

โครงการเลขที่ วศ.คพ. P042-2/2566

เรื่อง

แคปสแนป : ระบบจัดการการซื้อ-ขายในร้านค้าปลีกอัตโนมัติด้วยตนเองโดยใช้ปัญญา
ประดิษฐ์

โดย

นายพงศกร รัตนพันธ์ รหัส 630610749
นางสาวศุภาริญญา ศิลปะลิทธิ์ รหัส 630610765

โครงการนี้

เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่
ปีการศึกษา 2566

PROJECT No. CPE P042-2/2566

**CapSnap : Retail self-checkout system using Computational Intelligence
Technique**

**Pongsakorn Rattanapan 630610749
Suparida Silpasith 630610765**

**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
2023**

หัวข้อโครงการ : แคปสแนป : ระบบจัดการการซื้อ-ขายในร้านค้าปลีกอัตโนมัติด้วยตนเองโดยใช้ปัญญาประดิษฐ์
โดย : นายพงศกร รัตนพันธุ์ รหัส 630610749
นางสาวศุภาริยา ศิลปสิทธิ์ รหัส 630610765
ภาควิชา : วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา : รศ.ดร. ศันสนีย์ เอื้อพันธ์วิริยะกุล
ปริญญา : วิศวกรรมศาสตรบัณฑิต
สาขา : วิศวกรรมคอมพิวเตอร์
ปีการศึกษา : 2566

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

คณะกรรมการสอบโครงการ

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์
(รศ.ดร. สันติ พิทักษ์กิจนุกร)
..... ประธานกรรมการ
(รศ.ดร. ศันสนีย์ เอื้อพันธ์วิริยะกุล)

..... กรรมการ
(ผศ.ดร. เกษมสิทธิ์ ตียพันธ์)

..... กรรมการ
(รศ.ดร. นิพนธ์ ธีร正宗)

หัวข้อโครงการ	: แคปسنป : ระบบจัดการการซื้อ-ขายในร้านค้าปลีกอัตโนมัติด้วยตนเองโดยใช้ปัญญาประดิษฐ์
	: CapSnap : Retail self-checkout system using Computational Intelligence Technique
โดย	: นายพงศกร รัตนพันธ์ รหัส 630610749 นางสาวศุภริภา ศิลปสิทธิ์ รหัส 630610765
ภาควิชา	: วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	: รศ.ดร. ศันสนีย์ เอื้อพันธ์วิริยะกุล
ปริญญา	: วิศวกรรมศาสตรบัณฑิต
สาขา	: วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	: 2566

บทคัดย่อ

ในปัจจุบันการซื้อสินค้าปลีก ณ จุดจำหน่ายสินค้าทั่วไปในประเทศไทย เช่น ชูปเปอร์มาเก็ต ห้างสรรพสินค้า หรือ ร้านค้าปลีกต่าง ๆ นั้น มีการใช้ระบบการจ่ายเงินที่จุดชำระสินค้า หรือแคชเชียร์กับพนักงาน ซึ่งระบบดังกล่าวมีไม่ตอบโจทย์กับความต้องการ และความสะดวกสบายของผู้ซื้อสินค้า เนื่องจากมีข้อเสีย คือการรอและต่อแถวเพื่อชำระสินค้า นอกจากนี้จะเห็นว่าปัจจุบันบริษัทต่าง ๆ ได้เปลี่ยนรูปแบบการบริการให้ผู้บริโภคสามารถทำธุรกรรมได้ ผ่านโทรศัพท์มือถือ จึงมีความสมควรที่จะพัฒนาระบบซอฟแวร์ที่เป็นตัวช่วยให้ลูกค้าสามารถซื้อสินค้าได้ด้วยตนเอง โดยไม่ต้องผ่านจุดบริการของร้านค้า

แคปسنป (CapSnap) เป็นระบบจัดการการซื้อ-ขายสินค้าในร้านค้าปลีก เพื่อยกระดับความสะดวกและลดความยุ่งยากในการซื้อ-ขายสินค้าทั้งทางฝั่งลูกค้า และฝั่งร้านค้า โดยลูกค้าสามารถจัดการการซื้อสินค้าได้ด้วยตนเองผ่านทางแอปพลิเคชัน บนโทรศัพท์มือถือ โดยที่ระบบใช้หลักการความฉลาดเชิงคำนวณ หรือ ปัญญาประดิษฐ์ (Computational Intelligence) ในการประมวลผล ทั้งนี้ลูกค้าสามารถสแกนสินค้าที่ต้องการแบบเรียลไทม์ เพื่อบอกรายละเอียดและราคาของสินค้าแต่ละชนิดที่ถูกแสดง ซึ่งจะเป็นประโยชน์ต่อคนที่ไม่ปัญหาในการมองเห็นในการเลือกซื้อสินค้า ลูกค้าสามารถระบุรายการสินค้าได้ด้วยตนเองผ่านแอปพลิเคชัน นอกจากนี้ระบบยังมี เว็บไซต์สำหรับร้านค้า (Website Dashboard) ให้บริการสำหรับผู้รับผู้ร้านค้า เพื่อช่วยให้ร้านค้าปลีกสามารถจัดการร้านได้อย่างมีประสิทธิภาพมากขึ้นโดยสามารถจัดการ และดูข้อมูลคลังสินค้า รวมถึงข้อมูลการขายสินค้าได้ในเว็บไซต์เดียว

Project Title : CapSnap : Retail self-checkout system using Computational Intelligence Technique
Name : Pongsakorn Rattanapan 630610749
 Suparida Silpasith 630610765
Department : Computer Engineering
Project Advisor : Assoc. Prof. Sansanee Auephanwiriyakul, Ph.D.
Degree : Bachelor of Engineering
Program : Computer Engineering
Academic Year : 2023

ABSTRACT

Nowadays, purchasing retail products at general retail stores in Thailand, such as supermarkets, department stores, or small retailers offer payment at the checkout point or cashier, which does not satisfy the requirements and convenience of customers because there are disadvantages, namely waiting and queuing up to pay. In addition, it can be seen at present that different companies have modified their services permitting consumers to conduct transactions using mobile phones. Thus, it is appropriate to develop a software system that enables clients to independently purchase things and checkout without having to go through the store's service counter. In order to simplify and lessen the complexity of purchasing and selling products on both the consumer side and the store side, we have designed the CapSnap system for handling purchasing in retail establishments via mobile application and allows customers to make their own purchases. Moreover, customers can live-stream product images using the function in the mobile application in real-time. The details and costs of each scanned item are then retrieved by the system using a computational intelligence technique, which is advantageous for users with visual impairments. The shopping cart feature of the application allows users to add and remove products. Subsequently, customers can handle their own payments through the application. Eventually, the system offers a Website Dashboard service for the retailer side which enable them to monitor and view selling and stock information, which helps them run their stores more effectively.

สารบัญ

บทคัดย่อ	๑
Abstract	๒
สารบัญ	๓
สารบัญรูป	๔
สารบัญตาราง	๕
1 บทนำ	๑
1.1 ที่มาของโครงการ	๑
1.2 วัตถุประสงค์ของโครงการ	๑
1.3 ขอบเขตของโครงการ	๑
1.4 ประโยชน์ที่ได้รับ	๒
1.5 เทคโนโลยีและเครื่องมือที่ใช้	๒
1.6 แผนการดำเนินงาน	๒
1.7 บทบาทและความรับผิดชอบ	๓
1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม	๓
2 ทฤษฎีที่เกี่ยวข้อง	๔
2.1 WebSocket for Streaming image [1]	๔
2.2 Artificial Neural Networks [2]	๕
2.2.1 Multilayer Perceptron [3]	๕
2.2.2 Multilayer perceptron - Feed forward [4]	๖
2.2.3 Classification	๖
2.2.4 Training	๖
2.3 Convolutional neural network	๗
2.4 Transfer learning	๘
3 โครงสร้างและขั้นตอนการทำงาน	๑๐
3.1 โครงสร้างของระบบ	๑๐
3.2 ระบบบริการจำแนกสินค้าผ่านรูปภาพ	๑๑
3.2.1 เตรียมชุดข้อมูลฝึกสอน	๑๑
3.2.2 K-Fold Cross Validation	๑๑
3.2.3 Data Augmentation การเพิ่ม traing data	๑๒
3.2.4 Model Architecture	๑๓
3.2.5 classification products	๑๔
3.3 ระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ	๑๕
3.3.1 ขอบเขตการพัฒนาระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ	๑๕
3.3.2 การออกแบบระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ	๑๕
3.4 ระบบเว็บไซต์สำหรับร้านค้า	๑๖
3.4.1 ขอบเขตการพัฒนาระบบเว็บไซต์สำหรับร้านค้า	๑๖
3.4.2 การออกแบบระบบเว็บไซต์สำหรับร้านค้า	๒๐
3.5 การจัดทำฐานข้อมูล	๒๔

4 การทดลองและผลลัพธ์	27
4.1 Classification model	27
4.2 ผลลัพธ์จากการฝึกสอน	28
4.3 ผลการทำงานของระบบแอปพลิเคชันบนโทรศัพท์มือถือ	29
4.3.1 หน้าแสดงสินค้า และ รายละเอียดของสินค้า (Stock page)	30
4.3.2 หน้าแสดงสินค้า หรือ อ่าน barcode โดยใช้กล้อง (Streaming page)	31
4.3.3 หน้าลงชื่อเข้าใช้งาน และ ลงทะเบียนเข้าใช้งาน (Login / Register page)	31
4.3.4 หน้าการตั้งค่าผู้ใช้ (Profile page / Edit profile page)	32
4.3.5 หน้าแสดงตะกร้าสินค้าที่ผู้ใช้เลือก (cart page)	32
4.3.6 หน้าแสดงประวัติการซื้อสินค้าของลูกค้า (History page)	33
4.4 ผลการ checkout	34
4.5 ผลการทำงานของระบบเบร์ไซต์สำหรับร้านค้า	35
4.5.1 หน้าลงชื่อเข้าใช้งาน (Login)	35
4.5.2 หน้าการตั้งค่าร้านค้า และผู้ดูแลร้านค้า (Store Settings)	36
4.5.3 หน้าการจัดการสินค้า (Stock Manager)	37
4.5.4 หน้าแสดงรายงานยอดขายสินค้า (Selling Report)	40
4.5.5 หน้าแสดงประวัติการซื้อสินค้าของลูกค้า (Purchase History)	43
4.5.6 หน้าแสดงข้อมูลลูกค้า (Customers)	44
5 บทสรุปและข้อเสนอแนะ	45
5.1 สรุปผล	45
5.2 Web Dashboard	45
5.3 ปัญหาที่พบและแนวทางการแก้ไข	45
5.3.1 Classification model	45
5.3.2 Machine learning scalability	46
5.3.3 Web Dashboard	47
5.4 ข้อเสนอแนะและแนวทางการพัฒนาต่อ	47
บรรณานุกรม	48
ก Data Flow	50

สารบัญรูป

2.1	WebSocket connection	4
2.2	Multilayer perceptron	5
2.3	Convolution	7
2.4	Convolutional Neural Network	8
2.5	The concept of transfer learning	8
3.1	โครงสร้างของระบบ CapSnap	10
3.2	Google Drive	11
3.3	Dataset	12
3.4	Dataset generator	12
3.5	Xception	13
3.6	Classification model	14
3.7	Mobile Application Use Case Diagram	16
3.8	Website Application Use Case Diagram	20
3.9	Administer System Use Case Diagram	24
3.10	System Database Schema	26
4.1	Train fold	27
4.2	Confusion matrix	28
n.1	Work table	54

