

โครงการเลขที่ วศ.คพ. S007-2/66/2567

เรื่อง

สกรีนเนอร์: ระบบสำรวจถนนสำหรับการจัดการสินทรัพย์เมือง

โดย

นายธนภัทร สมสิทธิ์ รหัส 640610639

โครงการนี้

เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

ปีการศึกษา 2567

PROJECT No. CPE S007-2/66/2567

Screenner: street scanner system for urban asset management

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 : สกรีทเนอร์: ระบบสำรวจถนนสำหรับการจัดการสินทรัพย์เมือง
: Sreetner: street scanner system for urban asset management
โดย : นายธนภัทร สมสิทธิ์ รหัส 640610639
ภาควิชา : วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา : รศ.ดร. สันติ พิทักษ์กิจนุญ
ปริญญา : วิศวกรรมศาสตรบัณฑิต
สาขา : วิศวกรรมคอมพิวเตอร์
ปีการศึกษา : 2567

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงการนี้เป็นส่วน
หนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์
(รศ.ดร. สันติ พิทักษ์กิจนุญ)

คณะกรรมการสอบโครงการ

..... ประธานกรรมการ
(รศ.ดร. สันติ พิทักษ์กิจนุญ)

..... กรรมการ
(ผศ.ดร. กานต์ ปทานุคม)

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

หัวข้อโครงการ : สกรีนเนอร์: ระบบสำรวจถนนสำหรับการจัดการสินทรัพย์เมือง
: Sreetner: street scanner system for urban asset management
โดย : 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 : 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 | 3 |
| 1.7 บทบาทและความรับผิดชอบ | 3 |
| 1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม | 4 |
| 2 ทฤษฎีที่เกี่ยวข้อง | 5 |
| 2.1 You Only Look Once Object Detection Algorithm (YOLO) | 5 |
| 2.2 Object Relational Mapping (ORM) | 5 |
| 2.3 Model–View–Controller design pattern (MVC) | 6 |
| 2.4 Hypertext Transfer Protocol (HTTP) | 7 |
| 2.5 Docker | 7 |
| 2.6 Interactive Website | 7 |
| 3 โครงสร้างและขั้นตอนการทำงาน | 8 |
| 3.1 การใช้งานพื้นฐาน | 8 |
| 3.2 การออกแบบระบบพื้นฐานของโครงการ | 8 |
| 3.2.1 Database Design | 8 |
| 3.2.2 System Design | 9 |
| 3.2.3 Web Application Flow Diagram | 10 |
| 3.2.4 Mobile Application Flow Diagram | 11 |
| 4 ข้อมูลทั่วไปของบริษัท | 13 |
| 4.1 สินค้าและบริการของบริษัท | 13 |
| 4.2 สถานะงบการเงิน | 13 |
| 5 การประเมินระบบ | 14 |
| 5.1 การประเมินประสิทธิภาพซอฟต์แวร์ | 14 |
| 5.2 การประเมินความพึงพอใจในการใช้งานระบบ | 14 |

สารบัญรูป

| | | |
|-----|---|----|
| 2.1 | YOLO Architecture | 5 |
| 2.2 | Object Relational Mapping | 6 |
| 2.3 | Model-View-Controller | 6 |
| 2.4 | Docker Architecture | 7 |
| 3.1 | Database Design | 9 |
| 3.2 | System Design | 9 |
| 3.3 | Web Application Flow Diagram | 10 |
| 3.4 | Mobile Application Flow Diagram | 11 |
| 3.5 | Login Flow Diagram | 11 |
| 3.6 | Transition Flow Diagram | 12 |

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. เพื่อพัฒนาแอปพลิเคชันที่ใช้ในการอัดวิดีโอเพื่อที่จะส่งให้ระบบประมวลผล
3. เพื่อพัฒนาเครื่องมือในการรายงานป้ายที่ค้นพบภายในพื้นที่การปกครองส่วนท้องถิ่นสำหรับการไปจัดเก็บภาษี

