

โครงการเลขที่ วศ.คพ. S007-2/67/2567

เรื่อง

สกรีนเนอร์: ระบบสำรวจถนนสำหรับการจัดการสินทรัพย์เมือง

โดย

นายชาญชล ภาณุสุนันต์ รหัส 640610626

นายณัฐพงษ์ เทพพิทักษ์ รหัส 640610634

นายธนภัทร สมสิทธิ์ รหัส 640610639

โครงการนี้

เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

ปีการศึกษา 2567

PROJECT No. CPE S007-2/67/2567

Screeetner: street scanner system for urban asset management

Charnchol Panusupanirun 640610626

Natthaphong Thepphithak 640610634

Thanapat Somsit 640610639

**A Project Submitted in Partial Fulfillment of Requirements
for the Degree of Bachelor of Engineering
Department of Computer Engineering
Faculty of Engineering
Chiang Mai University
2024**

หัวข้อโครงการ : สกรีนเนอร์: ระบบสำรวจถนนสำหรับการจัดการสินทรัพย์เมือง
: Sreetner: street scanner system for urban asset management
โดย : นายชาญชล ภาณุสุนันต์ รหัส 640610626
: นายณัฐพงษ์ เทพพิทักษ์ รหัส 640610634
: นายธนภัทร สมสิทธิ์ รหัส 640610639
ภาควิชา : วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา : รศ.ดร. สันติ พิทักษ์กัญญู
ปริญญา : วิศวกรรมศาสตรบัณฑิต
สาขา : วิศวกรรมคอมพิวเตอร์
ปีการศึกษา : 2567

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์
(รศ.ดร. สันติ พิทักษ์กัญญู)

คณะกรรมการสอบโครงการ

..... ประธานกรรมการ
(รศ.ดร. สันติ พิทักษ์กัญญู)

..... กรรมการ
(ผศ.ดร. กานต์ ปทานุคม)

..... กรรมการ
(ผศ.ดร. นวदनย์ คุณเลิศกิจ)

หัวข้อโครงการ : สกรีนเนอร์: ระบบสำรวจถนนสำหรับการจัดการสินทรัพย์เมือง
: Sreetner: street scanner system for urban asset management
โดย : นายชาญชล ภาณุสุนิรันดร์ รหัส 640610626
นายณัฐพงษ์ เทพพิทักษ์ รหัส 640610634
นายธนภัทร สมสิทธิ์ รหัส 640610639
ภาควิชา : วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา : รศ.ดร. สันติ พิทักษ์กัญญูร
ปริญญา : วิศวกรรมศาสตรบัณฑิต
สาขา : วิศวกรรมคอมพิวเตอร์
ปีการศึกษา : 2567

บทคัดย่อ

โครงการ Sreetner (Street Scanner System for Urban Asset Management) เป็นโครงการที่ถูกพัฒนาเพื่ออำนวยความสะดวกในการบริหารจัดการเกี่ยวกับการจัดเก็บภาษีป้าย ด้วยการใช้เทคโนโลยี Object Detection ในการตรวจจับป้ายที่จัดเก็บภาษีได้ โดยใช้แอปพลิเคชันบนโทรศัพท์มือถือในการบันทึกข้อมูลภาพในขณะเดียวกันก็จะมี server ที่คอยประมวลผลรูปภาพนั้น และสุดท้ายก็จะมีเว็บแอปพลิเคชันในการแสดงผลรายงานข้อมูลที่ได้จากการบันทึกจากบนโทรศัพท์มือถือ

Project Title : Sreetner: street scanner system for urban asset management
Name : Charnchol Panusupanirun 640610626
Natthaphong Thepphithak 640610634
Thanapat Somsit 640610639
Department : Computer Engineering
Project Advisor : Assoc. Prof. Santi Phithakkitnukoon, Ph.D.
Degree : Bachelor of Engineering
Program : Computer Engineering
Academic Year : 2024

ABSTRACT

The Sreetner project (Street Scanner System for Urban Asset Management) is a project developed to facilitate the management of taxable billboards utilizing Object Detection technology. This is achieved through the use of a mobile application on handheld devices to capture image data, while simultaneously having a server to process the image data. Lastly, there is a web application to display reports derived from the captured data.

สารบัญ

บทคัดย่อ	ข
Abstract	ค
สารบัญ	ง
สารบัญรูป	จ
1 บทนำ	1
1.1 ที่มาของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	1
1.3.1 ขอบเขตด้านฮาร์ดแวร์	1
1.3.2 ขอบเขตด้านซอฟต์แวร์	2
1.4 ประโยชน์ที่ได้รับ	2
1.5 เทคโนโลยีและเครื่องมือที่ใช้	2
1.5.1 เทคโนโลยีด้านซอฟต์แวร์	2
1.6 แผนการดำเนินงาน (แก้ม)	4
1.7 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม	5
2 ทฤษฎีที่เกี่ยวข้อง	6
2.1 You Only Look Once Object Detection Algorithm (YOLO)	6
2.2 Object Relational Mapping (ORM)	6
2.3 Hypertext Transfer Protocol (HTTP)	7
2.4 Docker	7
2.5 Interactive Website	8
2.6 Azure Public Cloud	8
2.7 Role-Based Access Control (RBAC)	8
2.8 Json Web Token (JWT)	9
2.9 Microservices Architecture	9
2.10 Cross Platform	10
3 โครงสร้างและขั้นตอนการทำงาน	11
3.1 การใช้งานพื้นฐาน	11
3.2 การออกแบบระบบพื้นฐานของโครงการ	11
3.2.1 Database Design	11
3.2.2 System Design	12
3.2.3 Web Application Flow Diagram	13
3.2.4 Mobile Application Flow Diagram	14
4 การประเมินระบบ	16
4.1 การประเมินประสิทธิภาพซอฟต์แวร์	16
4.2 การประเมินความพึงพอใจในการใช้งานระบบ	16
บรรณานุกรม	17

สารบัญรูป

2.1	YOLO Architecture	6
2.2	Object Relational Mapping	7
2.3	Docker Architecture	7
3.1	Database Design	12
3.2	System Design	12
3.3	Web Application Flow Diagram	13
3.4	Mobile Application Flow Diagram	14
3.5	Login Flow Diagram	14
3.6	Transition Flow Diagram	15

บทที่ 1

บทนำ

1.1 ที่มาของโครงการ

การจัดเก็บภาษีถือเป็นหนึ่งในรายได้หลักของประเทศไม่ว่าจะเป็นภาษีทางตรง อย่างเช่น ภาษีทางตรง ภาษีรายได้บุคคลธรรมดา ซึ่งจะจัดเก็บได้ จากประชาชนผู้มีเงินได้ทั่วไป ภาษีเงินได้นิติบุคคลซึ่งเป็นภาษีที่จัดเก็บได้จากเงินได้ของบริษัทหรือห้างหุ้นส่วนนิติบุคคล และยังมีภาษีทางอ้อม เช่น ภาษีมูลค่าเพิ่ม ภาษีสรรพสามิต ซึ่งเงินที่ได้จากการเก็บภาษีเหล่านี้นำไปให้รัฐบาลใช้ในการพัฒนาประเทศให้เจริญก้าวหน้า

