cluster Computing ในแอพพลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่ายนี้นั้นเป็นการแบ่งงานในแต่ละเครื่องเพื่อให้ใช้ทรัพยากรที่จำกัด และ จำกัดทรัพยากรในแต่ละงานประมวผล

3) การจัดการลำดับงานบนระบบขึ้นอยู่กับประสิทธิภาพฮาร์ดแวร์ของ Node ที่พร้อมใช้งานในช่วงเวลานั้น

4) แอพพลิเคชั่นในส่วนของฝากไฟล์ข้อมูล Structures ของ Directory แต่ละ User นั้นการแตกระดับขั้นของ Directory ได้ 1 ขั้น

5) แอพพลิเคชั่นในส่วนของการประมวลผลนั้นสามารถทำการประมวลผลภาพจำนวนมาก จึงปรับค่า Parameter ได้ 1 ค่าในงานประมวลผลภาพจำนวนมาก

6) ส่วนของ Market Place ในการนำเข้า Weight นั้นจะต้องรองรับกับตัว Model ที่มีให้บริการภายในแอพพลิเคชั่นเท่านั้น

**บทที่ 2**

**เอกสารและทฤษฎีที่เกี่ยวข้อง**

ในการจัดทําโครงงาน ผู้จัดทําได้ทําการศึกษาทฤษฎีที่เกี่ยวข้อง, งานที่เกี่ยวข้อง, เครื่องมือที่ใช้

งานในการพัฒนาระบบ และงานวิจัยที่เกี่ยวข้อง เพื่อนํามาปรับใช้และเป็นแนวทางในการทํา โครงงาน ดังต่อไปนี้

**2.1 ทฤษฎีที่เกี่ยวข้อง**

**2.1.1 Digital Image Processing**

การประมวลผลภาพดิจิทัล หรือ Digital Image Processing คือ การใช้คอมพิวเตอร์ดิจิทัลในการประมวลผลภาพดิจิทัลผ่านอัลกอริธึม ภาพในที่นี้รวมความหมายถึงสัญญาณในระบบดิจิทัล 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) การดึงภาพ (Image retrieval) ทำให้จุดที่น่าสนใจของภาพให้เด่นชัด

4) การวัดรูปแบบของภาพ (Measurement of pattern) วัตถุในภาพทั้งหมดนั้นจะถูกวัดค่ามาเพื่อหารูปแบบของวัตถุภายในภาพ

5) การรับจดจำภาพ (Image Recognition) สามารถแยกแยะวัตถุในภาพได้

**2.1.2 Cluster Computing**

ระบบคลัสเตอร์ หรือ คลัสเตอริ่ง เป็นการเชื่อมต่อระบบการทำงานของกลุ่มคอมพิวเตอร์เข้าด้วยกันภายใต้ระบบเครือข่าย มีความสามารถในการกระจายงานที่ทำไปยังเครื่อง ภายในระบบเพื่อให้การประมวลผลมีประสิทธิภาพสูงขึ้น โดยอาจเทียบเท่าซุปเปอร์คอมพิวเตอร์ หรือ สูงกว่าสำหรับการประมวลผลงานที่มีความซับซ้อนโดยเฉพาะงานด้านวิทยาศาสตร์ เช่น การจำลองโครงสร้างของโมเลกุลทางเคมี, การวิเคราะห์เกี่ยวกับตำแหน่งการเกิดพายุสุริยะ, การประมวลผลภาพ เป็นต้น ดังรูป 2.1.2

Diagram

Description automatically generated

ภาพที่ 2.1.2 Cluster Computing

**2.1.3 การประมวลผลแบบกลุ่มเมฆ (Cloud Computing)**

การประมวลผลแบบกลุ่มเมฆ หรือ การให้บริการที่ตามความต้องการของผู้ใช้งานโดยที่ผู้ใช้งานระบุความต้องการ หรือ เลือกรูปแบบการใช้งานที่เหมาะสมกับความต้องการไปยังระบบการประมวลผลแบบกลุ่มเมฆ 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

Diagram

Description automatically generated

ภาพที่ 2.1.3 การประมวลผลแบบกลุ่มเมฆ (Cloud Computing)

**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

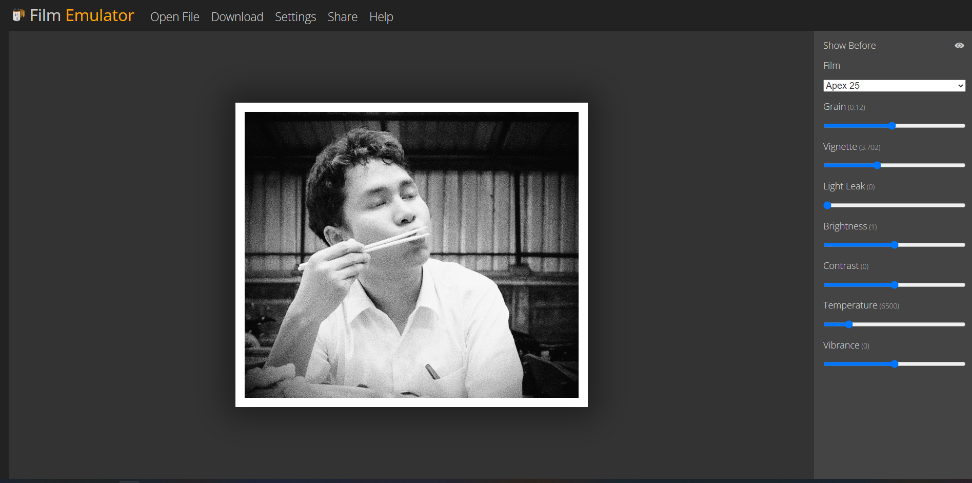
สิ่งที่ AnimeFilterยังไม่มี หรือ นำมาปรับปรุง มาพัฒนาต่อในแอพพลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย

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 ปรับแต่งสีของภาพตามที่ผู้ใช้ต้องการโดยมี parameter ที่กำหนดมาให้โดยมี Preset จากกล้องฟิล์มให้ผู้ใช้งานเข้ามาใช้บริการ ตัวอย่างแสดงดังรูปที่ 2.2.2



รูปที่ 2.2.2 ตัวอย่างของ Film Emulator

สิ่งที่ Film Emulator ยังไม่มี หรือ นำมาปรับปรุง มาพัฒนาต่อในแอพพลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย

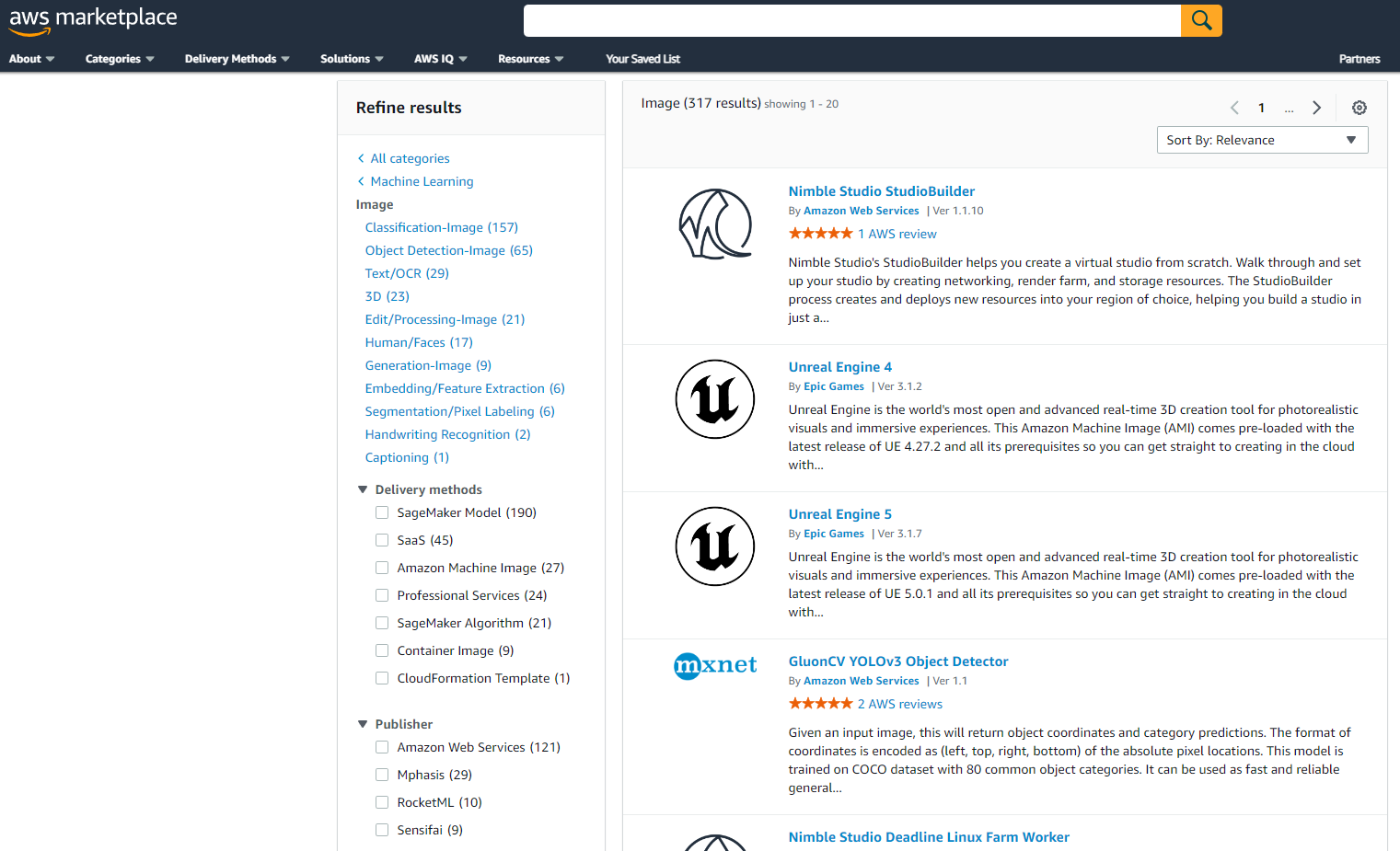
1. การรองรับการสั่งงานประมวลผลภาพจำนวนมาก

