บริการระบบทดสอบโมเดลปัญญาประดิษฐ์เพื่อการแบ่งปันโมเดลและสร้างความร่วมมือ

STREAMLINING AI INFERENCE FOR MODEL SHARING AND COLLABORATION

นวพล กรุดพันธ์

นิติพัฒน์ บุญเกตุ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาตร์บัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2565

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง บริการระบบทดสอบโมเดลปัญญาประดิษฐ์เพื่อการแบ่งปันโมเดลและสร้างความร่วมมือ
STREAMLINING AI INFERENCE FOR MODEL SHARING AND COLLABORATION

ผู้จัดทำ

1. นายนวพล กรุดพันธ์ รหัสนักศึกษา 62010474

2. นายนิติพัฒน์ บุญเกตุ รหัสนักศึกษา 62010494

อาจารย์ที่ปรึกษา

(ดร.อักฤทธิ์ สังข์เพ็ชร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บริการระบบทดสอบโมเดลปัญญาประดิษฐ์เพื่อการแบ่งปันโมเดลและสร้างความร่วมมือ

นายนวพล กรุดพันธ์ 62010474
นายนิติพัฒน์ บุญเกตุ 62010494
อาจารย์อักฤทธิ์ สังข์เพ็ชร อาจารย์ที่ปรึกษา
ปีการศึกษา 2565

บทคัดย่อ

โครงงานนี้จัดทำเพื่อพัฒนาเว็บแอพพลิเคชันที่สามารถใช้นำเสนอและทดสอบโมเดลปัญญาประดิษฐ์ผ่านบริการเว็บ ทั้งนี้เพื่อให้ผู้ที่สนใจสามารถศึกษาและทดลองใช้งานโมเดลปัญญาประดิษฐ์ได้ง่าย ส่งเสริมให้เกิดชุมชนนักพัฒนาให้สามารถทดสอบการใช้งานโมเดลปัญญาประดิษฐ์ในวงกว้างได้มากยิ่งขึ้น โดยนักพัฒนาสามารถอัพโหลดโมเดลปัญญาประดิษฐ์ผ่านการจัดการระบบคอนเทนเนอร์ เช่น Kubernetes หรือ Docker เป็นต้น ทำให้สามารถจัดการทรัพยากรและไลบรารีที่จำเป็นต่อการใช้งาน รวมถึงให้บริการในลักษณะของบริการซอฟต์แวร์เพื่อทดแทนการติดตั้งทดสอบโมเดลโดยตรงซึ่งอาจจะยุ่งยาก โดยทีมงานวางแผนจะใช้เครื่องมือเช่น COG ซึ่งเป็นซอฟต์แวร์โอเพนซอร์สเป็นพื้นฐานในการใช้รันโมเดลปัญญาประดิษฐ์ มาใช้งานร่วมกับหน่วยรับข้อมูลเข้า เช่น กล้องที่เชื่อมต่อกับคอมพิวเตอร์ รวมถึงโมเดลประเภทการแบ่งประเภท หรือการจัดแบ่งพื้นที่ภาพ เพื่อให้นักพัฒนาหรือผู้ที่สนใจสามารถทดสอบโมเดลที่เกี่ยวกับการใช้งานด้าน Computer Vision ได้สะดวกมากขึ้น ทำให้สามารถแลกเปลี่ยนเรียนรู้และศึกษาทำงานร่วมกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STREAMLINING AI INFERENCE FOR MODEL SHARING AND COLLABORATION

Mr. Nawapol Krudpun 62010474

Mr. Nitipat Boongate 62010494

Dr. Akkarit Sangpetch Advisor

Academic Year 2022

ABSTRACT

This project aims to develop a web application that enables the presentation and testing of artificial intelligence (AI) models through a web service. The objective is to make it easier for interested individuals to study and experiment with AI models, promoting the creation of a developer community capable of extensively testing the use of AI models. Developers can upload AI models through container management systems like Kubernetes or Docker, allowing for efficient resource management and necessary library dependencies. Additionally, the project provides software-as-a-service to replace the direct installation and testing of AI models, which can be complex. The team plans to utilize tools like COG, an open-source software, as a foundation for running AI models and integrate it with data input units such as cameras connected to computers. Furthermore, the project includes models for tasks like object classification or image segmentation, enabling developers or interested individuals to conveniently test AI models related to computer vision. This facilitates knowledge exchange, collaborative learning, and cooperative work among participants.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานเรื่องบริการระบบทดสอบโมเดลปัญญาประดิษฐ์เพื่อการแบ่งปันโมเดลและสร้างความร่วมมือ นี้สำเร็จลุล่วงไปได้ด้วยดี โดยได้รับความกรุณาจากอาจารย์อักฤทธิ์ สังข์เพ็ชร และพี่เสก ที่ให้คำแนะนำ และตรวจแก้ไขในการจัดทำโครงงานนี้ ทำให้ข้าพเจ้ามีความผูกพัน ซาบซึ้งและถือเป็นพระคุณอย่างยิ่งจึงขอขอบพระคุณทุกท่านที่กล่าวมาข้างต้นเป็นอย่างสูง ณ โอกาสนี้ ขอขอบคุณ นางสาววรรณรวี นาวีสัมพันธ์ และ นางสาวนฤนันท์ ตาปนานนท์ ที่คอยเป็นกำลังใจให้ข้าพเจ้าเสมอมา

นวพล กรุดพันธ์

นิติพัฒน์ บุญเกต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญรูปภาพ VII

บทที่ 1 บทนำ .....

1.1 ที่มาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ 1

1.3 ขอบเขตของโครงงาน 2

1.4 ประโยชน์ที่คาดว่าจะได้รับ 2

1.5 แผนการดำเนินงาน 2

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง .....

2.1 เครื่องมือที่เกี่ยวข้อง 3

2.2 งานวิจัยที่เกี่ยวข้อง 7

บทที่ 3 การออกแบบ และการพัฒนา .....

3.1 วิเคราะห์ความต้องการ 9

3.2 USE CASE DIAGRAM 9

3.3 ภาพรวมการทำงานของระบบ 10

3.4 SOFTWARE ARCHITECTURE 13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

3.5 ER DIAGRAM 14
3.6 CLASS DIAGRAM 22
บทที่ 4 ผลการดำเนินงาน .....
4.1 หน้าหลัก (HOMEPAGE) 34
4.2 หน้าเข้าสู่ระบบ (LOGIN) 35
4.3 หน้าสำรวจโมเดล (EXPLORE) 38
4.4 หน้าการใช้งานโมเดล (MODEL PAGE) 38
4.5 หน้าการอัพโหลดโมเดล (UPLOAD MODEL PAGE) 43
4.6 หน้าการอัพเดตโมเดล (UPDATE MODEL PAGE) 43
4.7 หน้าการจัดการของผู้ใช้ (DASHBOARD) 45
4.8 หน้าคู่มือการใช้งาน (DOCUMENTATION PAGE) 46
บทที่ 5 สรุป .....
5.1 ผลที่ได้จากโครงงาน 46
5.2 ปัญหาและอุปสรรค 47
5.3 เสนอแนะการพัฒนาต่อ 47
5.4 สรุป 47
บรรณานุกรม 50
ภาคผนวก 52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง หน้า

1 **แผนการดำเนินงาน** 2

3.1 User 15

3.2 Model Data 16

3.3 Model Image 17

3.4 Model Pod 18

3.5 Model Input 19

3.6 Model Input Detail 20

3.8 Model Output Data 21

3.9 Model Input Data 21

3.10 Data Input 21

3.11 Model Report 22

3.12 Attributes ของ User 24

3.13 Method ของ User 24

3.14 Attributes ของ Model 25

3.15 Method ของ Model 26

3.16 Attributes ของ Model Container Image 27

3.17 Method ของ Model Container Image 27

3.18 Attributes ของ Model Pod 28

3.19 Method ของ Model Pod 28

3.20 Attributes ของ Model Input 29

3.21 Method ของ Model Input 29

3.22 Attributes ของ Model Container Image 30

3.23 Method ของ Model Container Image 30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปภาพ หน้า

2.1 การทำงานของ OAuth ระหว่าง Client กับ Github1 4
2.2 ตัวอย่างการเขียนไฟล์ cog.yaml 6
2.3 โครงสร้างระบบในการเรียกใช้ Machine Learning Model ผ่าน Web-Interface[2] 8
2.4 องค์ประกอบหลักของการทำโครงสร้างของระบบ3 9
3.1 Use Case Diagram....10
3.2 การทำงานของระบบ Upload 11
ร3.3 การทำงานของระบบ Inference 12
ร3.4 Software Architecture 13
3.5 ER Diagram....14
3.6 User ER Diagram 15
3.7 Model ER Diagram 16
3.8 Model Image ER Diagram 17
3.9 Model Pod ER Diagram 18
3.10 Model Input ER Diagram....19
3.11 Model Input Detail และ Optional ER Diagram 20
3.12 Model Output Data และ Model Input Data และ Data Input ER Diagram 21
3.13 Model Report ER Diagram 22
3.14 Class Diagram 23
3.15 User Class Diagram 24
3.16 Model Class Diagram....25
3.17 Model Container Image Class Diagram....27
3.18 Model Pod Class Diagram 28
3.19 Model Input Class Diagram 29
3.20 Model Container Image Class Diagram....30
3.21 ความสัมพันธ์ของ Class User....31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

รูปภาพ หน้า

3.22 ความสัมพันธ์ของ Class Model 32
3.23 ความสัมพันธ์ของ Class Model Container Image 33
4.1 หน้าหลัก 35
4.2 หน้าเข้าสู่ระบบเว็บแอพพลิเคชั่น 36
4.3 หน้าเข้าสู่ระบบของ Github 37
4.4 หน้าการขอสิทธิอ่านข้อมูลของ Github 37
4.5 หน้าสำรวจโมเดล 38
4.6 หน้าการใช้งานโมเดล 39
4.7 หน้าการใช้งาน Streaming Mode 41
4.8 หน้าการใช้งานประวัติการใช้งานของโมเดล 42
4.9 หน้าการอัพโหลดโมเดล 423
4.10 หน้าการอัพเดตโมเดล 424
4.11 หน้าการจัดการของผู้ใช้ 445
4.12 หน้าคู่มือการใช้งาน 456

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ปัจจุบันโลกของเรามีผู้คนที่มีความสนใจศึกษาเกี่ยวกับทางด้าน Machine Learning Model เป็นจำนวนมาก ซึ่งการจะศึกษาทางด้าน Machine Learning จำเป็นต้องติดตั้ง Dependencies เป็นจำนวนมาก และหลากหลายเวอร์ชันที่จำเป็นต้องใช้บน Machine Learning Model และถ้าทำการทดลองหลาย ๆ Models ในแต่ละ Model ก็มีการใช้ Dependencies ที่แตกต่างกัน หรือถ้าเป็นตัวเดียวกันก็จะแตกต่างเวอร์ชัน ทำให้มีความยุ่งยากในการแบ่งปัน Machine Learning Model อีกทั้งผู้ใช้จำเป็นต้องคำนึงถึงทรัพยากรว่าเพียงพอต่อการใช้งาน Model นั้น ๆ หรือไม่ เช่น พื้นที่ว่างในที่เก็บข้อมูล และอุปกรณ์ที่ใช้สามารถรองรับการทำงานหรือไม่ เช่น การ์ดจอรองรับการทำงานของ Machine Learning Model ไหม

ในการที่จะนำ Machine Learning Model ขึ้นใช้งานบน Server หรือ Virtual Machine ทั่วไปนั้น จำเป็นต้องติดตั้ง Dependencies หรือ Tools ที่หลากหลายในแต่ละ Machine Learning Model และมีปัญหาเมื่อต้องทำการ Scaling ระบบให้รองรับไปเยอะขึ้น

นอกจากนั้นพื้นที่การแลกเปลี่ยนข้อมูลเกี่ยวกับ Machine Learning Model ในไทยยังพบเห็นได้น้อย และไม่สามารถทดลอง และแบ่งปันได้อย่างสะดวก

ทางเราได้เห็นถึงความสำคัญในการสร้างโครงงานนี้ขึ้นมาเพื่อทำให้การเข้ามาศึกษา ค้นคว้า แบ่งปัน Machine Learning Model และได้ทดลองหลาย ๆ Machine Learning Model ให้มีความง่ายดาย และมีประสิทธิภาพมากขึ้น โดยเฉพาะกลุ่มคนที่สนใจทางด้านนี้ หรือนักวิจัยจะสามารถทดลอง และแบ่งปัน Machine Learning Model ให้ผู้อื่นได้อย่างมีความสะดวกสบาย

1.2 วัตถุประสงค์

1) เพื่อพัฒนาเว็บแอพพลิเคชันที่สามารถ Inference Machine Learning Model
2) เพื่อพัฒนาเว็บแอพพลิเคชันที่สามารถ Upload Machine Learning Model
3) เพื่อที่จะสามารแบ่งปัน Machine Learning Model ได้
4) เพื่อที่จะลดเวลา และขั้นตอนในการ Deploy Machine Learning Model เพื่อใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงงาน

1) Upload Machine Learning Model ผ่านเว็บไซต์
2) Inference Machine Learning Model ผ่านเว็บไซต์
3) Machine learning Model ที่สามารถใช้ได้จะรับ Input เป็นรูปภาพเท่านั้น
4) Machine learning Model ที่สามารถใช้ได้จะมี Output เป็นรูปภาพเท่านั้น
5) Machine learning Model ที่สามารถ Streamingได้จะต้องใช้เวลาการประมวลผลไม่เกิน 5 วินาที
6) ความละเอียดของ Input ภาพจากการ Streaming จะลดขนาดเหลือ 360p

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) สามารถสร้างเว็บแอพพลิเคชันที่สามารถใช้งาน Machine Learning Model ได้อย่างง่ายดาย
2) สามารถสร้างเว็บแอพพลิเคชันที่สามารถ Upload Machine Learning Model และแบ่งปัน
3) ลดเวลา และขั้นตอนในการ Deploy Machine Learning Model เพื่อใช้งาน

1.5 แผนการดำเนินงาน

ตาราง 1 แผนการดำเนินงาน

หัวข้อ	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
กำหนดขอบเขต								
ศึกษาและออกแบบ								
พัฒนาระบบ								
ทดสอบและปรับปรุง								
จัดทำเล่มรายงานและนำเสนอ								

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 เครื่องมือที่เกี่ยวข้อง

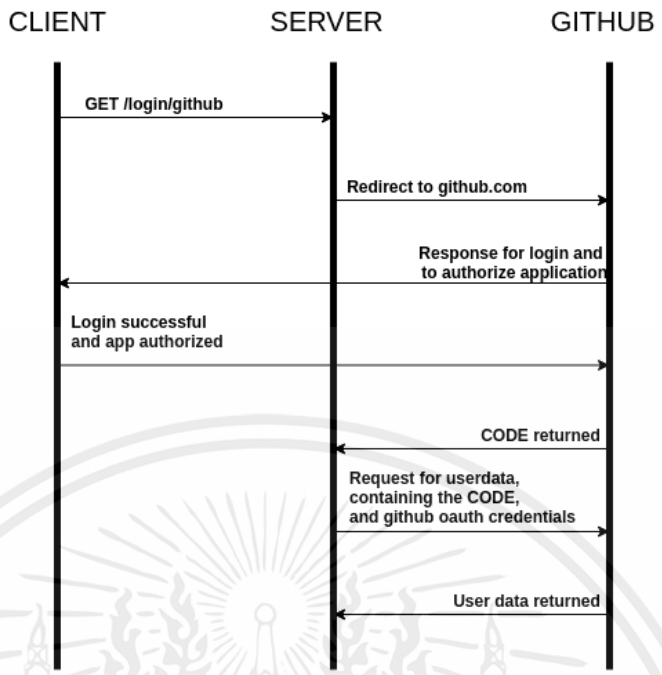
2.1.1 การยืนยันตัวตนด้วย Open Authentication (OAuth)