ภาษีป้ายก็เป็นส่วนหนึ่งของรายได้ท้องถิ่นที่สามารถจัดเก็บได้โดยองค์กรปกครองส่วนท้องถิ่น โดยที่ภาษีลักษณะนี้เมื่อจัดเก็บได้แล้ว ทางท้องถิ่น ไม่จำเป็นต้องส่งคืนให้ทางรัฐ สามารถนำไปใช้จัดการบริหารพัฒนาภายในท้องถิ่นของตนเองได้ แต่ด้วยความสามารถในการจัดเก็บภาษีป้ายขององค์กรปกครองส่วนท้องถิ่นในแต่ละที่ ขึ้นอยู่กับปัจจัยหลาย ๆ อย่าง เช่น การที่ไม่สามารถรู้ได้ว่าป้ายที่สามารถจัดเก็บภาษีได้นั้นอยู่ที่ตำแหน่งใดในเขตปกครอง ซึ่งมีส่วนทำให้ประสิทธิภาพในการค้นหาป้ายภายในท้องถิ่นที่มีอยู่ทำได้ยุ่งยาก และเป็นขั้นตอนที่ต้องใช้กำลังคนในการตรวจสอบเป็นอย่างมาก ดังนั้นจากปัญหาในจุดที่กล่าวมาทำให้เกิดโครงการที่เป็นเครื่องมือที่ช่วยในการตรวจจับหาป้ายที่คาดว่าจะสามารถนำไปจัดเก็บภาษี และรายงานผลให้กับแต่ละองค์กรปกครองส่วนท้องถิ่นให้ไปจัดเก็บภาษีจากป้ายเหล่านี้

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อสร้างระบบครบวงจรในการรับวิดีโอแล้วประมวลผลตรวจจับหาป้ายอัตโนมัติ
2. เพื่อพัฒนาแอปพลิเคชันที่ใช้ในการอัปวิดีโอเพื่อที่จะส่งให้ระบบประมวลผล
3. เพื่อพัฒนาเครื่องมือในการรายงานป้ายที่ค้นพบภายในพื้นที่การปกครองส่วนท้องถิ่นสำหรับการไปจัดเก็บภาษี

1.3 ขอบเขตของโครงการ

1.3.1 ขอบเขตด้านฮาร์ดแวร์

กล้องถ่ายของโทรศัพท์แต่ละเครื่องจะมีคุณภาพและลักษณะการถ่ายที่แตกต่างกัน ซึ่งอาจส่งผลกระทบต่อตรวจจับวัตถุทำให้เวลานำรูปภาพที่ได้นำไปประมวลผลได้ผลลัพธ์ที่แตกต่างกัน ซึ่งโทรศัพท์ที่ได้ใช้ในการเก็บข้อมูลก็นำไปสร้างโมเดลมีอยู่ด้วยกัน 2 เครื่อง โดยมีคุณภาพของกล้องถ่ายรูปดังนี้

- Xiaomi 11T Pro ความละเอียด 108 ล้านพิกเซล
- Samsung Galaxy A50s ความละเอียด 48 ล้านพิกเซล

ความสูงของรถแต่ละคัน และมุมกล้องในการถ่ายภาพก็มีความแตกต่างกันไป ซึ่งอาจส่งผลกระทบต่อประสิทธิภาพในการตรวจจับวัตถุได้ไม่เท่ากัน โดยรถยนต์ที่ใช้ในการอัปวิดีโอสำหรับในการเทรนโมเดลเป็น Honda City 2024

Mobile application ที่เป็นส่วนของการส่งข้อมูลภาพไปยังเซิร์ฟเวอร์จำเป็นต้องเชื่อมต่ออินเทอร์เน็ตอยู่ตลอดเวลาทั้งการใช้งาน เนื่องจากต้องมีการส่งข้อมูลตลอดเวลา ทั้งนี้สื่อก็จะมีเรื่องของการใช้งานทรัพยากรแบตเตอรี่

เตอร์มากตามไปด้วย และในการของการแสดงผลที่เป็นเว็บแอปพลิเคชันจะสามารถใช้งานได้เฉพาะ ในคอมพิวเตอร์เท่านั้น

1.3.2 ขอบเขตด้านซอฟต์แวร์

ในการเก็บภาษีนั่นจะถูกแบ่งออกเป็นป้ายหลาย ๆ ประเภท อย่างเช่น ป้ายที่มีอักษรไทยล้วน ป้ายที่มีอักษรไทยปนกับอักษรต่างประเทศหรือปนกับภาพ และหรือเครื่องหมาย ป้ายที่ไม่มีอักษรไทย ไม่ว่าจะมีความหมายและหรือเครื่องหมายใด ๆ ซึ่งแต่ละประเภทนั้นจะมีอัตราการเก็บภาษีที่แตกต่างกันออกไป แต่ในการประมวลผลในเซิร์ฟเวอร์นั้นจะไม่มี การตรวจสอบและแบ่งแยกประเภทของป้าย และจะรวบรวมเป็นคลาสประเภทเดียวกันแทน อีกทั้งป้ายที่สามารถจัดเก็บภาษีได้บางประเภทมีลักษณะคล้ายกับป้ายบอกทางและป้ายจราจร จึงอาจทำให้ข้อผิดพลาดเกิดขึ้นในการตรวจจับในบางสถานการณ์

1.4 ประโยชน์ที่ได้รับ

1. ได้เครื่องมือที่ช่วยอำนวยความสะดวกในการเก็บภาษีนั่นให้มีประสิทธิภาพมากยิ่งขึ้น

1.5 เทคโนโลยีและเครื่องมือที่ใช้

1.5.1 เทคโนโลยีด้านซอฟต์แวร์

1. JetBrain IDEs เป็นชุดเครื่องมือพัฒนาโปรแกรมจาก JetBrains ที่ประกอบด้วย IDEs หลายตัว เช่น IntelliJ IDEA, PyCharm, และ WebStorm ซึ่งช่วยในการพัฒนาโปรแกรมในภาษาต่าง ๆ อย่างมีประสิทธิภาพ
2. Data Grip เป็นเครื่องมือจัดการฐานข้อมูลจาก JetBrains ที่ช่วยในการเชื่อมต่อและจัดการฐานข้อมูลหลายประเภท เช่น MySQL, PostgreSQL, และ SQLite ซึ่งช่วยให้นักพัฒนาสามารถทำงานกับฐานข้อมูลได้ง่ายขึ้น
3. Python เป็นภาษาโปรแกรมมิ่งที่มีความยืดหยุ่นสูงและสามารถนำมาใช้ในการพัฒนาโปรแกรมต่าง ๆ ได้หลากหลาย ซึ่งมีความเหมาะสมในการใช้งานในโครงการที่ต้องการประมวลผลข้อมูลที่ซับซ้อนและมีขนาดใหญ่ อย่างเช่น โมเดลการเรียนรู้เชิงลึก ที่พวกเราจะนำไปใช้กับการตรวจจับวัตถุ
4. Typescript คือภาษาคอมพิวเตอร์ที่ใช้ในการพัฒนาเว็บร่วมกับ HTML เพื่อให้เว็บมีลักษณะแบบไดนามิก หมายถึง เว็บสามารถตอบสนองกับ ผู้ใช้งานหรือแสดงเนื้อหาที่แตกต่างกันไปโดยจะอ้างอิงตามเว็บเบราว์เซอร์ที่ผู้เข้าชมเว็บใช้งานอยู่
5. Golang เป็นภาษาการเขียนโปรแกรมที่พัฒนาโดย Google ซึ่งมีประสิทธิภาพสูงและเหมาะสำหรับการพัฒนาแอปพลิเคชันที่ต้องการความเร็วและความเสถียร
6. Tusd เป็นเซิร์ฟเวอร์ที่ใช้ในการอัปโหลดไฟล์ขนาดใหญ่แบบต่อเนื่อง (resumable file uploads) ซึ่งช่วยให้การอัปโหลดไฟล์มีความเสถียรและไม่ขาดตอน
7. Azure Logic Apps เป็นบริการของ Microsoft Azure ที่ช่วยในการสร้างและจัดการเวิร์กโฟลว์อัตโนมัติสำหรับการรวมระบบและการประมวลผลข้อมูล

