โครงงานเลขที่ วศ.คพ. S007-2/67/2567

เรื่อง

สกรีทเนอร์: ระบบสำรวจถนนสำหรับการจัดการสินทรัพย์เมือง

โดย

นายชาญชล ภานุศุภนิรันดร์ รหัส 640610626
นายณัฐพงษ์ เทพพิทักษ์ รหัส 640610634
นายธนภัทร สมสิทธิ์ รหัส 640610639

โครงงานนี้
เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่
ปีการศึกษา 2567

PROJECT No. CPE S007-2/67/2567

Screetner: 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

หัวข้อโครงงาน	: สกรีทเนอร์: ระบบสำรวจถนนสำหรับการจัดการสินทรัพย์เมือง	
	: Screetner: street scanner system for urban asset management	
โดย	: นายชาญชล ภานุศุภนิรันดร์	รหัส 640610626
	นายณัฐพงษ์ เทพพิทักษ์	รหัส 640610634
	นายธนภัทร สมสิทธิ์	รหัส 640610639
ภาควิชา	: วิศวกรรมคอมพิวเตอร์	
อาจารย์ที่ปรึกษา	: รศ.ดร. สันติ พิทักษ์กิจนุกูร	
ปริญญา	: วิศวกรรมศาสตรบัณฑิต	
สาขา	: วิศวกรรมคอมพิวเตอร์	
ปีการศึกษา	: 2567	

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์
(รศ.ดร. สันติ พิทักษ์กิจนุกูร)

คณะกรรมการสอบโครงงาน

..... ประธานกรรมการ
(รศ.ดร. สันติ พิทักษ์กิจนุกูร)

..... กรรมการ
(ผศ.ดร. กานต์ ปทานุคม)

..... กรรมการ
(ผศ.ดร. นวดนย์ คุณเลิศกิจ)

หัวข้อโครงงาน	: สกรีทเนอร์: ระบบสำรวจถนนสำหรับการจัดการสินทรัพย์เมือง
	: Screetner: street scanner system for urban asset management
โดย	: นายชาญชล ภานุศุภนิรันดร์ รหัส 640610626
	นายณัฐพงษ์ เทพพิทักษ์ รหัส 640610634
	นายธนภัทร สมสิทธิ์ รหัส 640610639
ภาควิชา	: วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	: รศ.ดร. สันติ พิทักษ์กิจนุกูร
ปริญญา	: วิศวกรรมศาสตรบัณฑิต
สาขา	: วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	: 2567

บทคัดย่อ

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

Project Title	: Screetner: 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 Screetner 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

สารบัญรูป

บทที่ 1
บทนำ

1.1 ที่มาของโครงงาน

การจัดเก็บภาษีถือเป็นหนึ่งในรายได้หลักของประเทศไม่ว่าจะเป็นภาษีทางตรง อย่างเช่น ภาษีทางตรง ภาษีรายได้บุคคลธรรมดา ซึ่งจะจัดเก็บได้ จากประชาชนผู้มีเงินได้ทั่วไป ภาษีเงินได้นิติบุคคลซึ่งเป็นภาษีที่จัดเก็บได้จากเงินได้ของบริษัทหรือห้างหุ้นส่วนนิติบุคคล และยังมีภาษีทางอ้อม เช่น ภาษีมูลค่าเพิ่ม ภาษีธุรกิจเฉพาะ ซึ่งเงินที่ได้จากการเก็บภาษีเหล่าล้วนนำไปให้รัฐบาลใช้ในการพัฒนาประเทศให้เจริญก้าวหน้า

ภาษีป้ายก็เป็นส่วนหนึ่งของรายได้ท้องถิ่นที่สามารถจัดเก็บได้โดยองค์กรปกครองส่วนท้องถิ่น โดยที่ภาษีลักษณะนี้เมื่อจัดเก็บได้แล้ว ทางท้องถิ่น ไม่จำเป็นต้องส่งคืนให้ทางรัฐ สามารถนำไปใช้จัดการบริหารพัฒนาภายในท้องถิ่นของตนเองได้ แต่ด้วยความสามารถในการจัดเก็บภาษีป้ายขององค์กรปกครองส่วนท้องถิ่นในแต่ละที่ ขึ้นอยู่กับปัจจัยหลาย ๆ อย่าง เช่น การที่ไม่สามารถรู้ได้ว่าป้ายที่สามารถจัดเก็บภาษีได้นั้นอยู่ที่ตำแหน่งใดในเขตปกครอง ซึ่งมีส่วนทำให้ประสิทธิภาพในการค้นหาป้ายภายในท้องถิ่นที่มีอยู่ทำได้อยู่จำกัด และเป็นขั้นตอนที่ต้องใช้กำลังคนในการตรวจสอบเป็นอย่างมาก ดังนั้นจากปัญหาในจุดที่กล่าวมาทำให้เกิดโครงงานที่เป็นเครื่องมือที่ช่วยในการตรวจจับหาป้ายที่คาดว่าจะสามารถนำไปจัดเก็บภาษี และรายงานผลให้กับแต่ละองค์กรปกครองส่วนท้องถิ่นให้ไปจัดเก็บภาษีจากป้ายเหล่านี้