OAuth คือโปรโตคอลที่ใช้สำหรับการยืนยันตัวตน และการเข้าถึงข้อมูลระหว่าง Application โดยการใช้ OAuth จะลดความเสี่ยงในการเปิดเผยข้อมูลของผู้ใช้งาน และเพิ่มความปลอดภัยในการเข้าถึงข้อมูลของผู้ใช้งาน

เมื่อผู้ใช้งานเข้าสู่ระบบด้วย โปรโตคอล OAuth เช่น Google หรือ Facebook ระบบที่ใช้ OAuth จะขออนุญาตให้เข้าถึงข้อมูลบัญชีผู้ใช้ โดยไม่เปิดเผยรหัสผ่านของผู้ใช้ โดยผู้ใช้จะได้รับ Token เพื่อให้ระบบสามารถนำ Token ไปใช้ในการเข้าถึงข้อมูลที่ผู้ใช้ได้ยินยอมให้เข้าถึงได้ [1]

โดยระบบในโครงงานนี้จะใช้ Github OAuth ที่ใช้สำหรับการยืนยันตัวตน และการเข้าถึงข้อมูล Repository ของผู้ใช้งาน โดยจะได้รับ Accesstoken จากทาง Github เพื่อยืนยันสิทธิ์ในการร้องขอข้อมูล [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ **2.1** การทำงานของ **OAuth** ระหว่าง **Client** กับ **Github1**

หมายเหตุ. จาก Github OAuth with Go | Dakshraj Sharma (sharmarajdaksh.github.io)

2.1.2 การทำงานของ Frontend Framework React

React เป็น Framework สำหรับการพัฒนา Web Application แบบ Single Page Application มีหลักการทำงานที่เกี่ยวกับ Component Based Architecture ซึ่งสามารถสร้าง UI ได้ยืดหยุ่น และเป็นโมดูล ๆ ทำให้สามารถนำ Components ไปใช้งานใน Modules ที่มี Logic หรือการทำงานคล้ายๆ กันได้ และยังสามารถจัดการ State และ Props จาก Component หนึ่งไปอีก Component ได้ ทำให้สามารถพัฒนา Web Application ขึ้นมาได้รวดเร็ว [3]

ด้วยความที่เป็น Open-Source ทำให้มีเครื่องมือที่หลากหลาย สามารถใช้งานร่วมกับ Tool Redux ที่เป็น State Management มาจัดการกับ User ในแต่ละ Component [4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 การทำงานของ Typescript

Typescript เป็นภาษาโปรแกรมมิ่งที่ถูกพัฒนาโดย Microsoft โดยเป็นการเพิ่มความสามารถจาก Javascript เกี่ยวกับการตรวจสอบประเภทข้อมูล เพื่อช่วยลดข้อมูลพลากในการพัฒนาโปรแกรมขึ้นมาได้ ซึ่งเหมาะสมกับการพัฒนาโครงงานที่มีขนาดใหญ่ มีการเชื่อมต่อหลายส่วน [5]

2.1.4 การใช้งาน Golang

Golang คือ ภาษา Programming แบบ Open-Source ถูกพัฒนาขึ้นโดยบริษัท Google ในปี 2007 และเป็นภาษาที่ได้รับความนิยมมากขึ้นเรื่อยๆ Go นั้นจะมีจุดเด่นในเรื่องของ Performance ที่ สามารถทำงานได้อย่างรวดเร็วเทียบกับภาษาอื่น ๆ อีกทั้งยังมีจุดเด่นในเรื่องของ Simplicity ที่เน้นความง่าย ในการเขียนและการอ่าน และยังสามารถทำ Concurrent Programming ได้ง่าย เพราะภาษา Golang ถูก ออกแบบมาเพื่อทำให้ Application ที่ต้องใช้ multi-Threading หรือ Distributed Systems เป็นเรื่องที่ง่ายขึ้น

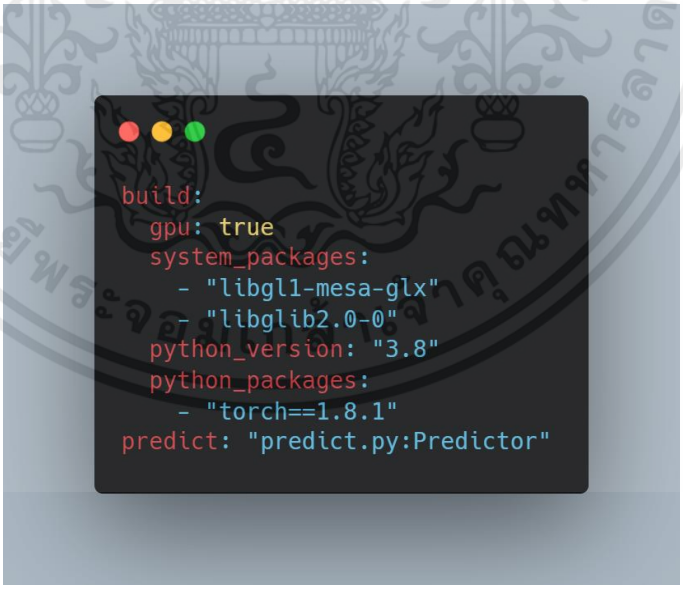
Golang เป็นภาษาที่นิยมในการพัฒนา Application ที่เป็น Client เพื่อไปติดต่อ Tools ต่างๆ รวมถึงการทำงานร่วมกับระบบปฏิบัติการ ซึ่งระบบของโครงงานนี้ต้องมีการติดต่อกับทั้ง Local File, Operation System, Docker Runtime และ Kubernetes [7][8]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 Replicate & Containers for machine learning (COG)

Replicate เป็นเว็บไซต์ที่เปิดโอกาสให้ผู้ที่พัฒนา AI หรือ Machine Learning Model สามารถแบ่งปัน Machine Learning Model ของตนเองบนเว็บไซต์ได้ และสามารถให้ผู้อื่นสามารถทดลองใช้ Machine Learning Model ของตัวเอง โดยสามารถทดลองการ Inference Machine Learning Model ผ่าน Containers for machine learning tool ที่เป็น Tool ของผู้พัฒนาเว็บไซต์, Docker, API และหน้าเว็บไซต์ได้ [9]

เบื้องหลังการทำงานของ Replicate คือใช้ Containers for machine learning (COG) ซึ่งเป็น Open Source tool มาช่วยในการจัดการ Machine Learning Models ของผู้พัฒนาที่ต้องการจะแบ่งปัน Machine Learning Models โดย COG นั้นเป็น Tool ที่จะช่วยจัดการ Config Environment และ Library ที่จำเป็นของ Machine Learning Model และสำหรับ Machine Learning Models ที่ต้องใช้ CUDA ในการทำงาน COG จะจัดการเรื่อง Version CUDA ให้อัตโนมัติ โดยการจะทำงานนั้นต้องเขียนไฟล์ YAML สำหรับ COG ซึ่งรูปแบบจะคล้ายกับ Dockerfile แต่มีรูปแบบที่สามารถอ่านได้ง่ายกว่า หลังจากนั้นจะทำการสร้าง Docker Image ที่มีการสร้าง Endpoint สำหรับการเรียกใช้งาน Inference Machine Learning Model ไว้ [10]



รูปที่ **2.2** ตัวอย่างการเขียนไฟล์ **cog.yaml**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 Docker

Docker เป็น Software Platform ที่ช่วยในการสร้าง และจัดการ Application ได้อย่างสะดวก และมีประสิทธิภาพ โดยการทำงานนั้น Docker จะแยก Application ออกจากระบบการทำงานของเครื่อง [11]

หลักการทำงานของ Docker คือการสร้างไฟล์ Docker Image ขึ้นมาเพื่อเป็นต้นแบบของ Container โดยจะเป็นการกำหนด Dependencies และ Environment ของ Application ขึ้นมา และเมื่อต้องการจะทำการนำ Application ขึ้นระบบ Production เพียงแค่สร้าง Container เพื่อจำลอง Environment ของ Docker Image ทำให้ช่วยลดเวลาในการตั้งค่าบนเครื่อง Server หรือ Virtual Machine [12]

2.1.7 Kubernetes

Kubernetes เป็น Open-Source Tool ในการจัดการ Container ที่สามารถ Automates Deployment, Scaling และ จัดการ Containerized Applications ได้

ข้อดีของการ Deploy และScaling แบบ Containerized จะสามารถทำได้รวดเร็ว และยืดหยุ่นกว่าการ Deploy และScaling Application บน Virtual Machine [13]ยกตัวอย่างกรณีของเกม Pokémon GO ที่ใช้ Kubernetes ในการ Deploy ในวันเปิดใช้งานวันแรก มี Users เข้ามาใช้งานอย่างต่อเนื่องมากว่า 1 ล้าน Users ซึ่งมากกว่าที่คาดการไว้ถึง 50 เท่า เมื่อระบบ Kubernetes สังเกตได้ถึง Traffic การใช้งานที่เพิ่มขึ้นอย่างรวดเร็ว Kubernetes ได้สร้าง Service ของ Pokémon GO อัตโนมัติทำให้สามารถรองรับ Users ได้ทันที [14]

องค์ประกอบของ Kubernetes จะประกอบไปด้วย

1) Kubernetes Cluster เปรียบเสมือน Server ขนาดใหญ่หลาย ๆเครื่องรวมเป็น 1 Cluster

2) Pod เป็นสิ่งที่รวม Container ไว้

3) Node เปรียบเสมือนระบบนึงที่เอาไว้ให้ Pod ทำงาน

4) Worker Node เป็น Node ที่เอาไว้ Run Pod ที่มี Application

5) Master Node เป็น Node ที่เอาไว้จัดการ Worker Node [13]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 งานวิจัยที่เกี่ยวข้อง

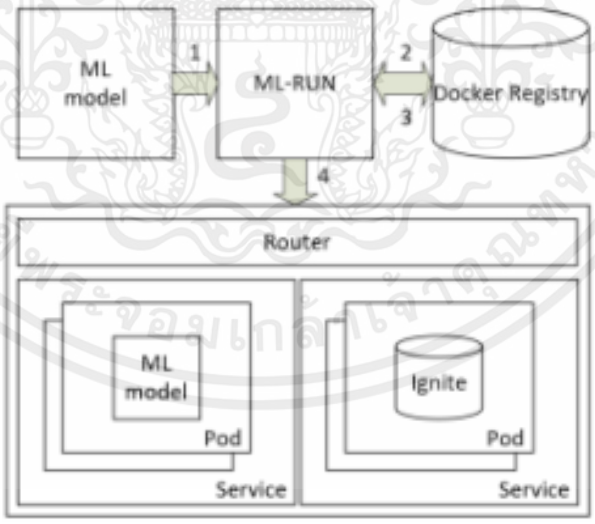
2.2.1 Cloud Computing Architecture for High-volume ML-based Solutions[15]

บทความวิชาการนี้จะกล่าวถึงวิธีการนำ Machine Learning Model ไปประยุกต์ใช้ในงานจริง ซึ่งจะมี 2 วิธีหลักคือ

1) สร้าง Machine Learning Model จากการทำ Data analysis ต่าง ๆ ด้วยภาษา R หรือ Python จากนั้นนำไปพัฒนาต่อในภาษาโปรแกรมมิ่งต่าง ๆ ที่ใช้ในทางธุรกิจ

2) ใช้งาน Machine Learning Model ผ่าน Web-interface และ Integrate เข้ากันได้เลย

บทความวิชาการนี้จะนำวิธีที่ 2 มาทดลอง โดยจะเป็นการสร้างโครงสร้างสำหรับการ Deploy Machine Learning model ขึ้นสู่ Cloud จะนำ Containerization technologies มาใช้งาน โดยในแต่ละ Machine learning model จะถูกแยกเป็นแต่ละ Container เพื่อให้สามารถจัดการ Dependencies และconfiguration file ในแต่ละ machine learning model หลังจากนั้นนำ Container Machine Learning Model ไปพัฒนาต่อให้สามารถเรียกการ Inference ผ่าน REST Service หลังจากนั้นก็จะสามารถ Deploy ขึ้นและเรียกใช้ Model ได้



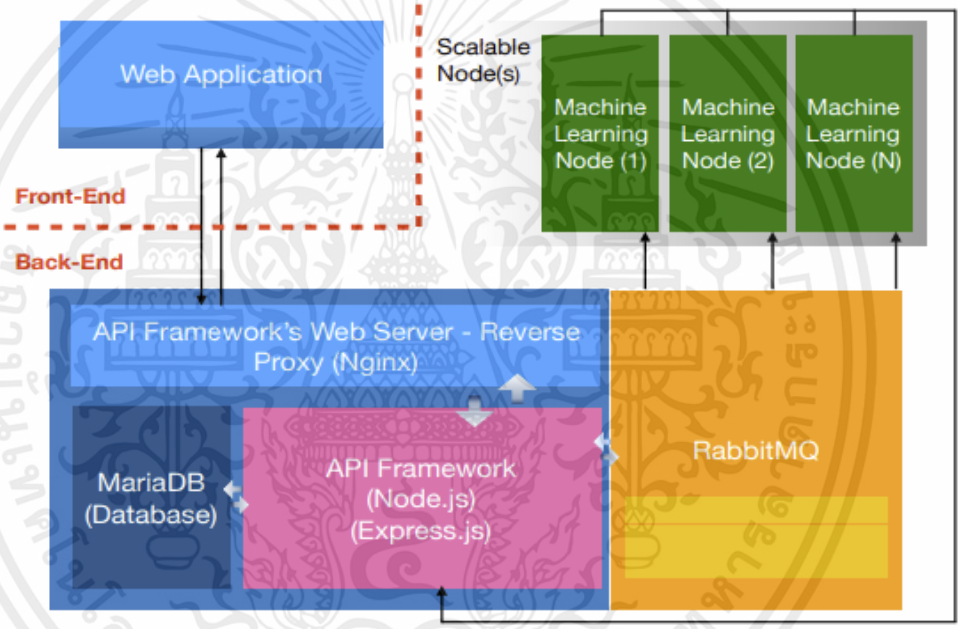
รูปที่ **2.3** โครงสร้างระบบในการเรียกใช้ **Machine Learning Model** ผ่าน **Web-Interface**[2]

หมายเหตุ. จาก M. M. Rovnyagin, K. V. Timofeev, A. A. Elenkin and V. A. Shipugin, "Cloud Computing Architecture for High-volume ML-based Solutions," 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2019, pp. 315-318, doi: 10.1109/EIConRus.2019.8656765.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 A Scalable Service Architecture with Request Queuing for Resource-Intensive[16]

งานวิจัยนี้เป็นการขยายการทำงานของระบบ Web API ในแต่ละ Request ด้วยการใช้ Queuing system ของ open-source message broker เช่น RabbitMQ เป็น Service Architecture อีกทั้งยังใช้งาน Docker และKubernetes กับระบบเพื่อใช้งานในรูปแบบ Container เพื่อเป็นระบบแบบ automate ที่มีการ deployment, scaling, และ จัดการ ML container ตอบสนองต่อ High-scale และ High-availability ซึ่งจะถูกนำมาประยุกต์ใช้งานในโครงงานของส่วนการจัดการ Queuing System ด้วย Service Architecture ของการ Request ใช้งานModel และการจัดการระบบแบบ automate



รูปที่ **2.4** องค์ประกอบหลักของการทำโครงสร้างของระบบ**3**

หมายเหตุ. จาก K. Chaowvasin, P. Sutanchaiyanonta, N. Kanungsukkasem and T. Leelanupab, "A Scalable Service Architecture with Request Queuing for Resource-Intensive Tasks," 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2020, pp. 67-70, doi: 10.1109/ECTICON49241.2020.9158114