สารบัญตาราง

1.1	แผนการดำเนินงาน	2
3.1	แสดงการอธิบายกรณีการใช้งานกรณีสินค้าของระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ	17
3.2	แสดงการอธิบายกรณีการใช้งานกรณีดูรายการสินค้าของระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ	17
3.3	แสดงการอธิบายกรณีการใช้งานกรณีดูข้อมูลสินค้าของระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ	17
3.4	แสดงการอธิบายกรณีการใช้งานกรณีเพิ่มสินค้าลงตะกร้าของระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ	18
3.5	แสดงการอธิบายกรณีการใช้งานกรณีเพิ่มสินค้าลงตะกร้าของระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ	18
3.6	แสดงการอธิบายกรณีการใช้งานกรณีเพิ่มสินค้าลงตะกร้าของระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ	19
3.7	แสดงการอธิบายกรณีการใช้งานกรณีเพิ่มสินค้าลงตะกร้าของระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ	19
3.8	แสดงการอธิบายกรณีการใช้งานกรณีดูข้อมูลสินค้าของระบบเว็บไซต์สำหรับร้านค้า	21
3.9	แสดงการอธิบายกรณีการใช้งานกรณีลบข้อมูลของระบบเว็บไซต์สำหรับร้านค้า	21
3.10	แสดงการอธิบายกรณีแก้ไขข้อมูลสินค้าของระบบเว็บไซต์สำหรับร้านค้า	22
3.11	แสดงการอธิบายกรณีเพิ่มข้อมูลสินค้าของระบบเว็บไซต์สำหรับร้านค้า	22
3.12	อธิบายกรณี ดูประวัติการซื้อของลูกค้าบนระบบเว็บไซต์สำหรับร้านค้า	23
3.13	อธิบายกรณี ดูรายงานยอดขายสินค้าของระบบเว็บไซต์สำหรับร้านค้า	23
3.14	แสดงการอธิบายกรณีการใช้งานกรณีดูข้อมูลลูกค้าบนระบบเว็บไซต์สำหรับร้านค้า	25
4.1	ผลลัพธ์จากการทดสอบ Model โดยใช้ Train Dataset	29
4.2	ผลลัพธ์จากการทดสอบ Model โดยใช้ Blind Dataset	29
4.3	Camera classification process	34
4.4	Camera barcode process	34
4.5	Line Checkout process	34
5.1	เปรียบเทียบการจัดการร้านค้าเมื่อไม่มีและมีระบบเว็บไซต์สำหรับร้านค้า	45

บทที่ 1

บทนำ

1.1 ที่มาของโครงการ

จากระบบการขายสินค้าในร้านค้าปลีกส่วนใหญ่ของประเทศไทย ไม่ว่าจะเป็นห้างสรรพสินค้า ชุปเปอร์มาร์เก็ต หรือ ร้านค้าปลีกรายย่อยต่าง ๆ พบร่วมกันการนำร่อง คือ การนำร่องการขายสินค้าที่ใช้เทคโนโลยีพัฒนา ซึ่งข้อเสียแรกของวิธีการขายสินค้าดังกล่าว คือ การรอชำระเงินของร้านนั้นเสียเวลา และไม่สะดวกรวดเร็ว ยิ่งหากต้องมีการต่อแวรอชำระเงิน ก็จะทำให้ผู้ใช้บริการเสียเวลามากขึ้น และเสียความพึงพอใจในการใช้บริการ รวมถึงต้องมีการจัดพื้นที่สำหรับการต่อแวรอีกด้วย ด้วยเหตุนี้ จึงควรพัฒนาเทคโนโลยีที่มาช่วยการแก้ปัญหาอย่างตรงจุด โดยให้ลูกค้าสามารถชำระเงินได้ด้วยตนเอง ผ่านการเลือกซื้อสินค้าได้อย่างสะดวกสบายผ่านอุปกรณ์โทรศัพท์มือถือ

นอกจากนี้สำหรับร้านค้าปลีกรายย่อยที่มีรูปแบบการขายเป็นแบบบริการตนเอง หรือแม้กระทั่งร้านค้าปลีกที่มีพนักงานนำร่อง แต่ไม่มีระบบช่วยจัดการยอดขาย ก็สามารถพับปัญหาในการจัดการยอดขาย และคลังสินค้าได้ เนื่องจากพนักงาน หรือเจ้าของร้านต้องติดตามการขายสินค้าด้วยตนเองทั้งหมด โดยจะต้องคอยนับจำนวนสินค้าที่เหลืออยู่ภายในร้านเพื่อตรวจสอบว่ามีสินค้าใดถูกขายไปแล้ว ทำให้เกิดความยากลำบากและผิดพลาดได้ด้วย

ผู้จัดทำจึงพัฒนาระบบ CapSnap self-service เพื่อลดปัญหาที่เกิดจากการรอชำระเงินที่เคยเขียนโดยพัฒนาแอปพลิเคชันสำหรับผู้ใช้งานทั่วไปที่สามารถเชื่อมต่อกับร้านค้าที่เข้าร่วมเพื่อใช้บริการการซื้อสินค้าแบบ Self-service ที่ร้านค้าใน ฯ โดยลูกค้าสามารถเข้าใช้งานแอปพลิเคชันในโทรศัพท์มือถือแล้วทำการเลือกซื้อสินค้าผ่านฟังก์ชันการสั่งซื้อสินค้าเพื่อ ทราบถึงรายละเอียดของสินค้า และเพิ่มสินค้าเข้าตะกร้า โดยระบบจะใช้หลักการทำงานความฉลาดเชิงคำนวณ (Computational Intelligence) ในการจำแนกสินค้า ผ่านการถ่ายทอดสตอรุสินค้า ลูกค้าสามารถชำระเงินได้ด้วยตนเองพร้อมนำสินค้าที่ซื้อออกจากร้านค้าได้เลย ซึ่งระบบจะเก็บบันทึกประวัติการขายสินค้าลงในฐานข้อมูลเพื่อประมวลผล และแสดงบนหน้าเว็บไซต์สำหรับร้านค้า (Website Dashboard) ซึ่งสามารถบริการจัดการคลังสินค้า และข้อมูลการขายได้ในที่เดียว เพิ่มประสบการณ์การใช้บริการที่ดีให้ลูกค้า และเพิ่มประสิทธิภาพในการบริหารจัดการสินค้าในร้านค้าสำหรับร้านค้าที่ต้องการให้บริการแบบ Self-service ลดภาระว่าจ้างพนักงาน และต้องการเครื่องมือในการบริหารร้านค้า

1.2 วัตถุประสงค์ของโครงการ

- เพื่อพัฒนาระบบ จำแนกสินค้าจากรูปภาพโดยใช้เทคนิคความฉลาดเชิงคำนวณ
- เพื่อสร้างระบบเว็บไซต์ และแอปพลิเคชันสำหรับโทรศัพท์มือถือเพื่อให้ผู้ใช้งานสามารถใช้งานระบบได้

1.3 ขอบเขตของโครงการ

ข้อมูลที่ใช้ในการฝึกอบรมความฉลาดเชิงคำนวณในการแยกประเภทชนิดสินค้า เป็นชุดข้อมูลชนิดอาหาร และเครื่องดื่มสำเร็จรูปที่เก็บจากร้านค้าห้อง 422 ตึก 30 ปี คณะวิศวกรรมศาสตร์ สาขาวิชาศึกษาคอมพิวเตอร์ มหาวิทยาลัยเชียงใหม่ และจากร้านค้าอื่น ๆ เพิ่มเติม

1.4 ประโยชน์ที่ได้รับ

1. แอปพลิเคชันสำหรับโทรศัพท์มือถือที่สามารถจำแนกสินค้าผ่านการแสดงผลรูปสินค้า และสามารถซื้อสินค้าได้
2. เว็บไซต์ที่สามารถแสดง และจัดการข้อมูลสินค้า รวมถึงรายงานข้อมูลการขาย เพื่อให้ร้านค้าสามารถจัดการการขายได้สะดวกยิ่งขึ้น

1.5 เทคโนโลยีและเครื่องมือที่ใช้

1. Python และ Aiortc : สำหรับพัฒนาในส่วนของ Backend การรับข้อมูลสตรีมมิ่งจากแอปพลิเคชัน การฝึกสอนโมเดล และการจำแนกสินค้าโดยไม่ต้องคุยกันโดยใช้ภาษา Python
2. Next.js: สำหรับการพัฒนาเว็บไซต์สำหรับร้านค้า (Website Dashboard)
3. Flutter: สำหรับพัฒนาแอปพลิเคชันสำหรับโทรศัพท์มือถือ (Mobile Application)
4. WebSocket: เชื่อมต่อกับแอปพลิเคชันสำหรับโทรศัพท์มือถือ เพื่อถ่ายทอดสดภาพจากกล้องโทรศัพท์มือถือไปยังส่วนระบบบริการจำแนกสินค้า
5. Supabase: สำหรับเก็บฐานข้อมูลทั้งหมดทั้งในส่วนของแอปพลิเคชันสำหรับโทรศัพท์มือถือ และเว็บไซต์สำหรับร้านค้า

1.6 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	ม.ค. 2566	ก.พ. 2566	มี.ค. 2566	เม.ย. 2566	พ.ค. 2566	มิ.ย. 2566	ก.ค. 2566	ส.ค. 2566	ก.ย. 2566	ต.ค. 2566	พ.ย. 2566	ธ.ค. 2566	ม.ค. 2567	ก.พ. 2567	มี.ค. 2567
Planning															
Document															
Collect data															
Back-end development															
App development															
Dashboard development															
Payment development															
Testing															

ตารางที่ 1.1: แผนการดำเนินงาน

1.7 บทบาทและความรับผิดชอบ

นายพงศกร รัตนพันธ์ รหัส 630610749 มีหน้าที่ที่รับผิดชอบ ดังนี้

1. การพัฒนาแอปพลิเคชันสำหรับโทรศัพท์มือถือ
2. การรับ-ส่งข้อมูลระหว่างแอปพลิเคชันสำหรับโทรศัพท์มือถือ และระบบบริการจำแนกสินค้าผ่าน WebSocket
3. การจัดเก็บข้อมูลรูปภาพของสินค้า และการฝึกสอนโมเดลเพื่อทำระบบบริการจำแนกสินค้า

นางสาวศุภาริญ่า ศิลปสิทธิ์ รหัส 630610765 มีหน้าที่ที่รับผิดชอบ ดังนี้

1. การพัฒนาเว็บไซต์สำหรับร้านค้า (Website Dashboard)
2. จัดทำระบบฐานข้อมูลทั้งหมด ทั้งในส่วนของแอปพลิเคชันสำหรับโทรศัพท์มือถือ และเว็บไซต์สำหรับร้านค้า

1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม

โครงการนี้ลดความซับซ้อน และเวลาที่ลูกค้าจะต้องรอต่อແ/qu เพื่อจ่ายเงินของสินค้า และ เป็นจุดเริ่มต้นให้สังคมสามารถเข้าถึงเทคโนโลยีที่ก้าวหน้าไปจากเดิมในการใช้กิจวัตรประจำวันอย่างการซื้อสินค้า โดยเปลี่ยนมาใช้การบริการตนเองผ่านแอปพลิเคชันที่อำนวยความสะดวกอย่างอุปกรณ์มือถือที่ใช้งานกันอย่างแพร่หลาย ทำให้สังคมมีความมั่นคงและความหลากหลายมากขึ้น ตอบโจทย์ความต้องการ และรูปแบบการใช้ชีวิตของผู้คนในยุคสมัยใหม่ ช่วยเหลือเลี้ยงปัญหาทางสุขภาพกาย ที่อาจเกิดจากการยืนรอชำระสินค้า หรือการใช้สายตาในการหาข้อมูลสินค้า และพัฒนาสุขภาพจิตจากประสบการณ์การซื้อสินค้าที่ดีขึ้น