1.3 ขอบเขตของโครงการ

1.3.1 ขอบเขตด้านฮาร์ดแวร์

กล้องถ่ายของโทรศัพท์แต่ละเครื่องจะมีคุณภาพและลักษณะการถ่ายที่แตกต่างกัน ซึ่งอาจส่งผลต่อการตรวจจับวัตถุทำให้เวลานำรูปภาพที่ได้นำไปประมวลผลได้ผลลัพธ์ที่แตกต่างกัน ซึ่งโทรศัพท์ที่ได้ใช้ในการเก็บข้อมูลก็นำไปสร้างโมเดลมีอยู่ด้วยกัน 2 เครื่อง โดยมีคุณภาพของกล้องถ่ายรูปดังนี้

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

ความสูงของรถแต่ละคัน และมุมกล้องในการถ่ายภาพก็มีความแตกต่างกันไป ซึ่งอาจส่งผลให้ประสิทธิภาพในการตรวจจับวัตถุได้ไม่เท่ากัน โดยรถยนต์ที่ใช้ในการอัดวิดีโอสำหรับในการเทรนโมเดลเป็น 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 แต่ในการประมวลผลในเซิร์ฟเวอร์นั้นจะไม่มีตรวจสอบและแบ่งแยกประเภทของป้าย และจะรวบรวมเป็นคลาสประเภทเดียวกันแทน

1.4 ประโยชน์ที่ได้รับ

1. ได้เครื่องมือที่ช่วยอำนวยความสะดวกในการเก็บภาษีนั่นให้มีประสิทธิภาพมากยิ่งขึ้น

1.5 เทคโนโลยีและเครื่องมือที่ใช้

1.5.1 เทคโนโลยีด้านซอฟต์แวร์

1. Visual Studio Code หรือ VSCode เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ดจากค่ายไมโครซอฟท์ มีการพัฒนาออกมาในรูปแบบของ OpenSource ซึ่ง Visual Studio Code นั้นเหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานข้ามแพลตฟอร์ม สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือส่วนขยายต่าง ๆ ให้เลือกใช้อย่างมากมาย
2. Python เป็นภาษาโปรแกรมมิ่งที่มีความยืดหยุ่นสูงและสามารถนำมาใช้ในการพัฒนาโปรแกรมต่าง ๆ ได้หลากหลาย ซึ่งมีความเหมาะสมในการใช้งานในโครงการที่ต้องการประมวลผลข้อมูลที่ซับซ้อนและมีขนาดใหญ่ อย่างเช่น โมเดลการเรียนรู้เชิงลึก ที่พวกเราจะนำไปใช้กับการตรวจจำวัตถุ และใช้เป็นระบบการส่งผ่านข้อมูล
3. Typescript คือภาษาคอมพิวเตอร์ที่ใช้ในการพัฒนาเว็บร่วมกับ HTML เพื่อให้เว็บมีลักษณะแบบไดนามิก หมายถึง เว็บสามารถตอบสนองกับ ผู้ใช้งานหรือแสดงเนื้อหาที่แตกต่างกันไปโดยจะอ้างอิงตามเว็บเบราว์เซอร์ที่ผู้เข้าชมเว็บใช้งานอยู่
4. React.js เป็นไลบรารีจาวาสคริปต์ที่เป็นที่ยอมรับกันว่าเป็นตัวช่วยให้สามารถสร้าง UI (User Interface หรือองค์ประกอบของเว็บที่เชื่อมต่อ กับผู้ใช้งานโดยตรง) ได้แม่นยำและรวดเร็วมากยิ่งขึ้น และส่งผลให้การแสดงผลมีความเป็นระบบคงเส้นคงวามากขึ้นไปพร้อม ๆ กัน
5. YOLOv8 เป็นระบบที่ใช้ในการพัฒนาโมเดลตรวจจำวัตถุความเร็วสูงแบบเวลาจริง ด้วยการเรียนรู้เชิงลึกและการมองเห็นคอมพิวเตอร์
6. Figma เครื่องมือออกแบบเว็บไซต์ แอปพลิเคชัน โลโก้ และอื่น ๆ ทำให้นักออกแบบ UX/UI สะดวกมากขึ้น ผ่านการใช้ฟีเจอร์ต่าง ๆ ซึ่งมีจุดเด่นอยู่ที่การใช้งานบนได้ทุกระบบปฏิบัติการ และยังมี Community ที่ผู้ใช้สามารถแชร์ไฟล์งาน Prototype หรือ Plug-in ต่าง ๆ แล้วนำไปปรับใช้กับงานของตัวเองได้