8. Azure Blob Storage เป็นบริการจัดเก็บข้อมูลแบบออบเจกต์ของ Microsoft Azure ที่ใช้ในการจัดเก็บข้อมูลขนาดใหญ่ เช่น ไฟล์วิดีโอและรูปภาพ
9. Azure App Instance เป็นบริการของ Microsoft Azure ที่ใช้ในการโฮสต์และจัดการแอปพลิเคชันบนคลาวด์
10. Azure Container Registry เป็นบริการของ Microsoft Azure ที่ใช้ในการจัดเก็บ จัดการ และเรียกใช้งานคอนเทนเนอร์
11. Azure Log Analytics workspace เป็นบริการของ Microsoft Azure ที่ใช้ในการจัดการ จัดเก็บ และวิเคราะห์ข้อมูลของระบบเช่น Log และ Metric
12. Azure Email Communication Service เป็นบริการของ Microsoft Azure ที่ใช้ในการส่งอีเมล และการสื่อสารอื่น ๆ ระหว่างระบบ
13. Flutter เป็นเฟรมเวิร์กที่พัฒนาโดย Google ที่ใช้ในการพัฒนาแอปพลิเคชันข้ามแพลตฟอร์ม (cross-platform) ทั้งบน iOS และ Android ด้วยโค้ดเบสเดียว
14. Next.js เป็นเฟรมเวิร์กที่ใช้ในการพัฒนาเว็บแอปพลิเคชันแบบเซิร์ฟเวอร์ไซด์เรนเดอร์ริง (SSR) และสแตติกไซต์เจเนอเรชัน (SSG) ซึ่งช่วยให้การพัฒนาเว็บมีประสิทธิภาพและความเร็วสูงขึ้น และยังมีฟีเจอร์ที่ช่วยในการทำ SEO ได้ดีขึ้น
15. YOLOv8 เป็นระบบที่ใช้ในการพัฒนาโมเดลตรวจจับวัตถุความเร็วสูงแบบเวลาจริง ด้วยการเรียนรู้เชิงลึกและการมองเห็นคอมพิวเตอร์
16. Figma เครื่องมือออกแบบเว็บไซต์ แอปพลิเคชัน โลโก้ และอื่น ๆ ทำให้นักออกแบบ UX/UI สะดวกมากขึ้น ผ่านการใช้ฟีเจอร์ต่าง ๆ ซึ่งมีจุดเด่นอยู่ที่การใช้งานบนได้ทุกระบบปฏิบัติการ และยังมี Community ที่ผู้ใช้สามารถแชร์ไฟล์งาน Prototype หรือ Plug-in ต่าง ๆ แล้วนำไปปรับใช้กับงานของตัวเองได้
17. Linux เป็นระบบปฏิบัติการ (Operating System) ที่เป็น Open Source และเป็นพื้นฐานบนหลักการทำงานของ Unix ซึ่งถูกพัฒนาขึ้นโดย Linus Torvalds ในปี ค.ศ. 1991 ซึ่งเป็นระบบปฏิบัติการที่เราจะนำมาใช้งาน
18. Kong เป็น API Gateway ที่ช่วยในการจัดการ API และการเชื่อมต่อระหว่างบริการต่าง ๆ ในระบบ ซึ่งช่วยเพิ่มความปลอดภัยและประสิทธิภาพในการทำงานของ API
19. Docker เป็นแพลตฟอร์มที่ใช้ในการสร้าง จัดส่ง และรันแอปพลิเคชันในคอนเทนเนอร์ ซึ่งช่วยให้การพัฒนาและการนำแอปพลิเคชันไปใช้งานมีความยืดหยุ่นและรวดเร็ว
20. Github Action เป็นเครื่องมือที่ใช้ในการทำ CI/CD (Continuous Integration/Continuous Deployment) บนแพลตฟอร์ม GitHub ซึ่งช่วยให้การทดสอบและการนำโค้ดไปใช้งานเป็นไปอย่างอัตโนมัติและมีประสิทธิภาพ
21. Draw.io เป็นเครื่องมือออนไลน์ที่ใช้ในการสร้างไดอะแกรมและแผนภาพต่าง ๆ เช่น แผนภาพการไหล (flowchart) และแผนภาพสถาปัตยกรรมระบบ ซึ่งช่วยให้การออกแบบและสื่อสารข้อมูลเป็นไปอย่างมีประสิทธิภาพ

22. Postman เป็นเครื่องมือที่ใช้ในการทดสอบ API ซึ่งช่วยให้นักพัฒนาสามารถส่งคำขอ (request) และ ดูผลลัพธ์ (response) ของ API ได้อย่างง่ายดาย
23. PostgreSQL เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS) ที่มีความเสถียรและมีประสิทธิภาพสูง ซึ่งใช้ในการจัดการและเก็บข้อมูลในโครงการ
24. MongoDB เป็นระบบจัดการฐานข้อมูลแบบ NoSQL ที่มีความยืดหยุ่นสูงและสามารถจัดการข้อมูลที่ไม่มีโครงสร้าง (unstructured data) ได้อย่างมีประสิทธิภาพ
25. Redis เป็นฐานข้อมูลแบบ key-value ที่ทำงานในหน่วยความจำ (in-memory) ซึ่งมีความเร็วสูงและเหมาะสำหรับการจัดเก็บข้อมูลที่ต้องการการเข้าถึงอย่างรวดเร็ว
26. Roboflow เป็นเครื่องมือที่สามารถใช้ทำการ Labeling ข้อมูล และสร้าง Dataset สำหรับการเทรนโมเดล Computer Vision ได้อย่างง่ายดาย

1.6 แผนการดำเนินงาน (แก๊)

ขั้นตอนการดำเนินงาน	ต.ค. 2566	พ.ย. 2566	ธ.ค. 2566	ม.ค. 2567	ก.พ. 2567	มี.ค. 2567	เม.ย. 2567	พ.ค. 2567	มิ.ย. 2567
เลือกอาจารย์ที่ปรึกษา และ เลือกหัวข้อโครงการ									
ออกแบบระบบการทำงานโดยคร่าว และ เครื่องมือที่ใช้ในการทำโครงการ									
ศึกษาข้อมูลเกี่ยวกับขอบเขตพื้นที่ที่จะใช้ทำโครงการ									
เก็บข้อมูลเพื่อใช้ในกระบวนการเทรนโมเดลสำหรับการตรวจจำวัตถุ									
คัดเลือก ข้อมูล และ พัฒนา โมเดล สำหรับ กระบวนการเทรนโมเดล									
ออกแบบระบบ									

ขั้นตอนการดำเนินงาน	ก.ค. 2567	ส.ค. 2567	ก.ย. 2567	ต.ค. 2567	พ.ย. 2567	ธ.ค. 2567	ม.ค. 2568	ก.พ. 2568
พัฒนากับ ทดสอบ แอปพลิเคชัน ที่ใช้ในการอัด วิดีโอ และ เว็บแอปพลิเคชันในการรายงานข้อมูล								
ดีพลอยระบบโดยรวม								
ตรวจสอบความถูกต้องสมบูรณ์หลังการนำไปใช้								
เขียนรายงาน								

1.7 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม

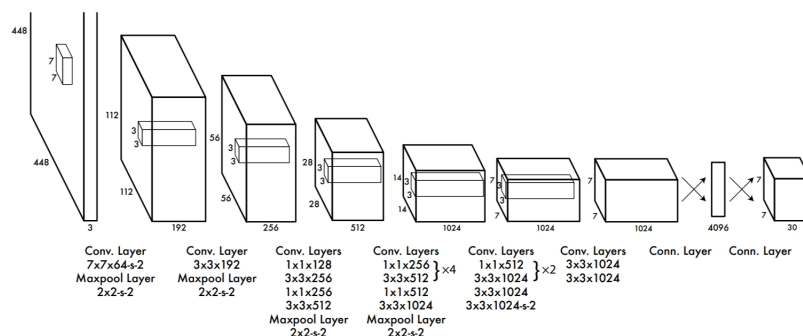
การพัฒนาระบบในการตรวจจับป้ายที่สามารถนำไปเก็บภาษีได้นั้น จะช่วยอำนวยความสะดวกให้สามารถจัดการได้ง่ายและสะดวกยิ่งขึ้น ซึ่งมีผลกระทบในด้านกฎหมายเพราะภาษีป้ายเป็นภาษีที่จัดเก็บจากป้ายที่แสดงชื่อ ยี่ห้อ หรือเครื่องหมายที่ใช้ในการประกอบ การค้า หรือประกอบกิจการอื่นเพื่อหารายได้ หรือ โฆษณาการค้า ซึ่งในส่วนของการเสียนั้นก็ขึ้นอยู่กับประเภทของป้ายตามที่กฎหมายกำหนด และรายได้ที่ได้จากการจัดเก็บภาษีก็จะถูกนำไปพัฒนาบ้านเมืองต่อไป