อีกทั้งยังเป็นอักเสบทางเลือกหนึ่งในการที่ร้านค้าจะมาใช้ระบบบริการตนเองที่มีการจัดการที่ดี เจ้าของกิจการร้านค้าปลีกสามารถจัดการบริหารร้านค้าได้สะดวก และมีประสิทธิภาพมากขึ้น ทำให้เกิดความคุ้นเคยกับวัฒนธรรมการซื้อของแบบบริการตนเองในสังคมประเทศไทยมากขึ้น รวมถึงเป็นต้นแบบในการพัฒนาโครงการในลักษณะเดียวกันเพื่อความก้าวหน้าทางเทคโนโลยีในประเทศต่อไป

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

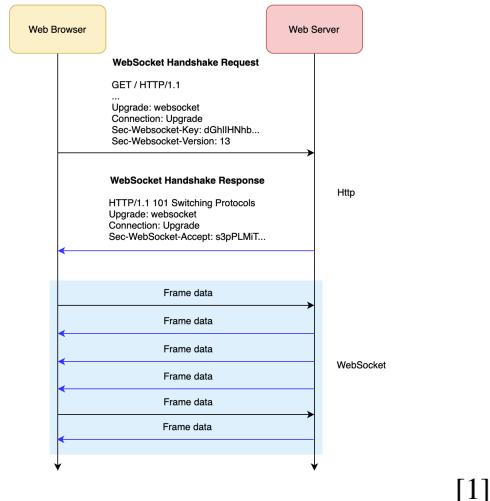
ทฤษฎีที่สำคัญของโครงงานนี้ คือ องค์ความรู้ในด้านของ ความฉลาดเชิงคำนวณ(Computational Intelligence) และ การส่งข้อมูลจากแอปพลิเคชัน(Streaming) ไปยัง Backend ผ่าน WebSocket เพื่อให้ Backend ที่เป็น Computational Intelligence ทำการจำแนกสินค้าผ่านรูปภาพ(Classification products)

2.1 WebSocket for Streaming image [1]

WebSocket (Computer Communications Protocol)

เป็น Protocol ชนิดหนึ่ง ซึ่งทำงานบน Socket ที่เชื่อมต่อ กันระหว่าง อุปกรณ์ที่เป็นการเชื่อมต่อแบบ TCP (Transmission Control Protocol) ดังรูป 2.1

โดย WebSocket สามารถรองรับการทำ Full Duplex หรือ Bidirectional Communication (การสื่อสารแบบสองทิศทาง หมายถึง เป็นผู้รับและผู้ส่งได้ในเวลาเดียวกัน) ซึ่งทำให้นิยมนำมาใช้กับระบบที่ต้องการการอัปเดทข้อมูลแบบ Realtime เช่น ระบบ Chat, Notification, Game, Developer Tools ต่างๆ โดยในโครงงานนี้ได้ใช้ในการรับข้อมูลสตรีมมิ่งจากระหว่าง Mobile application และ Classification server เพื่อส่งข้อมูลแบบ Realtime



รูปที่ 2.1: WebSocket connection

2.2 Artificial Neural Networks [2]

เป็นแขนงหนึ่งของ (Computational Intelligence) ซึ่งได้รับแรงบรรดาลใจมาจากการทำงานของสมองของมนุษย์ ซึ่งประกอบด้วยหน่วยเล็กๆ เรียกว่า cell Neuron แต่ละ Neuron ก็จะเชื่อมต่อโยงกันด้วยเส้นประสาทเรียกว่า ไซแนปส์ (Synapse) เพื่อส่งสัญญาณไฟฟ้า ที่เกิดจากสิ่งเร้าต่างๆ ว่าจะตอบสนองต่อสิ่งเร้านั้นอย่างไร โดยแต่ละ Neuron จะได้รับ Input หลาย ๆ ส่วนจากกิ่งก้านสาขาของ Dendrite และนำมาประมวลผล ออกมายield 1 Output ออกไปที่ Axon เพื่อส่งต่อไปให้ Dendrite ของ Neuron อื่น ๆ ใช้เป็น Input ต่อไป

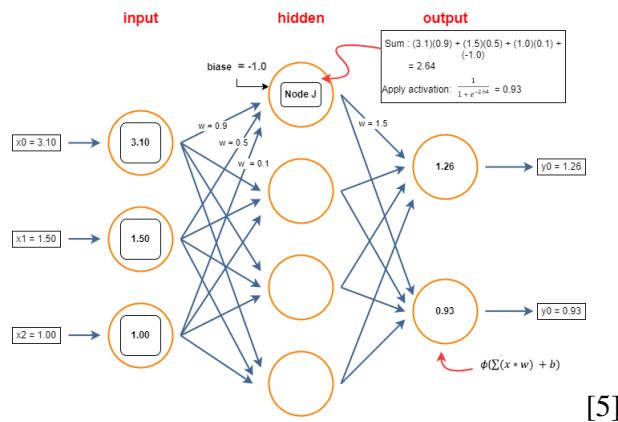
เมื่อมนุษย์เติบโตขึ้น หาก Neuron ไหนตอบสนองต่อสิ่งเร้าประเภทไหนได้ดี ก็จะสามารถส่งสัญญาณไฟฟ้าที่มีความแรงมากกว่า Neuron อื่นๆ เมื่อ Neuron หลายๆ อันต่อกันหลายๆ Layer ก็จะกลายเป็น Neural network ของมนุษย์

โดยในทาง Computational Intelligence จะใช้ node เป็นตัวแทนของ Neuron โดยจะเรียกเป็นชั้นๆ (Layer) โดยสัญญาณที่ส่งออกจากแต่ละ node จะมี weight ที่กำหนดความแรงของสัญญาณนั้นๆ เมื่อมีหลายๆ node และต่อกันหลายๆ layers ก็จะกลายเป็น neural network ขนาดใหญ่ เรียกว่า deep learning ซึ่งเลียนแบบการการทำงานของสมองมนุษย์ ทำให้ computers สามารถ process ข้อมูลในลักษณะเดียวกับที่ สมองของมนุษย์ทำการประมวลผลข้อมูล

2.2.1 Multilayer Perceptron [3]

เป็นส่วนพื้นฐานของ neural network เป็นการ ที่ node ในแต่ละ layer เชื่อมต่อกับ ทุก node ของ layer+1 (fully connected) โดยทุกเส้นการเชื่อมต่อของ $node_i$ กับ $node_j$ จะมี weight w_{ji} ซึ่งเป็นความแรงของสัญญาณอยู่

โดยจะมี Input layer สำหรับข้อมูล (สิ่งเร้า) hidden layer ในการตัดสินใจ และ output layer ในการเลือกการกระทำกับสิ่งเร้านั้นๆ



รูปที่ 2.2: Multilayer perceptron

2.2.2 Multilayer perceptron - Feed forward [4]

โดยในแต่ละ node ที่ไม่ใช่ Input layer จะรับค่าผลรวมจาก node ก่อนหน้า เป็นผลรวมจากทุก Input (ทุก Dendrite) ของ node นั้นๆ

โดยสมการผลรวม Input ของ $node_j$

$$node_j = v_j = \sum (\forall w_{j_i}) + biase \quad (2.1)$$

Activation function [6] คือ พังก์ชันที่รับ ผลรวมจากทุก Input (v_j) และคำนวณว่าจะส่งต่อเป็น Output เท่าไร ซึ่งมี function มากมายที่นิยมใช้ทำ Activation function ตัวอย่างเช่น

- Linear activation: $\phi(\mathbf{v}) = a + \mathbf{v}'\mathbf{b}$,
- ReLU activation: $\phi(\mathbf{v}) = \max(0, a + \mathbf{v}'\mathbf{b})$,
- Heaviside activation: $\phi(\mathbf{v}) = 1_{a+\mathbf{v}'\mathbf{b}>0}$,
- Logistic activation: $\phi(\mathbf{v}) = (1 + \exp(-a - \mathbf{v}'\mathbf{b}))^{-1}$.

โดยในทุกๆ $node_j$ จะมี Activation function สำหรับคำนวณค่า output ที่จะส่งต่อไปยัง layer ถัดไป

$$outputnode_j = Y_j = y(v_i) \quad (2.2)$$

ตัวอย่าง ReLU Activation function ค่าที่ส่งออกไปยัง layer ถัดไปจะเป็น

$$y(v_i) = \max(0, v_i) \quad (2.3)$$

2.2.3 Classification

จาก Input layer ข้อมูลจะถูกส่งต่อไปยัง Layer ถัดๆไป เรียกว่า Feed forward และเมื่อถึง Layer สุดท้ายของ Neural Network เรียกว่า Output layer โดยส่วนใหญ่ใน output layer นี้จะมีจำนวนของ node เท่ากับจำนวนของ class ของข้อมูลที่จะทำการ classification โดยแต่ละ node ใน Output layer จะเป็นตัวแทนของ class ซึ่ง หาก node ได้ให้ค่า Output node เ酵ะที่สุด Input data ก็ถูก classify เป็น class ของ node นั้นๆ

2.2.4 Training

คือการฝึกสอนโดยใช้ dataset เปลี่ยน weight ในแต่ละเส้นของ MLP เพื่อให้ตอบสนองต่อ Input ให้ใกล้เคียงกับสิ่งที่ควรจะเป็นมากขึ้น โดยหนึ่งในวิธีการเปลี่ยน weight คือ backpropagation

โดยค่า error ที่ได้มาจากการ output layer กับ ค่าที่ควรได้จาก Input ที่ได้เข้าไป เพื่อเปลี่ยน weight ให้ค่า error มีค่าน้อยลงกว่าเดิม

$$outputnode_j = Y_j = y(v_i) \quad (2.4)$$

สามารถคำนวณหาค่า error ในแต่ละ node ของ output layer ได้โดย

$$e_j(n) = d_j(n) - y_j(n) \quad (2.5)$$

where $d_j(n)$ is the desired target value , and $y_j(n)$ value produced by the perceptron at node $_j$

$$\mathcal{E}(n) = \frac{1}{2} \sum_{\text{output node } j} e_j^2(n). \quad (2.6)$$

และใช้ gradient descent ในการเปลี่ยน weight ในแต่ละเส้น

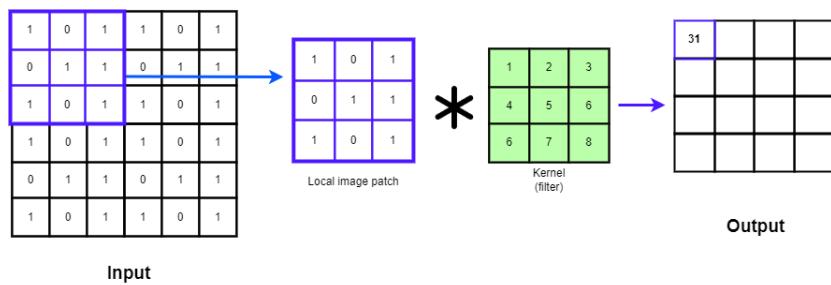
$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial v_j(n)} y_i(n) \quad (2.7)$$