2. การรองรับการประมวลผลภาพจำนวนมากในการปรับค่า Parameter เดียวกัน

3. มี Application ในการปรับแต่งภาพหลากหลายแบบมากกว่า

**2.2.3** **aws marketplace**

aws marketplace หรือ aws.amazon.com/marketplace/ เป็นผู้ให้บริการระบบ Cloud ที่ให้ผู้ใช้เลือกใช้บริการ Model ต่าง ๆ ที่ทาง aws รองรับโดยให้ผู้ใช้จ่ายค่าบริการเป็นรายชั่วโมงเพื่อเข้าใช้บริการงานต่าง ๆ เช่น YOLO, Unreal Engine 4 และ อื่น ๆ อีกมากมาย ดังรูป 2.2.3



รูปที่ 2..2.3 ตัวอย่างของ aws marketplace

สิ่งที่ 3 aws marketplace ยังไม่มี หรือ นำมาปรับปรุง มาพัฒนาต่อในแอพพลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย

1. การให้บริการเป็นจำนวนภาพที่ประมวลผลแทนการใช้บริการรายชั่วโมง

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

Diagram

Description automatically generated  
รูปที่ 2.3.1 Component system ของ Vue.js

**2.3.2 Docker**

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

**2.3.3 Kubernetes**

คือ ถูกพัฒนาโดย Google เพื่อนำมา จัดการระบบคลัสเตอร์ (Cluster management software) สำหรับ Docker Container โดยตัว Kubernetes นั้นใช้สำหรับการ จัดการปล่อยซอฟต์แวร์อัตโนมัติ (automating software deployment) และ ทำการ Scaling ตรวจสอบความถูกต้องของ Container บนระบบ Cluster ได้ตลอดเวลา ตัว Kubernetes นั้นเนื่องจากเป็น Open-Source จึงมีการนำไปดัดแปลงแก้ไขเป็น Kubernetes เวอร์ชั่นอื่น ๆ มาอีกมากมายไม่ว่าจะเป็น MicroK8S, K3D, MiniKube โดยทั้งหมดที่กล่าวมามีเงื่อนไข และ การทำงาน การใช้งานอยู่ในรูปแบบเดียวกันโดยมีหลักการทำงานพื้นฐานเหมือนกันดังรูป 2.3.3

Diagram

Description automatically generated

รูปที่ 2.3.2 Component system ของ Vue.js

**2.3.4 Python**

Python เป็นภาษาในการเขียนโปรแกรมที่ใช้อย่างแพร่หลายในการพัฒนาซอฟต์แวร์ เว็บแอปพลิเคชั่น วิทยาศาสตร์ข้อมูล การประมวลผลภาพ และ แมชชีนเลิร์นนิง (ML) มีการเรียนรู้ง่าย และ สามารถทำงานบนแพลตฟอร์มได้มากมายไม่ว่าจะเป็นบน Window, MacOS, Linux ทั้งนี้ Python เป็นซอฟต์แวร์ที่สามารถดาวน์โหลด และ ใช้งานได้ฟรี

**2.3.5 Django**

Django เป็น Framework ที่ถูกเขียนด้วยภาษา Python สำหรับพัฒนาเว็บแอพลิเคชั่น โดยที่สามารถพัฒนาได้อย่างรวดเร็ว และ มีประสิทธิภาพ เหมาะกับการพัฒนาเว็บที่มีการใช้งานบ่อย เช่น การตรวจสอบการดึงข้อมูลจากฐานข้อมูล การจัดการคุกกี้ และ การทำงานควบคู่ไปกับ Python ซอฟต์แวร์อื่น ๆ มาปรับให้ใช้งานภายในเว็บแอพลิเคชั่นได้ ตัว Django เป็น open-source ที่สามารถนำมาใช้งานได้ฟรี และ พัฒนาได้ฟรีโดยมี Community ที่ช่วยดูแลรักษาซอฟต์แวร์ตัวนี้ และมีองค์กรไม่แสวงหาผลกำไรชื่อว่า Django Software Foundation ที่ทำการพัฒนา ปรับปรุง และ บำรุงรักษา ตัว Django ให้มีคุณภาพสูง และ ฟีเจอร์อีกมากมายให้ใช้งานฟรีตลอดมา

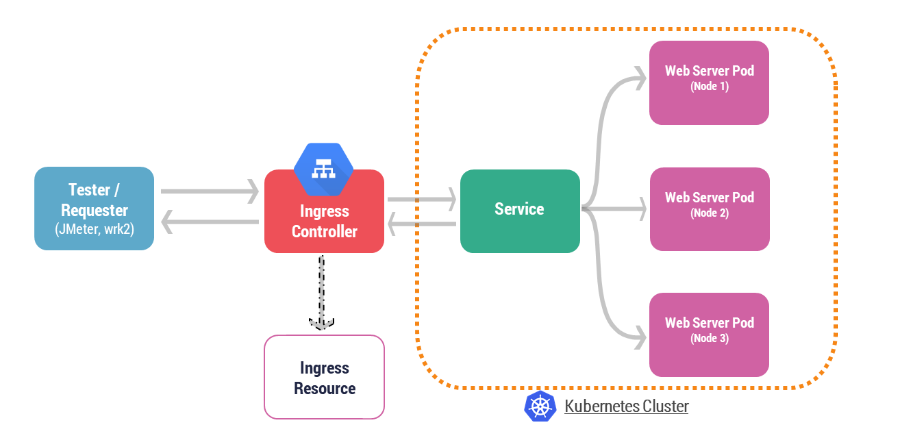
**2.3.6 MongoDB**

MongoDB เป็น NoSQL Database ที่สามารถใช้งานข้าม Platform ต่าง ๆ ได้ โดยตัว MongoDB ได้ใช้ NoSQL เป็นการเข้าถึงข้อมูลโดยใช้เทคนิคของการดึงข้อมูลผ่าน Key Pair Value โดยเหมาะกับงานที่ต้องการเข้าถึงข้อมูลได้อย่างรวดเร็ว 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 คุณอาธิป พวงลำไย และ คุณชัยพร เขมะภาตะพันธ์ กล่าวถึงเรื่องการทดลองเพื่อประเมินสมรรถนะ และเปรียบเทียบประสิทธิภาพของ ระบบ Load balance แบบ Layer – 7 ที่ถูกใช้งานในสภาพแวดล้อมการทำงานที่อยู่ในรูปแบบของ Container cluster ด้วย Kubernetes โดยแผนผังการวางระบบในการวิจัยจะเป็นดังรูป 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 กล่าวถึงเรื่องการ กระบวนการแยกแยะพื้นผิวของรูปภาพโดยการใช้อัลกอริทึม gray-level difference (GLD) โดยจะมี 4 ขั้นตอนคือ 1. ประเมินความแตกต่างของพื้นผิวรูปภาพโดยเปรียบทียบกับข้อมูลที่เตรียมเอาไว้ก่อนหน้า 2.ใช้ตัวกรองแบบ Wiener เพื่อลบ Gaussian blur noise ที่ที่อาจเกิดขึ้นหากค่าความแตกต่างมีน้อย 3. ทำซ้ำวิธีการที่ 2. ไปจนกระทั่งค่าความแตกต่างสูงกว่าข้อมูลที่เตรียมเอาไว้ 4. ใช้อัลกอริทึม gray-level difference แบบดั้งเดิม เพื่อเก็บข้อมูลพื้นผิวจากรูปภาพที่ผ่านการประมวลผลแล้ว เพื่อนำมาวัดประสิทธิภาพเทียบกับอัลกอริทึม gray-level difference แบบใหม่