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อสร้างระบบครบวงจรในการรับวิดีโอแล้วประมวลผลตรวจจับหาป้ายอัตโนมัติ
2. เพื่อพัฒนาแอปพลิเคชันที่ใช้ในการอัดวิดีโอเพื่อที่จะส่งให้ระบบประมวลผล
3. เพื่อพัฒนาเครื่องมือในการรายงานป้ายที่ค้นพบภายในพื้นที่การปกครองส่วนท้องถิ่นสำหรับการไปจัดเก็บภาษี

1.3 ขอบเขตของโครงงาน

1.3.1 ขอบเขตด้านฮาร์ดแวร์

กล้องถ่ายของโทรศัพท์แต่ละเครื่องจะมีคุณภาพและลักษณ์การร่ายที่แตกต่างกัน ซึ่งอาจส่งผลต่อการตรวจจับวัตถุทำให้เวลานำรูปภาพที่ได้นำไปประมวลจะได้ผลลัพธ์ที่แตกต่างกัน ซึ่งโทรศัพท์ที่ได้ใช้ในการเก็บข้อมูลที่นำไปสร้างโมเดลมีอยู่ด้วยกัน 2 เครื่อง โดยมีคุณภาพของกล้องถ่ายรูปดังนี้

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

ความสูงของรถแต่ละคัน และมุมกล้องในการถ่ายภาพมักมีความแตกต่างกันไป ซึ่งอาจส่งผลให้ประสิทธิภาพในการตรวจจับวัตถุได้ไม่เท่ากัน โดยรถยนต์ที่ใช้ในการอัดวิดีโอสำหรับในการเทรนโมเดลเป็น Honda City 2024

Mobile application ที่เป็นส่วนของการส่งข้อมูลภาพไปยังเซิฟเวอร์จำเป็นต้องเชื่อมต่ออินเทอร์เน็ตอยู่ตลอดทั้งการใช้งาน เนื่องจากต้องมีการส่งข้อมูลตลอดเวลา ทั้งนี้สืบก็จะมีเรื่องของการใช้งานทรัพยากรแบต

เตอร์มากตามไปด้วย และในการของการแสดงผลที่เป็นเว็บแอปพลิเคชันจะสามารถใช้งานได้เฉพาะ ในคอมพิวเตอร์เท่านั้น

1.3.2 ขอบเขตด้านซอฟต์แวร์

ในการเก็บภาษีป้ายนั้นจะถูกแบ่งออกเป็นป้ายหลาย ๆ ประเภท อย่างเช่น ป้ายที่มีอักษรไทยล้วน ป้ายที่มีอักษรไทยปนกับอักษรต่างประเทศหรือปนกับภาพ และหรือเครื่องหมาย ป้ายที่ไม่มีอักษรไทย ไม่ว่าจะมีภาพและหรือเครื่องหมายใด ๆ ซึ่งแต่ละประเภทนั้นจะมีอัตราการเก็บภาษีที่แตกต่างกันออกไป แต่ในการประมวลผลในเซิฟเวอร์นั้นจะไม่มีการตรวจสอบและแบ่งแยกประเภทของป้าย และจะรวบรวมเป็นคลาสประเภทเดียวกันแทน อีกทั้งป้ายที่สามารถจัดเก็บภาษีได้บางประเภทมีลักษณะคล้ายกับป้ายบอกทางและป้ายจราจร จึงอาจทำให้มีข้อผิดพลาดเกิดขึ้นในการตรวจจับในบางสถานการณ์

1.4 ประโยชน์ที่ได้รับ

1. ได้เครื่องมือที่ช่วยอำนวยความสะดวกในการเก็บภาษีป้ายให้มีประสิทธิภาพมากยิ่งขึ้น

1.5 เทคโนโลยีและเครื่องมือที่ใช้

1.5.1 เทคโนโลยีด้านซอฟต์แวร์

1. JetBrainIDEs 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 ซึ่งมีความเหมาะสมในการ ใช้งานในโครงการที่ต้องการประมวลผลข้อมูลที่ซับซ้อนและมีขนาดใหญ่ อย่างเช่น โมเดลการเรียนรู้เชิงลึก ที่พวกเราจะนำไปใช้กับการตรวจจับวัตถุ
4. Typescript คือภาษาคอมพิวเตอร์ที่ใช้ในการพัฒนาเว็บร่วมกับ HTML เพื่อให้เว็บมีลักษณะแบบไดนามิก หมายถึง เว็บสามารถตอบสนองกับ ผู้ใช้งานหรือแสดงเนื้อหาที่แตกต่างกันไปโดยจะอ้างอิงตามเว็บบราวเซอร์ที่ผู้เข้าชมเว็บใช้งานอยู่
5. Golang เป็นภาษาการเขียนโปรแกรมที่พัฒนาโดย Google ซึ่งมีประสิทธิภาพสูงและเหมาะสำหรับการพัฒนาแอปพลิเคชันที่ต้องการความเร็วและความเสถียร
6. Tusd เป็นเซิร์ฟเวอร์ที่ใช้ในการอัปโหลดไฟล์ขนาดใหญ่แบบต่อเนื่อง (resumable file uploads) ซึ่งช่วยให้การอัปโหลดไฟล์มีความเสถียรและไม่ขาดตอน
7. Azure Logic Apps เป็นบริการของ Microsoft Azure ที่ช่วยในการสร้างและจัดการเวิร์กโฟลว์อัตโนมัติสำหรับการรวมระบบและการประมวลผลข้อมูล