2.3 Convolutional neural network

เป็น Computational intelligence ที่ออกแบบมาในงานด้านรูปภาพ ใช้การดำเนินการที่เรียกว่า ”convolution” (สัญลักษณ์ *) เป็นการดำเนินการทางคณิตศาสตร์ ระหว่างสองฟังก์ชัน $f * g$ เพื่อคุณการเปลี่ยนแปลงของฟังก์ชัน (f) เมื่อมีฟังก์ชัน (g) เข้ามา ใช้ในการเพื่อประมวลผลข้อมูลภาพ

$$(f * g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau. \quad (2.8)$$

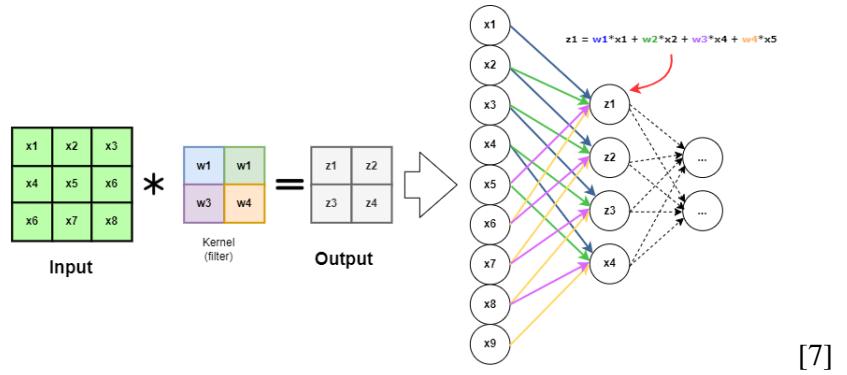
โดยจะแบ่งส่วนของภาพเป็นส่วนย่อยๆ ไปทั่วทั้งรูปภาพ และนำไป convolution กับ ”filters” หรือที่เรียกว่า ”kernels” เพื่อหา patterns ของรูปภาพ ที่เรียกว่า feature map



รูปที่ 2.3: Convolution

CNNs ต่างจาก Neural Networks อื่นๆ ตรงที่ shared-weight ร่วมกัน ซึ่งทำให้มีความสามารถในการแยกแยะ patterns ได้ดี

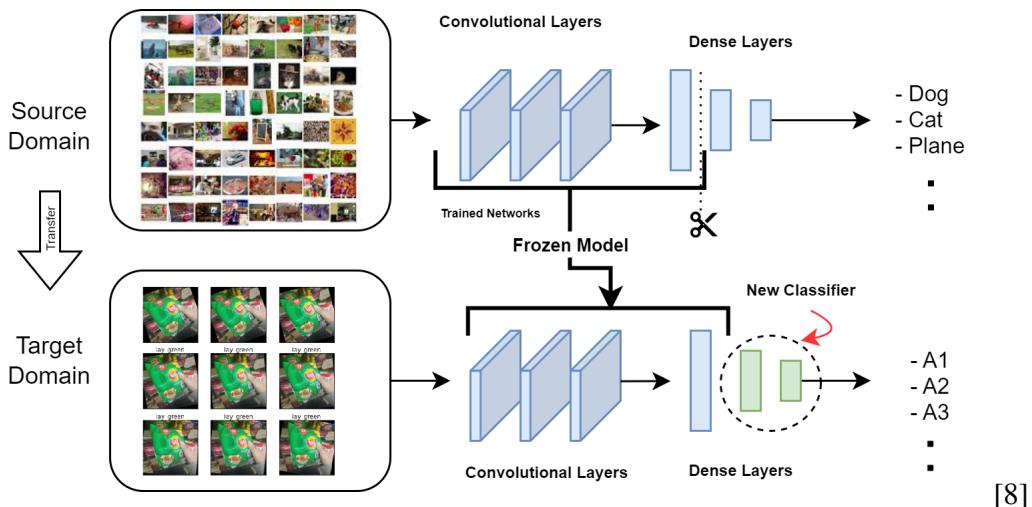
หากเขียนในโครงสร้าง Neural Network จะได้ดังรูป 2.4



รูปที่ 2.4: Convolutional Neural Network

2.4 Transfer learning

เป็นเทคนิคที่นำโมเดล ที่ผ่านการฝึกฝนจนเก็บัญหา ในงานอื่นๆที่มีความคล้ายคลึงกัน นำมาเป็นโมเดลตั้งต้น สำหรับโมเดลในการเก็บัญหาใหม่ๆ ตัวอย่างเช่น โมเดลที่ได้รับการฝึกฝนให้จดจำวัตถุในภาพสามารถใช้ เพื่อรับบุตถุที่คล้ายกันในภาพต่างๆ ได้ เมื่อภาพใหม่จะมีสภาพแสงหรือพื้นหลังต่างกันก็ตาม กุญแจสำคัญคือการระบุคุณสมบัติที่ร่วมกันระหว่างโมเดลต้นทางและโมเดลเป้าหมาย วิธี Transfer Learning ที่ได้รับความนิยมมากที่สุดวิธีหนึ่งคือ fine-tuning คือการใช้โมเดลที่ผ่านการ pre-trained มาแล้ว นำมา train ต่อบน ชุดข้อมูลใหม่ และใช้ learning rate น้อยๆ เพื่อป้องกัน weight ที่เคยผ่านการฝึกสอนจนมีความแม่นยำเปลี่ยนแปลงไปมาก จนไม่มีความแม่นยำ อีกวิธีหนึ่งคือ feature extraction ซึ่งเกี่ยวข้องกับการใช้ โมเดลที่ผ่านการ pre-trained เป็นตัวแยกคุณลักษณะของ ข้อมูล และสร้าง โมเดลใหม่เพื่อ train จากคุณลักษณะเหล่านี้ที่โมเดลตั้งต้นแบ่งแยกออกมาได้



รูปที่ 2.5: The concept of transfer learning

Transfer learning สามารถอธิบายได้ในรูปของ domains และ tasks.

โดย domain \mathcal{D} ประกอบไปด้วย: feature space \mathcal{X} และ marginal probability distribution $P(X)$, ซึ่ง $X = \{x_1, \dots, x_n\} \in \mathcal{X}$.

จะได้ว่าหากมี domain ใด, $\mathcal{D} = \{\mathcal{X}, P(X)\}$, หน้าที่ของ domain นั้นจะมี 2 ส่วนคือการ label space \mathcal{Y} และเป็น objective predictive function $f : \mathcal{X} \rightarrow \mathcal{Y}$.

ซึ่งฟังction f จะถูกใช้ในการ ทำนาย ข้อมูลที่มีความสอดคล้องกันของ $f(x)$ ของข้อมูลใหม่ x . โดยหน้าที่ของ domain ,โดยเขียนแทนด้วยสัญลักษณ์ $\mathcal{T} = \{\mathcal{Y}, f(x)\}$, ทำการเรียนรู้จากชุดข้อมูลฝึกสอนที่ เป็นคู่อันดับ $\{x_i, y_i\}$, โดยที่ $x_i \in X$ และ $y_i \in Y$.

หากมี source domain ตั้งต้น \mathcal{D}_S และ learning task ตั้งต้น \mathcal{T}_S , target domain \mathcal{D}_T และ learning task เป้าหมาย \mathcal{T}_T โดยที่ $\mathcal{D}_S \neq \mathcal{D}_T$, หรือ $\mathcal{T}_S \neq \mathcal{T}_T$,

Transfer learning นั้นช่วยในการพัฒนาการเรียนรู้ของ target predictive function ที่ต้องการ $f_T(\cdot)$ ใน \mathcal{D}_T จากการใช้ความรู้ที่ได้รับจาก \mathcal{D}_S และ \mathcal{T}_S

บทที่ 3

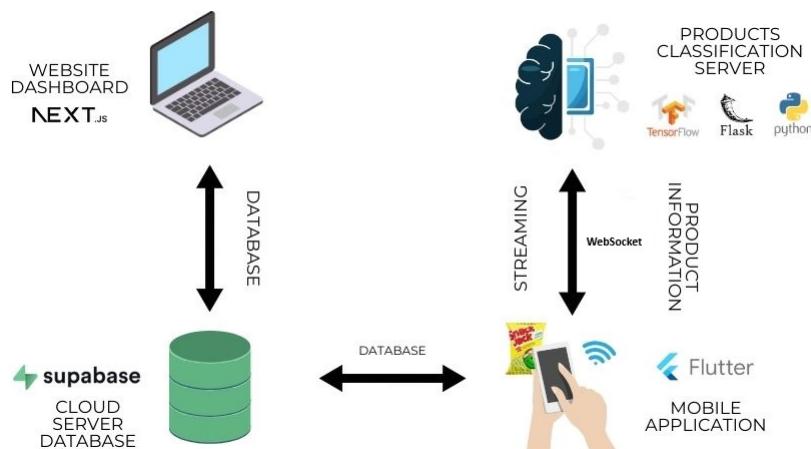
โครงสร้างและขั้นตอนการทำงาน

ในบทนี้จะกล่าวถึงหลักการนำทฤษฎีที่เกี่ยวข้องมาประยุกต์ใช้ในการออกแบบ และพัฒนาระบบ

3.1 โครงสร้างของระบบ

ระบบจัดการการซื้อขายปลีกอัตโนมัติด้วยตนเอง โดยใช้เทคนิคความฉลาดเชิงคำนวณนั้นประกอบไปด้วยส่วนต่าง ๆ หลายส่วน เริ่มตั้งแต่ส่วนของแอปพลิเคชันบนโทรศัพท์มือถือ ที่จะถูกใช้งานโดยลูกค้าที่เข้ามาเลือกซื้อสินค้าในร้านค้า โดยมีจุดประสงค์หลักคือ การอำนวยความสะดวกให้ลูกค้าสามารถชำระเงินได้ด้วยตนเองโดยไม่ต้องผ่านจุดชำระเงินในร้าน ซึ่งการทำงานหลัก ๆ ของแอปพลิเคชันคือการให้ลูกค้าเลือกสินค้าลงตระกร้าเพื่อชำระเงิน นอกจากนี้ยังมีฟังก์ชันสำคัญในการจำแนกสินค้าผ่านการสตรีมเมิ่ง ซึ่งอำนวยความสะดวกให้ลูกค้าสามารถหยิบสินค้าในร้าน แล้วสแกนรูปสินค้าเพื่อดูข้อมูลรายละเอียด หรือกดเข้าตรวจสอบได้โดยไม่ต้องเลื่อนหน้าจอแอปพลิเคชันเอง โดยในการจำแนกประเภทสินค้า จะมีการสื่อสารไปยังระบบให้บริการจำแนกสินค้าซึ่งพัฒนาจากเทคโนโลยีความฉลาดเชิงคำนวณ จนนั้นข้อมูลธุรกิจต่าง ๆ ที่ลูกค้ากระทำบนแอปพลิเคชันจะถูกเก็บไว้บนฐานข้อมูลแบบคลาวด์ ซึ่งเป็นฐานข้อมูลที่ใช้ร่วมกับระบบเว็บไซต์สำหรับร้านค้า ทำให้เว็บไซต์ดังกล่าวสามารถนำข้อมูลการขายมาประมวลผล และแสดงให้ผู้ดูแลร้านค้าติดตามได้แบบเรียลไทม์ ในทางเดียวกันผู้ดูแลร้านค้าก็จะมีการจัดการเรื่องของข้อมูล และคลังสินค้าผ่านเว็บไซต์ร้านค้า ซึ่งข้อมูลดังกล่าวก็จะไปแสดงผลบนแอปพลิเคชันมือถือสำหรับลูกค้าเข่นกัน