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบ และการพัฒนา

3.1 วิเคราะห์ความต้องการ

ระบบการแบ่งปัน และทดสอบ Machine Learning Model นั้นจะต้องรองรับ Machine Learning Model ของ User และสามารถทดสอบผ่านบริการของเราได้

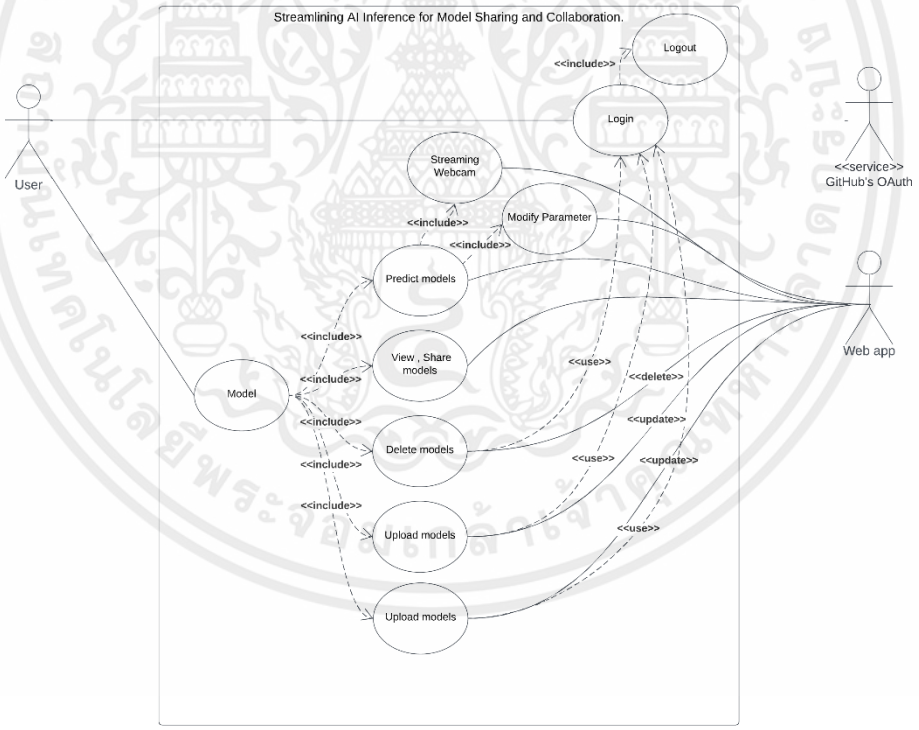
1) รองรับการ Upload Machine Learning Model ผ่าน Git URL

2) รองรับการ Inference Machine Learning Model ผ่านหน้าเว็บไซต์ และ Web API

3) จำกัดความละเอียดการ Streaming Prediction ไว้ที่ 360p

4) รองรับเฉพาะ Machine Learning Model ที่เป็น Image Input เท่านั้น

3.2 Use Case Diagram



รูปที่ 3.1 Use Case Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการแบ่งปัน และทดสอบ Machine Learning Model จะมี Use Case ดังนี้

1) การ Inference Machine Learning Model

2) การ View Sharing Machine Learning Model

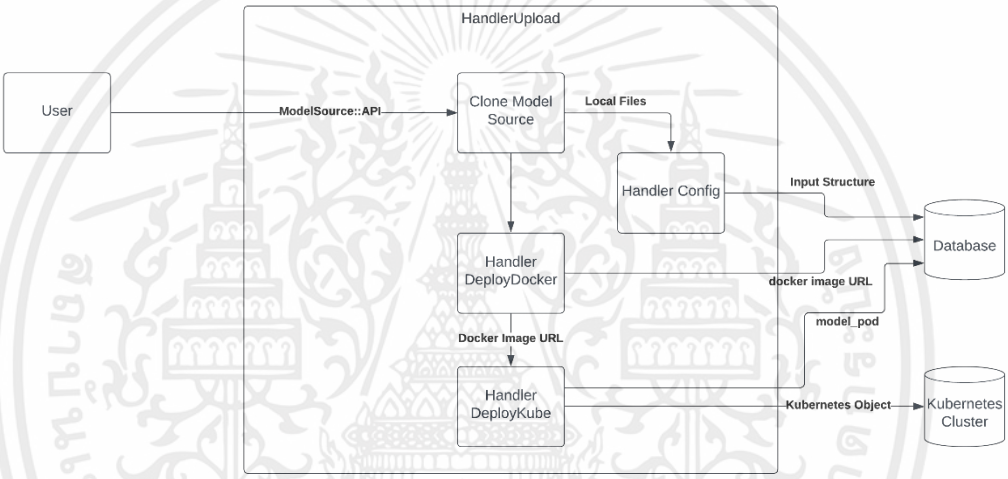
3) การ Delete Machine Learning Model

4) การ Upload Machine Learning Model

5) การ Update Machine Learning Model

6) การเข้าใช้งานระบบ

3.3 ภาพรวมการทำงานของระบบ



รูปที่ **3.2** การทำงานของระบบ **Upload**

3.3.1 การทำงานของระบบ Upload Machine Learning Model

1) รับ Input ของผู้ใช้งานเข้ามาเป็น Git URL

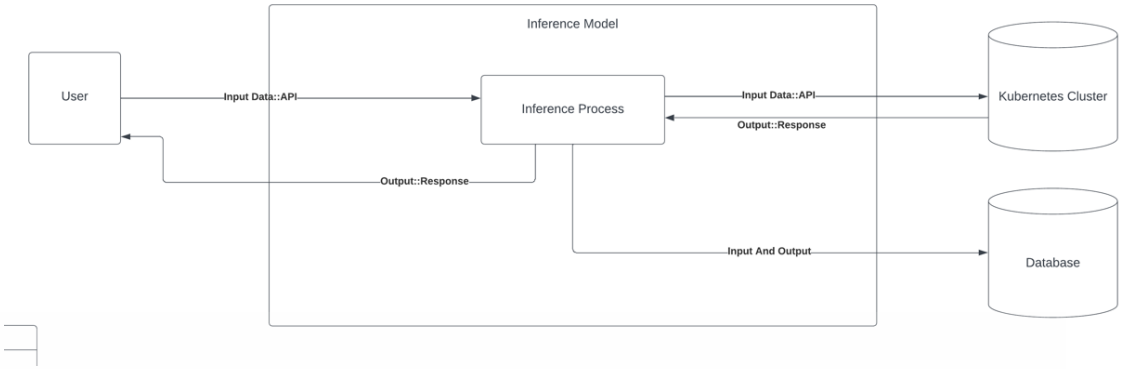
2) Clone Source Code ลงระบบ

3) สร้าง Docker Image

4) สร้าง Kubernetes Object

5) สร้าง Machine Learning Model Input Structure

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



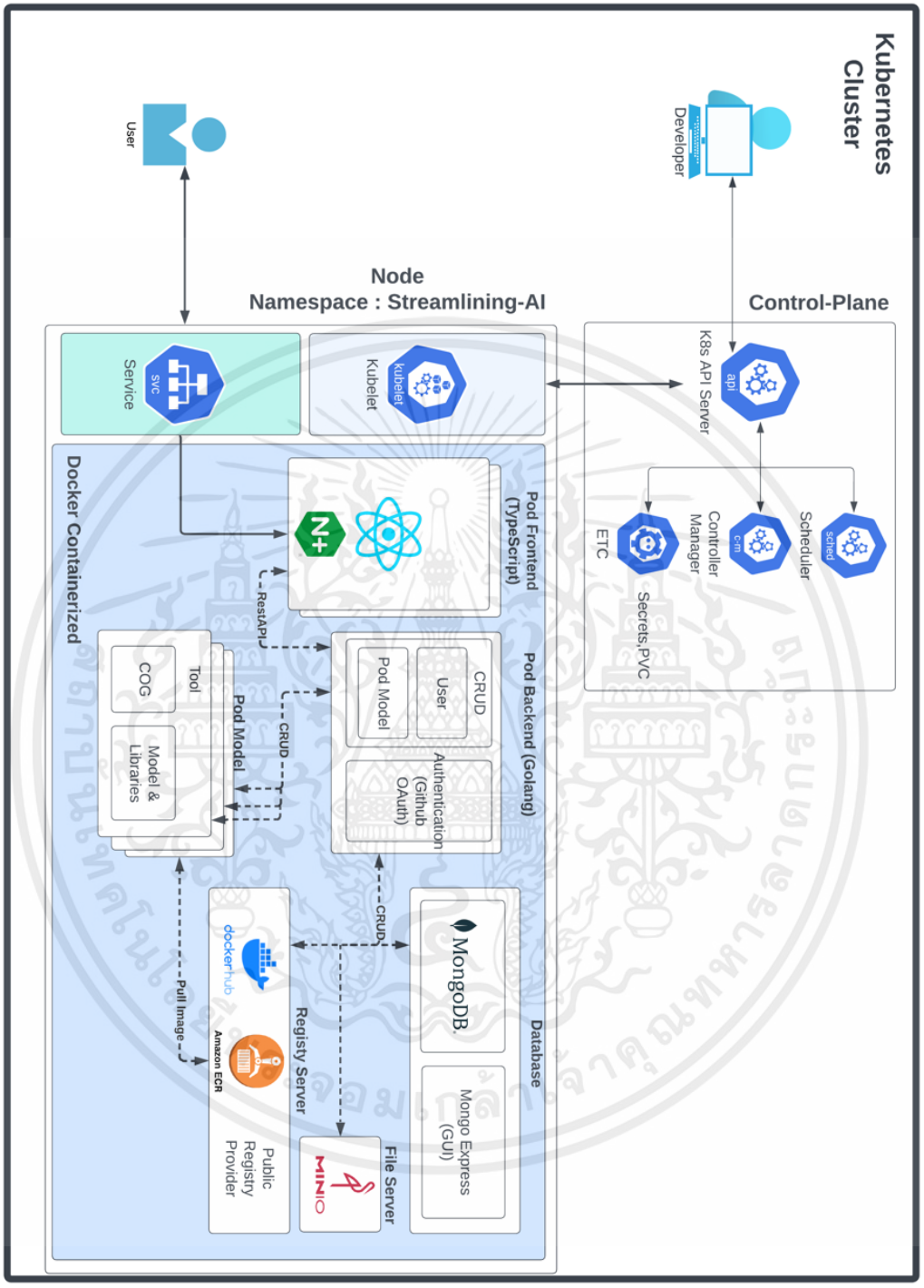
รูปที่ **3.3** การทำงานของระบบ **Inference**

3.3.2 การทำงานของระบบ Inference Machine Learning Model

1) รับ Input ของผู้ใช้งานเข้ามาเป็นผ่าน API
2) ตรวจสอบข้อมูล Input
3) Inference Machine Learning Model
4) ตอบกลับ Output ไปให้ผู้ใช้งาน
5) บันทึกผลลัพธ์ลงฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

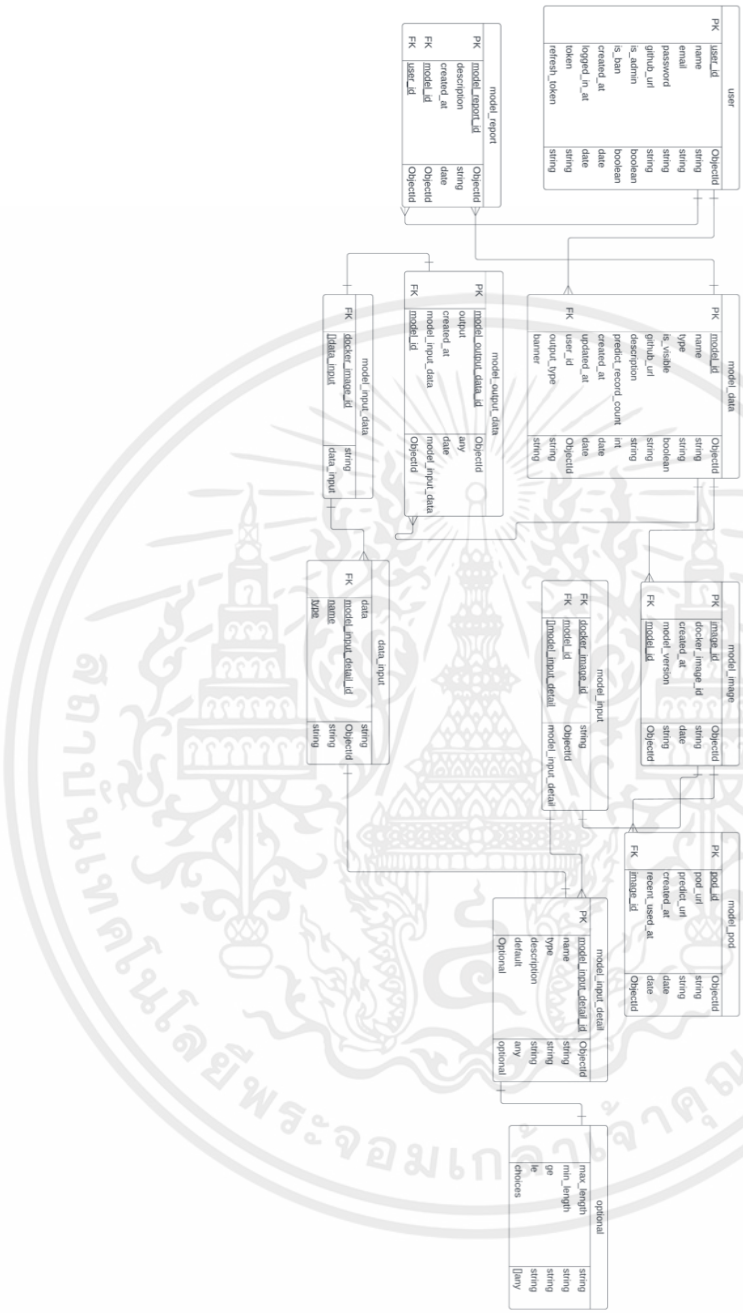
3.4 Software Architecture



รูปที่ 3.4 Software Architecture

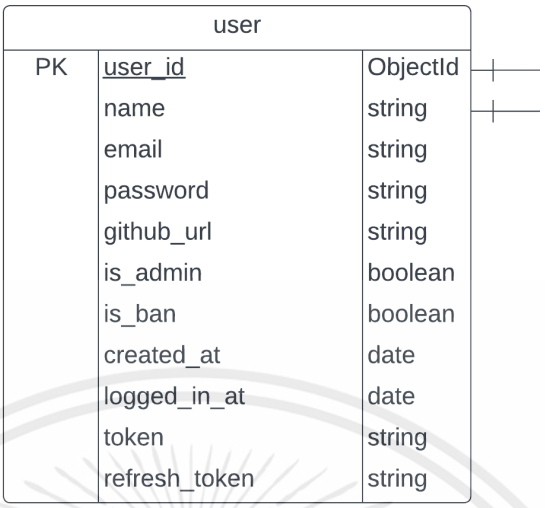
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ER Diagram



รูปที่ 3.5 ER Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

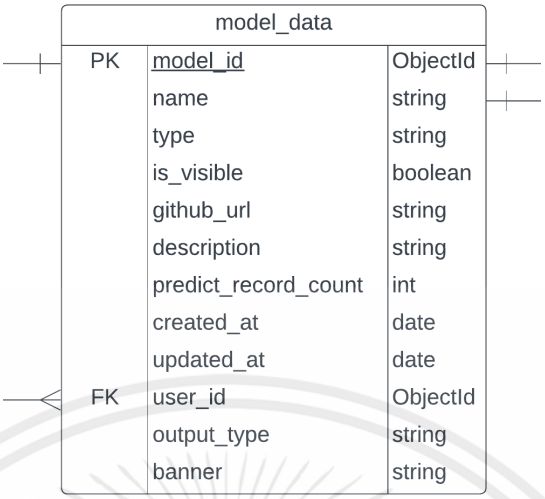


รูปที่ 3.6 User ER Diagram

ตาราง **3.1 User**

Name	**Type**	**Key**	**Description**
user_id	ObjectId	PK	User ID
name	String	-	Name of user
email	String	-	User Email
password	String	-	User Password
github_url	String	-	User github url
is_admin	Boolean	-	User Role
is_ban	Boolean	-	User Ban Status
created_at	date	-	User Creation Date
logged_in_at	date	-	Last User Login Date
token	string	-	user’s login token
refresh_token	string	-	user’s refresh token