8. Azure Blob Storage เป็นบริการจัดเก็บข้อมูลแบบออบเจ็กต์ของ 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 จัดเก็บและวิเคราะห์ข้อมูลของระบบเช่น Log และ Metric
12. Azure Email Communication Service เป็นบริการของ Microsoft Azure ที่ใช้ในการส่งอีเมลและการสื่อสารอื่น ๆ ระหว่างระบบ
13. Flutter เป็นเฟรมเวิร์กที่พัฒนาโดย Google ที่ใช้ในการพัฒนาแอปพลิเคชันข้ามแพลตฟอร์ม (cross-platform) ทั้งบน iOS และ Android ด้วยโค้ดเบสเดียว
14. Next.js เป็นเฟรมเวิร์กที่ใช้ในการพัฒนาเว็บแอปพลิเคชันแบบเซิร์ฟเวอร์ไซด์เรนเดอริ่ง (SSR) และสเตติกไซต์เจเนอเรชัน (SSG) ซึ่งช่วยให้การพัฒนาเว็บมีประสิทธิภาพและความเร็วสูงขึ้น และยังมีฟีเจอร์ที่ช่วยในการทำ SEO ได้ดีขึ้น
15. YOLOv8 เป็นระบบที่ใช้ในการพัฒนาโนโมเดลตรวจจับวัตถุความเร็วสูงแบบเวลาจริง 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 และเป็นพื้นฐานบนหลักการของ 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) และดูผลลัพธ์ (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 (แก้)

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 เลือกหัวข้อโครงงาน									
ออกแบบระบบการทำงานโดยคร่าว และ เครื่องมือที่ใช้ในการทำโครงงาน									
ศึกษาข้อมูลเกี่ยวกับขอบเขตพื้นที่ที่จะใช้ทำโครงงาน									
เก็บข้อมูลเพื่อใช้ในกระบวนการเทรนโมเดลสำหรับการตรวจจับวัตถุ									
คัด เลือก 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
พัฒนา กับ ทดสอบ แอปพลิเคชัน ที่ใช้ใน การ อัด วิดีโอ และ เว็บแอปพลิเคชันในการรายงานข้อมูล								
ดิพลอยระบบโดยรวม								
ตรวจสอบความถูกต้องสมบูรณ์หลังการนำไปใช้								
เขียนรายงาน								

1.7 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม

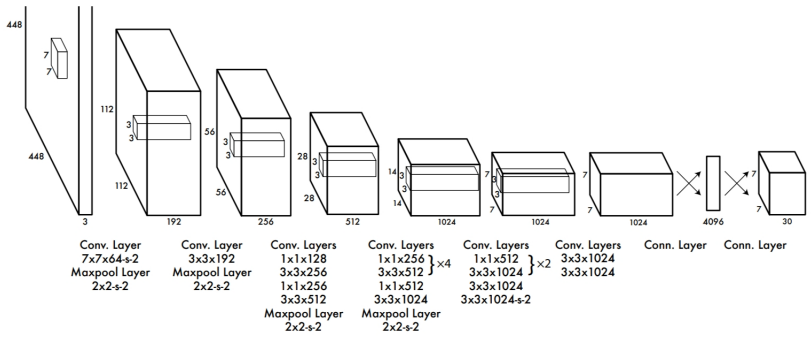
การพัฒนาระบบในการตรวจจับป้ายที่สามารถนำไปเก็บภาษีได้นั้น จะช่วยอำนวยความสะดวกให้สามารถจัดการได้ง่ายและสะดวกยิ่งขึ้น ซึ่งมีผลกระทบในด้านกฎหมายเพราะภาษีป้ายเป็นภาษีที่จัดเก็บจากป้ายที่ แสดงชื่อ ยี่ห้อ หรือเครื่องหมายที่ใช้ในการประกอบ การค้า หรือประกอบกิจการอื่นเพื่อหารายได้ หรือ โฆษณาการค้า ซึ่งในส่วนของการเสียนั้นก็ขึ้นอยู่กับประเภทของป้ายตามที่กฎหมายกำหนด และรายได้ที่ได้จากการจัดเก็บภาษีก็จะถูกนำไปพัฒนาบ้านเมืองต่อไป

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

การทำโครงงานเริ่มต้นด้วยการศึกษาค้นคว้าทฤษฎีที่เกี่ยวข้อง หรืองานวิจัย/โครงงานที่เคยมีผู้พัฒนาและนำเสนอไว้แล้ว ซึ่งเนื้อหาในบทนี้ก็จะเกี่ยวกับ การอธิบายถึงทฤษฎีที่นำไปประยุกต์ใช้กับโครงงานนี้ เพื่ออำนวยให้ผู้อ่านทำความเข้าใจกับตัวระบบของโครงงานได้ง่ายขึ้น