โดยสรุปแล้ว ระบบจัดการการซื้อขายปลีกอัตโนมัติด้วยตนเอง โดยใช้เทคนิคความฉลาดเชิงคำนวณนั้นประกอบไปด้วยระบบ 4 ส่วนที่ทำงานร่วมกัน คือ 1. แอปพลิเคชันบนโทรศัพท์มือถือ (Mobile Application) 2. ระบบให้บริการจำแนกสินค้า (Classification Server) 3. ระบบฐานข้อมูล (Database) 4. เว็บไซต์สำหรับร้านค้า (Website Dashboard) โดยสามารถแสดงส่วนประกอบต่าง ๆ และการทำงานร่วมกันได้ ดังแผนภาพต่อไปนี้ 3.1



รูปที่ 3.1: โครงสร้างของระบบ CapSnap

3.2 ระบบบริการจำแนกสินค้าผ่านรูปภาพ

3.2.1 เตรียมชุดข้อมูลฝึกสอน

ข้อมูลที่ใช้ในการฝึกสอนโมเดล(Train model) ใช้รูปภาพจากสินค้าสินค้าจากห้อง 422 ทำการจัดเก็บข้อมูลรูปภาพ โดยใช้กล้องมือถือ ในการถ่ายภาพโดยสินค้า 1 ชนิด(class) ทำการถ่ายภาพ 8 รูป ในมุมที่แตกต่างกัน โดยโมเดลในโครงงานนี้จะใช้สินค้าทั้งหมด 51 class และเพิ่มอีก 1 class สำหรับพื้นหลัง (Background class) รวมเป็น 52 class จัดเก็บใน Google Drive ดังรูป 3.2

โดยทำการแยก รูปภาพตามหมวดหมู่ และทำการดึงข้อมูลมา ฝึกสอนโมเดล ผ่าน Google Colab โดยใน

```
directory/  
    class_label_1/  
        image1  
        image2  
        .  
        .  
    class_label_2/  
        .
```

รูปที่ 3.2: Google Drive

แต่ละรูป จะเป็นข้อมูล Array ขนาด 224x224x3 ตามข้อมูลของรูปภาพ

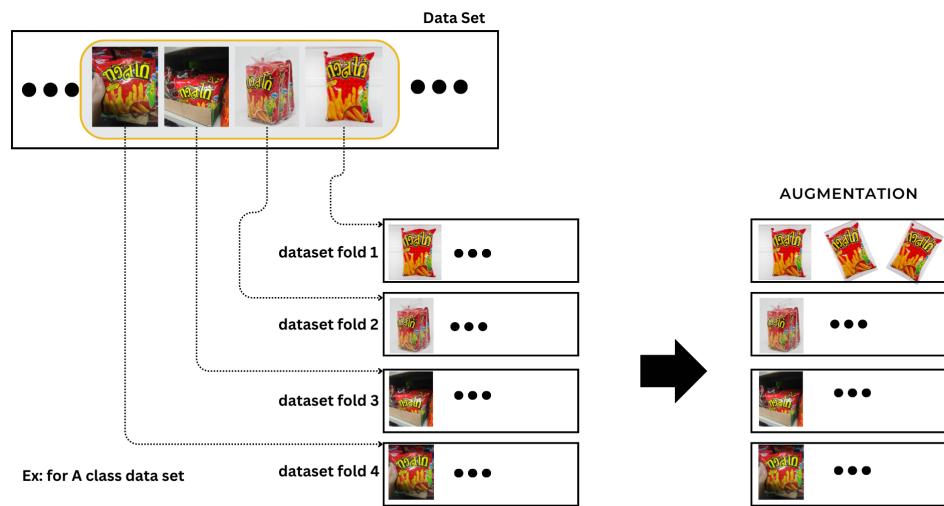
3.2.2 K-Fold Cross Validation

เพื่อตรวจสอบว่า โมเดล ที่เราทำการฝึกสอนมานั้นมีความแม่นยำเพียงพอ ไม่ว่าจะเจอชุดข้อมูล dataset ที่ข้อมูลแตกต่างกัน ในแต่ละ Fold ก็ยังสามารถมี Accuracy ที่มีความใกล้เคียงกัน ในแต่ละ fold โดยที่ โมเดลนั้นไม่แม่นยำแค่บาง dataset ซึ่งเป็นถือว่ามีการ bias ต่อ dataset

4-fold validation (k=4)



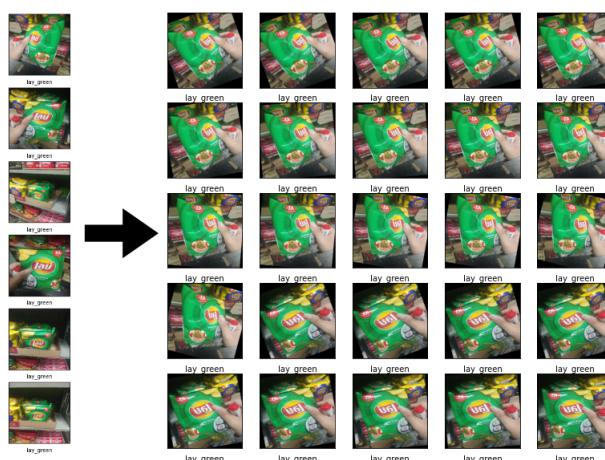
ซึ่งในโครงการนี้ ได้ทำการแบ่ง dataset เป็น ทั้งหมด 4 fold ดังรูป 3.3 โดยใน dataset แต่ละ fold ก็ จะทำการ Augmentation เพื่อเพิ่มจำนวนข้อมูลที่ใช้ในการฝึกสอนไม่เดล ของแต่ละ fold



รูปที่ 3.3: Dataset

3.2.3 Data Augmentation การเพิ่ม traing data

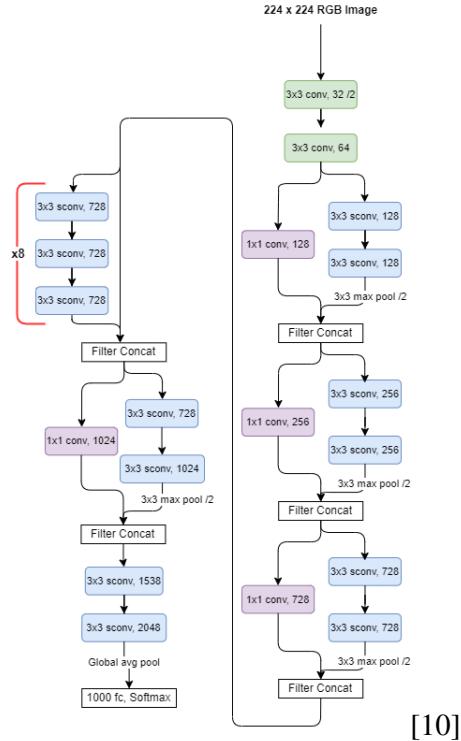
เพื่อให้มี train dataset ในจำนวนมาก โดยในแต่ละ 1 รูปภาพที่ถ่าย ซึ่งเป็นรูปภาพ RGB ขนาด 224x224 pixel นำไปสร้างเป็นรูปภาพใหม่ โดยใช้วิธีการ หมุน และ กลับด้าน ด้วยมุม -20,-15,-10,-5,0,5,10,15,20 องศา โดย 1 รูปภาพที่ผ่านการ Generate datasets จะกลายเป็น 18 รูปภาพซึ่งมีความแตกต่างกันเล็กน้อย ตามรูป 3.4



รูปที่ 3.4: Dataset generator

3.2.4 Model Architecture

โมเดลในโครงการนี้จะใช้ Xception Pre-trained ซึ่งเป็น Convolutional Neural Network ซึ่งมีความลึก 71 layers. เป็นโมเดลที่ผ่านการจากุปภาพต่าง ๆ มากกว่าล้านรูปภาพจาก ImageNet database ซึ่งจะมีโครงสร้างดังรูป 3.5



รูปที่ 3.5: Xception

โดยการนำ Xception Model มาใช้ในการแยกคุณลักษณะเด่นของรูปภาพ และ สร้างโมเดล มาต่อท้ายเพื่อ เรียนรู้ลักษณะเด่นจากที่ Xception ทำการแยกออกมามาได้

ซึ่ง โมเดล ที่จะใช้ก็จะเป็น Xception Model ที่รับ Input ขนาด 224 x 224 x 3 ตามขนาดรูปภาพ 224x224 pixel และตาม Layer RGB

โดย Xception เป็นโมเดล ตั้งต้นที่ผ่าน Pre-train จะทำหน้าที่เป็น Feature extraction ลักษณะเด่นของรูปภาพ โดย Xception จะล็อก lock ค่าของ weight เอาไว้เพื่อไม่ให้ค่าของ weight ที่ได้เรียนรู้ข้อมูลมาจาก ImageNet หายไป ซึ่ง Xception มี weight ทั้งหมด 20 ล้าน parameter

และนำ layer มาต่อห้ายโมเดล ด้วย Fully Connected Node ที่เรียกว่า Dense layer จำนวน 128 , 64 , 32 , 64 node และ layer สุดท้าย ซึ่งมีขนาดเท่ากับจำนวน ชนิดของสินค้า(Class) ซึ่งทำหน้าที่เป็น Classifier โดย Dense layer ซึ่งเชื่อมกันแบบ Fully Connected ที่สามารถเรียนรู้และปรับค่าของ weight ได้ ในโครงสร้างทดสอบนี้จะมี weight ที่สามารถเรียนรู้ หรือ เปลี่ยนแปลงได้ 12 ล้าน parameters โดยโครงสร้างของโมเดล จะเป็นดังรูป 3.6

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
xception (Functional)	(None, 7, 7, 2048)	20861480
flatten (Flatten)	(None, 100352)	0
dense (Dense)	(None, 128)	12845184
dense_1 (Dense)	(None, 64)	8256
dropout (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 32)	2080
dropout_1 (Dropout)	(None, 32)	0
dropout_2 (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 64)	2112
dropout_3 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 52)	3380

Total params:	33722492 (128.64 MB)
Trainable params:	12861812 (49.06 MB)
Non-trainable params:	20861480 (79.58 MB)

รูปที่ 3.6: Classification model

3.2.5 classification products

จากรูปภาพใน 52 class ผ่านการ generate datasets จะมี dataset ทั้งหมด 4432 sample ต่อ 1 Fold ทำการแบ่งเป็น train 3546 sample และ 886 sample สำหรับการ evaluate โดยจาก train 3546 แบ่ง 50% สำหรับการ validation ในระหว่างการ train model

สำหรับเป็น service ในการ classify products ของ application ผ่าน aiortc และเว็บ WebSocket

3.3 ระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ

3.3.1 ขอบเขตการพัฒนาระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ

1. ผู้คนทั่วไปสามารถลงทะเบียนเข้าใช้งานได้
2. สามารถใช้กล้องโทรศัพท์มือถือสแกนสินค้าเพื่อสติ๊กเกอร์ระบบบริการจำแนกสินค้า (Classification Server) เพื่อจำแนกสินค้าที่สแกน แล้วแสดงผล บนแอปพลิเคชัน
3. สามารถดูรายการสินค้าทั้งหมดในร้านได้
4. สามารถดูข้อมูลสินค้าที่เลือก หรือสแกนได้
5. สามารถเพิ่มหรือลดสินค้าในตะกร้าได้
6. สามารถเติมเงินเป็นเครดิตในแอปพลิเคชันได้
7. สามารถชำระสินค้าผ่านเครดิตเงินเดือนในตัวแอปพลิเคชัน