7. Linux เป็นระบบปฏิบัติการ (Operating System) ที่เป็น Open Source และเป็นพื้นฐานบนหลักการทำงานของ Unix ซึ่งถูกพัฒนาขึ้นโดย Linus Torvalds ในปี ค.ศ. 1991 ซึ่งเป็นระบบปฏิบัติการที่เราจะนำมาใช้งาน

1.6 แผนการดำเนินงาน

| 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 | | | | | | | | | |
| เก็บข้อมูลเพื่อใช้ในการกระบวนการเทรนโมเดลสำหรับการตรวจจับวัตถุ | | | | | | | | | |
| 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 บทบาทและความรับผิดชอบ

- นาย ชานูชล ภาณุศุภนิรันดร์: ทำหน้าที่ศึกษาค้นคว้าข้อมูลที่จะนำมาใช้ในโครงการ และจัดการเก็บข้อมูลและเชื่อมต่อ ส่วนต่อประสานเชิงประยุกต์ (API: Application Programming Interface)
- นาย ณัฐพงษ์ เทพพิทักษ์: ทำหน้าที่ศึกษาค้นคว้าข้อมูลที่จะนำมาใช้ในโครงการ และพัฒนาโมบายแอปพลิเคชันถ่ายวิดีโอสำหรับการตรวจจับวัตถุ
- นาย ธนภัทร สมสิทธิ์: ทำหน้าที่ศึกษาค้นคว้าข้อมูลที่จะนำมาใช้ในโครงการ และพัฒนาเว็บแอปพลิเคชันรายงานผลข้อมูลหลังจากการประมวลผลตรวจจับข้อมูล

1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม

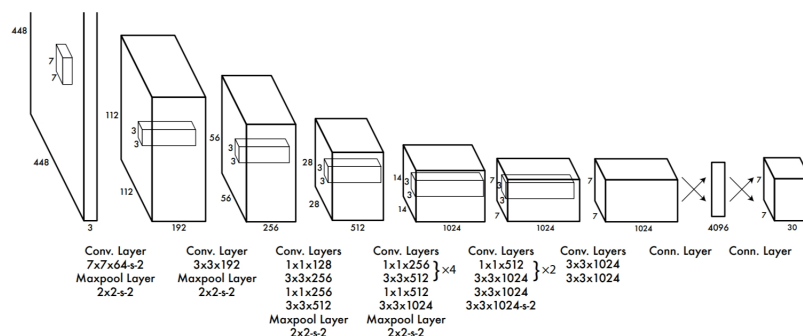
การพัฒนาระบบในการตรวจจับป้ายที่สามารถนำไปเก็บภาษีได้นั้น จะช่วยอำนวยความสะดวกให้สามารถจัดการได้ง่ายและสะดวกยิ่งขึ้น ซึ่งมีผลกระทบในด้านกฎหมายเพราะภาษีป้ายเป็นภาษีที่จัดเก็บจากป้ายที่แสดงชื่อ ยี่ห้อ หรือเครื่องหมายที่ใช้ในการประกอบ การค้า หรือประกอบกิจการอื่นเพื่อหารายได้ หรือ โฆษณาการค้า ซึ่งในส่วนของการเสียนั้นก็ขึ้นอยู่กับประเภทของป้ายตามที่กฎหมายกำหนด และรายได้ที่ได้จากการจัดเก็บภาษีก็จะถูกนำไปพัฒนาบ้านเมืองต่อไป

ทฤษฎีที่เกี่ยวข้อง

การทำโครงการเริ่มต้นด้วยการศึกษาค้นคว้าทฤษฎีที่เกี่ยวข้อง หรืองานวิจัย/โครงการที่เคยมีผู้พัฒนาและนำเสนอไว้แล้ว ซึ่งเนื้อหาในบทนี้ก็จะเกี่ยวกับ การอธิบายถึงทฤษฎีที่นำไปประยุกต์ใช้กับโครงการนี้ เพื่ออ่านทำความเข้าใจกับตัวระบบของโครงการได้ง่ายขึ้น