โดยโครงงานนี้ได้นำเอกสารนี้มาเพื่อศึกษาใช้ในการพัฒนาส่วนของการพัฒนา 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 และนอกจากนี้ก็ยังนำเสนอวิธีการอื่นที่ใช้ในการปรับแต่งความคมชัดเช่น the bi-histogram equalization (BHE) และ Global histogram equalization เป็นต้น โดยรูปที่ 2.4.3 จะเป็นรูปผลลัพธ์จากการปรับต่างด้วยวิธีที่แตกต่างกัน

รูปภาพประกอบด้วย ข้อความ, ต้นไม้, ป่า, ไม้

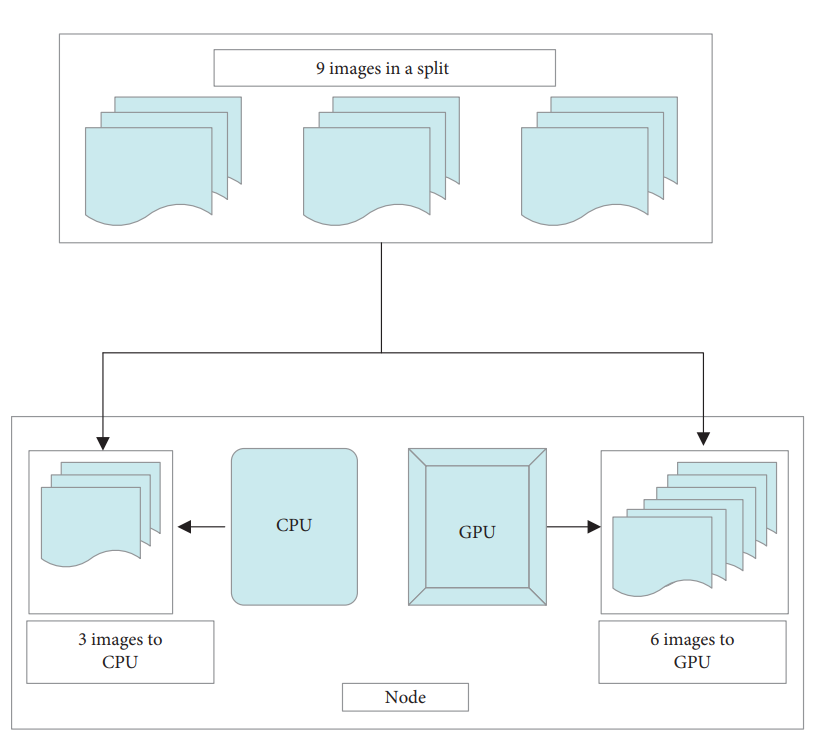
คำอธิบายที่สร้างโดยอัตโนมัติ

รูปที่ 2.4.3 รูปผลลัพธ์จากการปรับต่างความตมชัดด้วยวิธีที่แตกต่างกันโดย (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

โดยโครงงานนี้ได้นำเอกสารนี้มาศึกษาเพื่อใช้ในการพัฒนาส่วน Image processing application เพื่อสร้าง Filtering และการปรับความคมชัดของรูปภาพที่จะถูกปรับแต่งภายใน Image processing application

**2.4.4 Efficient Processing of Image Processing Applications on CPU/GPU**

งานวิจัยนี้จัดทำขึ้นในปี 2020 โดย Najia Naz, Abdul Haseeb Malik, Abu Bakar Khurshid, Furqan Aziz, Bader Alouffi, M. Irfan Uddin และ Ahmed AlGhamdi กล่าวถึงเรื่องการกระบวนการ การแบ่งงานที่มีประสิทธิภาพสำหรับ Image processing application จากปกติการที่ประมวลผลงานด้าน image processing สิ่งที่สามารถพบเจอได้เป็นปกติคือ การกระจายภาระงานที่ไม่สมดุล ซึ่งทำให้ประสิทธิภาพของการทำงานนั้นลดลง โดยการแบ่งนั้นจะแบ่งออกเป็น 2 ช่วง ช่วงแรกจะเป็นการแบ่งและแยกแยะข้อมูลของรูปภาพเป็นส่วนๆ ที่เหมาะสมต่อการกระจายไปยังหน่วยประมวลผลต่าง จากนั้นในช่วงที่สอง ทำการกระจายเพิ่มเติมไปยัง CPU และ GPU ขึ้นออยู่กับความเร็วในการประมวลผลดังรูป 2.4.3

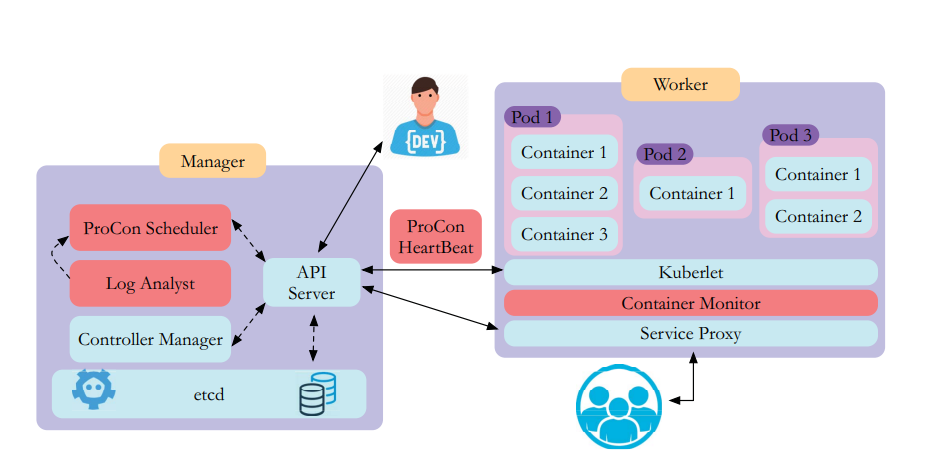


รูปที่ 2.4.3 แผนผังรูปแบบการกระจายงานในช่วงที่สองที่มีการกระจายภาระงานระหว่าง CPU และ GPU

โดยโครงงานนี้ได้นำเอกสารนี้มาศึกษาเพื่อใช้ในการพัฒนาส่วน Load balance ที่จะทำหน้าที่แบ่งงานภาระงาน Image processing ไปยัง CPU และ GPU

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

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



รูปที่ 2.4.4 แผนผัง System Architecture ของ ProCon

โดยโครงงานนี้ได้นำงานวิจัยมาศึกษาเพื่อสร้างรูปแบบการจัดการคอนเทนเนอร์ที่มีการจัดทรัพยากรให้แต่ละ Worker อย่างมีประสิทธิภาพ