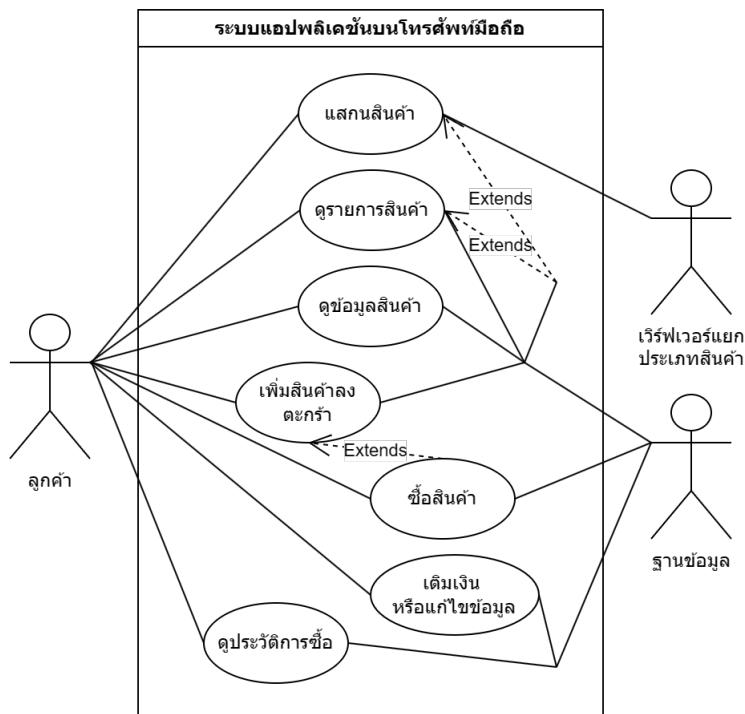
3.3.2 การออกแบบระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ

สำหรับระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือนี้ ได้ออกแบบให้เป็นแอปพลิเคชันที่ให้บริการบนระบบปฏิบัติการแอนดรอยด์ ทำการพัฒนาโดยใช้ Flutter ร่วมกับ Supabase ในการเชื่อมต่อ กับฐานข้อมูลที่ได้เตรียมไว้ รวมถึงใช้ในการระบบบัญชีตัวตน ลูกค้าทั่วไปสามารถติดตั้งแอปพลิเคชันบนโทรศัพท์มือถือ และลงทะเบียนใช้งาน เพื่อนำมาใช้ซื้อสินค้าในร้านด้วยตนเอง ผ่านการกดเลือกสินค้าในรายการ หรือสแกนสินค้าจริง ซึ่งระบบจะจำแนกสินค้าได้ผ่านการถ่ายทอดสด โดยมีการใช้บริการผ่าน WebSocket ซึ่งติดต่อ กับระบบบริการจำแนกสินค้า (Classification server) ที่ได้ทำการฝึกสอนไว้แล้ว จากนั้นจะมีการเชื่อมต่อ กับฐานข้อมูล แล้วตอบกลับมายังแอปพลิเคชันเพื่อแสดงผลให้กับผู้ใช้งาน ซึ่งหน้าการใช้งานทั้งหมดของแอปพลิเคชัน ประกอบไปด้วย

1. หน้าลงทะเบียน และลงชื่อเข้าใช้งาน (login & Register)
2. หน้าโปรไฟล์ผู้ใช้งาน และหน้าประวัติการซื้อสินค้า (Profile & Purchase History)
3. หน้าแก้ไขโปรไฟล์ผู้ใช้งาน และเติมเงิน (Edit Profile & Credit Topup)
4. หน้าแสดงรายการสินค้า (Stock)
5. หน้าแสดงข้อมูลสินค้า (Product Details)
6. หน้าตะกร้าสินค้า และจ่ายเงิน (Cart & Checkout)

Use Case

3.7



รูปที่ 3.7: Mobile Application Use Case Diagram

3.4 ระบบเว็บไซต์สำหรับร้านค้า

3.4.1 ขอบเขตการพัฒนาระบบเว็บไซต์สำหรับร้านค้า

- สามารถเข้าใช้งานได้ผ่านการลูกเพิ่มเป็นผู้ดูแลระบบร้านค้า (Administer) โดย Super Administer เท่านั้น ผู้คนทั่วไปไม่สามารถลงทะเบียนเข้าใช้ได้
- สามารถดูรายการสินค้าในแต่ละหมวดหมู่ และแก้ไขข้อมูลสินค้าในแต่ละชนิดได้แบบเรียลไทม์
- สามารถสถิติยอดขายสินค้าได้ทั้งแบบรายวัน รายสัปดาห์ รายเดือน และรายปี โดยแบ่งได้ 2 แบบ คือตามหมวดหมู่ของสินค้า และตามสินค้าทั้งหมด
- สามารถดูประวัติการซื้อของลูกค้าในแต่ละครั้ง พร้อมรายชื่อสินค้าที่ซื้อ จำนวนที่ซื้อ และจำนวนเงินที่จ่าย
- สามารถดูข้อมูลลูกค้าที่ลงทะเบียนเข้าใช้งานผ่านแอปพลิเคชันได้

ตารางที่ 3.1: แสดงการอธิบายกรณีการใช้งานกรณีสแกนสินค้าของระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ

Use Case ID:	1
Use Case Name:	สแกนสินค้า
Actor:	1. ผู้ใช้งานแอปพลิเคชัน 2. ฐานข้อมูล 3. ระบบบริการจำแนกสินค้า
Pre-Condition:	1. เข้าสู่ระบบ 2. ระบบจำแนกสินค้าเปิดให้บริการ
Post-Condition:	สินค้าที่สแกนแสดงผลบนแอปพลิเคชัน
Event flow:	1. กดเมนูสแกนสินค้า 2. เชื่อมต่อระบบบริการจำแนกสินค้า 3. ลูกค้าส่องกล้องมือถือไปที่สินค้า 4. ดึงข้อมูลจากฐานข้อมูล 5. ระบบแสดงสินค้าที่สแกนบนแอปพลิเคชัน
Alternative flow:	5. หากไม่ใช้สินค้าที่ขายในร้าน ระบบจะแจ้งเตือนว่าไม่มีข้อมูลสินค้าที่สแกนในระบบ

ตารางที่ 3.2: แสดงการอธิบายกรณีการใช้งานกรณีดูรายการสินค้าของระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ

Use Case ID:	2
Use Case Name:	ดูรายการสินค้า
Actor:	1. ผู้ใช้งานแอปพลิเคชัน 2. ฐานข้อมูล
Pre-Condition:	เข้าสู่ระบบ
Post-Condition:	ลูกค้าสามารถดูรายการสินค้าที่มีในร้านได้
Event flow:	1. กดเมนูรายการสินค้า 2. ดึงข้อมูลจากฐานข้อมูล 3. แสดงรายการสินค้าพร้อมจำนวนในคลัง
Alternative flow:	-

ตารางที่ 3.3: แสดงการอธิบายกรณีการใช้งานกรณีดูข้อมูลสินค้าของระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ

Use Case ID:	3
Use Case Name:	ดูข้อมูลสินค้า
Actor:	1. ผู้ใช้งานแอปพลิเคชัน 2. ฐานข้อมูล
Pre-Condition:	1. เข้าสู่ระบบ 2.1 กดเมนูรายการสินค้า แล้วกดเลือกสินค้าที่ต้องการ 2.2 กดเมนูสแกนสินค้า แล้วสแกนสินค้าสำเร็จ
Post-Condition:	แสดงหน้าต่างรายละเอียดข้อมูลของสินค้าที่เลือก หรือสแกน
Event flow:	1. กดปุ่มดูรายละเอียดข้อมูล 2. แสดงข้อมูล
Alternative flow:	2. ไม่พบข้อมูลรายละเอียดของสินค้าในฐานข้อมูล

ตารางที่ 3.4: แสดงการอธิบายกรณีการใช้งานกรณีเพิ่มสินค้าลงตะกร้าของระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ

Use Case ID:	4
Use Case Name:	เพิ่มสินค้าลงตะกร้า
Actor:	1. ผู้ใช้งานแอปพลิเคชัน 2. ฐานข้อมูล
Pre-Condition:	1. เข้าสู่ระบบ 2.1 กดเมนูรายการสินค้า แล้วกดเลือกสินค้าที่ต้องการ 2.2 กดเมนูແສກสินค้า แล้วແສກสินค้าสำเร็จ
Post-Condition:	แสดงหน้าต่างรายละเอียดข้อมูลของสินค้าที่เลือก หรือແສກ
Event flow:	1. เลือกจำนวนที่ต้องการซื้อ 2. กดปุ่มตะกร้า 3. ระบบเพิ่มสินค้า ในจำนวนที่เลือกเข้าตะกร้า
Alternative flow:	-

ตารางที่ 3.5: แสดงการอธิบายกรณีการใช้งานกรณีเพิ่มสินค้าลงตะกร้าของระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ

Use Case ID:	5
Use Case Name:	ซื้อสินค้า
Actor:	1. ผู้ใช้งานแอปพลิเคชัน 2. ฐานข้อมูล
Pre-Condition:	1. เข้าสู่ระบบ 2. มีสินค้าอย่างน้อย 1 ชิ้นในตะกร้า
Post-Condition:	ลูกค้าซื้อสินค้าสำเร็จ ระบบตัดเงินเครดิตของลูกค้า และเก็บรายการสินค้าที่ซื้อ ในประวัติการซื้อสินค้า
Event flow:	1. เลือกเมนู Cart 2. เช็คสินค้าในตะกร้าว่าถูกต้องตามที่ต้องการหรือไม่ 3. กดปุ่มซื้อสินค้า 4. บันทึก และแก้ไขข้อมูลบนฐานข้อมูล 5. แสดงผลการซื้อสินค้าสำเร็จ
Alternative flow:	กรณีที่ 1: หากต้องการลบสินค้าในตะกร้า 2. เช็คสินค้าในตะกร้าว่าถูกต้องตามที่ต้องการหรือไม่ กรณีที่ 2: หากสินค้าในคลังไม่พอจำนวนที่ซื้อ หรือหมด 4. เช็คจำนวนสินค้าที่มีบนฐานข้อมูลไม่พอ กับที่ต้องการซื้อ 5. แสดงผลการซื้อสินค้าไม่สำเร็จ

ตารางที่ 3.6: แสดงการอธิบายกรณีการใช้งานกรณีเพิ่มสินค้าลงตะกร้าของระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ

Use Case ID:	6
Use Case Name:	ดูประวัติการซื้อสินค้า
Actor:	1. ผู้ใช้งานแอปพลิเคชัน 2. ฐานข้อมูล
Pre-Condition:	เข้าสู่ระบบ
Post-Condition:	ลูกค้ารับรู้ข้อมูลผู้ใช้งาน และรายการสินค้าที่เคยซื้อ
Event flow:	<ol style="list-style-type: none"> 1. กดที่ชื่อผู้ใช้งาน 2. เข้าสู่หน้าໂປຣັບໝູນຜູ້ໃຊ້ 3. แสดงชื่อผู้ใช้ เครดิตເຈີນເຕີມ และรายการประวัติสินค้าที่เคยซื้อ
Alternative flow:	-