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

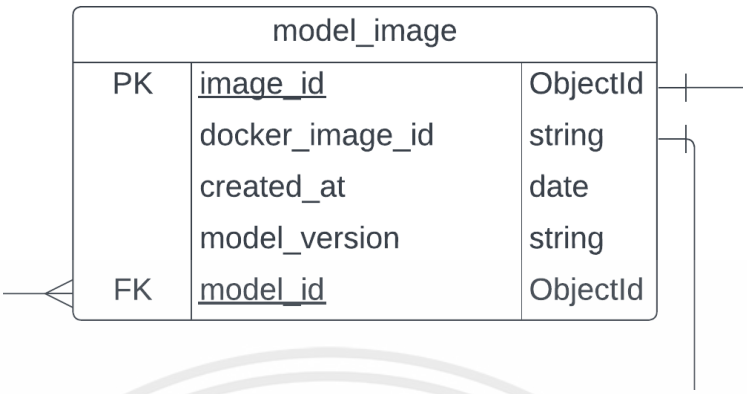


รูปที่ 3.7 Model ER Diagram

ตาราง **3.2 Model Data**

Name	**Type**	**Key**	**Description**
model_id	ObjectId	PK	Model ID
name	String	-	Name of Model
type	String	-	Type of Model
is_visible	Boolean	-	Model Visibility
github_url	String	-	Git URL of Model
description	String	-	Description
predict_record_count	int	-	Count
created_at	date	-	Model Creation Date
updated_at	date	-	Model Update Date
user_id	ObjectID	FK	Owner's ID
output_type	string	-	Output type
banner	string	-	Banner URL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

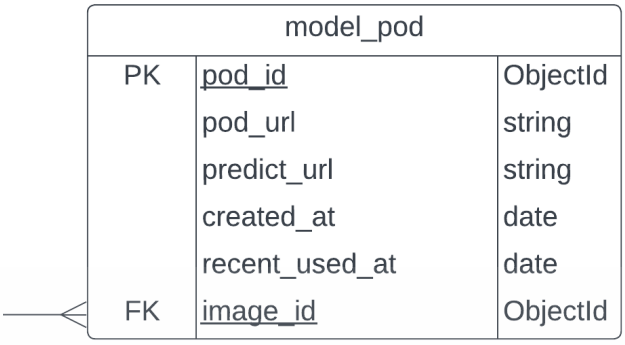


รูปที่ **3.**8 **Model Image ER Diagram**

ตาราง **3.3 Model Image**

Name	**Type**	**Key**	**Description**
image_id	ObjectId	PK	Image ID
docker_image_id	String	-	Docker Image Name
created_at	date	-	Image Creation Date
model_version	String	-	Version of Image
model_id	ObjectID	FK	Model ID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

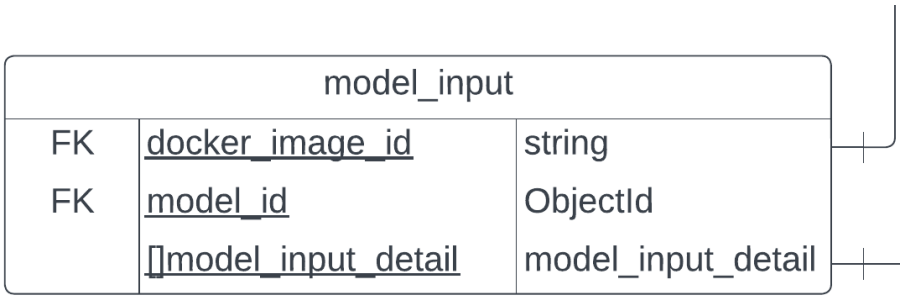


รูปที่ 3.9 Model Pod ER Diagram

ตาราง **3.4 Model Pod**

Name	**Type**	**Key**	**Description**
pod_id	ObjectId	PK	Pod ID
pod_url	String	-	Service Pod URL
predict_url	String	-	Predict URL
created_at	date	-	Pod Creation Date
recent_used_at	date	-	Latest used Date
image_id	ObjectID	FK	Image ID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

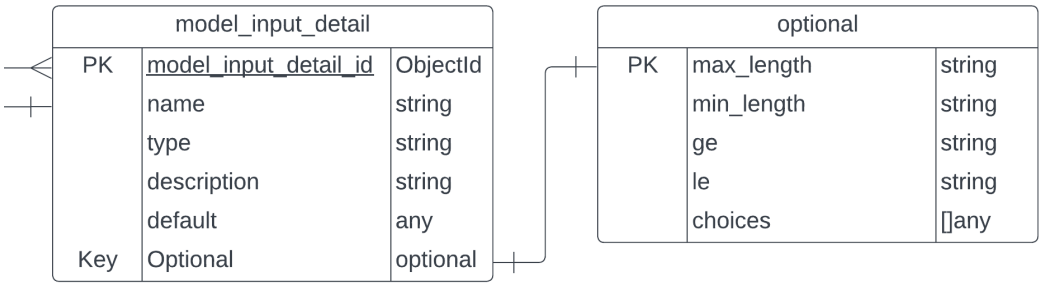


รูปที่ **3.**10 **Model Input ER Diagram**

ตาราง **3.5 Model Input**

Name	**Type**	**Key**	**Description**
docker_image_id	string	FK	Docker Image ID
model_id	ObjectID	FK	Model ID
model_input_detail	[]model_input_detail	-	List of Input Detail

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ **3.**11 **Model Input Detail** และ **Optional ER Diagram**

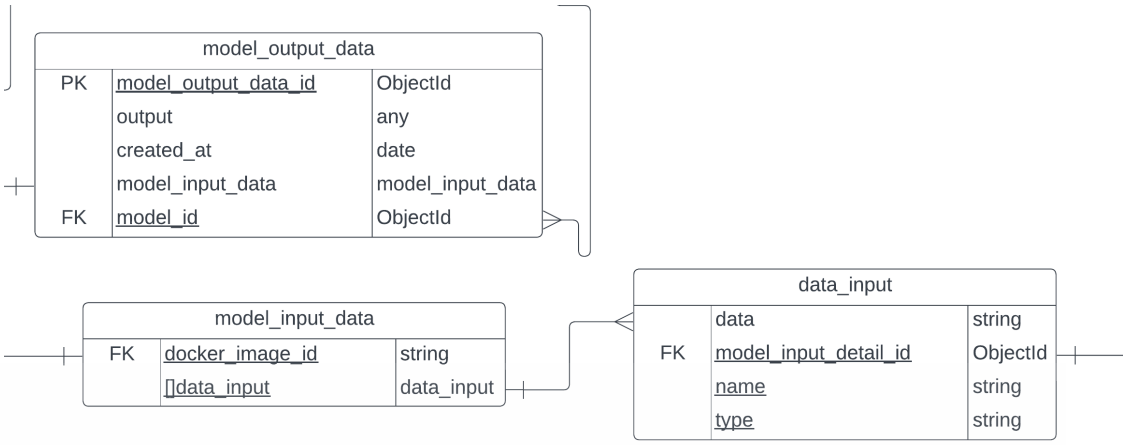
ตาราง **3.6 Model Input Detail**

Name	**Type**	**Key**	**Description**
model_input_detail_id	ObjectId	PK	Input Detail ID
name	String	-	Name of input
type	String	-	Data type
description	string	-	Description
default	any	-	Default of Input
Optional	optional	-	Optional Type

ตาราง 3.7 Optional

Name	**Type**	**Key**	**Description**
max_length	string	-	Max length of Input
min_length	string	-	Min length of Input
ge	String	-	Value Greater than
le	string	-	Value Lower than
choices	[]any	-	Choice of Input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ **3.**12 **Model Output Data** และ **Model Input Data** และ **Data Input ER Diagram**

ตาราง **3.8 Model Output Data**

Name	**Type**	**Key**	**Description**
model_output_data_id	ObjectID	-	ID of output
output	any	-	output
created_at	date	-	Creation Date
model_input_data	model_input_data	-	Data of Input
model_id	ObjectID	FK	Model ID

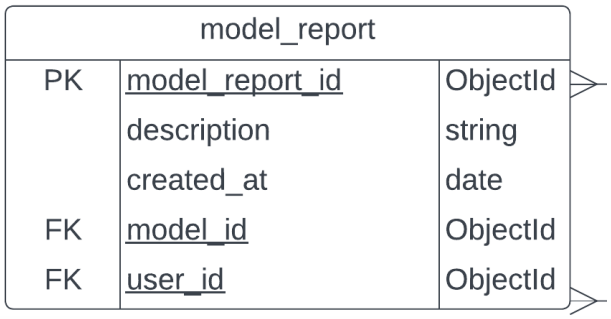
ตาราง **3.9 Model Input Data**

Name	**Type**	**Key**	**Description**
docker_image_id	string	-	Docker Image ID
data_input	[]data_input	-	List of Input Data

ตาราง **3.10 Data Input**

Name	**Type**	**Key**	**Description**
data	string	-	Data
model_input_detail_id	ObjectID	FK	ID of input Detail
name	String	-	name of input
type	String	-	Type of Input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



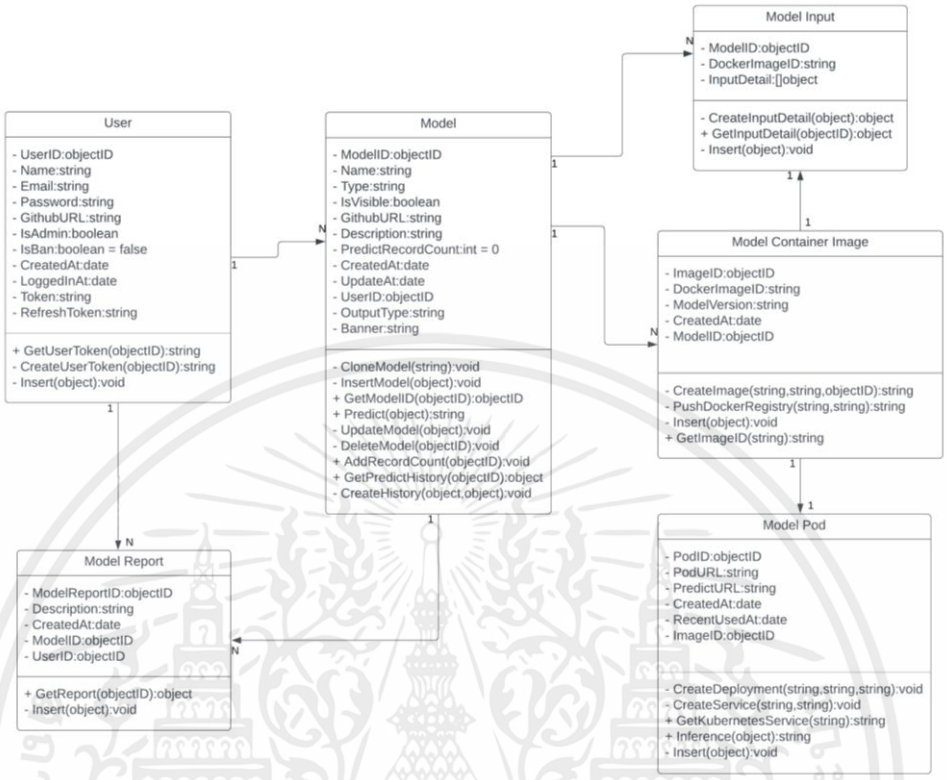
รูปที่ **3.**13 **Model Report ER Diagram**

ตาราง **3.11 Model Report**

Name	**Type**	**Key**	**Description**
model_report_id	ObjectID	-	ID of Report
description	string	-	Reason
created_at	date	-	Creation Date
model_id	ObjectID	FK	Model to Report
user_id	ObjectID	FK	User who reported

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 Class Diagram



รูปที่ 3.14 Class Diagram

ระบบจะประกอบไปด้วย 6 Classes

1) Class User

2) Class Model

3) Class Model Container Image

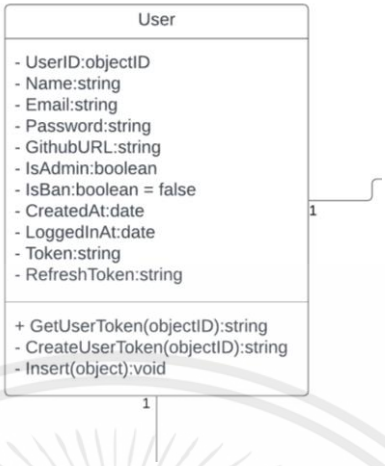
4) Class Model Pod

5) Class Model Input

6) Class Model Report

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.1 Class User



รูปที่ 3.15 User Class Diagram

ตาราง **3.12 Attributes ของ User**

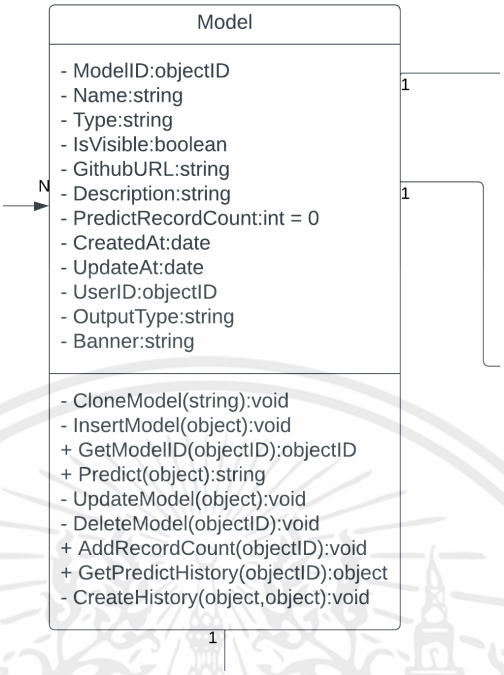
Accessible	**Name**	**Type**	**Description**
Private	UserID	ObjectID	User ID
Private	Name	string	Name
Private	Email	string	Email
Private	Password	string	Password
Private	GithubURL	string	Github Profile
Private	IsAdmin	Boolean	Role
Private	IsBan	Boolean	Default = false
Private	CreatedAt	date	Created Date
Private	LoggedInAt	date	User Last Logged in
Private	Token	string	User Token
Private	RefreshToken	string	User Refresh Token

ตาราง **3.13 Method ของ User**

Accessible	**Name**	**Parameter**	**Return Type**
Public	GetUserToken	objectID	string
Private	CreateUserToken	objectID	string
Private	Insert	object	void

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.2 Class Model



รูปที่ 3.16 Model Class Diagram

ตาราง **3.14 Attributes ของ Model**

Accessible	**Name**	**Type**	**Description**
Private	ModelID	ObjectID	Model ID
Private	Name	string	Name
Private	Type	string	Type of Model
Private	GithubURL	string	Github URL of Model
Private	Description	string	Description
Private	PredictRecordCount	int	Count
Private	CreatedAt	date	Created Date
Private	UpdateAt	date	Updated Date
Private	UserID	objectID	Owner ID
Private	OutputType	string	Output Type
Private	Banner	string	Model Banner Image