2.1 You Only Look Once Object Detection Algorithm (YOLO)

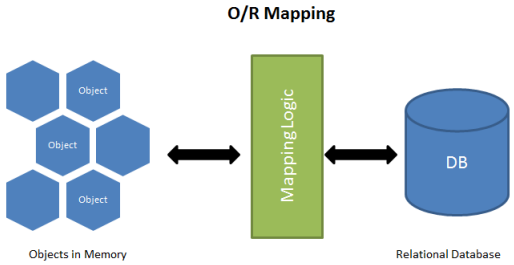
YOLO [1] เป็นอัลกอริทึมสำหรับการระบุบริเวณที่สนใจภายในภาพ และจำแนกประเภทของวัตถุบนแต่ละบริเวณแบบเวลาจริงเหมือนกับตัวจำแนกภาพปกติ โดยที่ภาพหนึ่งสามารถประกอบด้วยบริเวณที่สนใจหลายบริเวณ แล้วแต่ละบริเวณจะนำไปจำแนกวัตถุที่แตกต่างกันได้ ซึ่งทำให้เกิดความซับซ้อนสูงในการ จำแนกภาพระหว่างการตรวจจับวัตถุ ต่างจากอัลกอริทึมตรวจจับวัตถุทั่วไปที่จะใช้อัลกอริทึมแบบ Two-stage Object Detection YOLO นั้นจะใช้แบบ Single-shot Object Detection แทน ซึ่งใช้การสแกนภาพแต่ละภาพเพียงครั้งเดียวสำหรับการพยากรตำแหน่งของวัตถุที่ต้อง 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

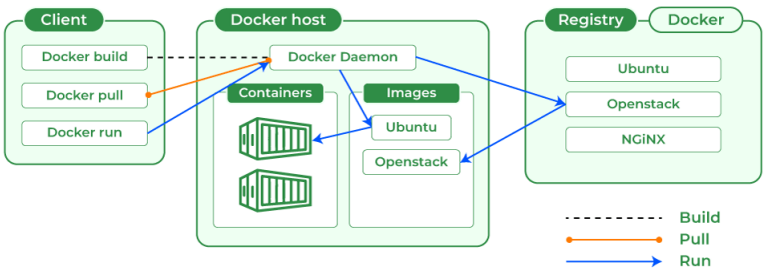
2.3 Hypertext Transfer Protocol (HTTP)

HTTP (Hypertext Transfer Protocol) เป็นโปรโตคอลสื่อสารที่ใช้ในการส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์บนเครือข่ายอินเทอร์เน็ต 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 จะรองรับการร้องขอและการตอบกลับโดยใช้ฟอแมตที่เป็นรูปแบบมาตรฐาน เช่น JSON (JavaScript Object Notation) หรือ XML (Extensible Markup Language)

2.4 Docker

Docker [3] เป็นเทคโนโลยีคอนเทนเนอร์แพลตฟอร์มที่ช่วยในการสร้างและทำการงานร่วมกับคอนเทนเนอร์อย่างมีประสิทธิภาพ ด้วย Docker ผู้ใช้สามารถแยกแยะและแพคเกจแอปพลิเคชันพร้อมกับสิ่งที่เกี่ยวข้องทั้งหมด 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 ในลักษระที่เป็นมิตรต่อผู้ใช้ 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 โดยเฉพาะในองค์กรที่มีผู้ใช้จำนวนมากและมีการเปลี่ยนแปลงบทบาทของผู้ใช้อยู่บ่อยครั้ง

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

2.9 Microservices Architecture

Microservices Architecture [8] เป็นสถาปัตยกรรมในการออกแบบระบบซอฟต์แวร์ ที่แบ่งแอปพลิเคชันออกเป็นบริการขนาดเล็ก ๆ ที่สามารถพัฒนา ทดสอบ และปรับใช้ได้อย่างอิสระ โดยแต่ละบริการจะทำงานร่วมกันผ่าน API และสามารถสื่อสารกันได้ผ่านโปรโตคอลต่าง ๆ เช่น HTTP หรือ AMQP

ข้อดีของ Microservices Architecture ได้แก่:

- **Scalability (การขยายขนาด)**: สามารถขยายขนาดบริการแต่ละตัวได้อย่างอิสระตามความต้องการของระบบ
- **Flexibility (ความยืดหยุ่น)**: สามารถใช้เทคโนโลยีและภาษา.ในการเขียนโปรแกรมที่แตกต่างกันในแต่ละเซอร์วิสโดยที่ไม่มีผลกระทบต่อระบบโดยรวม
- **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 สถาปัตยกรรมระบบ

โครงงานนี้ได้ออกแบบสถาปัตยกรรมระบบให้เป็นแบบ Microservices ทั้งนี้ ในส่วนนี้ได้ได้อธิบายถึงเรื่องการออกแบบทั้งระบบ ไม่ว่าจะเป็น ระบบการพัฒนาระบบ เน็ตเวิร์ค รวมถึงการสื่อสารระหว่างระบบ และการจัดการข้อมูล โดยที่สถาปัตยกรรมทั้งหมดสามารถอธิบายได้ดังรูป 3.1

โดยที่จะแบ่งออกเป็น 3 ส่วนหลัก ๆ ได้แก่

1. **ส่วนของผู้ใช้งานระบบ**: ไม่ว่าจะเป็นทางเว็บไซต์และแอปพลิเคชัน ซึ่งการใช้งานเหล่านั้นก็จะต้องติดต่อสื่อสารไปยังเซอร์วิสหลังบ้าน
2. **ส่วนของเซอร์วิสของระบบ**: ซึ่งเป็นส่วนที่ดูแลการประมวลผลและตรรกะของระบบ
3. **ส่วนของคราวด์เซอร์วิส**: ที่ระบบของเรานำมาใช้อย่าง Azure