2.1 You Only Look Once Object Detection Algorithm (YOLO)

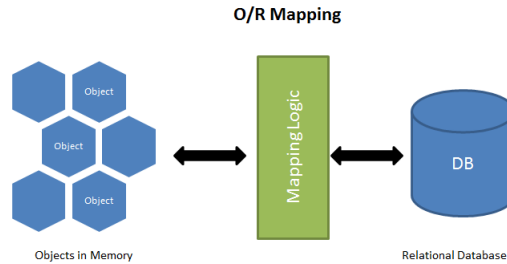
YOLO [?] เป็นอัลกอริทึมสำหรับการระบุบริเวณที่สนใจภายในภาพ และจำแนกประเภทของวัตถุบนแต่ละบริเวณแบบเวลาจริงเหมือนกับตัวจำแนกภาพปกติ โดยที่ภาพหนึ่งสามารถประกอบด้วยบริเวณที่สนใจหลายบริเวณ แล้วแต่ละบริเวณจะนำไปจำแนกวัตถุที่แตกต่างกันได้ ซึ่งทำให้เกิดความซับซ้อนสูงในการ จำแนกภาพระหว่างการตรวจจับวัตถุ ต่างจากอัลกอริทึมตรวจจับวัตถุทั่วไปที่จะใช้อัลกอริทึมแบบ 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 [?] 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

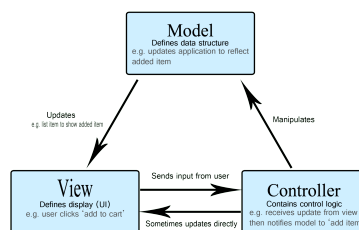


รูปที่ 2.2: Object Relational Mapping

2.3 Model–View–Controller design pattern (MVC)

Model-View-Controller [?] เป็นรูปแบบโครงสร้างที่แยกแอปพลิเคชันออกเป็น 3 ส่วนหลักคือ: โมเดล (model), มุมมอง (view), และคอนโทรลเลอร์ (controller) แต่ละส่วนมีการสร้างขึ้นเพื่อจัดการด้านพัฒนาส่วนแอปพลิเคชันที่เฉพาะเจาะจง ตามรูปที่ 2.3 MVC เป็นหนึ่งในรูปแบบการพัฒนาเว็บตามมาตรฐานอุตสาหกรรมที่ถูกใช้บ่อยที่สุดเพื่อสร้างโครงการที่สามารถเพิ่มและขยายขนาดในอนาคตได้ โดยที่เพื่อให้โปรแกรมเมอร์ง่ายต่อการแก้ไขจัดการ ซึ่งความหมายในแต่ละส่วนของ MVC นั้นได้แก่

1. **Model** คือส่วนที่รับผิดชอบเกี่ยวกับข้อมูลและการประมวลผลทางด้านข้อมูลในแอปพลิเคชัน เช่น การเชื่อมต่อกับฐานข้อมูล การจัดการข้อมูล และการประมวลผลทางข้อมูล เป็นต้น โดยที่ **model** มักจะเป็นตัวแทนของข้อมูลและสถานะของแอปพลิเคชัน
2. **View** คือส่วนที่จะเป็นหน้าต่างของโปรแกรมที่ผู้ใช้จะใช้งานจากส่วนนี้ ไม่ว่าจะเป็นการกรอกข้อมูล, ดูผลลัพธ์ หรือการมีปฏิสัมพันธ์กับผู้ใช้ (User Interface) **view** จริง ๆ แล้วก็คือส่วนที่เรียกว่า GUI (Graphic User Interface)
3. **Controller** เป็นส่วนที่รับผิดชอบในการควบคุมและจัดการกับการกระทำที่เกิดขึ้นจากผู้ใช้ เช่น การรับข้อมูลจากผู้ใช้, การส่งข้อมูลไปยังโมเดลเพื่อประมวลผล, และการอัปเดตสถานะของ **view** ซึ่งโดยทั่วไปแล้ว **controller** จะเป็นตัวกลางที่เชื่อมต่อระหว่าง **model** และ **view** โดยการควบคุมการทำงานของทั้งสอง