ตารางที่ 3.7: แสดงการอธิบายกรณีการใช้งานกรณีเพิ่มสินค้าลงตะกร้าของระบบแอปพลิเคชันสำหรับโทรศัพท์มือถือ

Use Case ID:	7
Use Case Name:	เติมเงิน และแก้ไขข้อมูล
Actor:	1. ผู้ใช้งานแอปพลิเคชัน 2. ฐานข้อมูล
Pre-Condition:	เข้าสู่ระบบ
Post-Condition:	ข้อมูลที่แก้ไข และเงินที่เติมถูกบันทุกลงฐานข้อมูล
Event flow:	<ol style="list-style-type: none"> 1. กดที่ชื่อผู้ใช้งาน 2. เข้าสู่หน้าໂປຣັບໝູນຜູ້ໃຊ້ 3. กดแก้ไข 4. กรอกข้อมูลที่ต้องการแก้ไข 5. กดຢືນຢັນ 6. แสดงผลข้อมูลที่แก้ไขสำเร็จ
Alternative flow:	-

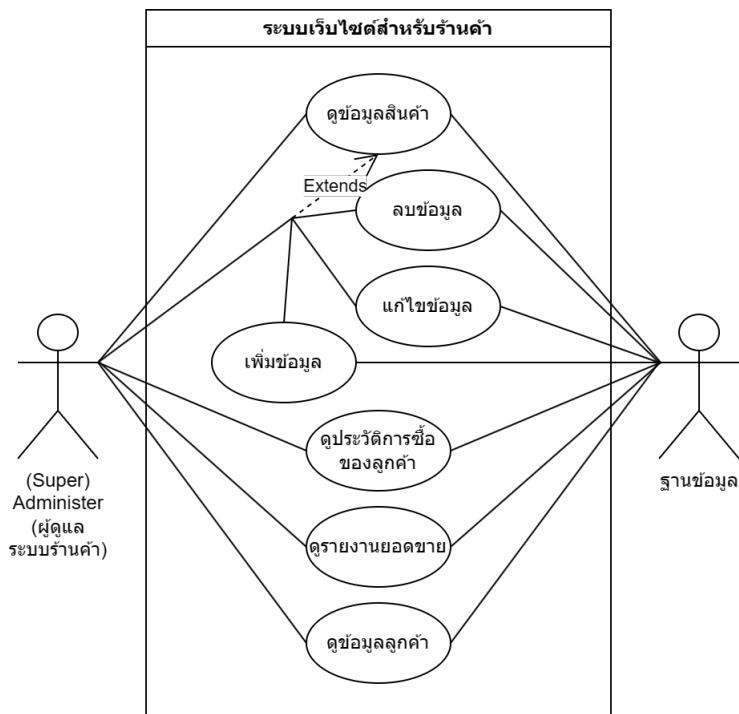
3.4.2 การออกแบบระบบเว็บไซต์สำหรับร้านค้า

ออกแบบในส่วนของเว็บไซต์ทางผู้ขายโดยใช้ Next.js ในการพัฒนาเว็บแอปพลิเคชัน (web application) แบบ React ร่วมกับ Supabase ในการโฮสต์เว็บไซต์ และเชื่อมต่อกับฐานข้อมูลที่ได้เตรียมไว้ รวมถึงใช้ในการทำ Authentication ในผู้ดูแลระบบร้านค้า และใช้ Tailwind ในการตกแต่ง และจัดระเบียบส่วนประสานผู้ใช้ ในส่วนของหน้าการใช้งาน มีหน้าต่าง ๆ ดังนี้

1. หน้าลงชื่อเข้าใช้งาน (Login)
2. หน้าการตั้งค่าร้านค้า และผู้ดูแลร้านค้า (Store Settings)
3. หน้าการจัดการสินค้า (Stock Manager)
4. หน้าแสดงรายงานยอดขายสินค้า (Selling Report)
5. หน้าแสดงข้อมูลลูกค้า (Customers)
6. หน้าแสดงประวัติการซื้อสินค้าของลูกค้า (Purchase History)

Use Case

3.83.9



รูปที่ 3.8: Website Application Use Case Diagram

ตารางที่ 3.8: แสดงการอธิบายกรณีการใช้งานกรณีดูข้อมูลสินค้าของระบบเว็บไซต์สำหรับร้านค้า

Use Case ID:	1
Use Case Name:	ดูข้อมูลสินค้า
Actor:	1. ผู้ดูแลระบบร้านค้า 2. ฐานข้อมูล
Pre-Condition:	เข้าสู่ระบบ
Post-Condition:	แสดงข้อมูลสินค้าตามหมวดหมู่ พร้อมรายละเอียดทั้งหมด
	1. เข้าสู่ระบบ
Event flow:	2. กดเมนู Stock Manager
Alternative flow:	-

ตารางที่ 3.9: แสดงการอธิบายกรณีการใช้งานกรณีลบข้อมูลของระบบเว็บไซต์สำหรับร้านค้า

Use Case ID:	2
Use Case Name:	ลบข้อมูลสินค้า
Actor:	1. ผู้ดูแลระบบร้านค้า 2. ฐานข้อมูล
Pre-Condition:	1. เข้าสู่ระบบ 2. กดเมนู Stock Manager
Post-Condition:	ลบหมวดหมู่สินค้า และ/หรือสินค้าออกจากฐานข้อมูล
Event flow:	<p>1. กดเมนู Stock Manager</p> <p>กรณีที่ 1: ลบหมวดหมู่สินค้า</p> <p>2. กดที่การ์ดหมวดหมู่สินค้า</p> <p>3. แสดงปุ่มลบหมวดหมู่สินค้าในการ์ดหมวดหมู่นั้น ๆ</p> <p>4. กดลบหมวดหมู่สินค้า และกดยืนยัน</p> <p>5. แสดงผลการลบหมวดหมู่สินค้าสำเร็จ</p> <p>กรณีที่ 2: ลบสินค้า</p> <p>2. กดที่การ์ดหมวดหมู่สินค้า</p> <p>3. แสดงปุ่มสินค้าทั้งหมดหมวดหมู่นั้น ๆ</p> <p>4. กดที่การ์ดสินค้า เพื่อแสดงปุ่มลบสินค้า</p> <p>5. กดลบสินค้า และกดยืนยัน</p> <p>6. แสดงผลการลบสินค้าสำเร็จ</p>
Alternative flow:	-

ตารางที่ 3.10: แสดงการอธิบายกรณีการใช้งานกรณีแก้ไขข้อมูลสินค้าของระบบเว็บไซต์สำหรับร้านค้า

Use Case ID:	3
Use Case Name:	แก้ไขข้อมูลสินค้า
Actor:	1. ผู้ดูแลระบบร้านค้า 2. ฐานข้อมูล
Pre-Condition:	1. เข้าสู่ระบบ 2. กดเมนู Stock Manager
Post-Condition:	แก้ไขข้อมูลสินค้าบนฐานข้อมูล
Event flow:	<ul style="list-style-type: none"> 1. กดที่การ์ดหมวดหมู่สินค้าที่แล้วกดการ์ดสินค้าที่ต้องการ เพื่อแสดงปุ่มแก้ไขสินค้า 2. กดปุ่มแก้ไขสินค้า 3. แสดงหน้าต่างการแก้ไขข้อมูล พร้อมข้อมูลเดิม 4. ผู้ดูแลระบบร้านค้ากรอกข้อมูลที่ต้องการแก้ไข และกดยืนยัน 5. แสดงผลการแก้ไขสินค้าสำเร็จ
Alternative flow:	-

ตารางที่ 3.11: แสดงการอธิบายกรณีการใช้งานกรณีเพิ่มข้อมูลสินค้าของระบบเว็บไซต์สำหรับร้านค้า

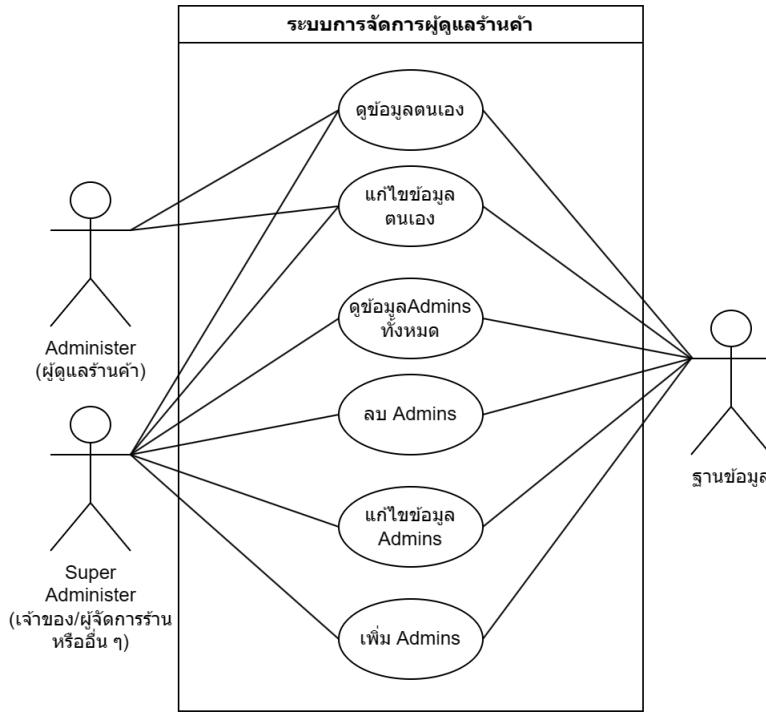
Use Case ID:	4
Use Case Name:	เพิ่มข้อมูลข้อมูลสินค้า
Actor:	1. ผู้ดูแลระบบร้านค้า 2. ฐานข้อมูล
Pre-Condition:	1. เข้าสู่ระบบ 2. กดเมนู Stock Manager
Post-Condition:	เพิ่มข้อมูลสินค้าบนฐานข้อมูล
Event flow:	<p>กรณีที่ 1: เพิ่มหมวดหมู่สินค้า</p> <ul style="list-style-type: none"> 1. กดปุ่มหมวดหมู่สินค้า 2. กรอกข้อมูลหมวดหมู่สินค้าใหม่ และกดยืนยัน 3. แสดงผลการเพิ่มหมวดหมู่สินค้าสำเร็จ <p>กรณีที่ 2: ลบสินค้า</p> <ul style="list-style-type: none"> 1. กดที่การ์ดหมวดหมู่สินค้า 2. แสดงปุ่มเพิ่มสินค้าในหมวดหมู่นั้น ๆ 3. กรอกข้อมูลสินค้าใหม่ และกดยืนยัน 4. แสดงผลการเพิ่มสินค้าสำเร็จ
Alternative flow:	-

ตารางที่ 3.12: อธิบายกรณี ดูประวัติการซื้อของลูกค้าบนระบบเว็บไซต์สำหรับร้านค้า

Use Case ID:	5
Use Case Name:	ดูประวัติการซื้อของลูกค้า
Actor:	1. ผู้ดูแลระบบร้านค้า 2. ฐานข้อมูล
Pre-Condition:	เข้าสู่ระบบ
Post-Condition:	ผู้ดูแลระบบร้านค้ารับรู้ประวัติการซื้อของลูกค้า
Event flow:	<ol style="list-style-type: none"> 1. กดเมนู Purchase History 2. แสดงรายการการซื้อของลูกค้าแต่ละครั้งในรูปแบบกริด ประกอบไปด้วยชื่อวัน และเวลาที่ซื้อ พร้อมรายการสินค้าที่ซื้อ และจำนวนเงินที่จ่าย
Alternative flow:	<ol style="list-style-type: none"> 2. ไม่มีรายการแสดงหากยังไม่มีลูกค้าซื้อสินค้าผ่านระบบแอปพลิเคชันบนโทรศัพท์มือถือ