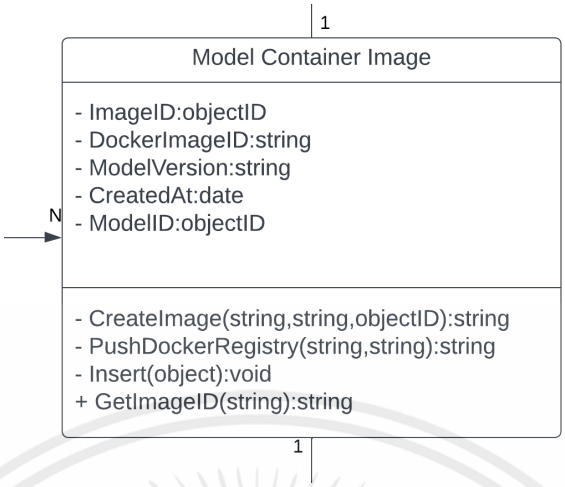
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง **3.15 Method** ของ **Model**

Accessible	**Name**	**Parameter**	**Return Type**
Private	CloneModel	string	void
Private	Insert	object	void
Public	GetModelID	objectID	objectID
Public	Predict	object	string
Private	UpdateModel	object	void
Private	DeleteModel	objectID	void
Public	AddRecordCount	objectID	void
Public	GetPredictHustory	objectID	object
Private	CreateHistory	object, object	void

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.3 Class Model Container Image



รูปที่ 3.17 Model Container Image Class Diagram

ตาราง **3.16 Attributes ของ Model Container Image**

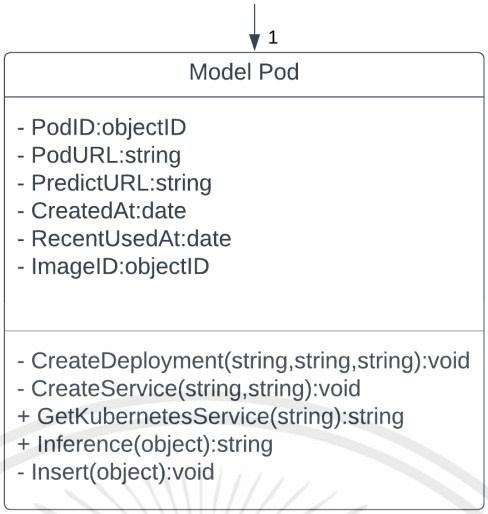
Accessible	**Name**	**Type**	**Description**
Private	ImageID	objectID	ImageID
Private	DockerImageID	string	Docker Image
Private	ModelVersion	string	Version of Model
Private	CreatedAt	date	Created Date
Private	ModelID	objectID	Model ID

ตาราง **3.17 Method ของ Model Container Image**

Accessible	**Name**	**Parameter**	**Return Type**
Private	CreateImage	string, string, objectID	string
Private	PushDockerRegistry	string, string	string
Private	Insert	object	void
Public	GetImageID	string	string

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.4 Class Model Pod



รูปที่ **3.18 Model Pod Class Diagram**

ตาราง **3.18 Attributes** ของ **Model Pod**

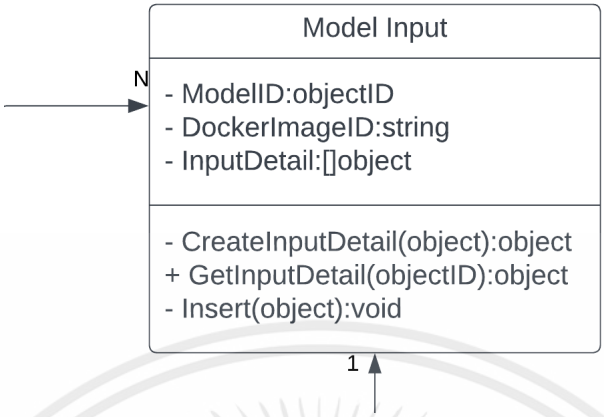
Accessible	**Name**	**Type**	**Description**
Private	PodID	objectID	Pod ID
Private	PodURL	string	URL of Pod
Private	PredictURL	string	URL to Predict
Private	CreatedAt	date	Created Date
Private	RecentUsedAt	date	Recent used Date
Private	ImageID	objectID	Image ID

ตาราง **3.19 Method** ของ **Model Pod**

Accessible	**Name**	**Parameter**	**Return Type**
Private	CreateDeployment	string, string, string	void
Private	CreateService	string, string	void
Publuc	GetKubernetesService	string	string
Public	Inference	object	string
Private	Insert	object	void

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.5 Class Model Input



รูปที่ 3.19 Model Input Class Diagram

ตาราง **3.20 Attributes ของ Model Input**

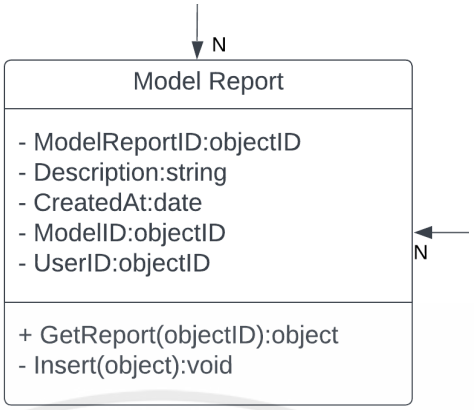
Accessible	**Name**	**Type**	**Description**
Private	ModelID	objectID	ModelID
Private	DockerImageID	string	Docker Image
Private	InputDetail	[]object	List of Input Details

ตาราง **3.21 Method ของ Model Input**

Accessible	**Name**	**Parameter**	**Return Type**
Private	CreateInputDetail	object	object
Public	GetInputDetail	objectID	object
Private	Insert	object	void

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.6 Class Model Report



รูปที่ 3.20 Model Container Image Class Diagram

ตาราง **3.22 Attributes ของ Model Container Image**

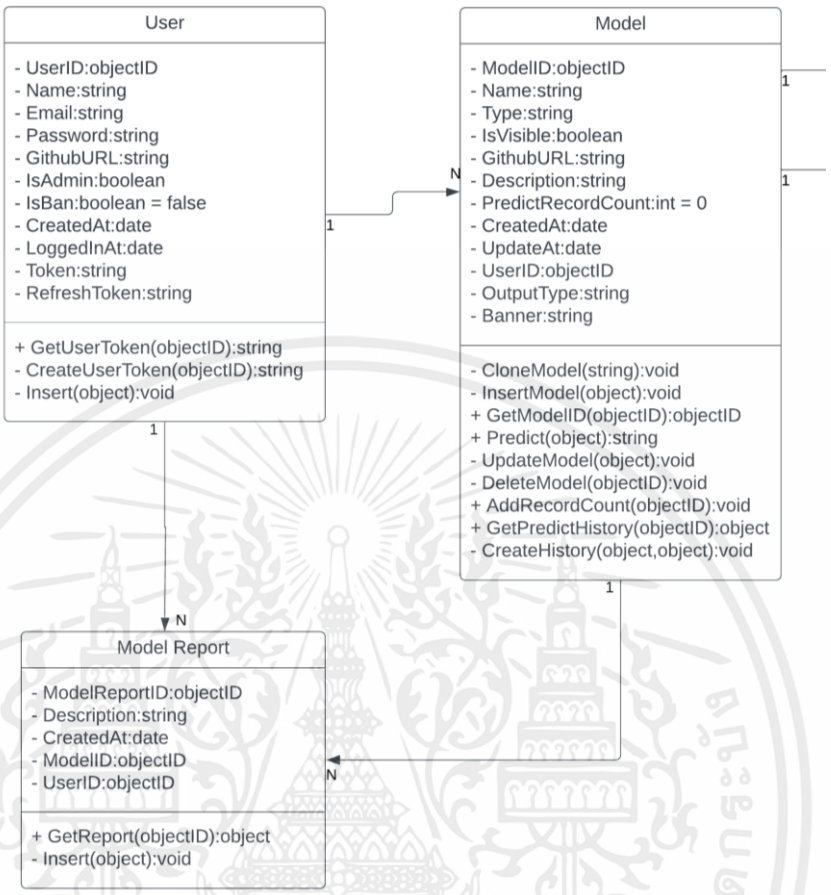
Accessible	**Name**	**Type**	**Description**
Private	ModelReportID	objectID	Report ID
Private	Description	string	Description
Private	CreatedAt	date	Created Date
Private	ModelID	objectID	Model ID to Report
Private	UesrID	objectID	User who Report

ตาราง 3.23 Method ของ Model Container Image

Accessible	**Name**	**Parameter**	**Return Type**
Public	GetReport	objectID	object
Private	Insert	object	void

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.7 ความสัมพันธ์ของ **Class User**



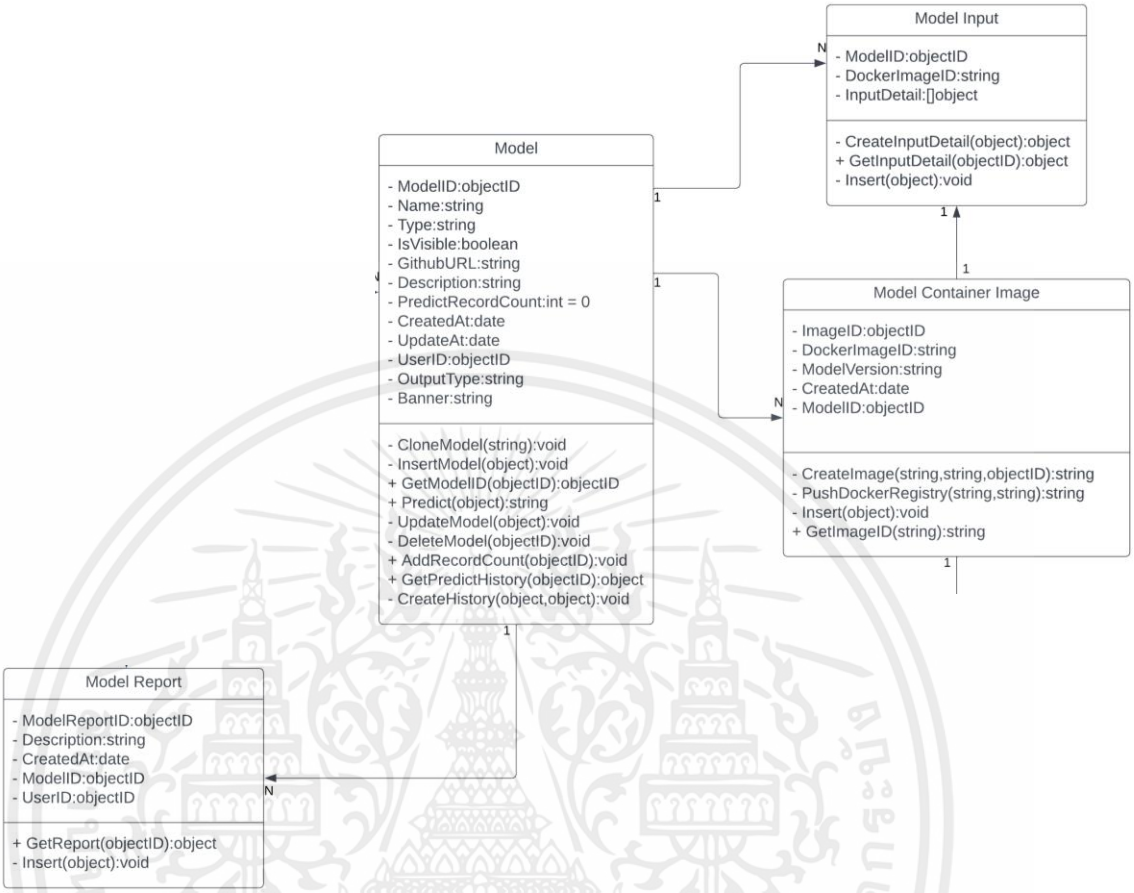
รูปที่ **3.21** ความสัมพันธ์ของ **Class User**

1) Class User มีความสัมพันธ์กับ Class Model แบบ One - Many

2) Class User มีความสัมพันธ์กับ Class Model Report แบบ One – Many

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.8 ความสัมพันธ์ของ **Class Model**



รูปที่ 3.22 ความสัมพันธ์ของ **Class Model**

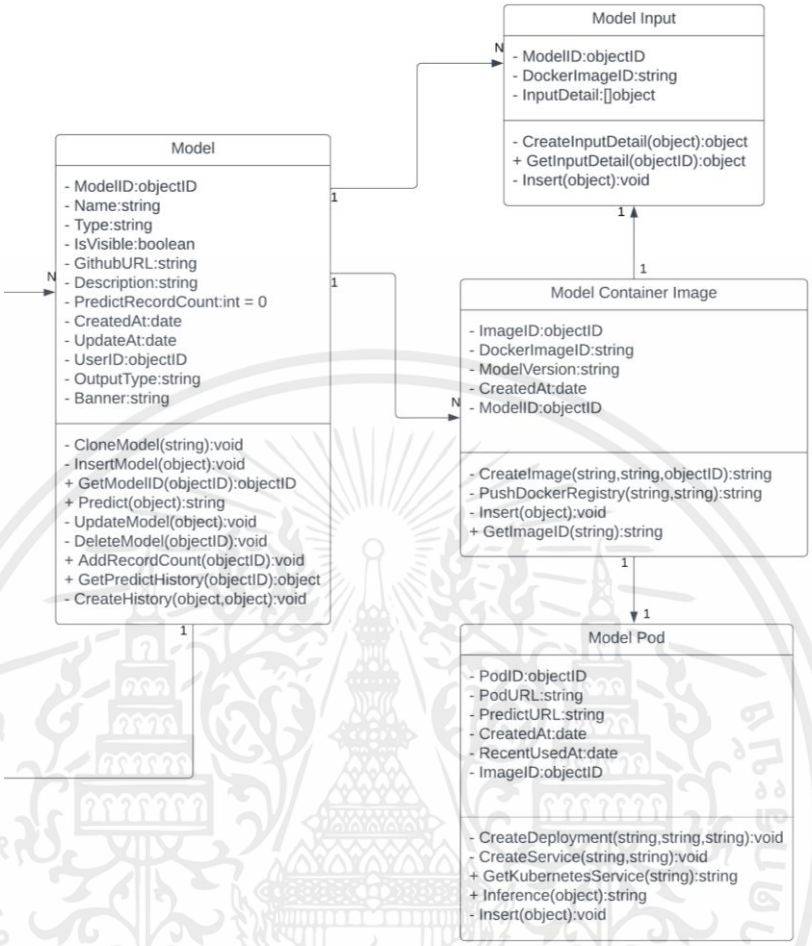
1) Class Model มีความสัมพันธ์กับ Class Model Container Image แบบ One - Many

2) Class Model มีความสัมพันธ์กับ Class Model Input แบบ One – Many

3) Class Model มีความสัมพันธ์กับ Class Model Report แบบ One – Many

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.9 ความสัมพันธ์ของ **Class Model Container Image**



รูปที่ **3.23** ความสัมพันธ์ของ **Class Model Container Image**

1) Class Model Container Image มีความสัมพันธ์กับ Class Model แบบ Many – One

2) Class Model Container Image มีความสัมพันธ์กับ Class Model Pod แบบ One – One

3) Class Model Container Image มีความสัมพันธ์กับ Class Model Input แบบ One – One

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการดำเนินงาน

4.1 หน้าหลัก (Homepage)

โดยใช้งานบน Web Application จะแสดงหน้านี้เป็นหน้าแรกโดยจะมีองค์ประกอบของ Navigator Bar และ Side Bar(เมื่อทำการเข้าสู่ระบบ) เป็นส่วนเชื่อมโยงไปหน้าอื่นๆ ได้แก่

4.1.1 หน้าหลัก (Home)

4.1.2 หน้าเข้าสู่ระบบ (Login)

4.1.3 หน้าสำรวจโมเดล (Explore)