3.1.1 ผู้ใช้งานระบบ

ในส่วนของผู้ใช้งานจะถูกแบ่งออกเป็นสองกลุ่มผู้ใช้งานหลัก ๆ คือ กลุ่มผู้ใช้งานผ่านเว็บไซต์และกลุ่มผู้ใช้งานผ่านแอปพลิเคชัน ซึ่งการใช้งานในกลุ่มที่แตกต่างกันก็จะต้องมีการเรียกใช้เซอร์วิสที่แตกต่างกันเช่นกัน ทั้งนี้จากรูป 3.1 จะเห็นได้ว่าไม่ว่าจะเป็นผู้ใช้งานจากส่วนใดก็ตามที่จะเรียกใช้งานเซอร์วิสจะต้องเรียกใช้งานผ่าน Nginx Reverse Proxy ซึ่งทำหน้าที่เป็นตัวจัดการ Traffic ว่าควรจะนำ Request นั้นเรียกใช้งานในเซอร์วิสใดของระบบ

ส่วนของแอปพลิเคชันนั้นในขณะที่ต้องเรียนใช้งานเซอร์วิสก็จะสามารถทำได้โดยการส่ง Request เป็น REST API ผ่านตัว Nginx Reverse Proxy เพื่อใช้งานได้โดยตรง

สองของเว็บไซต์จากรูป 3.1 จะเห็นได้ว่าเว็บไซต์จะ Deploy อยู่บนเซอร์วิสหนึ่งที่ชื่อว่า Heroku ซึ่งเปรียบเสมือนผู้ช่วยที่จะคอยจัดการเรื่องการ Deploy เว็บไซต์ให้เรา โดยจะทำงานเมื่อมีการเปลี่ยนแปลงของ Source Code ของ Repository เว็บไซต์ ทำให้การ Deploy เว็บของระบบเป็นไปได้โดยง่ายโดยทำงานผ่านสิ่งที่เรียกว่า CI/CD (continuous integration continuous deployment) อีกทั้งการใช้งานระบบสามารถเข้าใช้งานได้จาก Domain name ที่ชื่อว่า www.screetner.studio

3.1.2 ระบบหลังบ้าน (Backend Service)

ระบบหลังบ้านของโครงงานนี้ได้ถูกออกแบบให้เป็น microservice architecture ซึ่งจะประกอบด้วยหลาย ๆ เซอร์วิสทำงานด้วยกันไม่ว่างจะเป็น Main Service (เซอร์วิสหลังบ้านตัวหลัก), Log Service (เซอร์วิสที่ใช้จัดเก็บ Audit Logs ของการใช้งาน), Tusd Reuseable Upload (เซอร์วิสที่ใช้ในการอัพโหลดไฟล์ที่มีความสามารถในการอัพโหลดต่อจากเดิมถึงเมื่อการเชื่อต่ออินเทอร์เน็ตขัดข่องระหว่างการอัพโหลด)

โดยที่เซอร์วิสทั้งหมดจะถูก Deploy บน Docker ซึ่งจะมีการจัดแบ่ง Private Network ภายใน Docker ไว้ตามรูป 3.1 ซึ่งจะประกอบไปด้วยสองเน็ตเวิร์คหลัก ๆ คือ

- Default : ทำหน้าเป็นเน็ตเวิร์คที่จะ Deploy เซอร์วิสทั่วไปที่จำเป็นต่อการทำงานของระบบเช่น Portainer, Nginx Reverse Proxy, Zero Trust Client และ Tusd Reuseable Upload

- scn-service : ทำหน้าเป็นเน็ตเวิร์คเฉพาะที่ใช้ Deploy microservice ซึ่งจะประกอบไปด้วยสองเซอร์วิสหลัก และ API Gateway ซึ่งทำหน้าที่ Route เส้นทางของการเรียกใช้งานแต่ละเซอร์วิสจากผู้ใช้