**บทที่ 3**

**การออกแบบและการพัฒนา**

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

**3.1 รายละเอียดของระบบ**

1) Web Application

1) ผู้ใข้สามารถสมัครสมาชิก เข้าสู่ระบบได้

2) สามารถกู้คือ และ เปลี่ยนรหัสผ่านได้

3) สามารถจัดเก็บไฟล์รูปได้

?) สามารถสั่งงานประมวลผลภาพได้

?) สามารถทดลองดูตัวอย่างงานประมวลผลภาพได้

?) สามารถจัดเก็บ Weight Model ได้

?) สามารถซื้อขาย Weight Model ได้

?) สามารถ

2) Image Processing Application

1) สามารถประมวลผลภาพเป็นภาพ ASCII ได้

2) สามารถประมวผลภาพเป็นภาพ PixelArt ได้

3) สามารถประมวผลภาพเป็นภาพ Mosaic ได้

4) สามารถประมวผลภาพเป็นภาพขาวดำได้

5) สามารถประมวผลภาพเป็นภาพเพื่อลบพื้นหลังภาพได้

6) สามารถประมวผลภาพเป็นภาพจาก Model และ Weight ได้

3) Server

1) สามารถ LoadBalance งานประมวลผลได้

2) สามารถ LoadBalance การ Request เข้าถึง Service ได้

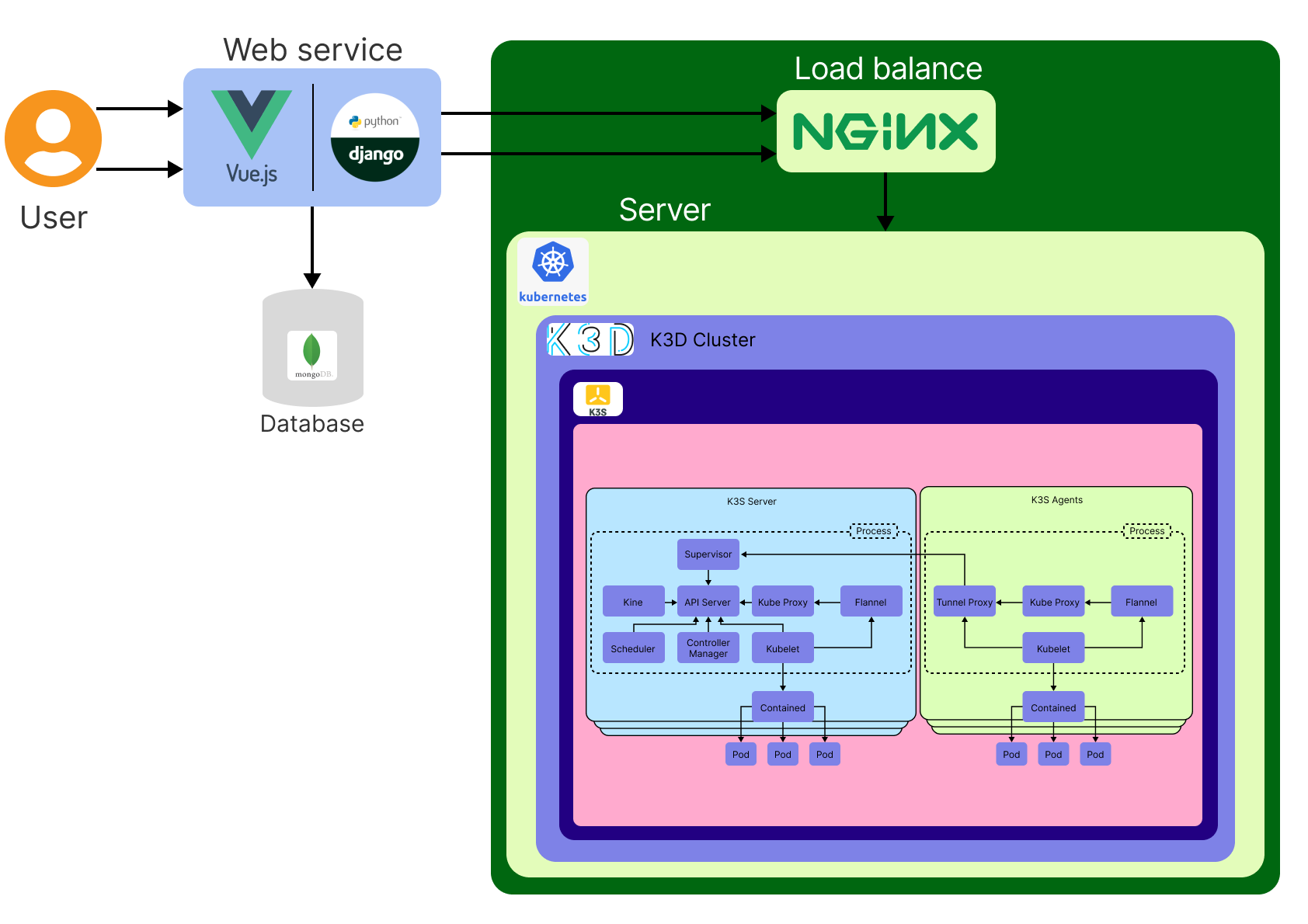
3) สามารถจัดเก็บข้อมูลไว้แยกจากตัว Server ได้