รูปที่ 2.3: Model-View-Controller Design Pattern

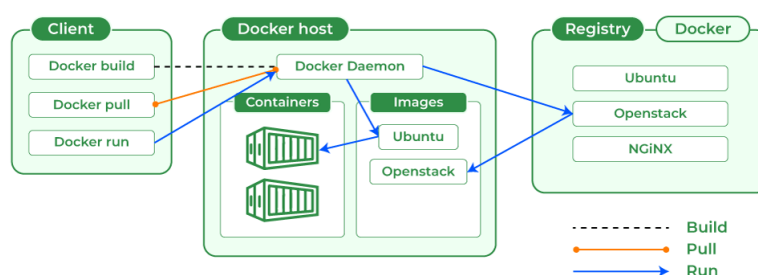
2.4 Hypertext Transfer Protocol (HTTP)

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

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

2.5 Docker

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



รูปที่ 2.4: Docker Architecture

2.6 Interactive Website

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

บทที่ 3

โครงสร้างและขั้นตอนการทำงาน

3.1 การใช้งานพื้นฐาน

ในส่วนของโมบายล์แอปพลิเคชัน เป็นเครื่องมือที่จะจำเป็นต้องใช้งานกล้องและบันทึกพิกัดตำแหน่งทาง GPS อยู่ตลอดเวลาเพื่อทำการส่งรูปภาพ พร้อมกับพิกัดตำแหน่ง แล้วนำไปประมวลผลในเซิร์ฟเวอร์ที่ได้ออกแบบเอาไว้ โดยที่เซิร์ฟเวอร์ดังกล่าวจะทำการประมวลผลรูปภาพเพื่อหาป้ายโฆษณาที่สามารถจัดเก็บภาษีได้ และหลังจากนั้นก็จะจัดเก็บลงฐานข้อมูลต่อไป

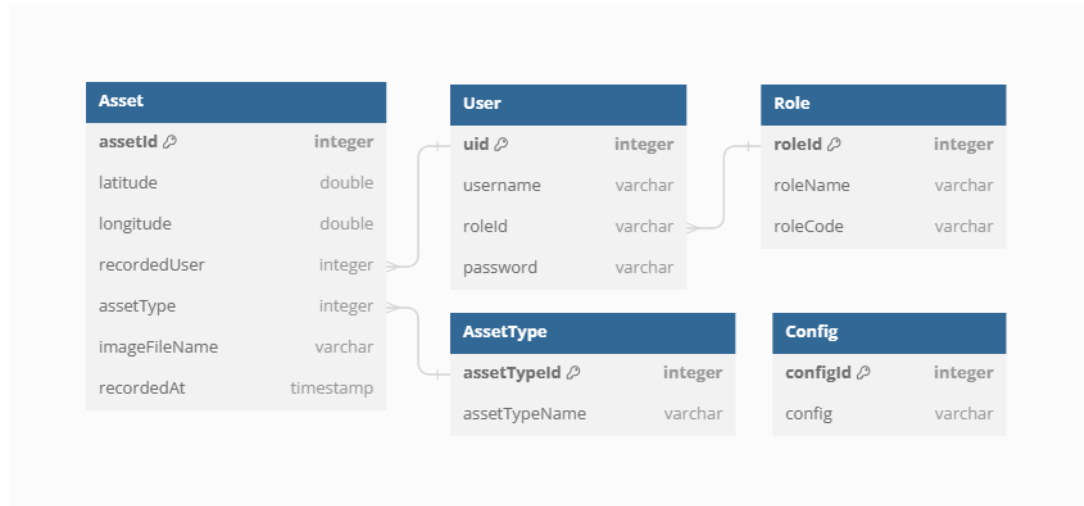
ในส่วนของเว็บแอปพลิเคชัน จะเป็นส่วนของการแสดงผลข้อมูลที่ได้บันทึกมาได้จากส่วนของโมบายล์แอปพลิเคชัน โดยจะแสดงในรูปแบบของแผนที่ในแผนที่ คล้าย ๆ กับการปักหมุดของ Google map โดยที่ในแต่ละหมุดสามารถกดเพื่อดูรายละเอียดต่าง ๆ ได้ เช่น พิกัดของหมุดนั้น และลักษณะรูปป้ายในตำแหน่งนั้นๆที่ได้บันทึกมาจากโมบายล์แอปฯ