ทฤษฎีที่เกี่ยวข้อง

การดำเนินงานเริ่มต้นด้วยการศึกษาค้นคว้าทฤษฎีที่เกี่ยวข้อง หรืองานวิจัย/โครงการที่เคยมีผู้พัฒนาและนำเสนอไว้แล้ว ซึ่งเนื้อหาในบทนี้ก็จะเกี่ยวกับ การอธิบายถึงทฤษฎีที่นำไปประยุกต์ใช้กับโครงการนี้ เพื่ออ่านให้ผู้อ่านทำความเข้าใจกับตัวระบบของโครงการได้ง่ายขึ้น

2.1 You Only Look Once Object Detection Algorithm (YOLO)

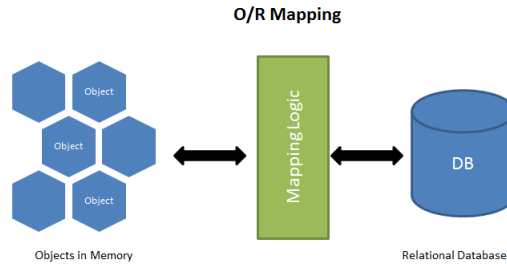
YOLO [1] เป็นอัลกอริทึมสำหรับการระบุบริเวณที่สนใจภายในภาพ และจำแนกประเภทของวัตถุบนแต่ละบริเวณแบบเวลาจริงเหมือนกับตัวจำแนกภาพปกติ โดยที่ภาพหนึ่งสามารถประกอบด้วยบริเวณที่สนใจหลายบริเวณ แล้วแต่ละบริเวณจะนำไปจำแนกวัตถุที่แตกต่างกันได้ ซึ่งทำให้เกิดความซับซ้อนสูงในการ จำแนกภาพระหว่างการตรวจจับวัตถุ ต่างจากอัลกอริทึมตรวจจับวัตถุทั่วไปที่จะใช้อัลกอริทึมแบบ Two-stage Object Detection YOLO นั้นจะใช้แบบ Single-shot Object Detection แทน ซึ่งใช้การสแกนภาพแต่ละภาพเพียงครั้งเดียวสำหรับการพยากรณ์ตำแหน่งของวัตถุที่ต้อง การจะตรวจจับ และเนื่องจากการประมวลผลภาพเพียงครั้งเดียวนั้น ส่งผลให้อัลกอริทึมดังกล่าวใช้ระยะเวลาในการประมวลผลต่ำ เหมาะกับการนำไปใช้แบบเวลาจริง แต่ก็แลกมากับข้อเสียที่ความแม่นยำในการตรวจจับภาพนั้นอาจไม่มากเท่าอัลกอริทึมแบบ Two-stage Object Detection โดยใช้เทคนิคการเรียนรู้แบบ Convolutional Neural Network ดังรูปที่ 2.1



รูปที่ 2.1: You Only Look Once Architecture

2.2 Object Relational Mapping (ORM)

Object-Relational Mapping [2] เป็นการสร้างการสัมพันธ์ระหว่างฐานข้อมูลแบบ **Relational** กับโครงสร้างข้อมูลแบบ **Object-Oriented** ตามรูปที่ 2.2 ในการพัฒนาซอฟต์แวร์ เช่น เว็บแอปพลิเคชัน โดยที่ไม่ต้องเขียน **SQL** โดยตรงแต่สามารถใช้ภาษาโปรแกรมเพื่อจัดการกับข้อมูลแทน ซึ่งสามารถป้องกันการโจมตีแบบ **SQL Injection** ได้ ในกรณีที่กำหนดให้มีการเปลี่ยนแปลงในโครงสร้างข้อมูล คุณสมบัติหรือโครงสร้างข้อมูลในฐานข้อมูลจะถูกปรับเปลี่ยนตามในโครงสร้างของ **Object** ในโปรแกรม เป็นฐานข้อมูลแบบเสมือนในโปรแกรม โดยที่การจัดเก็บข้อมูลยังคงเป็นแบบ **Relational** เหมือนเดิม โดยไม่ต้องใช้ **SQL Statements** โดยตรง



รูปที่ 2.2: Object Relational Mapping

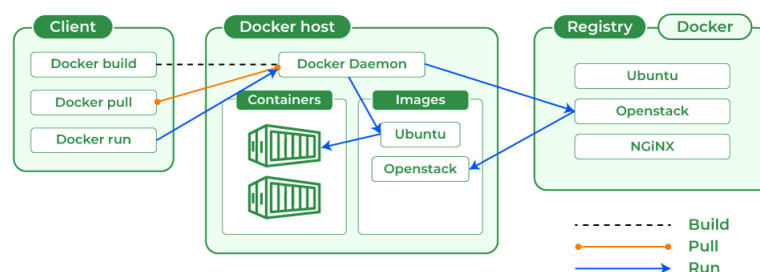
2.3 Hypertext Transfer Protocol (HTTP)

HTTP (Hypertext Transfer Protocol) เป็นโพรโตคอลสื่อสารที่ใช้ในการส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์บนเครือข่ายอินเทอร์เน็ต โดย HTTP มีหน้าที่เป็นตัวกลางในการร้องขอและส่งข้อมูลระหว่างเว็บไซต์ (web servers) และเบราว์เซอร์ (web browsers) หรือแอปพลิเคชันอื่น ๆ ที่ใช้เครือข่ายอินเทอร์เน็ต

- API (Application Programming Interface) เป็นชุดของกฎและโครงสร้างข้อมูลที่กำหนดโดยโปรแกรมคอมพิวเตอร์เพื่อให้แอปพลิเคชันอื่น ๆ สามารถสื่อสารและทำงานร่วมกันได้ ในเชิงพื้นฐาน API เป็นวิธีที่แอปพลิเคชันใช้เรียกใช้ฟังก์ชันหรือการบริการที่มาจากแหล่งข้อมูลหรือบริการ ซึ่งอาจเป็นเซิร์ฟเวอร์เว็บ ฐานข้อมูล หรือแหล่งข้อมูลอื่น ๆ โดยทั่วไป API จะรองรับการร้องขอและการตอบกลับโดยใช้พอร์มัตที่เป็นรูปแบบมาตรฐาน เช่น JSON (JavaScript Object Notation) หรือ XML (Extensible Markup Language)

2.4 Docker

Docker [3] เป็นเทคโนโลยีคอนเทนเนอร์แพลตฟอร์มที่ช่วยในการสร้างและทำงานร่วมกับคอนเทนเนอร์อย่างมีประสิทธิภาพ ด้วย Docker ผู้ใช้สามารถแยกแยะและแพ็คเกจแอปพลิเคชันพร้อมกับสิ่งที่เกี่ยวข้องทั้งหมด เช่น ไฟล์ ระบบปฏิบัติการ ไลบรารี และสิ่งอื่น ๆ ลงในคอนเทนเนอร์ได้อย่างเรียบง่าย โดยมีโครงสร้างการทำงานตามรูปที่ 2.4 ผู้ใช้สามารถสร้าง และรันคอนเทนเนอร์ได้โดยง่าย นอกจากนี้ Docker ยังช่วยลดปัญหาเกี่ยวกับสภาพแวดล้อมและการติดตั้งโปรแกรมที่ซับซ้อน ทำให้การพัฒนาและการทำงานของโปรแกรมมีประสิทธิภาพมากขึ้น