4) สามารถ High Availability Server ได้

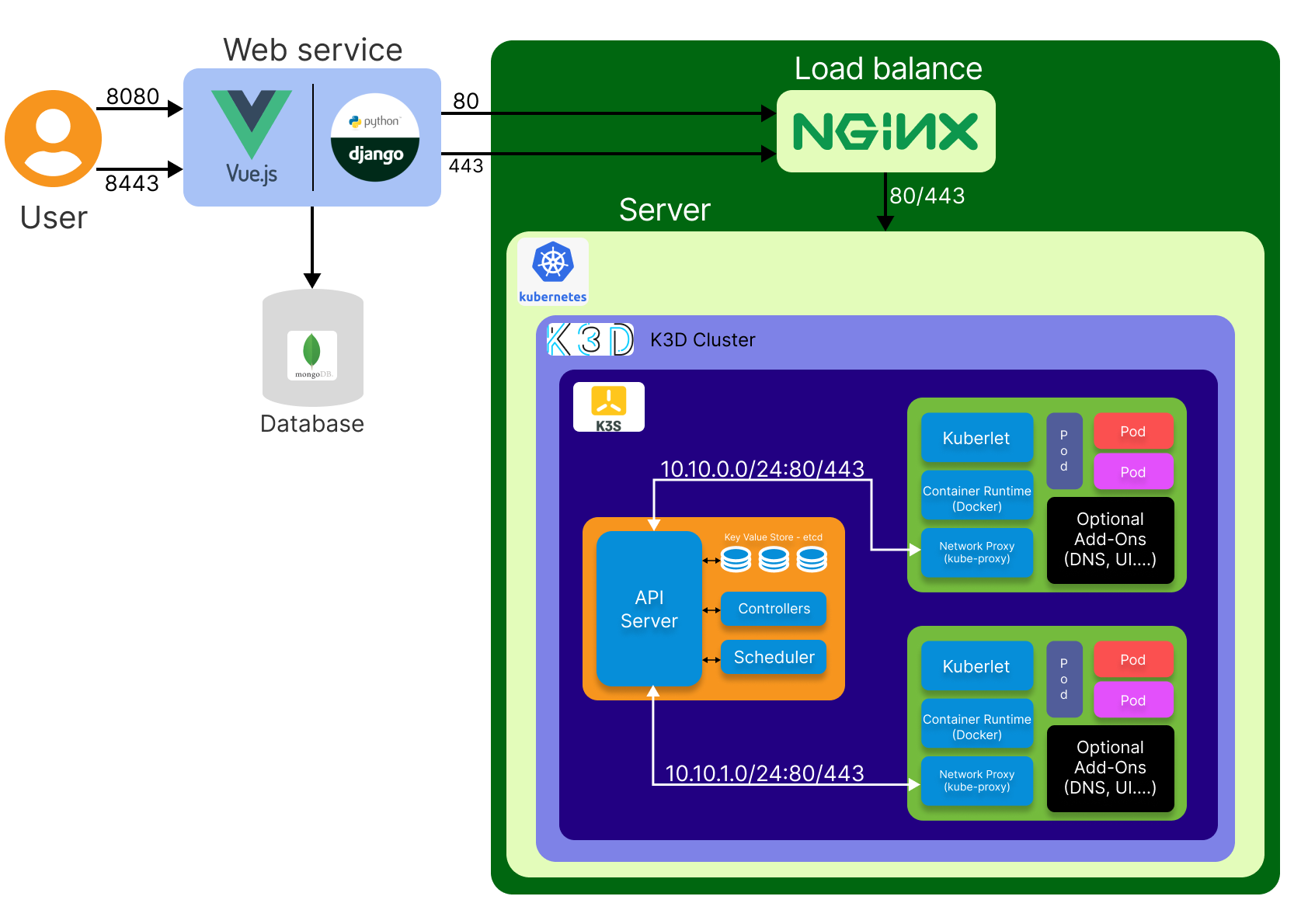
5) สามารถ Scale Service บน Server ได้

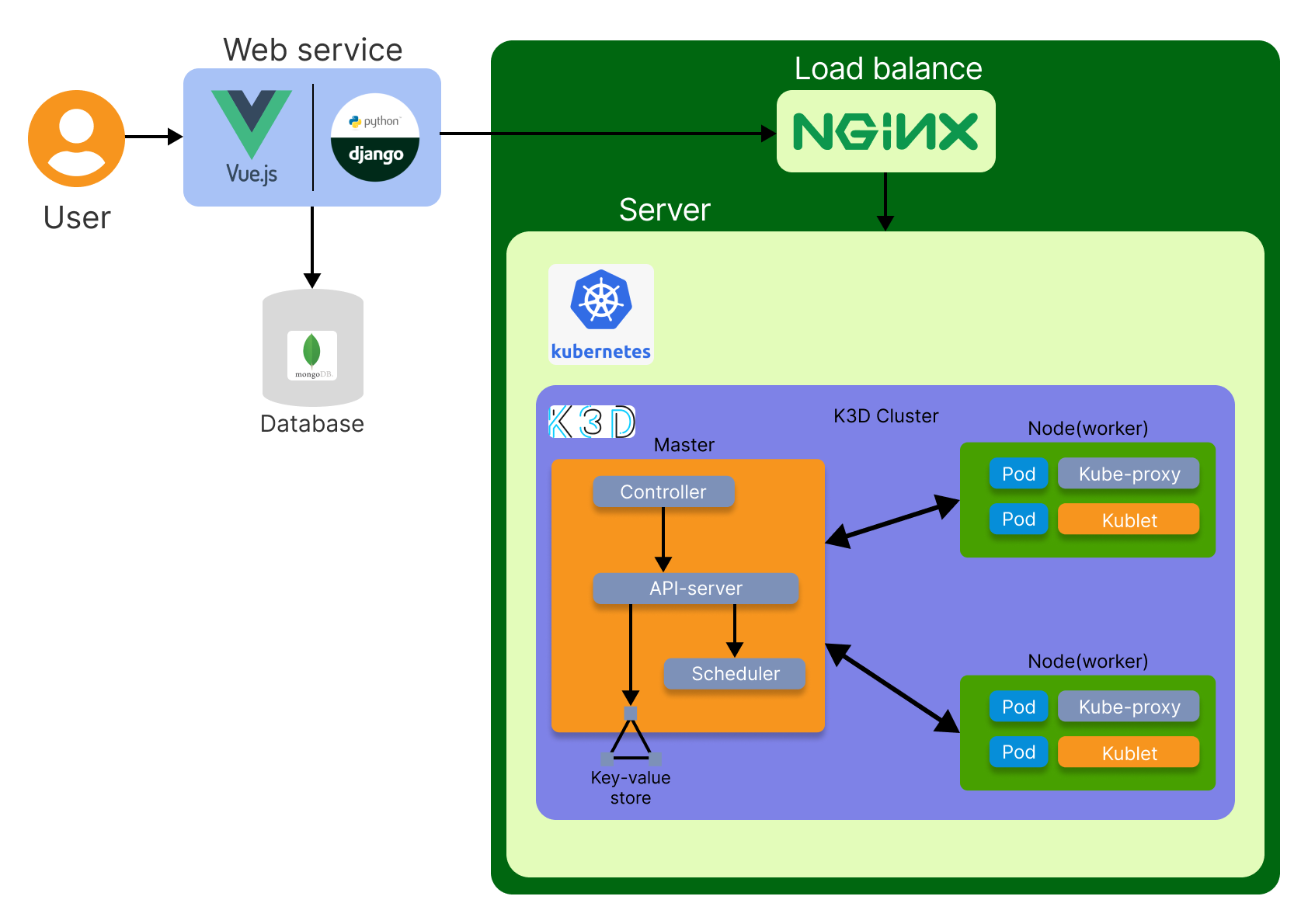
**3.2 ภาพรวมของระบบ**

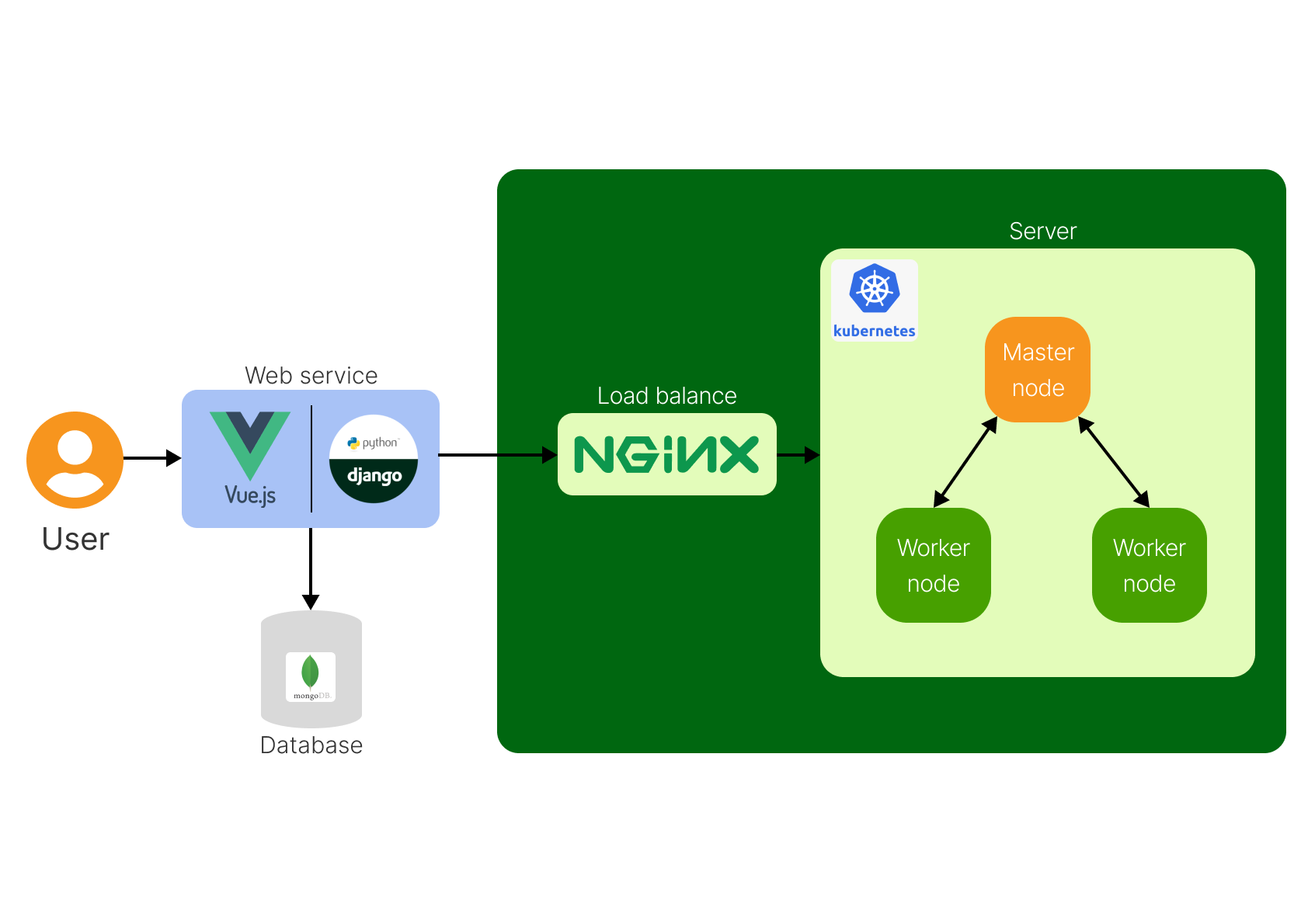
ภาพรวมในการพัฒนาระบบที่แสดงให้เห็นส่วนต่างๆที่อยู่ในแผนการพัฒนาโครงงานที่ประกอบไปด้วยส่วนประกอบหลักๆคือ Client Frontend (ส่วนติดต่อกับส่วนให้บริการ), ส่วน Docker host Backend (ส่วนบริการต่างๆ), ส่วน CI/CD (ส่วนของการบูรณาการโครงงาน ) ภาพรวมของระบบ จะแสดงดังรูปที่ 3.1



รูปที่ 3.1 ภาพรวมของระบบ







**3.3 แผนภาพผังงาน (Flowchart)**

**3.3.1 แผนภาพผังงาน (Flowchart) ในส่วนของระบบ Widget**

แผนภาพผังงานในส่วนของ Web Application บนเว็บ เมื่อผู้ใช้เริ่มยืนยันตัวตนโดยผู้ใช้จะได้รับการตอบกลับเป็นผลการยืนยันตัวตน โดยแผนภาพผังงานในส่วนของ โมดูล

ยืนยันตัวตน (Widget) บนเว็บ จะแสดงดังรูปที่ 3.2

**3.3.2 แผนภาพผังงาน (Flowchart) ในส่วนของระบบ Widget**

แผนภาพผังงานในส่วนของ Web Application บนเว็บ เมื่อผู้ใช้เริ่มยืนยันตัวตนโดยผู้ใช้จะได้รับการตอบกลับเป็นผลการยืนยันตัวตน โดยแผนภาพผังงานในส่วนของ โมดูล