3.2 การออกแบบระบบพื้นฐานของโครงการ

3.2.1 Database Design

ประกอบด้วย 4 ตารางดังรูปที่ 3.1 ได้แก่

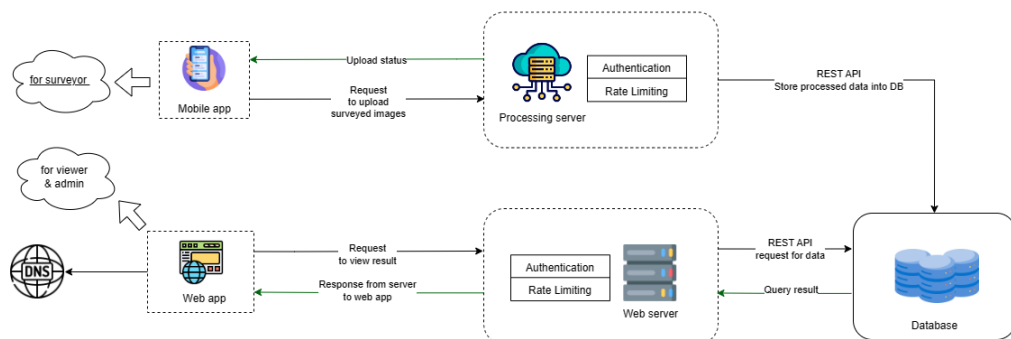
1. User table: เนื่องจากระบบต้องมีการ Authentication เพื่อเข้าใช้งานไม่ว่าจะเป็นทั้งส่วนของ โมบายล์แอปฯ หรือเว็บแอปฯ ดังนั้นตารางนี้จึงจะใช้เก็บข้อมูลพื้นฐานต่าง ๆ ที่จำเป็นต่อการยืนยันตัวตนทั้งหมด
2. Role table: ใช้ในการเก็บบทบาททั้งหมดที่มีของระบบ เช่น ผู้ดูแลระบบ ผู้สำรวจ และอื่น ๆ
3. Asset table: 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

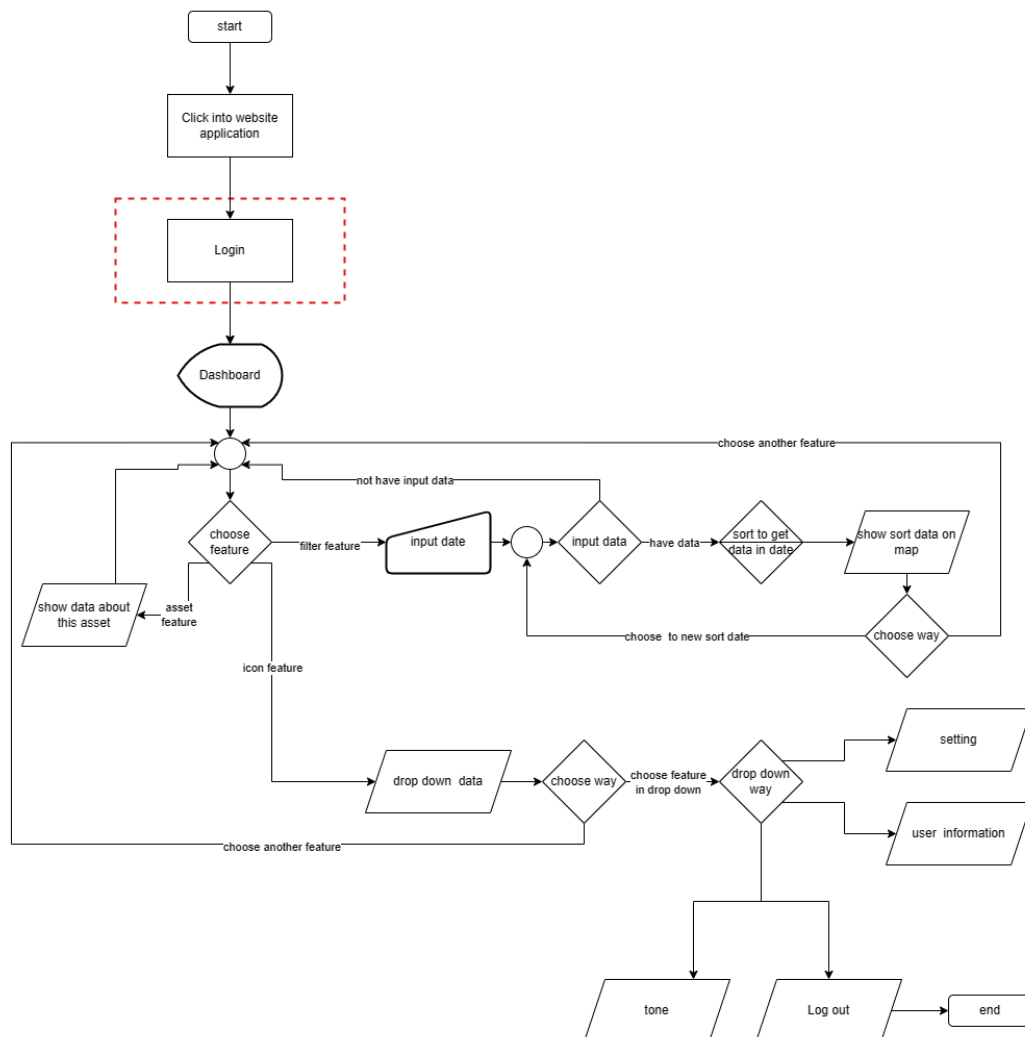
จากรูป จะอธิบายถึงโครงสร้างระบบของโครงการงานนี้ในรูปแบบ Flow diagram เพื่อให้เข้าใจถึงโครงสร้างการทำงานพอสังเขป โดยที่ซอฟต์แวร์จะประกอบด้วย 3 ส่วนหลัก ๆ ได้ Mobile application ซึ่งจะทำงานตามรูปที่ 3.2 Web application ซึ่งจะทำงานตามรูปที่ 3.3 และ Processing server โดยที่ลักษณะการทำงานร่วมกันระหว่างทั้งสามส่วนประกอบ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