รูปที่ 2.3: Docker Architecture

2.5 Interactive Website

Interactive website [4] คือ เว็บไซต์ที่สามารถให้ผู้ใช้งาน communicate หรือ interact เช่น การแสดงความคิดเห็น การตอบโต้กับตัวเว็บ การได้รับผลจากการกระทำในเว็บ ในลักษณะที่เป็นมิตรต่อผู้ใช้ โดยปัจจุบันมักใช้ animation sound picture audio etc. ประกอบ เพื่อให้มีความสนุกสนานและเพิ่มการเข้าถึงได้ง่ายของผู้ใช้ ทั้งนี้อาจทำเพื่อเก็บข้อมูลหลังจากการใช้งานเว็บไซต์ได้อีกด้วย ซึ่งดีกว่าเว็บที่มีแต่ตัวอักษร หรือ การแสดงผลเฉย ๆ ที่ได้รับข้อมูลทางฝ่ายเดียวอย่างแน่นอน

2.6 Azure Public Cloud

Azure Public Cloud [5] เป็นแพลตฟอร์มคลาวด์คอมพิวติ้งที่พัฒนาโดย Microsoft ซึ่งให้บริการหลากหลายประเภท เช่น การประมวลผล (compute), การจัดเก็บข้อมูล (storage), การเครือข่าย (networking), และการวิเคราะห์ข้อมูล (analytics) โดย Azure Public Cloud ช่วยให้องค์กรและนักพัฒนาสามารถสร้างและจัดการแอปพลิเคชันได้อย่างมีประสิทธิภาพและยืดหยุ่น

Azure Public Cloud มีบริการที่หลากหลาย เช่น:

- **Azure Virtual Machines (VMs):** บริการที่ช่วยให้ผู้ใช้สามารถสร้างและจัดการเครื่องเสมือนบนคลาวด์ได้
- **Azure App Services:** บริการที่ช่วยในการพัฒนาและโฮสต์เว็บแอปพลิเคชันและ API บนคลาวด์
- **Azure Storage:** บริการจัดเก็บข้อมูลที่มีความยืดหยุ่นและสามารถขยายขนาดได้ตามความต้องการ
- **Azure SQL Database:** บริการฐานข้อมูลเชิงสัมพันธ์ที่มีความเสถียรและปลอดภัย
- **Azure Kubernetes Service (AKS):** บริการที่ช่วยในการจัดการและปรับใช้คอนเทนเนอร์โดยใช้ Kubernetes

Azure Public Cloud ยังมีความสามารถในการรองรับการทำงานร่วมกับเครื่องมือและเทคโนโลยีอื่น ๆ เช่น Docker, Kubernetes, และ DevOps ซึ่งช่วยให้การพัฒนาและการจัดการแอปพลิเคชันเป็นไปอย่างราบรื่นและมีประสิทธิภาพ

2.7 Role-Based Access Control (RBAC)

Role-Based Access Control (RBAC) [6] เป็นวิธีการจัดการสิทธิ์การเข้าถึงระบบที่กำหนดสิทธิ์การเข้าถึงตามบทบาทของผู้ใช้ในองค์กร โดย RBAC ช่วยให้การจัดการสิทธิ์การเข้าถึงเป็นไปอย่างมีประสิทธิภาพและปลอดภัยมากขึ้น เนื่องจากสิทธิ์การเข้าถึงจะถูกกำหนดตามบทบาทที่ผู้ใช้มีในองค์กร ไม่ใช่ตามผู้ใช้แต่ละคน

RBAC มีองค์ประกอบหลักดังนี้:

- **Roles (บทบาท):** บทบาทที่กำหนดสิทธิ์การเข้าถึงตามหน้าที่หรือความรับผิดชอบของผู้ใช้ในองค์กร
- **Permissions (สิทธิ์):** สิทธิ์การเข้าถึงที่กำหนดให้กับบทบาทต่าง ๆ เช่น การอ่าน การเขียน หรือการลบข้อมูล

- **Users (ผู้ใช้):** ผู้ใช้ที่ได้รับการกำหนดบทบาทและสิทธิ์การเข้าถึงตามบทบาทนั้น ๆ
- **Sessions (เซสชัน):** การเชื่อมต่อระหว่างผู้ใช้และระบบที่กำหนดสิทธิ์การเข้าถึงตามบทบาทของผู้ใช้ในช่วงเวลาหนึ่ง

RBAC ช่วยลดความซับซ้อนในการจัดการสิทธิ์การเข้าถึงและเพิ่มความปลอดภัยในการเข้าถึงระบบ โดยเฉพาะในองค์กรที่มีผู้ใช้งานจำนวนมากและมีการเปลี่ยนแปลงบทบาทของผู้ใช้บ่อยครั้ง

2.8 Json Web Token (JWT)

JSON Web Token (JWT) [7] เป็นมาตรฐานเปิด (RFC 7519) ที่กำหนดวิธีการสร้างโทเค็นที่ใช้ในการส่งข้อมูลระหว่างฝ่ายต่าง ๆ อย่างปลอดภัยในรูปแบบของ JSON โดย JWT ประกอบด้วยสามส่วนหลัก ๆ คือ Header, Payload และ Signature ซึ่งถูกเข้ารหัสและเชื่อมต่อกันด้วยจุด (.) เพื่อสร้างโทเค็นที่สมบูรณ์ JWT มีการใช้งานที่หลากหลาย เช่น:

- **Authentication (การยืนยันตัวตน):** JWT ถูกใช้ในการยืนยันตัวตนของผู้ใช้ในระบบ โดยโทเค็นจะถูกส่งไปยังผู้ใช้หลังจากที่ผู้ใช้ทำการล็อกอินสำเร็จ และผู้ใช้จะต้องส่งโทเค็นนี้กลับมาในคำขอถัดไปเพื่อยืนยันตัวตน
- **Information Exchange (การแลกเปลี่ยนข้อมูล):** JWT สามารถใช้ในการส่งข้อมูลระหว่างฝ่ายต่าง ๆ อย่างปลอดภัย เนื่องจากข้อมูลในโทเค็นถูกเข้ารหัสและสามารถตรวจสอบความถูกต้องได้

JWT มีข้อดีหลายประการ เช่น:

- **Compact (กะทัดรัด):** โทเค็นมีขนาดเล็กและสามารถส่งผ่าน URL, POST parameters หรือใน HTTP headers ได้อย่างง่ายดาย
- **Self-contained (บรรจุข้อมูลในตัวเอง):** โทเค็นประกอบด้วยข้อมูลที่จำเป็นทั้งหมด เช่น ข้อมูลผู้ใช้และสิทธิ์การเข้าถึง ทำให้ไม่จำเป็นต้องเข้าถึงฐานข้อมูลในทุกคำขอ

2.9 Microservices Architecture

Microservices Architecture [8] เป็นสถาปัตยกรรมในการออกแบบระบบซอฟต์แวร์ที่แบ่งแอปพลิเคชันออกเป็นบริการขนาดเล็ก ๆ ที่สามารถพัฒนา ทดสอบ และปรับใช้ได้อย่างอิสระ โดยแต่ละบริการจะทำงานร่วมกันผ่าน API และสามารถสื่อสารกันผ่านโปรโตคอลต่าง ๆ เช่น HTTP หรือ AMQP

ข้อดีของ Microservices Architecture ได้แก่:

- **Scalability (การขยายขนาด):** สามารถขยายขนาดบริการแต่ละตัวได้อย่างอิสระตามความต้องการของระบบ
- **Flexibility (ความยืดหยุ่น):** สามารถใช้เทคโนโลยีและภาษาในการเขียนโปรแกรมที่แตกต่างกันในแต่ละเซอร์วิสโดยที่ไม่มีผลกระทบต่อระบบโดยรวม
- **Resilience (ความทนทาน):** หากบริการใดบริการหนึ่งล้มเหลว จะไม่ส่งผลกระทบต่อบริการอื่น ๆ ในระบบ
- **Continuous Deployment (การปรับใช้อย่างต่อเนื่อง):** สามารถปรับใช้และอัปเดตบริการแต่ละตัวได้อย่างรวดเร็วและง่ายดาย

2.10 Cross Platform

Cross Platform [9] เป็นแนวคิดในการพัฒนาแอปพลิเคชันที่สามารถทำงานได้บนหลายแพลตฟอร์ม เช่น iOS, Android, และ Windows โดยใช้โค้ดเบสเดียวกัน ซึ่งช่วยลดเวลาและค่าใช้จ่ายในการพัฒนาและบำรุงรักษาแอปพลิเคชัน

ข้อดีของ Cross Platform ได้แก่:

- **Code Reusability (การใช้โค้ดซ้ำ):** สามารถใช้โค้ดเบสเดียวกันในการพัฒนาแอปพลิเคชันสำหรับหลายแพลตฟอร์ม
- **Cost Efficiency (ประหยัดค่าใช้จ่าย):** ลดค่าใช้จ่ายในการพัฒนาและบำรุงรักษาแอปพลิเคชัน
- **Faster Time-to-Market (เวลาสู่ตลาดเร็วขึ้น):** สามารถเปิดตัวแอปพลิเคชันได้เร็วขึ้นเนื่องจากไม่ต้องพัฒนาแยกกันสำหรับแต่ละแพลตฟอร์ม
- **Consistency (ความสม่ำเสมอ):** ให้ประสบการณ์การใช้งานที่สม่ำเสมอบนทุกแพลตฟอร์ม

เครื่องมือที่นิยมใช้ในการพัฒนาแอปพลิเคชันแบบ Cross Platform ได้แก่ Flutter, React Native, และ Xamarin

บทที่ 3

โครงสร้างและขั้นตอนการทำงาน

3.1 การใช้งานพื้นฐาน

ในส่วนของโมบายล์แอปพลิเคชัน เป็นเครื่องมือที่จะจำเป็นต้องใช้งานกล้องและบันทึกพิกัดตำแหน่งทาง GPS อยู่ตลอดเวลาเพื่อทำการส่งรูปภาพ พร้อมกับพิกัดตำแหน่ง แล้วนำไปประมวลผลในเซิร์ฟเวอร์ที่ได้ออกแบบเอาไว้ โดยที่เซิร์ฟเวอร์ดังกล่าวจะทำการประมวลผลรูปภาพเพื่อหาป้ายโฆษณาที่สามารถจัดเก็บภาษีได้ และหลังจากนั้นก็จะจัดเก็บลงฐานข้อมูลต่อไป

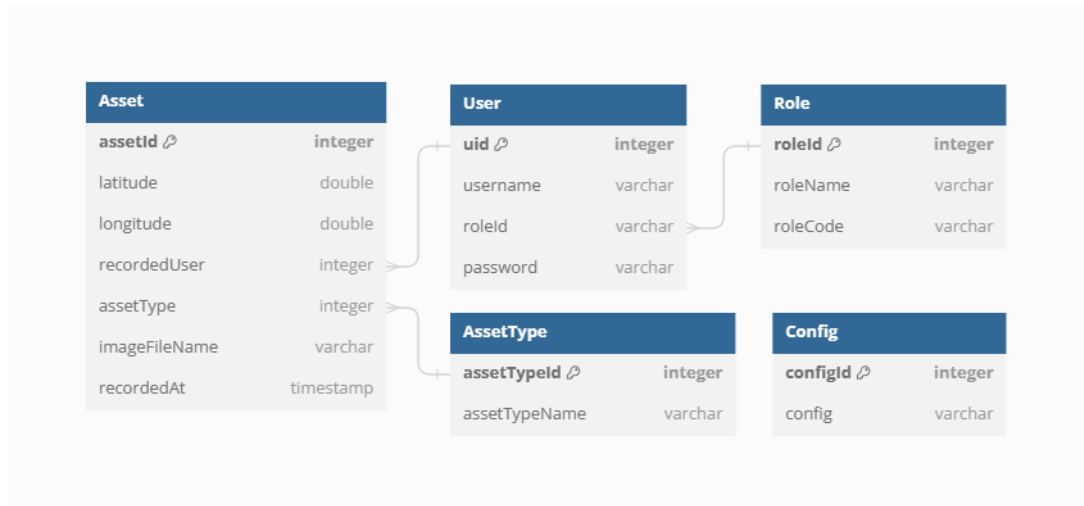
ในส่วนของเว็บแอปพลิเคชัน จะเป็นส่วนของการแสดงผลข้อมูลที่ได้บันทึกมาได้จากส่วนของโมบายล์แอปพลิเคชัน โดยจะแสดงในรูปแบบของแผนที่ในแผนที่ คล้าย ๆ กับการปักหมุดของ Google map โดยที่ในแต่ละหมุดสามารถกดเพื่อดูรายละเอียดต่าง ๆ ได้ เช่น พิกัดของหมุดนั้น และลักษณะรูปป้ายในตำแหน่งนั้นๆที่ได้บันทึกมาจากโมบายล์แอปฯ

3.2 การออกแบบระบบพื้นฐานของโครงการ

3.2.1 Database Design

ประกอบด้วย 4 ตารางดังรูปที่ 3.1 ได้แก่

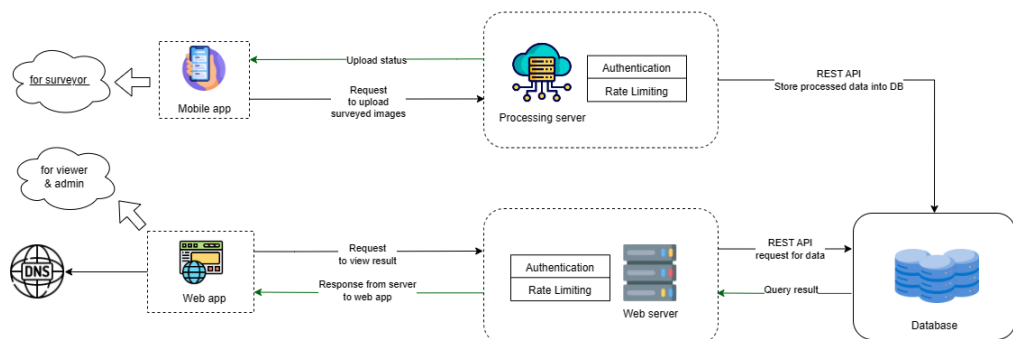
1. User table: เนื่องจากระบบต้องมีการ Authentication เพื่อเข้าใช้งานไม่ว่าจะเป็นทั้งส่วนของ โมบายล์แอปฯ หรือเว็บแอปฯ ดังนั้นตารางนี้จึงจะใช้เก็บข้อมูลพื้นฐานต่าง ๆ ที่จำเป็นต่อการยืนยันตัวตนทั้งหมด
2. Role table: ใช้ในการเก็บบทบาททั้งหมดที่มีของระบบ เช่น ผู้ดูแลระบบ ผู้สำรวจ และอื่น ๆ
3. Asset table: ใช้ในการเก็บข้อมูลที่ได้รับมาจาก โมบายล์แอปฯ ไม่ว่าจะเป็นตำแหน่งของรูป ชื่อของรูป และประเภทของ asset ที่ตรวจจับได้
4. Asset type table: ใช้ในการเก็บประเภทของ asset ต่างๆที่ระบบสามารถตรวจจับได้
5. Config table: ใช้เก็บการตั้งค่าพื้นฐานต่างๆเช่น ขอบเขตของแผนที่