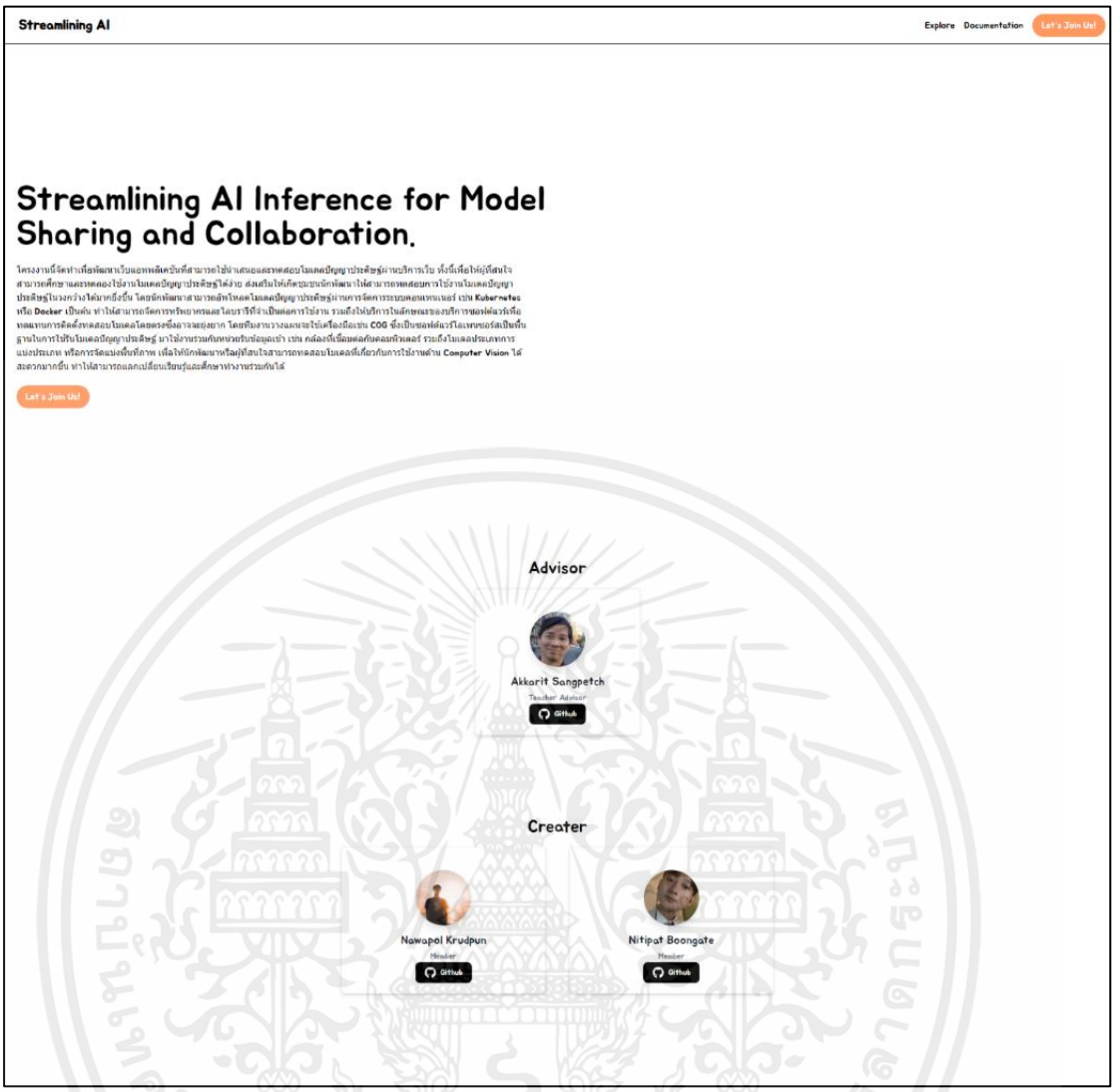
4.1.3 หน้าการอัพโหลดโมเดล (Upload Model Page)

4.1.3 หน้าการอัพเดตโมเดล (Update Model Page)

4.1.4 หน้าการจัดการของผู้ใช้ (Dashboard)

4.1.5 หน้าคู่มือการใช้งาน (Documentation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ **4.1** หน้าหลัก

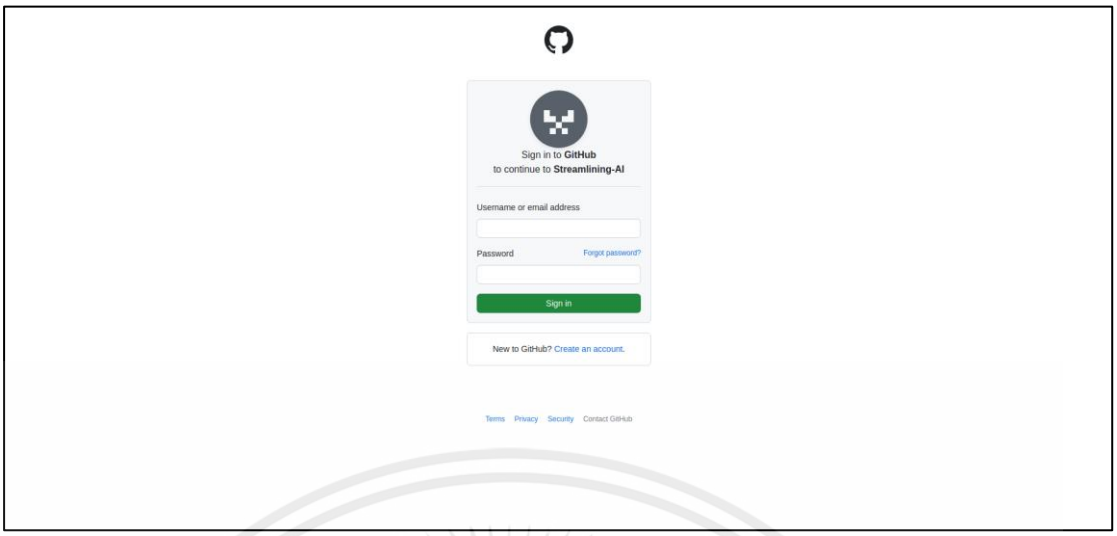
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 หน้าเข้าสู่ระบบ (Login)

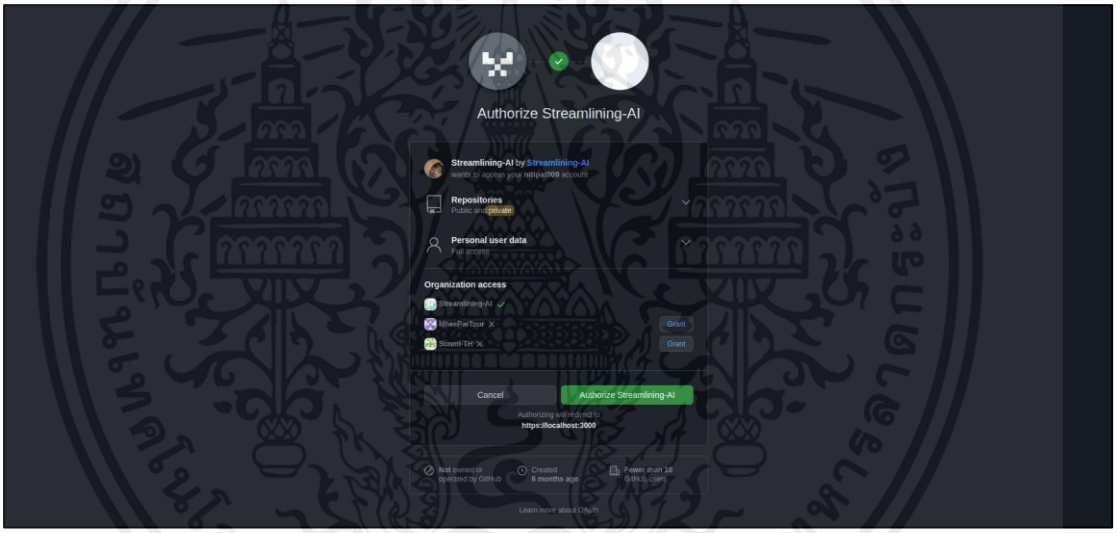
การเข้าใช้งานผ่านระบบโดยผู้ใช้กดปุ่ม Sign in with GitHub ก็จะแสดงหน้าต่างสำหรับกรอก Email และ Password ของ GitHub เพื่อเข้าสู่ระบบ โดยจะมีการขอสิทธิในการอ่านข้อมูล Repository ทั้ง Public และ Private อีกทั้งยังมีข้อมูลส่วนตัว เช่น email , profile , information ,followers เป็นต้น

รูปที่ **4.2** หน้าเข้าสู่ระบบเว็บแอพพลิเคชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ **4.**3 หน้าเข้าสู่ระบบของ **GitHub**



รูปที่ **4.4** หน้าการขอสิทธิอ่านข้อมูลของ **Github**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 หน้าสำรวจโมเดล (Explore)

หน้านี้จะแสดงโมเดลทั้งหมดที่มีอยู่บนระบบ ซึ่งจะสามารถเลือกในส่วนของ Dropdown เพื่อกรองชนิดของโมเดลและในช่อง Search เพื่อค้นหาชื่อโมเดลที่ต้องการ โดยรูปแบบการแสดงของแต่ละโมเดลจะมีองค์ประกอบดังนี้

แบนเนอร์ (Banner) รูปภาพแสดงหน้าปกของโมเดล

ชื่อ (Name) ประกอบไปด้วย เจ้าของโมเดล/ชื่อโมเดล

รายละเอียด (description)

เมื่อกดไปที่โมเดลจะเปลี่ยนหน้าไปยังหน้าต่างการใช้งานโมเดล

รูปที่ **4.5** หน้าสำรวจโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 หน้าการใช้งานโมเดล (Model Page)

จะแบ่งออกเป็น 3 ส่วนใหญ่ๆ คือ

หน้าต่างสำหรับกรอก Parameter และ Predict โมเดล (ส่วนบนซ้าย)

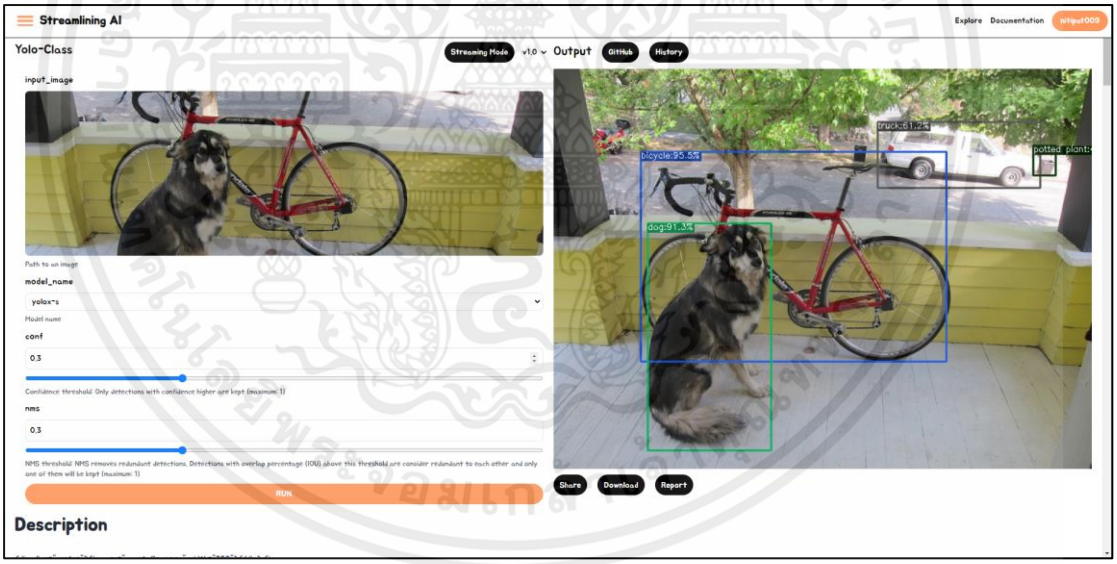
- จำแนก Input ต่างๆ สำหรับโมเดลนั้นๆตามข้อมูลที่เก็บไว้บนระบบฐานข้อมูล และเมื่อกรอกเสร็จเรียบร้อยก็สามารถกดปุ่ม Run เพื่อทดสอบโมเดล
- ปุ่ม Streaming Mode เพื่อปรับให้การกรอก Input ของรูปภาพเปลี่ยนเป็น Webcam และจะทำการส่งข้อมูลทุกๆ 5 วินาที โดยมีรูปภาพขนาด 640x360 ตามที่กำหนดไว้

หน้าต่างสำหรับแสดงผลลัพธ์ (ส่วนบนขวา)

- โดยจะมีปุ่ม History เพื่อปรับให้เห็นประวัติการใช้งานของโมเดล
- แสดงผลลัพธ์ออกมาเป็นรูปภาพ (Image)

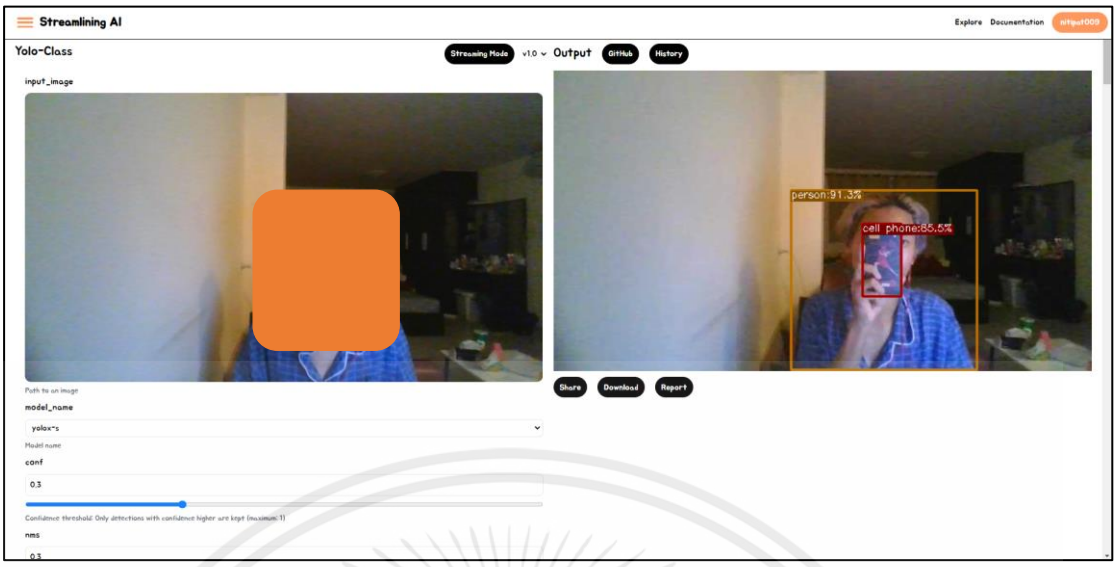
หน้าต่างสำหรับแสดงรายละเอียดของโมเดล (ส่วนล่าง)