รูปที่ 3.2: Overall System Design

3.2.3 Web Application Flow Diagram

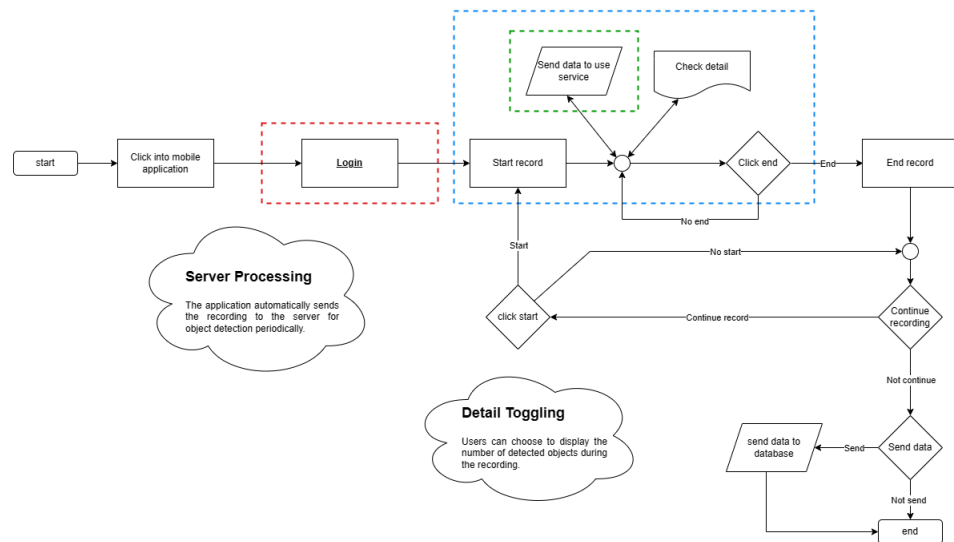
จากรูปที่ 3.3 จะอธิบายถึงลำดับการทำงานของเว็บแอปพลิเคชันในรูปแบบของ flow diagram เพื่อให้เข้าใจในลำดับการทำงานอย่างพอสังเขป พอหลังจากที่ได้เข้าระบบสู่หน้า dashboard จะมีตัวเลือกที่สามารถทำได้อยู่ 3 อย่างคือ filter เป็นการคัดกรองข้อมูลให้เหลือเพียงข้อมูลในช่วงเวลาที่เรต้องการ asset เป็นการกดที่รูปภาพเพื่อที่จะดูข้อมูลเกี่ยวกับ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