ยืนยันตัวตน (Widget) บนเว็บ จะแสดงดังรูปที่ 3.3

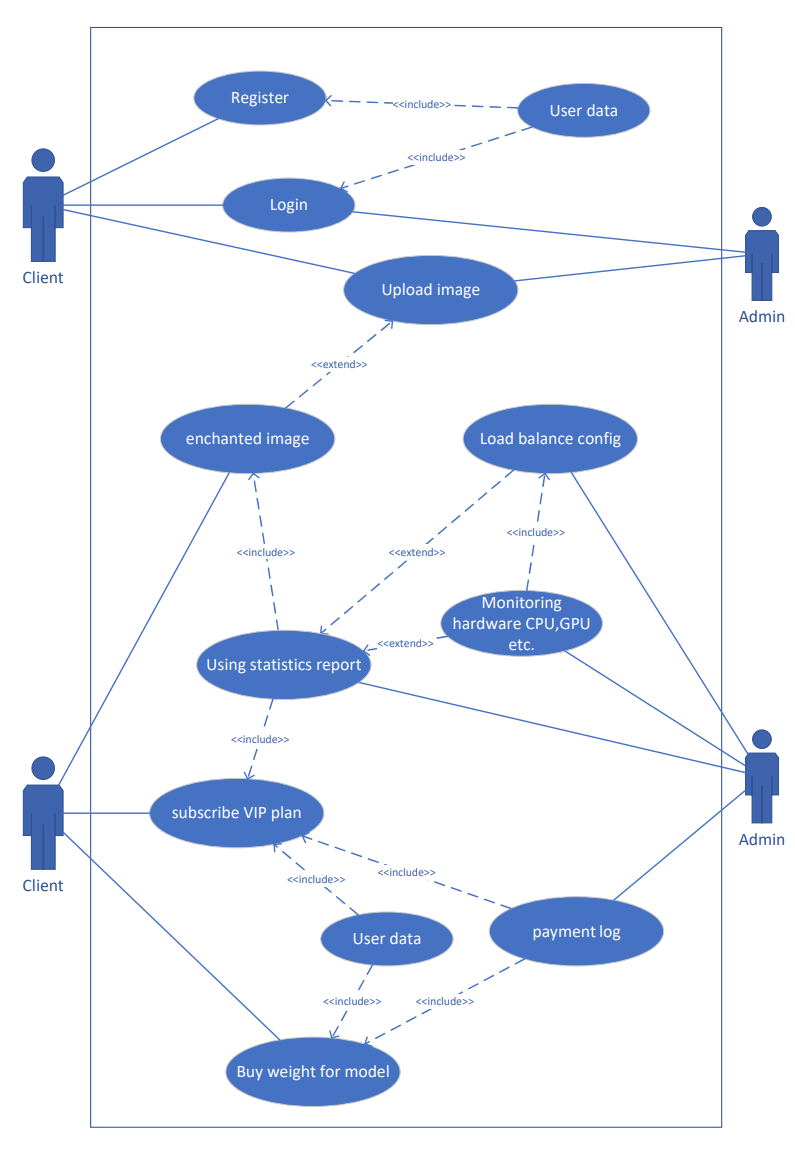
**3.3.3 แผนภาพผังงาน (Flowchart) ในส่วนของระบบ Widget**

แผนภาพผังงานในส่วนของ Web Application บนเว็บ เมื่อผู้ใช้เริ่มยืนยันตัวตนโดยผู้ใช้จะได้รับการตอบกลับเป็นผลการยืนยันตัวตน โดยแผนภาพผังงานในส่วนของ โมดูล

ยืนยันตัวตน (Widget) บนเว็บ จะแสดงดังรูปที่ 3.4

**3.4 แผนภาพยูสเคส (Use Case Diagram)**

แผนภาพยูสเคสจะแสดงให้เห็นถึงผู้ใช้ที่ถูกแบ่งเป็น 2 ประเภท คือ Client คือ ผู้ใช้ที่เข้ามาในรูปแบบลูกค้าที่ต้องการใช้บริการงานประมวลผลภาพต่างๆ และ Administer คือผู้ใช้ที่เป็นผู้ดูแลในส่วนของข้อมูลในระบบ และ ตั้งค่า ในส่วนของการประมวลผลภาพ รวมไปถึงการตรวจสอบการ payment ต่างๆ โดยแผนภาพ ยูสเคส จะแสดงดังรูปภาพที่ 3.5



รูปที่ 3.5 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 | |
| **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 | |
| **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 :** ได้รับผลอัปโหลดรูปและเก็บรูปภาพไว้ในฐานข้อมูล | |
| **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 | |
| **Trigger :** เลือกรูปภาพที่จะปรับแต่ง เลือกฟังค์ชันการปรับแต่ง 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 | |
| **Trigger :** เลือกฟังค์ชันการปรับแต่ง และกดปุ่มปรับแต่ง | |

**ตารางที่ 3.6 รายละเอียดของ Monitoring hardware**

|  |  |
| --- | --- |
| **Use Case Title :** Monitoring hardware | **Use case ID:** 6 |
| **Description :** ดูระดับการทำงานของฮาร์ดแวร์ต่างๆ เช่น CPU ,GPU เป็นต้น | |
| **Actor :** Administer | |
| **Precondition :** ยืนยันตัวตนลงชื่อเข้าใช้ | |
| **Postcondition :** ได้ผลการทำงานของฮาร์ดแวร์ | |
| **Trigger :** เลือกฟังค์ชันดูระดับการทำงาน | |

**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 | |
| **Trigger :** เลือกฟังค์ชันดูสถิติการใช้งาน | |

**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 | |
| **Trigger :** เลือกสมัครเข้าเป็นผู้ใช้ Client ระดับ VIP และชำระเงิน | |

**ตารางที่ 3.9 รายละเอียดของ Buy weight for model**

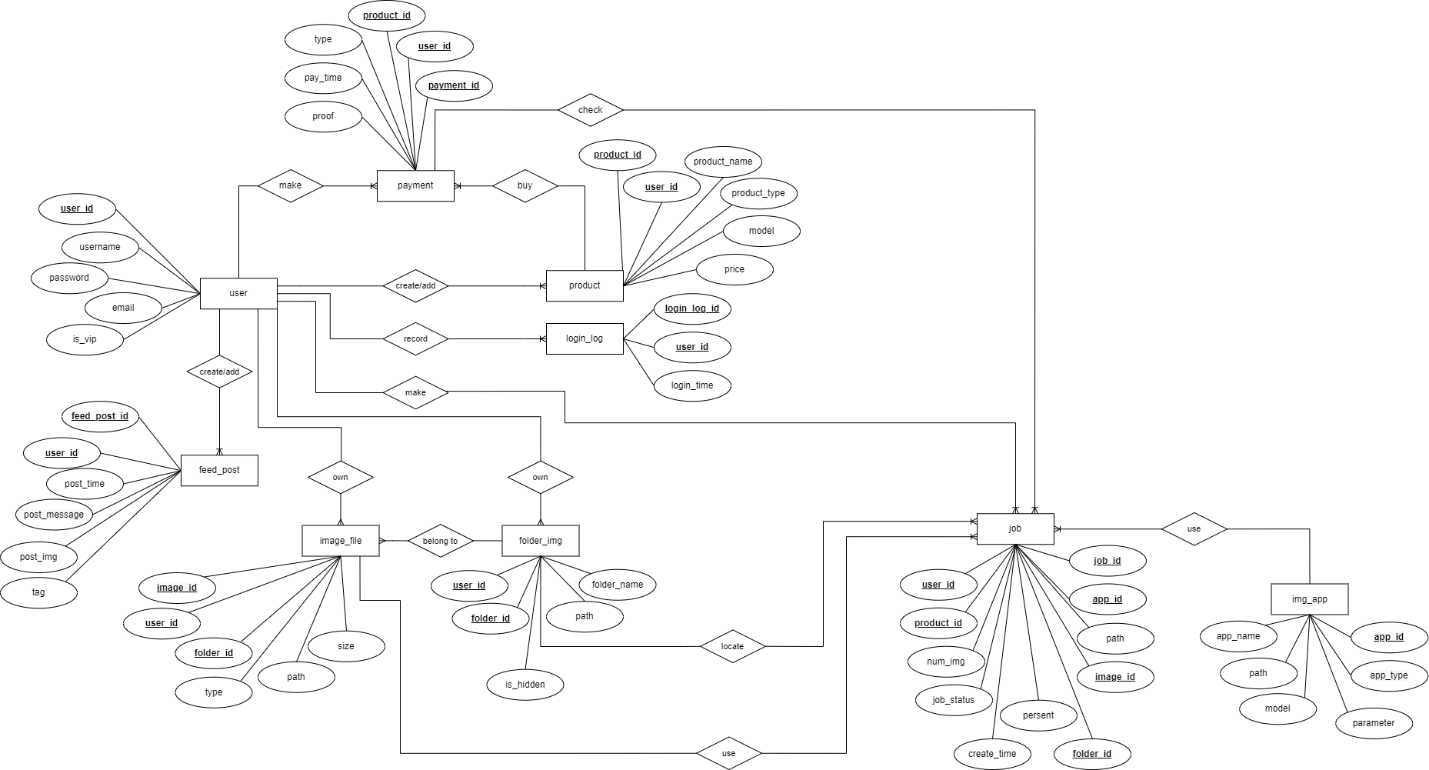
|  |  |
| --- | --- |
| **Use Case Title :** Buy weight for model | **Use case ID:** 9 |
| **Description :** ซื้อ weight สำหรับ Model เพื่อนำไปทำการประมวลผลภาพ | |
| **Actor :** Client | |
| **Precondition :** ยืนยันตัวตนลงชื่อเข้าใช้ | |
| **Postcondition :** ได้รับ weight สำหรับ Model ไว้ในฐานข้อมูล | |
| **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 :** ยืนยันตัวตนลงชื่อเข้าใช้ | |
| **Postcondition :** ได้ผลบันทึกการชำระเงิน | |
| **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 3.6