- แสดงรายละเอียดตามข้อมูลที่เก็บไว้บนระบบฐานข้อมูล



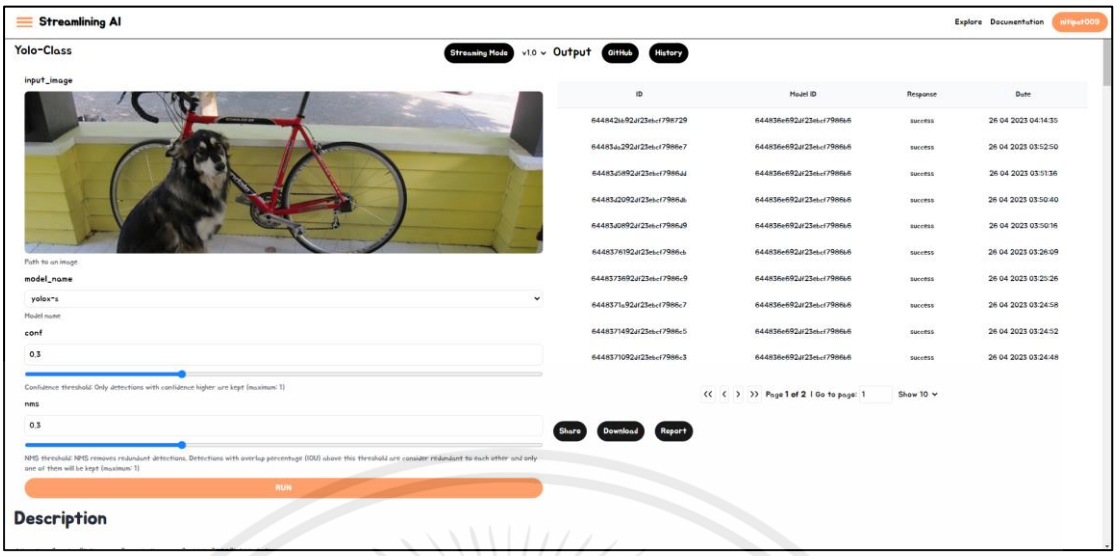
รูปที่ **4.6** หน้าการใช้งานโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ **4.7** หน้าการใช้งาน **Streaming Mode**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ **4.8** หน้าการใช้งานประวัติการใช้งานของโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 หน้าการอัพโหลดโมเดล (Upload Model Page)

ใช้สำหรับการกำหนดค่าที่จำเป็นในการอัพโหลดโมเดลขึ้นระบบโดยผู้ใช้จะต้องกรอกข้อมดังต่อไปนี้

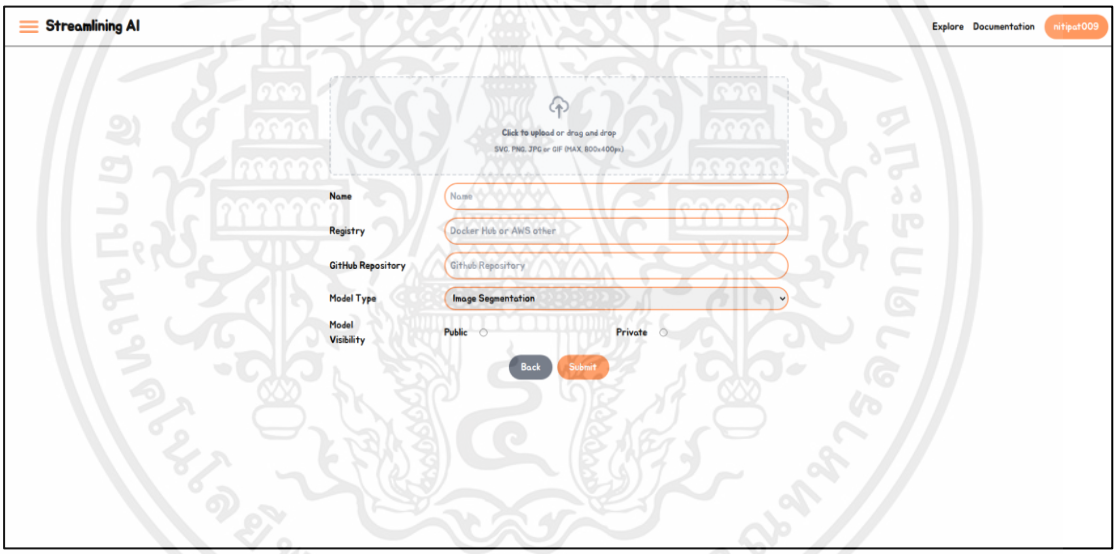
แบนเนอร์ (Banner) สำหรับแสดงรูปภาพตัวอย่างของโมเดล

ลิงค์สำหรับ GitHub Repository (GitHub Repository's URL) สำหรับ Clone ข้อมูลของโมเดลเพื่อใช้อัพโหลด

ลิงค์สำหรับ Docker Registry สำหรับการสร้างโมเดลเพื่อใช้อัพโหลด

ชนิดของโมเดล (Model Type) แบ่งออกเป็น 2 ชนิด คือ Image Classification และ Image Segmentation

การเผยแพร่ (Visibility) แบ่งออกเป็น 2 ชนิด คือ สาธารณะ (Public) และ ส่วนตัว (Private)

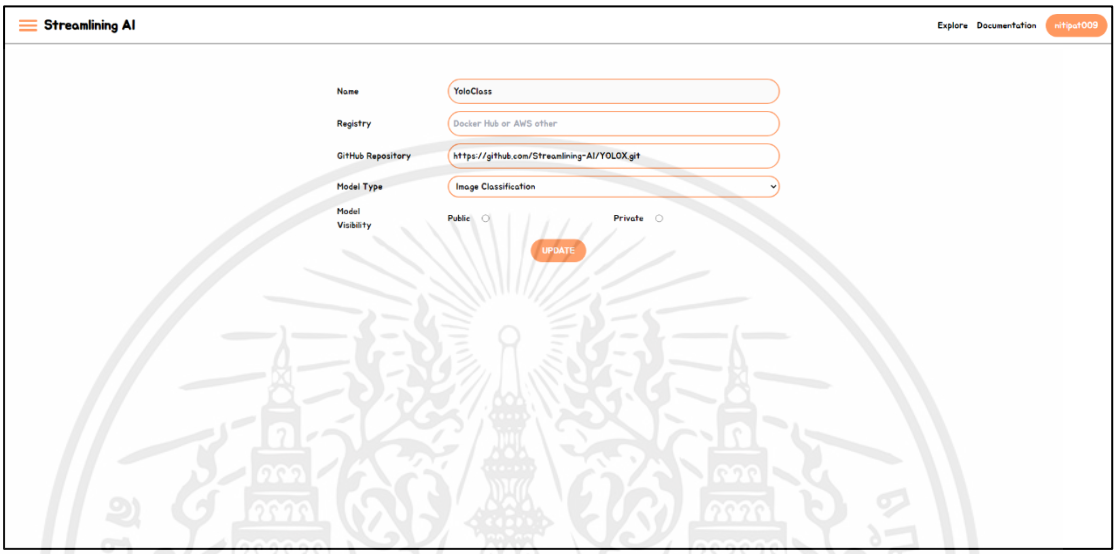


รูปที่ **4.9** หน้าการอัพโหลดโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 หน้าการอัพเดตโมเดล (Update Model Page)

หน้านี้จะมีลักษณะการรับข้อมูลเหมือนกับหน้าการอัพโหลดโมเดลเนื่องจากจะต้องการอัพเดตข้อมูลที่มีอยู่แล้ว โดยเมื่อเข้ามาในหน้านี้จะมีข้อมูลเก่ากรอกไว้ เพื่อให้ทราบรายละเอียดข้อมูลเดิมหลังจากแก้ไขเรียบร้อยสามารถกดปุ่ม Update เพื่อทำการอัพเดตข้อมูล

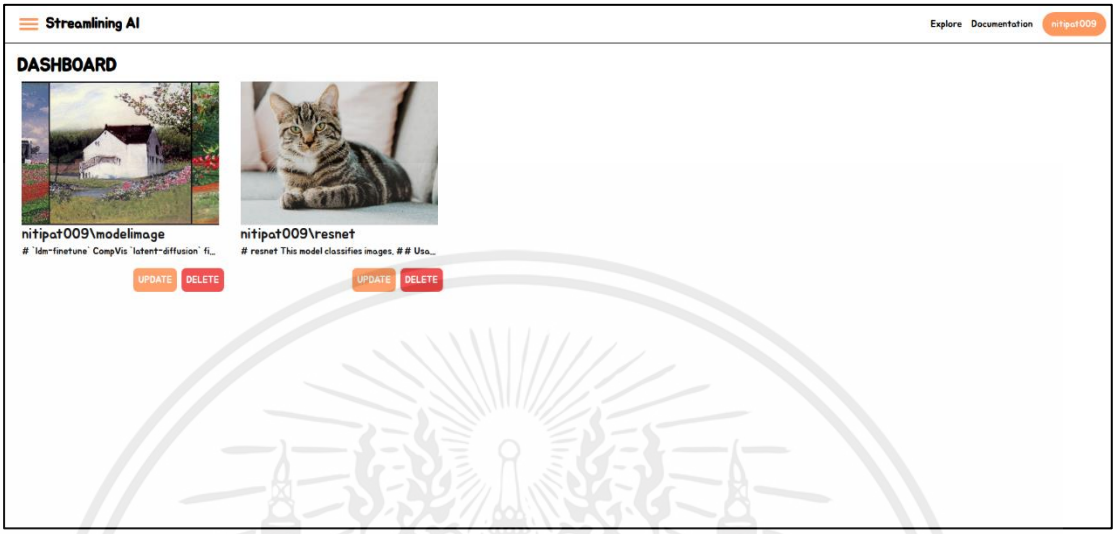


รูปที่ **4.10** หน้าการอัพเดตโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 หน้าการจัดการของผู้ใช้ (Dashboard)

แสดงโมเดลที่ผู้ใช้อัพโหลดและจัดการลบหรืออัพเดตได้ตามปุ่มด้านล่างของโมเดล

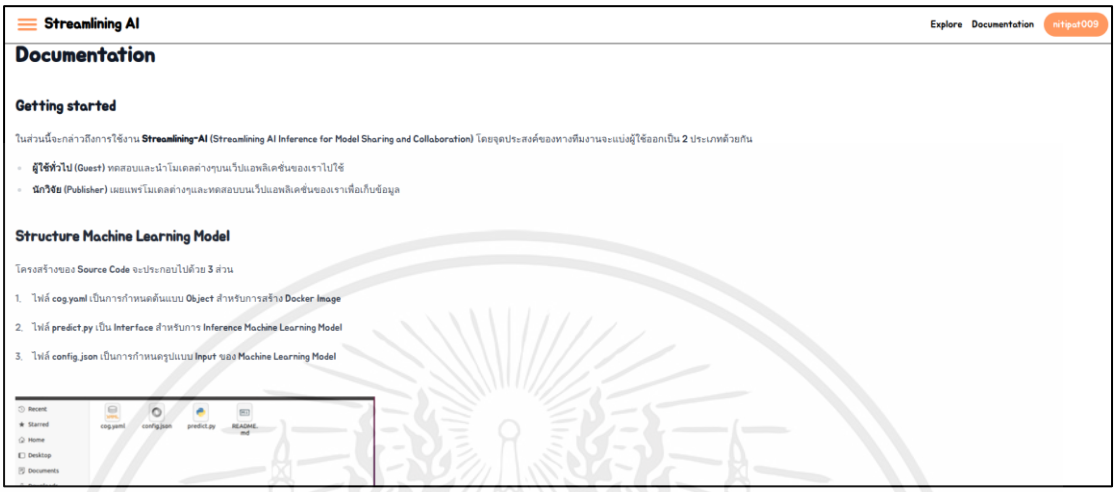


รูปที่ **4.11** หน้าการจัดการของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 หน้าคู่มือการใช้งาน (Documentation Page)

ใช้สำหรับให้ผู้ใช้ทำความเข้าใจในการใช้งานบน Web Application ได้อย่างถูกต้อง และให้นักวิจัยสามารถทำความเข้าใจและเตรียมตัวก่อนการจะอัพโหลดโมเดลของตนเองขึ้นสู่ระบบ



รูปที่ **4.12** หน้าคู่มือการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุป

5.1 ผลที่ได้จากโครงงาน

5.1.1 ระบบตรวจสอบผู้ใช้งาน

การเข้าใช้งานจะสามารถทำได้ 1 วิธี

1) เข้าใช้งานระบบผ่าน GitHub OAuth

5.1.2 ระบบการแบ่งปัน Machine Learning Model

1) รับไฟล์ Source Code ของ Machine Learning Model ผ่านทาง Git URL

2) สร้าง Docker Image สำหรับ Machine Learning Model ด้วย COG Tool

3) ดึง Docker Image จาก Public Docker Registry มา Deploy Service Machine Learning Model ผ่าน Kubernetes

4) การ Update Version ของ Machine Learning Model

5.1.3 ระบบ Inference Machine Learning Model

1) รับ Input ที่หลากหลายในแต่ละ Machine Learning Model สำหรับการ Inference

2) รองรับการ Inference Machine Learning Model ผ่าน Rest API

3) จัดเก็บ Input และ Output ที่ได้จากการ Inference Machine Learning Model

5.1.4 Database

1) สร้างไฟล์ YAML สำหรับ Kubernetes เพื่อการสร้าง Service ของ Mongo DB

2) สร้างไฟล์ YAML สำหรับ Kubernetes เพื่อการสร้าง Deployment ของ Mongo DB

3) สร้างไฟล์ YAML สำหรับ Kubernetes เพื่อการสร้าง Config Map ของ Mongo DB

4) สร้างไฟล์ YAML สำหรับ Kubernetes เพื่อการสร้าง Secret ของ Mongo DB และ Mongo Express

5) สร้างไฟล์ YAML สำหรับ Kubernetes เพื่อการสร้าง Service ของ Mongo Express

6) สร้างไฟล์ YAML สำหรับ Kubernetes เพื่อการสร้าง Deployment ของ Mongo Express

7) สร้างไฟล์ YAML สำหรับ Kubernetes เพื่อการสร้าง Config Map ของ Mongo Express

5.1.5 หน้าเว็บไซต์

1) หน้าหลัก

2) หน้าการเข้าใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) หน้าการ Upload Machine Learning Model

4) หน้าการเรียกใช้การ Inference Machine Learning Model

5) หน้าการดูประวัติผลลัพธ์ของการ Inference Machine Learning Model

6) หน้าการ Update Machine Learning Model

5.2 ปัญหาและอุปสรรค

1) Machine Learning Model มีขนาดใหญ่ส่งผลต่อการ Streaming

2) การตั้งค่า Kubernetes Cluster

5.3 เสนอแนะการพัฒนาต่อ

1) พัฒนาระบบให้สามารถรองรับ Machine Learning Model ที่หลากหลายมากขึ้น

2) พัฒนาระบบให้สามารถ Train Machine Learning Model ได้

3) พัฒนาระบบให้สามารถรองรับ Machine Learning Model จากหลากหลายช่องทาง

5.4 สรุป

จากแบบสอบถามจะพบได้ว่าปัญหาการ Deployment Machine Learning Models จะมี

1) การติดตั้ง CUDA ที่มีขั้นตอนยุ่งยาก

2) การติดตั้ง Libraries และ Tools ที่หลากหลายเวอร์ชัน

3) อุปกรณ์ไม่รองรับ CUDA

และจากแบบสอบถามจะพบได้ว่าเวลาเฉลี่ยในการ Deploy Machine Learning ขั้นตอนต่าง ๆ เป็นดังนี้

1) การติดตั้ง CUDA บนอุปกรณ์ของตนเองใช้เวลาเฉลี่ยประมาน 1 ชั่วโมง

2) การติดตั้ง Library ต่างๆ เช่น TensorFlow ใช้เวลาเฉลี่ยประมาน 4 ชั่วโมง

3) การตั้งค่า Network เผื่อให้สามารถเรียกใช้ Machine Learning Models ผ่าน Web API ใช้เวลาเฉลี่ยประมาน 1 ชั่วโมง