รูปที่ 3.1: Overall Database Design

3.2.2 System Design

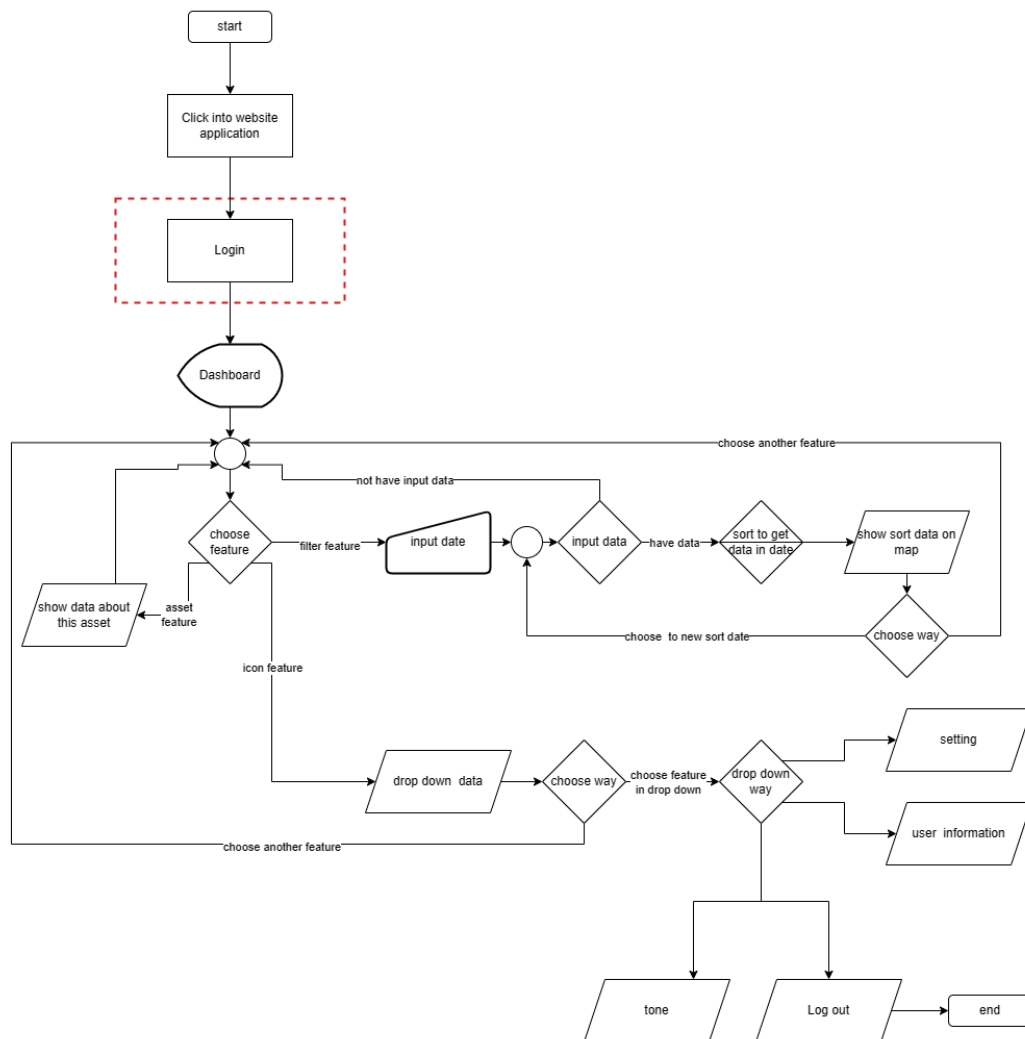
จากรูป จะอธิบายถึงโครงสร้างระบบของโครงการงานนี้ในรูปแบบ Flow diagram เพื่อให้เข้าใจถึงโครงสร้างการทำงานพอสังเขป โดยที่ซอฟต์แวร์จะประกอบด้วย 3 ส่วนหลัก ๆ ได้ Mobile application ซึ่งจะทำงานตามรูปที่ 3.2 Web application ซึ่งจะทำงานตามรูปที่ 3.3 และ Processing server โดยที่ลักษณะการทำงานร่วมกันระหว่างทั้งสามส่วนประกอบ แสดงตามรูปที่ 3.1



รูปที่ 3.2: Overall System Design

3.2.3 Web Application Flow Diagram

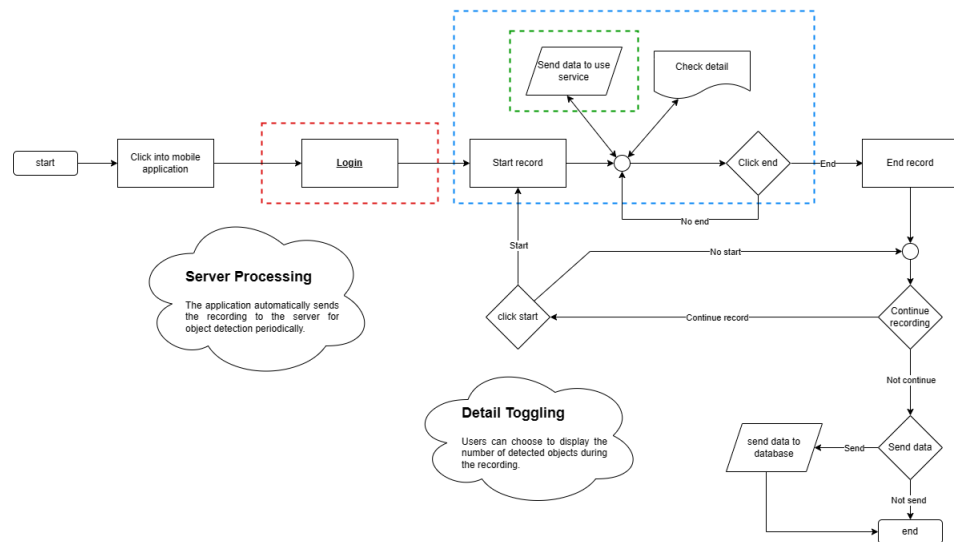
จากรูปที่ 3.3 จะอธิบายถึงลำดับการทำงานของเว็บแอปพลิเคชันในรูปแบบของ flow diagram เพื่อให้เข้าใจในลำดับการทำงานอย่างพอสังเขป พอหลังจากที่ได้เข้าระบบสู่หน้า dashboard จะมีตัวเลือกที่สามารถทำได้อยู่ 3 อย่างคือ filter เป็นการคัดกรองข้อมูลให้เหลือเพียงข้อมูลในช่วงเวลาที่เรต้องการ asset เป็นการกดที่รูปภาพเพื่อที่จะดูข้อมูลเกี่ยวกับ asset ดังกล่าว และ icon เป็นส่วนที่จะแสดงตัวเลือกเพิ่มเติมอีก 4 ทางเพื่อให้เราสามารถเลือกเข้าไปยังหน้าอื่นต่อไปได้



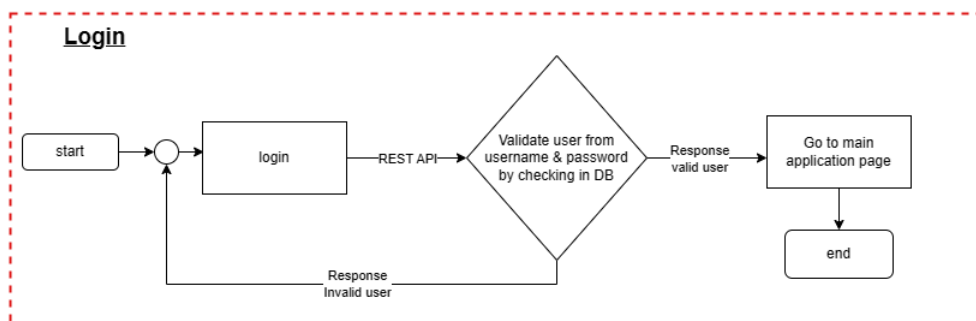
รูปที่ 3.3: Web Application Flow Diagram