****

รูปที่ 3.6 แผนภาพแสดงความสัมพันธ์ของข้อมูล

โดยรายละเอียดของแผนภาพแสดงความสัมพันธ์ของข้อมูลแสดงดังตารางที่ 3.12 ถึง 3.20

**ตารางที่ 3.12 แสดงรายละเอียดตารางข้อมูลผู้ใช้ (user)**

|  |  |  |  |
| --- | --- | --- | --- |
| **คีย์** | **ชื่อแอดทริบิวต์** | **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** | **ชื่อแอดทริบิวต์** | **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** | **ชื่อแอดทริบิวต์** | **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** | **ชื่อแอดทริบิวต์** | **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** | **ชื่อแอดทริบิวต์** | **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 |
| - | 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** | **ชื่อแอดทริบิวต์** | **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** | **ชื่อแอดทริบิวต์** | **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** | **ชื่อแอดทริบิวต์** | **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 |

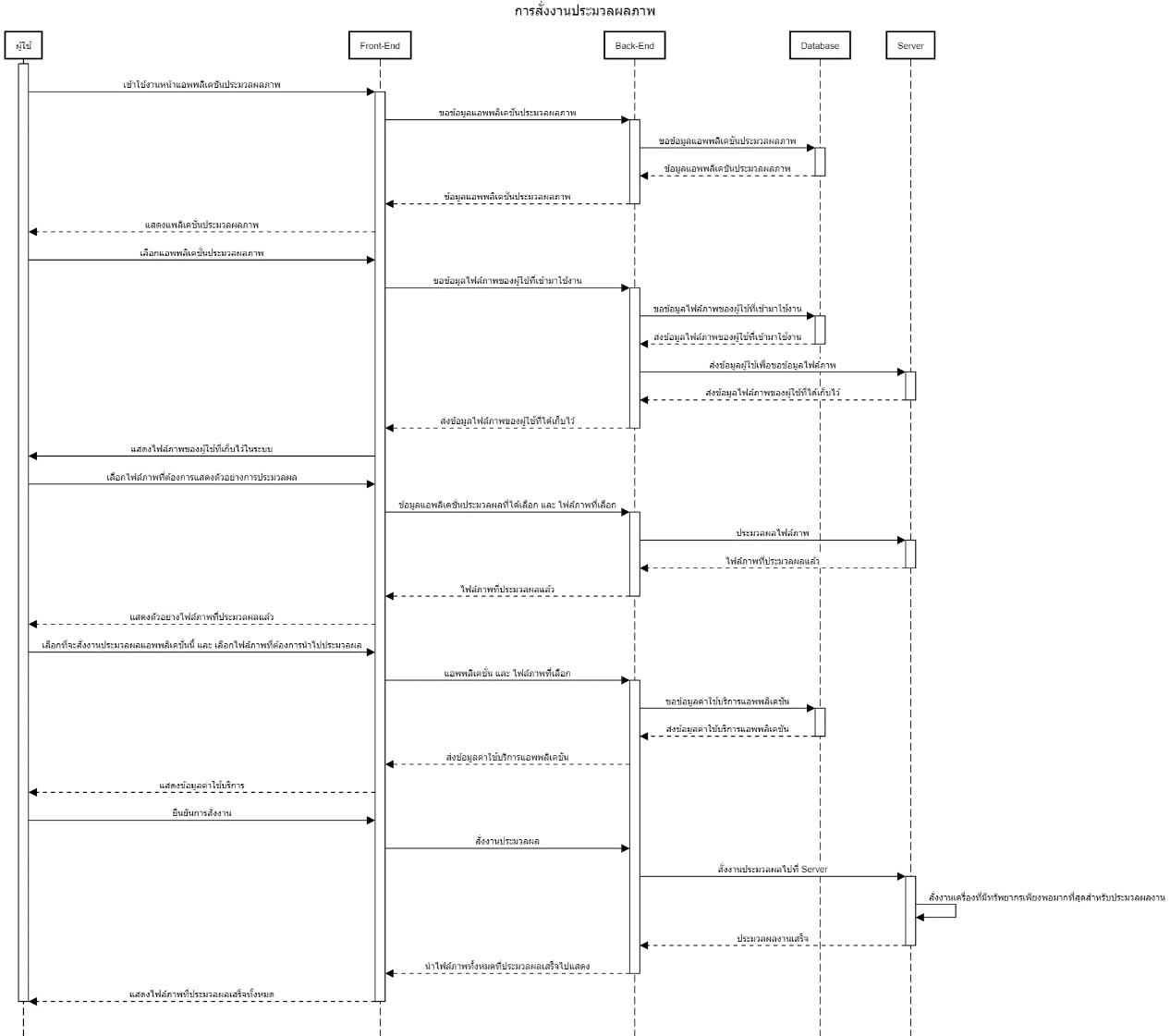
**ตารางที่ 3.20 แสดงรายละเอียดตารางข้อมูลการสั่งงานประมวลผล (job)**

|  |  |  |  |
| --- | --- | --- | --- |
| **คีย์** | **ชื่อแอดทริบิวต์** | **ชนิดตัวแปร** | **คำอธิบาย** |
| PK | job\_id | Varchar(50) | รหัสระบุ งานที่สั่ง |
| 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 แผนภาพการทํางานแบบลําดับปฏิสัมพันธ์ (Sequence Diagram)**