4) Configuration Machine Learning Models บน VM ใช้เวลาเฉลี่ย 30 นาที

5) หากทำการติดตั้งบน Virtual Machine ต้องคำนึงถึง Version ของ Library ต่าง ๆ ทั้งบนอุปกรณ์ และที่ Machine Learning Models ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งระบบที่ของผู้จัดทำสามารถลดขั้นตอนการติดตั้ง CUDA, Libraries, Tools และการตั้งค่า Network เพื่อให้เรียกใช้งานผ่าน Web API ด้วยขั้นตอนการเขียนไฟล์ cog โดยจากการทดลองใช้เวลาประมาน 60 นาทีในการเขียนไฟล์ cog และการ Build Docker Image

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Panuwat W. OAuth, Medium, Retrieved from https://medium.com/@panuwat.w/oauth-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-bde9e1f773f9

[2] GitHub Docs. (n.d.). Authorizing OAuth Apps. Retrieved from https://docs.github.com/en/apps/oauth-apps/building-oauth-apps/authorizing-oauth-apps

[3] React. (n.d.). Legacy React Documentation. Retrieved from https://legacy.reactjs.org/

[4] Devahoy. (2018, July 13). Introduction to Redux. Retrieved from https://devahoy.com/blog/2018/07/introduction-to-redux

[5] TypeScript. (n.d.). TypeScript - JavaScript that scales. Retrieved from https://www.typescriptlang.org/

[6] Mahidol University. (n.d.). TypeScript. Retrieved from https://il.mahidol.ac.th/th/i-learning-clinic/computerarticles/%E0%B8%97%E0%B8%B3%E0%B9%84%E0%B8%A1%E0%B8%96%E0%B8%B6%E0%B8%87%E0%B8%95%E0%B9%89%E0%B8%AD%E0%B8%87%E0%B9%83%E0%B8%8A%E0%B9%89typescript%E0%B8%97%E0%B8%B1%E0%B9%89%E0%B8%87%E0%B8%97%E0%B8%B5/

[7] Go Programming Language. (n.d.). Go. Retrieved from https://go.dev/

[8] Skooldio. (n.d.). What is Golang?. Retrieved from https://blog.skooldio.com/what-is-golang/

[9] Replicate. (n.d.). Replicate. Retrieved from https://replicate.com/

[10] Replicate. (n.d.). Cog. Retrieved from https://github.com/replicate/cog

[11] Amazon Web Services. (n.d.). Docker. Retrieved from https://aws.amazon.com/th/docker/

[12] Nakhon Si Thammarat Rajabhat University. (n.d.). Docker. Retrieved from https://race.nstru.ac.th/home_ex/blog/topic/show/4200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[13] Openlandscape.cloud. (n.d.). What is Kubernetes?. Retrieved from https://blog.openlandscape.cloud/what-is-kubernetes

[14] Medium. (2021, May 9). Catch ‘Em All With Kubernetes: Pokémon GO Case Study. Retrieved from https://medium.com/nerd-for-tech/catch-em-all-with-kubernetes-pokemon-go-case-study-37a43c5dce1b

[15] M. M. Rovnyagin, K. V. Timofeev, A. A. Elenkin and V. A. Shipugin, "Cloud Computing Architecture for High-volume ML-based Solutions," 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2019, pp. 315-318, doi:10.1109/EIConRus.2019.8656765.

[16] K. Chaowvasin, P. Sutanchaiyanonta, N. Kanungsukkasem and T. Leelanupab, "A Scalable Service Architecture with Request Queuing for Resource-Intensive Tasks," 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2020, pp. 67-70, doi: 10.1109/ECTI-CON49241.2020.9158114.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
ขั้นตอนการสร้าง Docker Image ด้วย COG

ก.1 โครงสร้างของไฟล์ Machine Learning Model

โครงสร้างของ Source Code จะประกอบไปด้วย 3 ส่วน

1) ไฟล์ cog.yaml เป็นการกำหนดต้นแบบ Object สำหรับการสร้าง Docker Image

2) ไฟล์ predict.py เป็น Interface สำหรับการ Inference Machine Learning Model

3) ไฟล์ config.json เป็นการกำหนดรูปแบบ Input ของ Machine Learning Model

รูปที่ ก.1 ตัวอย่างไฟล์ของ Machine Learning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    "input": [
      {
          "name": "image",
          "type" : "image",
          "description" : "example",
          "default" : "https://bucketeer-be99e627-94e7-4e5b-a292-
54eeb40ac303.s3.amazonaws.com/public/models_models_cover_image/8faf1a87-e796-4ee5-83e1-
385184220187/Screenshot_from_2022-06-17_13-12-.png"
      },
      {
          "name": "blur",
          "type" : "int",
          "description" : "example",
          "default" : 1,
          "optional" : {
              "ge" : 1,
              "le" : 10
          }
      }
  ],
    "output" :
      {
          "name": "output",
          "type" : "string"
      }
}
```

รูปที่ ก.2 ตัวอย่างไฟล์ config.json

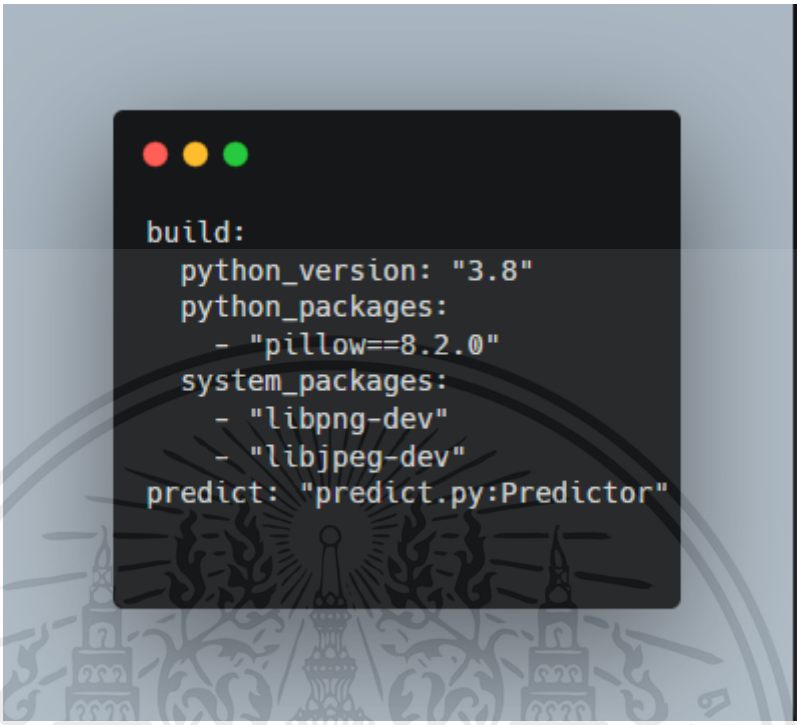
```
import tempfile

from cog import BasePredictor, Input, Path
from PIL import Image, ImageFilter


class Predictor(BasePredictor):
    def predict(
        self,
        image: Path = Input(description="Input image"),
        blur: float = Input(description="Blur radius", default=5),
    ) -> Path:
        if blur == 0:
            return input
        im = Image.open(str(image))
        im = im.filter(ImageFilter.BoxBlur(blur))
        out_path = Path(tempfile.mkdtemp()) / "out.png"
        im.save(str(out_path)
        return
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ก.3 ตัวอย่างไฟล์ predict.py



```
build:
  python_version: "3.8"
  python_packages:
    - "pillow==8.2.0"
  system_packages:
    - "libpng-dev"
    - "libjpeg-dev"
predict: "predict.py:Predictor"
```

รูปที่ ก.4 ตัวอย่างไฟล์ cog.yaml

ก.2 การ Build Docker Image ด้วย COG

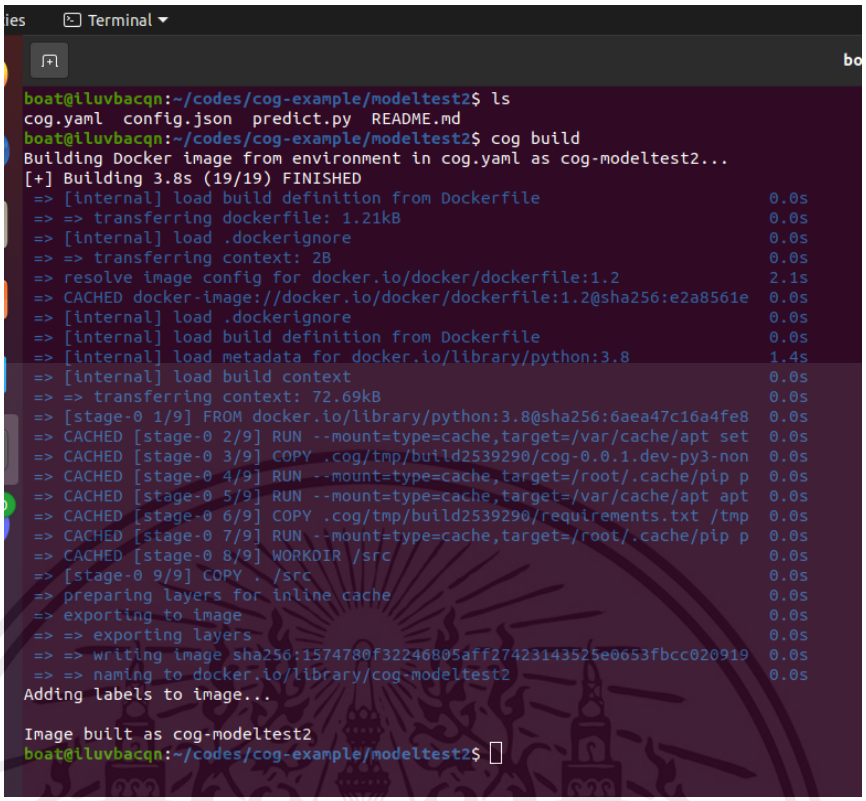
1) ใส่คำสั่ง cog build ใน Directory ของ Source code

2) ทำการ login เข้าสู่ Docker Registry

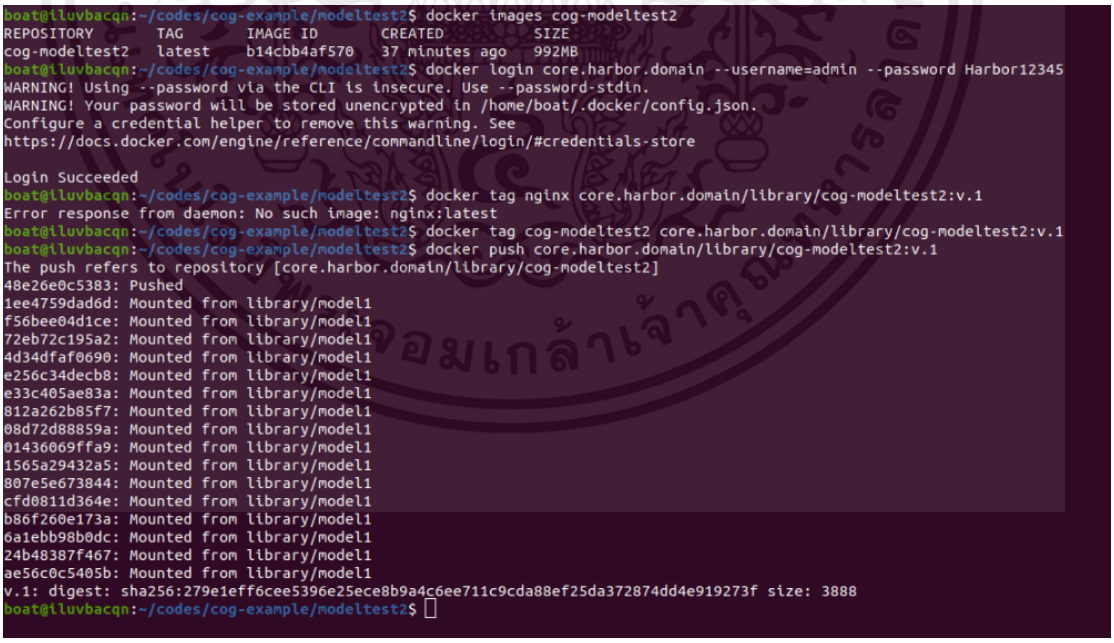
3) ทำการ tag docker image รูปแบบ <RegistryURL>/<library>/<dockername>:<version>

4) ใส่คำสั่ง docker push image name ที่ได้ทำการ tag ไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

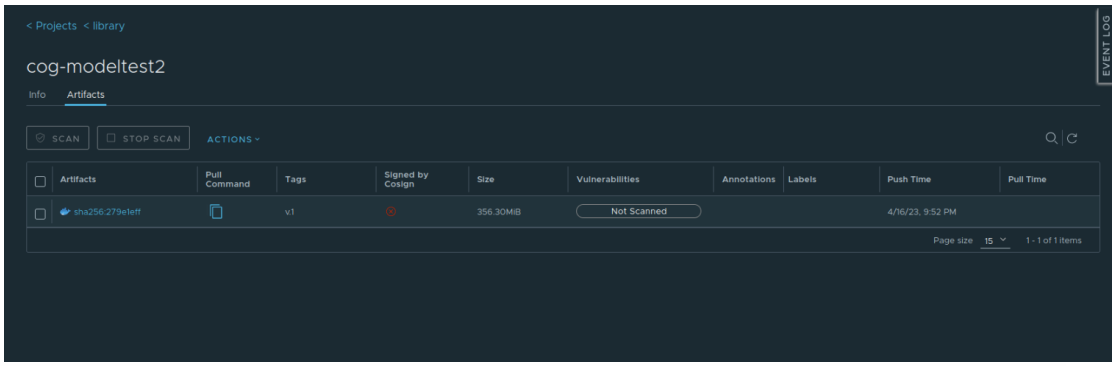


รูปที่ ก.5 การทำงาน cog build



รูปที่ ก.6 การ push docker image ขึ้น registry

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.7 ผลลัพธ์การ Push ขึ้น Registry

```
clientset, err := kubernetes.NewForConfig(config)
    if err != nil {
        return err
    }

service := &v1.Service{
    ObjectMeta: metav1.ObjectMeta{
        Name: serviceName + "-service",
        Labels: map[string]string{
            "app": name,
        },
    },
    Spec: v1.ServiceSpec{
        Type: "LoadBalancer",
        Ports: []v1.ServicePort{
            {
                Port:       5000,
                TargetPort: intstr.FromInt(5000),
                Protocol:   "TCP",
            },
        },
        Selector: map[string]string{
            "app": name,
        },
    },
}

_, err = clientset.CoreV1().Services("default").Create(context.TODO(), service, metav1.CreateOptions{})
```

รูปที่ ก.8 สร้างไฟล์ Kubernetes Service Object สำหรับ Machine Learning Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    clientset, err := kubernetes.NewForConfig(config)
    if err != nil {
        return err
    }

    // Define the Deployment object
    deployment := &v1d.Deployment{
        ObjectMeta: metav1.ObjectMeta{
            Name: serviceName + "-service",
            Labels: map[string]string{
                "app": name,
            },
        },
        Spec: v1d.DeploymentSpec{
            Replicas: int32Ptr(1),
            Selector: &metav1.LabelSelector{
                MatchLabels: map[string]string{
                    "app": name,
                },
            },
            Template: v1.PodTemplateSpec{
                ObjectMeta: metav1.ObjectMeta{
                    Labels: map[string]string{
                        "app": name,
                    },
                },
                Spec: v1.PodSpec{
                    Containers: []v1.Container{
                        {
                            Name:  name,
                            Image: imageURL,
                            Ports: []v1.ContainerPort{
                                {
                                    ContainerPort: 5000,
                                },
                            },
                        },
                    },
                    ImagePullSecrets: []v1.LocalObjectReference{
                        {
                            Name: "regcred",
                        },
                    },
                },
            },
        },
    }
    _, err = clientset.AppsV1().Deployments("default").Create(context.TODO(), deployment,
metav1.CreateOptions{})
```

รูปที่ ก.9 สร้างไฟล์ Kubernetes Deployment Object สำหรับ Machine Learning Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้