3.2.4 Mobile Application Flow Diagram

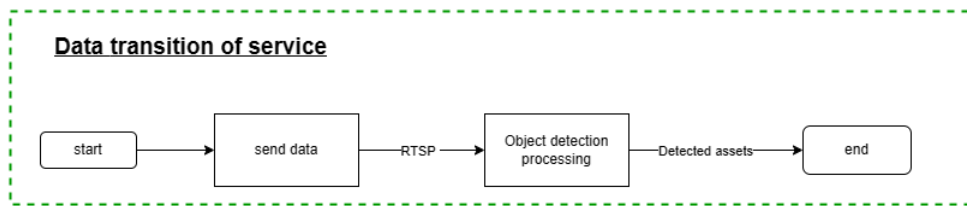
รูปที่ 3.4 จะอธิบายถึงลำดับการทำงานของแอปพลิเคชันในรูปแบบของ flow diagram เพื่อให้เข้าใจในลำดับการทำงานอย่างพอสังเขป โดยพอผู้ใช้จะเริ่มเข้าใช้งานแอปพลิเคชัน ผู้ใช้จะต้องผ่านการเข้าสู่ระบบ ซึ่งมีขั้นตอนการทำงานดังรูปที่ 3.5 เพื่อเป็นการยืนยันตัวตน หลังจากที่ได้เข้าสู่แอปพลิเคชันเรียบร้อยแล้ว ผู้ใช้งานจะสามารถเริ่มสตรีมวิดีโอเพื่อทำการส่งรูปภาพในช่วงเวลาหนึ่งพร้อมแนบตำแหน่งพิกัดในช่วงเวลาดังกล่าวไปยังเซิร์ฟเวอร์ที่ได้จัดเตรียมเอาไว้อยู่ตลอดเวลาที่ทำการสตรีม เพื่อให้ทางเซิร์ฟเวอร์ทำการคืนค่าออกมาว่าในตำแหน่งนี้จะมี asset อยู่เท่าไร โดยจะมีขั้นตอนการทำงานดังรูปที่ 3.6 ซึ่งสิ่งที่คืนค่ามาทุกคร้ังนั้นจะเอามาจัดเก็บเอาไว้บนมือถือชั่วคราวและยังไม่ได้ทำการบันทึกข้อมูลลงไปในฐานข้อมูลเพื่อให้ผู้ใช้งานสามารถดูข้อมูลได้ตลอดเวลาว่าปัจจุบันมี asset อยู่เท่าไรจนจบการทำงาน และในตอนท้ายของการทำงานผู้ใช้สามารถที่จะเลือกได้ว่าจะทำการสตรีมต่ออีกครั้งหรือไม่ หากไม่ทำการสตรีมต่อ ผู้ใช้งานต้องเลือกว่าจะทำการส่งข้อมูลทั้งหมดที่ได้มานั้นไปยังฐานข้อมูลหรือไม่



รูปที่ 3.4: Mobile Application Flow Diagram



รูปที่ 3.5: Login Flow Diagram



รูปที่ 3.6: Transition Flow Diagram

บทที่ 4

การประเมินระบบ

4.1 การประเมินประสิทธิภาพซอฟต์แวร์

ทดสอบประสิทธิภาพซอฟต์แวร์โดยจะมีการแบ่งส่วนในการทดสอบออกเป็น ส่วน ๆ เพื่อให้รู้ว่าในแต่ละส่วนของซอฟต์แวร์ของเรานั้น ทำงานได้อย่างมีประสิทธิภาพหรือไม่ จึงสามารถแบ่งออกการประเมินได้เป็นดังนี้

1. **Classification model** - เป็นการทดสอบเพื่อประเมินและตรวจสอบความเร็วในการประมวลผลเพื่อทำการ **classify** ว่า **object** ใดเป็นป้ายที่สามารถจัดเก็บภาษีได้ รวมถึงในเรื่องของความแม่นยำในการ **classify**
2. **Response time** - เป็นการทดสอบเพื่อประเมินในเรื่องของความเร็วในการรับส่งข้อมูลระหว่าง **client** กับ **application server**

4.2 การประเมินความพึงพอใจในการใช้งานระบบ

ทดสอบความพึงพอใจในการใช้งานจะมีการแบ่งออกเป็นสองส่วน คือ ส่วนของแอปพลิเคชันในโทรศัพท์มือถือ กับ ส่วนของเว็บแอปพลิเคชัน โดยจะมีเกณฑ์การให้คะแนนอยู่ที่ 1 ถึง 5 โดยจะมีการให้คะแนนในเรื่องดังต่อไปนี้

1. ความง่ายต่อการใช้งานของแอปพลิเคชัน
2. ความสะดวกในการใช้งานในตอนเริ่มต้นของแอปพลิเคชัน
3. ความดึงดูดในการใช้งานของแอปพลิเคชัน
4. ประโยชน์ที่มีของแอปพลิเคชัน

โดยที่ทั้ง 4 ข้อเป็นพิจารณาจากแนวคิดตาม **The Four Elements of User Experience [10]** ที่ประกอบไปด้วย

1. **Usability** ความง่ายในการใช้งาน เกี่ยวข้องกับความสามารถในการใช้งาน รวมไปถึงความเหมาะสมการใช้งานกับผู้ใช้งาน
2. **Adaptability** ความสามารถในการปรับตัว กล่าวถึงระดับความยากง่ายของการใช้งานตั้งแต่จุดเริ่มต้นจนถึงจุดสิ้นสุดของระบบ โดยที่ผู้ใช้งานสามารถใช้งานได้อย่างคล่องแคล่ว
3. **Desirability** ความพึงพอใจ คือเมื่อใช้งานแล้วผู้ได้รับประสบการณ์ที่ดีในจากใช้งานของระบบ
4. **Value** คุณค่าของระบบ คือระบบที่ผู้ใช้เข้ามาใช้งานมีความสอดคล้องกับความต้องการของผู้ใช้

บรรณานุกรม

- [1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” 2015, cite arxiv:1506.02640. [Online]. Available: <http://arxiv.org/abs/1506.02640>
- [2] M. Erica. (2019) Object relational mapping. Medium. Accessed: February 25, 2024. [Online]. Available: <https://medium.com/@emccul13/object-relational-mapping-9d84807f5536>
- [3] *Docker*, Docker Inc., 2020, version 18.03. Accessed: February 25, 2024. [Online]. Available: <https://docs.docker.com/manuals/>
- [4] (2021) The benefits an interactive website design offers your business. MEWS Agency. Accessed: February 25, 2024. [Online]. Available: <https://mews.agency/blog-post/the-benefits-an-interactive-website-design-offers-your-business/>
- [5] Azure devops. Microsoft. Accessed: February 25, 2024. [Online]. Available: <https://azure.microsoft.com/en-us/services/devops/>
- [6] (2014) Role-based access control (rbac). NIST. Accessed: February 25, 2024. [Online]. Available: <https://csrc.nist.gov/publications/detail/sp/800-162/final>
- [7] Json web token (jwt). JWT.io. Accessed: February 25, 2024. [Online]. Available: <https://jwt.io/introduction/>
- [8] Microservices architecture. AWS. Accessed: February 25, 2024. [Online]. Available: <https://aws.amazon.com/microservices/>
- [9] Cross platform development. Techopedia. Accessed: February 25, 2024. [Online]. Available: <https://www.techopedia.com/definition/3414/cross-platform-development>
- [10] (2024) Ux design elements: How to create an amazing user experience. Direct Images. Accessed: February 28, 2024. [Online]. Available: <https://directimages.com/insites/ux-design-elements-create-user-experience/>