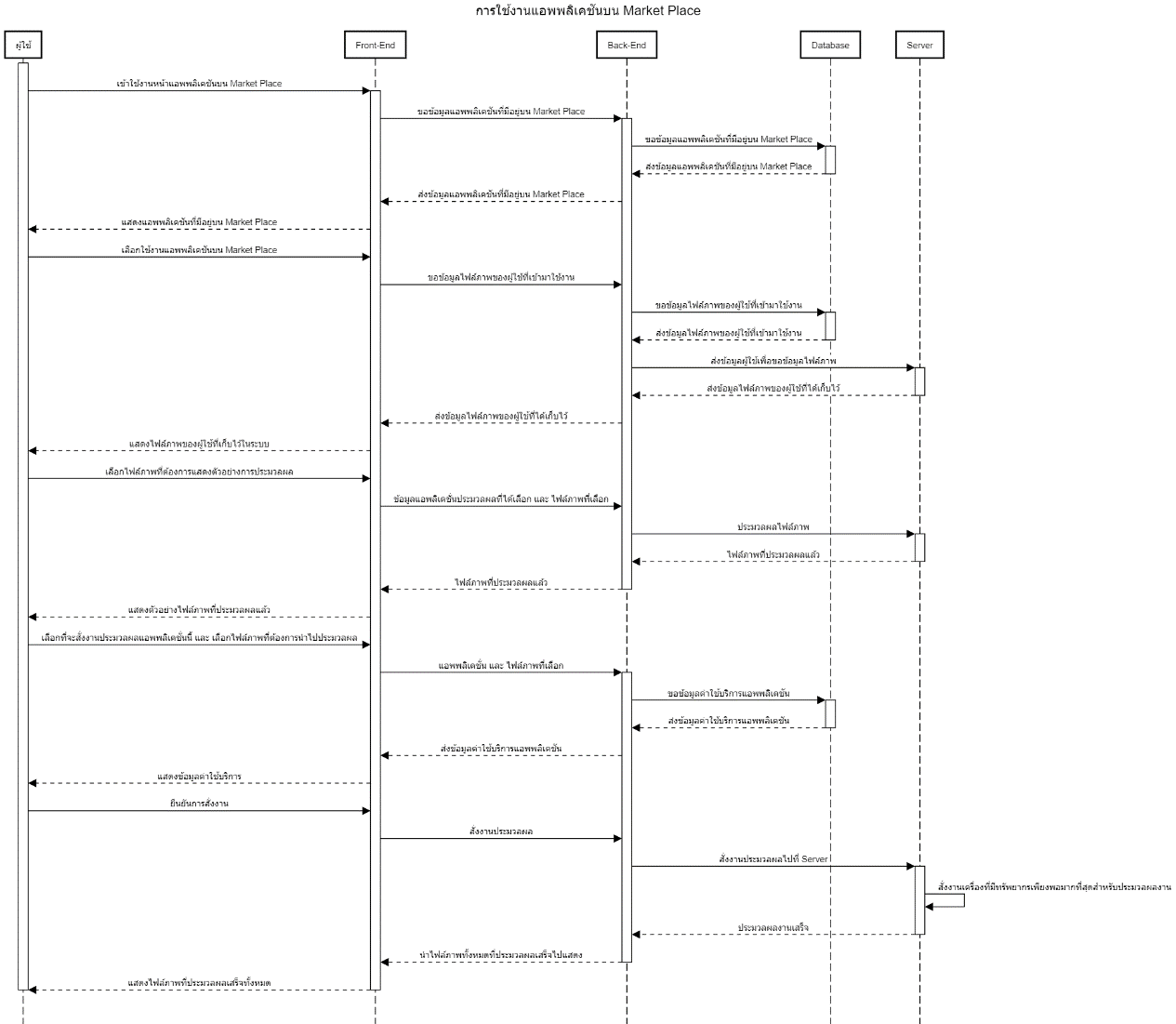
**3.6.1 แผนภาพการทํางานในส่วนของการสั่งงานประมวผลภาพ (Web Application)**

โดยรูปแบบลําดับปฏิสัมพันธ์นี้จะเริ่มตั้งแต่เหตุการณ์ที่ ผู้ใช้งานเริ่มทําการใช้งานการสั่ง งานประมวผลภาพจนถึงการประมวลผลภาพเสร็จสิ้น โดยรูปภาพแสดงการทํางานแบบลําดับปฏิสัมพันธ์โดยรวมในส่วนของการสั่งงานประมวผลภาพ จะแสดงดังรูปภาพที่ 3.7

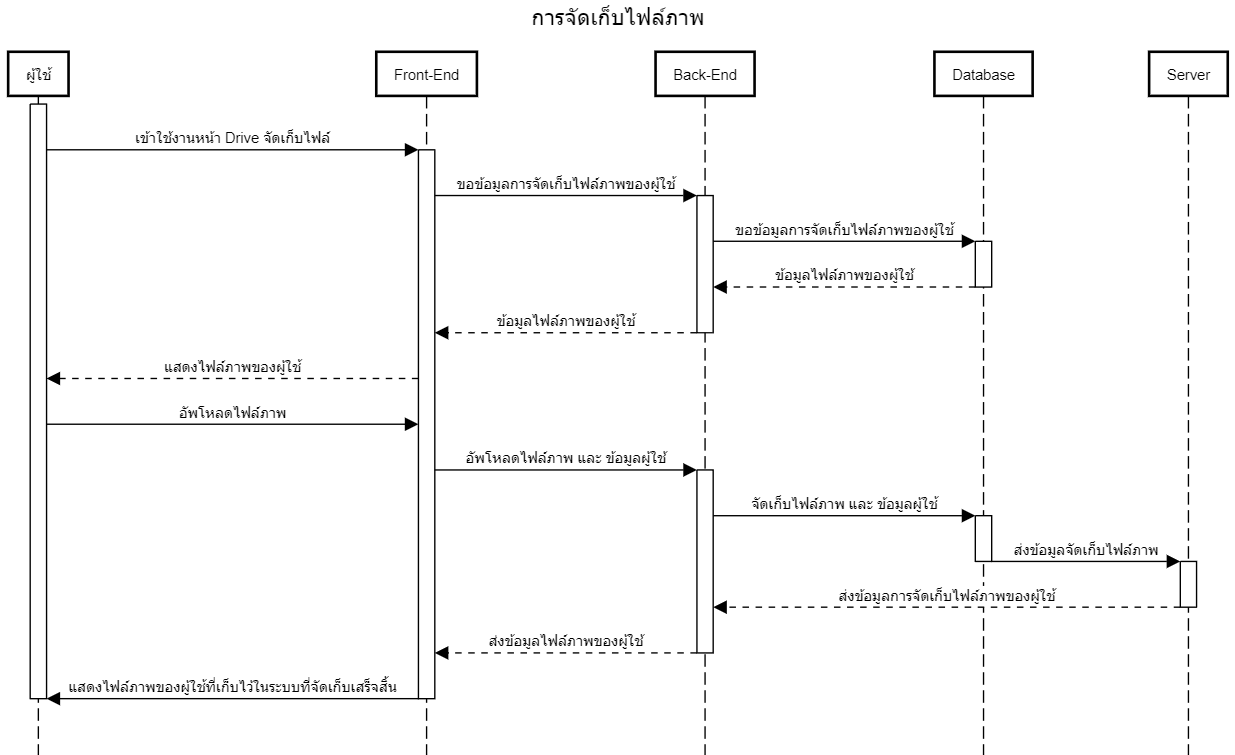
  
รูปที่ 3.7 รูปภาพแสดงการทํางานแบบลําดับปฏิสัมพันธ์โดยรวมในส่วนของการสั่งงานประมวผลภาพ

**3.6.2 แผนภาพการทํางานในส่วนของการสั่งงานประมวผลภาพบน Marketplace (Web Application)**

โดยรูปแบบลําดับปฏิสัมพันธ์นี้จะเริ่มตั้งแต่เหตุการณ์ที่ ผู้ใช้งานเริ่มทําการใช้งานการสั่ง งานประมวผลภาพบน Marketplace จนถึงการประมวลผลภาพเสร็จสิ้น โดยรูปภาพแสดงการทํางานแบบลําดับปฏิสัมพันธ์โดยรวมในส่วนของการสั่งงานประมวผลภาพ จะแสดงดังรูปภาพที่ 3.8

รูปที่ 3.8 รูปภาพแสดงการทํางานแบบลําดับปฏิสัมพันธ์โดยรวมในส่วนของการสั่งงานประมวผลภาพบน Marketplace

**3.6.3 แผนภาพการทํางานในส่วนของการจัดเก็บไฟล์ภาพไว้ใน Drive (Web Application)**

โดยรูปแบบลําดับปฏิสัมพันธ์นี้จะเริ่มตั้งแต่เหตุการณ์ที่ ผู้ใช้งานเริ่มทําการจัดเก็บไฟล์ภาพไว้ใน Drive จนถึงการการจัดเก็บไฟล์ภาพเสร็จสิ้น โดยรูปภาพแสดงการทํางานแบบลําดับปฏิสัมพันธ์โดยรวมในส่วนของการสั่งงานประมวผลภาพ จะแสดงดังรูปภาพที่ 3.9

รูปที่ 3.8 รูปภาพแสดงการทํางานแบบลําดับปฏิสัมพันธ์โดยรวมในส่วนของการจัดเก็บไฟล์ภาพไว้ใน Drive

**3.6.4 แผนภาพการทํางานในส่วนของการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train (Web Application)**

โดยรูปแบบลําดับปฏิสัมพันธ์นี้จะเริ่มตั้งแต่เหตุการณ์ที่ การใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train จนถึงการโพสต์งานเสร็จสิ้น โดยรูปภาพแสดงการทํางานแบบลําดับปฏิสัมพันธ์โดยรวมในส่วนของการสั่งงานประมวผลภาพ จะแสดงดังรูปภาพที่ 3.9

Diagram

Description automatically generated

รูปที่ 3.9 รูปภาพแสดงการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train

**3.7 การออกแบบส่วนต่อประสานกับผู้ใช้ (User Interface)**