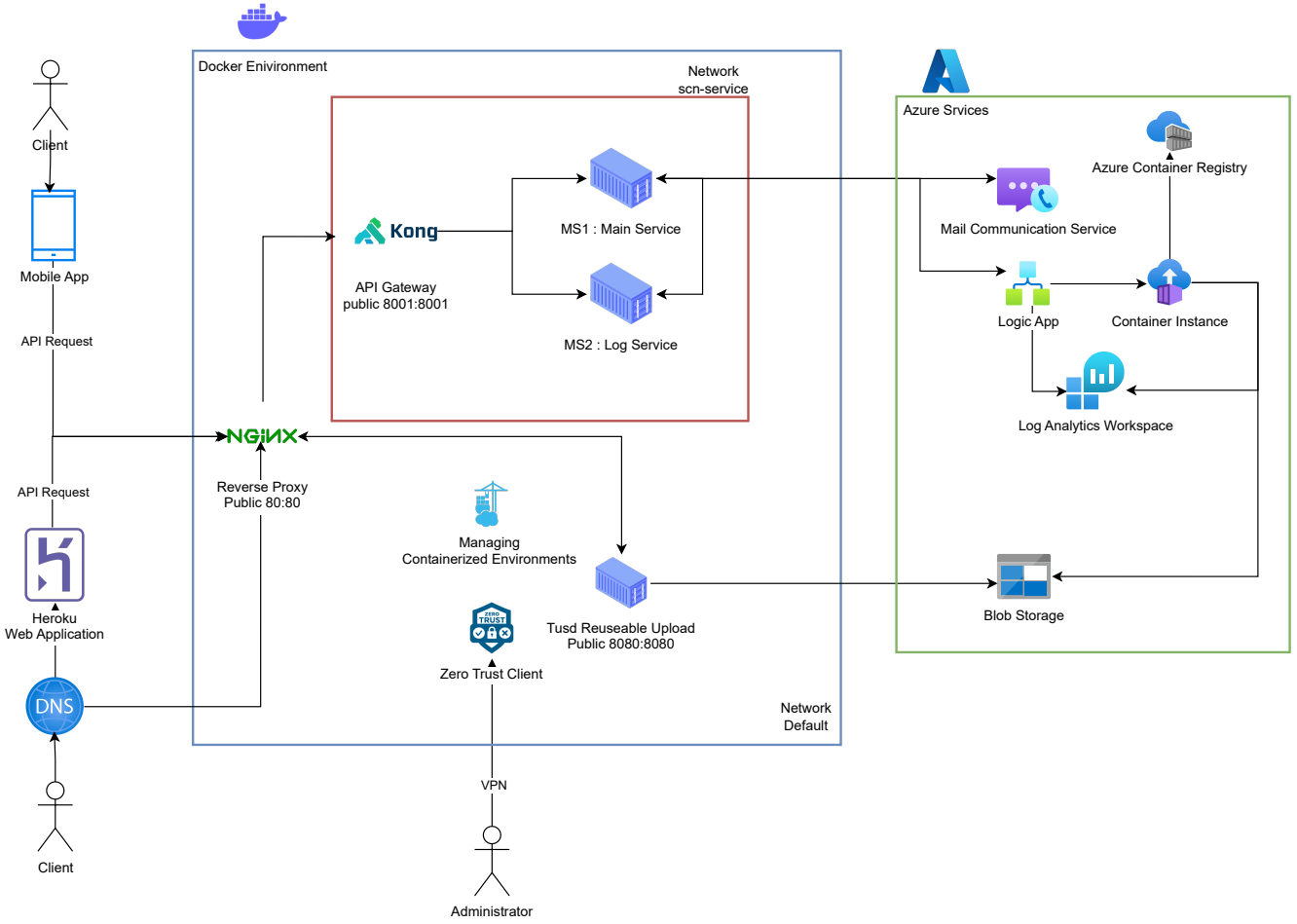
จากรูป 3.1 จะเห็นได้ว่าแต่ละเน็ตเวิร์คจะมีการแบ่งส่วนอยู่ในกล่องของตัวเองนั้นหมายความว่า แต่ละเน็ตเวิร์คจะไม่สามารถสื่อสารกันโดยตรงได้จะต้องสื่อสารผ่านเส้นทางที่ได้แสดงไว้ตามรูป 3.1 เท่านั้น

จากที่กล่าวมาข้างต้น จะเห็นได้ว่าระบบนี้ได้รับการออกแบบให้ทำงานภายใต้ Private Network โดยสมบูรณ์ ซึ่งหมายความว่าจะไม่มีช่องทางให้ผู้ดูแลระบบสามารถเข้าถึงหรือจัดการระบบได้โดยตรง อย่างไรก็ตาม จากภาพ 3.1 แสดงให้เห็นถึงเซอร์วิสที่เรียกว่า Zero Trust Client ซึ่งทำหน้าที่เป็น VPN Server เพื่อให้ผู้ดูแลระบบสามารถเชื่อมต่อเข้าสู่ระบบเพื่อจัดการและบำรุงรักษาได้ตามความจำเป็น

3.1.3 คลาวด์คอมพิวติงบน Microsoft Azure

โครงงานนี้ได้มีการนำเทคโนโลยีคลาวด์คอมพิวติงบนเข้ามาปรับใช้งานเพื่อความสะดวกสบายในการใช้งานในเซอร์วิสบางประเภทที่ไม่เหมากับการใช้งานบนเซิร์ฟเวอร์ทั่วไป โดยเซอร์วิสที่เลือกใช้งานนั้นมีอยู่หลายประเภทเพื่อตอบสนองต่อความต้องการของระบบที่แตกต่างกัน ประกอบด้วย

- **Azure Functions** โครงงานนี้ได้นำเซอร์วิสนี้มาช่วยในการทำ auto scaling ของระบบการทำการตรวจจับวัตถุ โดยเมื่อเซอร์วิสนี้ได้รับคำสั่งให้ทำงาน ก็จะทำการสร้างเครื่องเสมือนขึ้นมา เพื่อทำงานและเมื่อเสร็จสิ้นก็จะทำการลบเครื่องเสมือนทิ้งไป ทั้งนี้ประโยชน์ของการทำ auto scaling คือ สามารถปรับขนาดของระบบให้เหมาะสมกับการทำงานที่เข้ามาในแต่ละช่วงเวลา และลดค่าใช้จ่ายในการใช้งานเซิร์ฟเวอร์
- **Container Instance** หลังจากที่ Azure Functions ได้รับคำสั่งให้ทำงานแล้ว ก็จะสร้างเครื่องพิวเตอร์เสมือนขึ้นมาโดยเครื่องนั้นก็คือเซอร์วิส Container Instance ซึ่งเป็นเซอร์วิสที่ทำหน้าที่ในการรัน Docker Container โดยเซอร์วิสนี้จะทำการรัน Docker Container ที่มี Image ของโปรแกรมที่ต้องการให้ทำงาน มากไปกว่านั้นสามารถกำหนดจำนวน สเปค ตามความต้องการของผู้ใช้งานได้ สามารถทำงานเป็นกลุ่ม หรือแยกออกมาเป็นเครื่องเสมือนแยกต่างหากกันได้
- **Azure Container Registry** เป็นเซอร์วิสที่ใช้ในการเก็บ Image ของ Docker Container ที่ต้องการให้ Container Instance รัน โดยเซอร์วิสนี้จะทำการเก็บ Image ที่สร้างขึ้นมาจากการ Build โปรแกรม และเมื่อ Container Instance ได้รับคำสั่งให้ทำงาน ก็จะไปดึง Image จากเซอร์วิสนี้มาใช้งาน เนื่องจากที่จัดเก็บอยู่บนคลาวด์เดียวกับเซอร์วิสอื่น จึงทำให้การดึง Image มาใช้งานได้รวดเร็วและมีประสิทธิภาพ
- **Azure Blob Storage** เป็นเซอร์วิสที่ใช้ในการเก็บข้อมูลที่ใช้ในการทำงานของระบบ โดยเซอร์วิสนี้จะทำการเก็บข้อมูลที่ได้จากการตรวจจับวัตถุ และเมื่อต้องการใช้งานก็จะไปดึงข้อมูลจากเซอร์วิสนี้มาใช้งาน โดยเซอร์วิสนี้สามารถเก็บข้อมูลได้มาก และสามารถเข้าถึงข้อมูลได้ง่าย และมีความปลอดภัยสูง
- **Mail Comunication Service** ใช้ในการส่งอีเมลไปถึงผู้ใช้งานเพื่อใช้ในการสมัครสมาชิก และมากไปว่านั้นยังสามารถส่งการแจ้งเตือนในเรื่องต่าง ๆ ไปยังผู้ใช้งานได้ตามความต้องการของระบบ