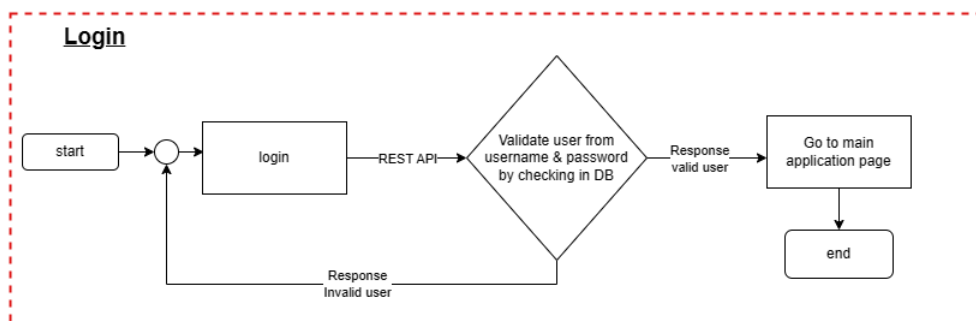
รูปที่ 3.3: Web Application Flow Diagram

3.2.4 Mobile Application Flow Diagram

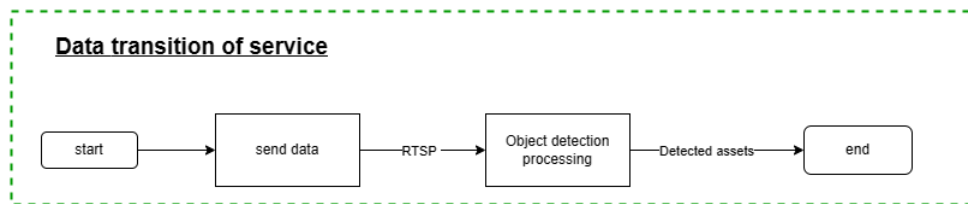
รูปที่ 3.4 จะอธิบายถึงลำดับการทำงานของแอปพลิเคชันในรูปแบบของ flow diagram เพื่อให้เข้าใจในลำดับการทำงานอย่างพอสังเขป โดยพอผู้ใช้จะเริ่มเข้าใช้งานแอปพลิเคชัน ผู้ใช้จะต้องผ่านการเข้าสู่ระบบ ซึ่งมีขั้นตอนการทำงานดังรูปที่ 3.5 เพื่อเป็นการยืนยันตัวตน หลังจากที่ได้เข้าสู่แอปพลิเคชันเรียบร้อยแล้ว ผู้ใช้งานจะสามารถเริ่มสตรีมวิดีโอเพื่อทำการส่งรูปภาพในช่วงเวลาหนึ่งพร้อมแนบตำแหน่งพิกัดในช่วงเวลาดังกล่าวไปยังเซิร์ฟเวอร์ที่ได้จัดเตรียมเอาไว้อยู่ตลอดเวลาที่ทำการสตรีม 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

ข้อมูลทั่วไปของบริษัท

การทำโครงการเริ่มต้นด้วยการศึกษาค้นคว้าทฤษฎีที่เกี่ยวข้อง หรืองานวิจัย/โครงการที่เคยมีผู้พัฒนาและนำเสนอไว้แล้ว ซึ่งเนื้อหาในบทนี้ก็จะเกี่ยวกับ การอธิบายถึงทฤษฎีที่นำไปประยุกต์ใช้กับโครงการนี้ เพื่ออำนวยความสะดวกให้อ่านทำความเข้าใจกับตัวระบบของโครงการได้ง่ายขึ้น

4.1 สินค้าและบริการของบริษัท

1. lorem
2. ipsum
3. solor

4.2 สถานะงบการเงิน

ipsum

บทที่ 5

การประเมินระบบ

5.1 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**

5.2 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 [?]** ที่ประกอบไปด้วย

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