รูปที่ 3.1: แผนภาพแสดงสถาปัตยกรรมระบบ

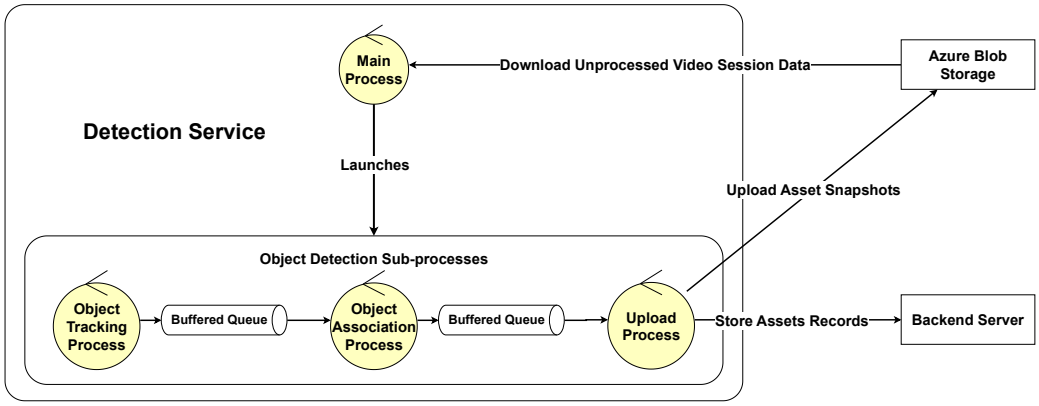
3.2 ระบบการตรวจจับวัตถุ

The Detection Service is a component of the application responsible for identifying objects within a video stream. It operates by downloading a video session directory from Azure Blob Storage, processing the video for object detection, and finally converting the results into structured records for storage in a database.

This pipeline is implemented in Python using Ultralytics' YOLO for object detection and tracking. The entire process consists of three sub-processes, orchestrated by the main process.

3.2.1 กระดำเนินการหลัก

The main process serves as the controller of the workflow. First, it downloads the necessary files from Azure Blob Storage. For each image location data in the video session, it launches three subprocesses to handle different tasks concurrently. These subprocesses include the Detection Process, the Object Tracking Process, and the Upload Process. These subprocesses interact through a buffered queue, following the producer-consumer pattern. This concurrent execution allows for efficient processing and handling of video frames.



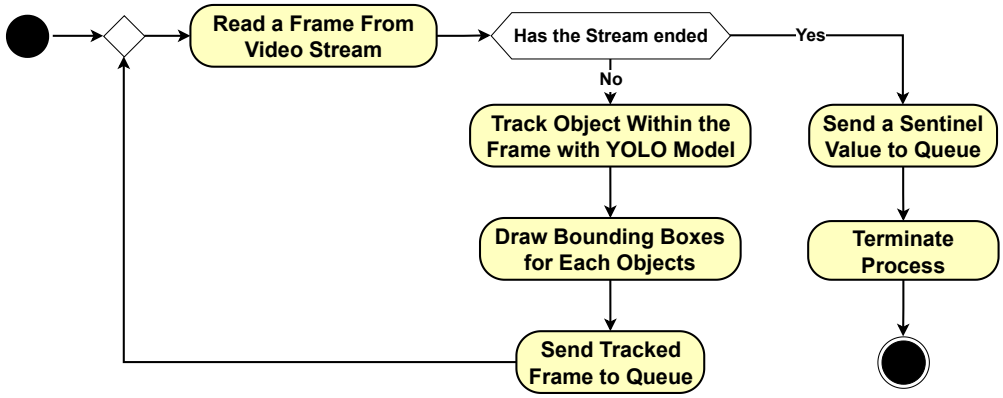
รูปที่ 3.2: องค์ประกอบระบบตรวจจับวัตถุ

3.2.2 การดำเนินการติดตามวัตถุ

The Tracking Process focuses on performing object detection and tracking across video frames using Ultralytics' YOLO model. The process is designed to identify objects in each frame and assign a unique tracking ID to each object. This ensures consistent tracking of objects across multiple frames. The output from this process includes several key components: the detected frame object from YOLO, the Unix timestamp indicating when the frame was recorded, and a list of detected objects. Each object is accompanied by its tracking ID and bounding box coordinates, allowing for precise identification and localization

within the frame.

```
{
    "frame": object,
    "recordedAt": long,
    "trackingBoxes": {
        "trackId": number,
        "box": (int, int, int, int)
    }[],
}
```



รูปที่ 3.3: ขั้นตอนการทำงานของการดำเนินการติดตามวัตถุ

3.2.3 การดำเนินการเชื่อมโยงวัตถุ

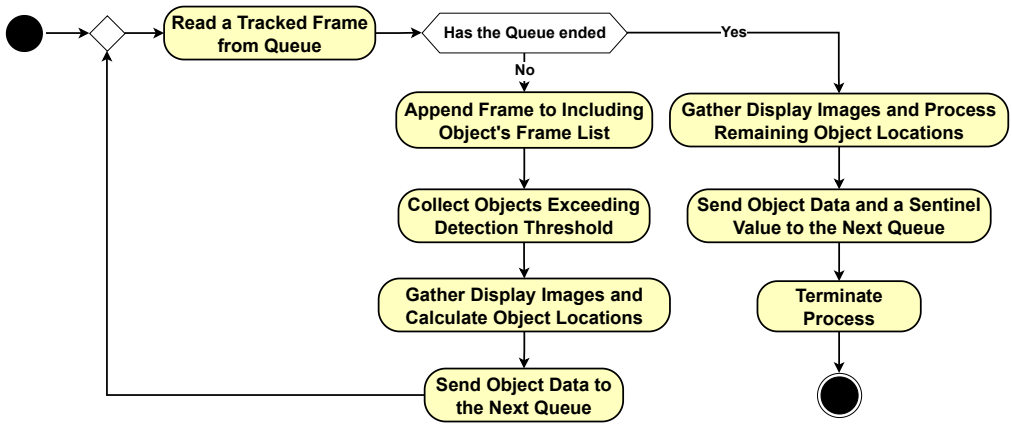
The Object Association Process groups identical objects across multiple frames and assigns them relevant attributes. To achieve this, the system maintains a mapping of object IDs to the frames in which they appear. When an object is not detected for a predefined number of consecutive frames, the process assumes that the recorder has moved past the object. At this point, the mapping of the object ID to its associated frames is used to determine its final location.

The object's location is inferred from the last detected frame by correlating its timestamp with the closest recorded location timestamp. Given that timestamps are stored alongside location coordinates, the location of the object is determined by finding the closest matching timestamp in the location data. Additionally, the image representing the object in the system's interface is chosen from the third quartile of its detected frames. This selection ensures that the image is sufficiently large for visibility while avoiding frames that might have the object partially out of view.

Finally, the processed data is sent to the queue in the following structured format:

```
{
```

```
    "frame": object,
    "tloc": {
        "timestamp": long,
        "latitute": float,
        "longitude": float
    },
    "recordedAt": long
}
```



รูปที่ 3.4: ขั้นตอนการทำงานของการดำเนินการเชื่อมโยงวัตถุ

3.2.4 กระดำเนินการอัปโหลด

The Upload Process serves as the final stage in the object detection pipeline, ensuring that both image data and metadata are stored. This process is responsible for handling the processed frames and corresponding object location data, transferring them to their respective storage destinations.

The workflow begins by retrieving the processed detection data from the queue. Each detection consists of a video frame and its associated metadata, including the object's timestamped location (tloc). The image frame is first encoded into a JPEG format before being converted into a byte stream, preparing it for upload.

The encoded image is then stored in Azure Blob Storage under a structured naming convention that includes the video name and frame index. Simultaneously, the object's metadata—including its latitude, longitude, and the timestamp at which the frame was recorded—is stored in a database. Each uploaded image is referenced within this metadata, ensuring a direct link between the visual representation and its recorded spatial data.

บทที่ 4
การประเมินระบบ

4.1 การประเมินประสิทธิภาพซอฟต์แวร์

ทดสอบประสิทธิภาพซอฟต์แวร์โดยจะมีการแบ่งส่วนในการทดสอบออกเป็นส่วน 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 คือส่วนของแอปพลิเคชันในโทรศัพท์มือถือกับส่วนของเว็บแอปพลิเคชัน โดยจะมีเกณฑ์การให้คะแนนอยู่ที่ 1 ถึง 5 โดยจะมีการให้คะแนนในเรื่องดังต่อไปนี้

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

โดยที่ทั้ง 4 ข้อเป็นพิจราณาจากแนวคิดตาม The Four Elements of User Experience [10] ที่ประกอบไป ด้วย

1. Usability ความใช้ง่ายในการใช้งาน เกี่ยวข้องกับสามารถในการใช้งาน รวมไปถึงความเหมาะสมการใช้งานกับผู้งานใช้
2. Adaptability ความสามารถในงานปรับตัว กล่าวถึงระดับความยากง่ายของการใช้งานตั้งแต่จุดเริ่มต้นจนถึงจุดสิ้นสุดของระบบ โดยที่ผู้งานสามารถใช้งานได้อย่างคล่องแคล่ว
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/