ตารางที่ 3.13: อธิบายกรณี ดูรายงานยอดขายสินค้าของระบบเว็บไซต์สำหรับร้านค้า

Use Case ID:	6
Use Case Name:	ดูรายงานยอดขายสินค้า
Actor:	1. ผู้ดูแลระบบร้านค้า 2. ฐานข้อมูล
Pre-Condition:	เข้าสู่ระบบ
Post-Condition:	ผู้ดูแลระบบร้านค้าทราบยอดขายโดยรวมของร้านตามที่ต้องการของ
Event flow:	<ol style="list-style-type: none"> 1. กดเมนู Selling Report 2. แสดงข้อมูลยอดขายของวันที่ปัจจุบันผ่านการ์ดข้อมูล แผนภูมิ และตาราง ในโหมดรายวัน ตามสินค้าทั้งหมด กรณีที่ 1: 3. กดเลือกวันที่ที่ต้องการ กรณีที่ 2: 3. กดเลือกแสดงตามรายวัน/สัปดาห์/เดือน/ปี กรณีที่ 3: 3. กดเลือกแสดงตามหมวดหมู่สินค้า หรือสินค้าทั้งหมด 4. แสดงข้อมูลยอดขายผ่านการ์ดข้อมูล แผนภูมิ และตาราง ตามตัวเลือกที่เลือกในข้อ 3
Alternative flow:	<ol style="list-style-type: none"> 2. และ 4. แสดงไม่มีข้อมูลหากไม่มีประวัติการซื้อสินค้าในวันที่ที่เลือกบนฐานข้อมูล



รูปที่ 3.9: Administer System Use Case Diagram

3.5 การจัดทำฐานข้อมูล

ในส่วนของการจัดทำฐานข้อมูล ได้เลือกใช้ Supabase ซึ่งเป็นเซอร์วิสสำหรับการสร้างและจัดการฐานข้อมูล (Database) แบบ Open Source ที่ให้บริการในรูปแบบหอสต์ (hosted) ซึ่งเป็นเซอร์วิสที่ให้บริการแบบคลาวด์ (cloud services) ประกอบด้วยฐานข้อมูลแบบ PostgreSQL และบริการอื่น ๆ เช่น Authentication, Realtime Database, Storage และ APIs ทำให้สามารถจัดการข้อมูลได้สะดวก และมีประสิทธิภาพ โดยมีการเก็บ Database Schema เพื่อครอบคลุมการเก็บข้อมูลที่ใช้ในผู้ใช้งานแอปพลิเคชันสำหรับโทรศัพท์มือถือ และเว็บไซต์สำหรับร้านค้า แสดงได้ดังนี้ 3.10

1. Admin: เก็บข้อมูลของแอดมิน ซึ่งเป็นผู้ที่สามารถเข้าถึงและแก้ไขข้อมูลร้านค้าได้ โดยในร้านหนึ่ง ๆ สามารถมีแอดมินได้มากกว่าหนึ่งคน ซึ่งจะเก็บข้อมูลชื่อจริง รหัสผ่าน ที่อยู่อีเมล และรูปภาพของแอดมิน โดยมี Primary key เป็นชื่อผู้ใช้ หรือ username
2. Permission: เก็บสถานะของแอดมิน ได้แก่ Administers และ Super Administers Foreign key จากตาราง admin คือ admin_id
3. Product: เก็บข้อมูลของสินค้าแต่ละชนิดได้แก่ ชื่อสินค้า รูปสินค้า จำนวนสินค้าในคลัง ราคาสินค้า คำอธิบายสินค้า และหมวดหมุนสินค้าซึ่งเป็น Foreign key จากตาราง category และมี Primary key คือ รหัสหรือ product_id
4. Category: เก็บข้อมูลของหมวดหมุนสินค้า ซึ่งประกอบไปด้วยชื่อ และคำอธิบายของหมวดหมุนสินค้า โดยมี Primary key คือ รหัสหมวดหมุนสินค้า หรือ category_id

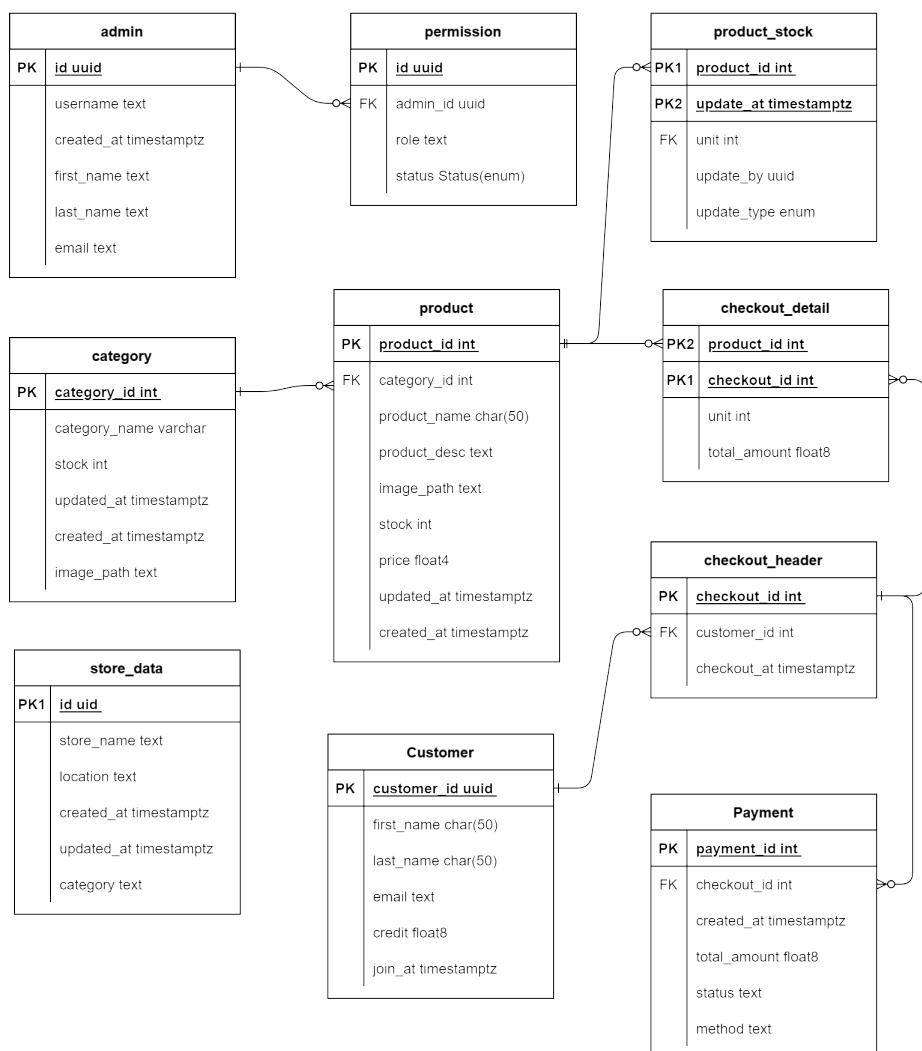
ตารางที่ 3.14: แสดงการอธิบายกรณีการใช้งานกรณีดูข้อมูลลูกค้าบนระบบเว็บไซต์สำหรับร้านค้า

Use Case ID:	7
Use Case Name:	ดูข้อมูลลูกค้า
Actor:	1. ผู้ดูแลระบบร้านค้า 2. ฐานข้อมูล
Pre-Condition:	เข้าสู่ระบบ
Post-Condition:	ผู้ดูแลระบบร้านค้ารับรู้ข้อมูลของลูกค้า
Event flow:	<ul style="list-style-type: none"> 1. กดเมนู Customers 2. แสดงข้อมูลของลูกค้าทั้งหมดผ่านตาราง และการดูข้อมูล ได้แก่ จำนวนลูกค้า ทั้งหมด จำนวนลูกค้าที่ลงทะเบียนใหม่ใน 7 วันที่ผ่านมา และข้อมูลของลูกค้าแต่ละคน
Alternative flow:	<ul style="list-style-type: none"> 2. ไม่มีรายการแสดงหากยังไม่มีลูกค้างานทะเบียนผ่านระบบแอปพลิเคชันบนโทรศัพท์มือถือ

5. Checkout_header: เก็บข้อมูลคำสั่งซื้อสินค้าทั้งหมดของร้านค้า ซึ่งประกอบไปด้วยวันที่ และเวลาที่สั่งซื้อ ยอดขายจากคำสั่งซื้อ หมายเลขระบุตัวตนลูกค้าหรือ customer_id จากตาราง customer และรหัสชำระเงิน หรือ payment_id จากตาราง Payment ซึ่งมี Primary key คือ หมายเลขระบุคำสั่งซื้อ หรือ checkout_id
6. Checkout_detail: เก็บข้อมูลสินค้าที่ถูกซื้อของแต่ละคำสั่งซื้อ โดยเก็บ Foreign key เป็นหมายเลขระบุคำสั่งซื้อ หรือ checkout_id จากตาราง checkout_header และรหัสสินค้า หรือ product_id จากตาราง product ควบคู่กันเป็น Primary key และเก็บข้อมูลอื่น ๆ ได้แก่ จำนวนสินค้าที่สั่งซื้อ และจำนวนเงินที่จ่ายทั้งหมด
7. Customer: เก็บข้อมูลลูกค้าของร้านค้า โดยเก็บชื่อ username ของลูกค้า และมี Primary key คือ หมายเลขระบุตัวตนลูกค้าหรือ customer_id
8. Product stock: เก็บข้อมูลการเปลี่ยนแปลงจำนวนสินค้าในคลัง โดยเก็บ Foreign key เป็นหมายเลขระบุสินค้า หรือ product_id จากตาราง Product และจำนวนสินค้าที่เพิ่ม หรือลด มี Primary key คือ ชื่อผู้ใช้งานและมีผู้เปลี่ยนจำนวนสินค้าในคลัง หรือ username จากตาราง Admin ควบคู่กับวันที่และเวลาที่ทำการเปลี่ยน หรือ timestamp
9. Payment: เก็บข้อมูลการชำระเงิน ซึ่งประกอบไปด้วยประเภทการชำระเงิน วันที่ และเวลาของการชำระเงิน และสถานะการชำระเงิน และเก็บ Foreign key เป็นหมายเลขระบุตัวตนลูกค้า หรือ customer_id มี Primary key คือ รหัสชำระเงิน หรือ payment_id
10. Store data: เก็บข้อมูลของร้านค้า ได้แก่ ชื่อ สถานที่ตั้ง และ ประเภทร้านค้า

เนื่องจาก Supabase นั้นสนับสนุนการใช้งาน PostgreSQL จึงได้มีการออกแบบฟังก์ชัน PostgreSQL เพื่อดำเนินการต่าง ๆ กับข้อมูลในฐานข้อมูลที่ซับซ้อน และต้องการการประมวลผลต่าง ๆ เช่น การค้นหาข้อมูล (query), การเพิ่มข้อมูล (insert), การปรับปรุงข้อมูล (update), และการลบข้อมูล (delete) ซึ่งได้ทำทั้งหมด 23 พังก์ชัน

System Database Schema



ຮູບທີ 3.10: System Database Schema

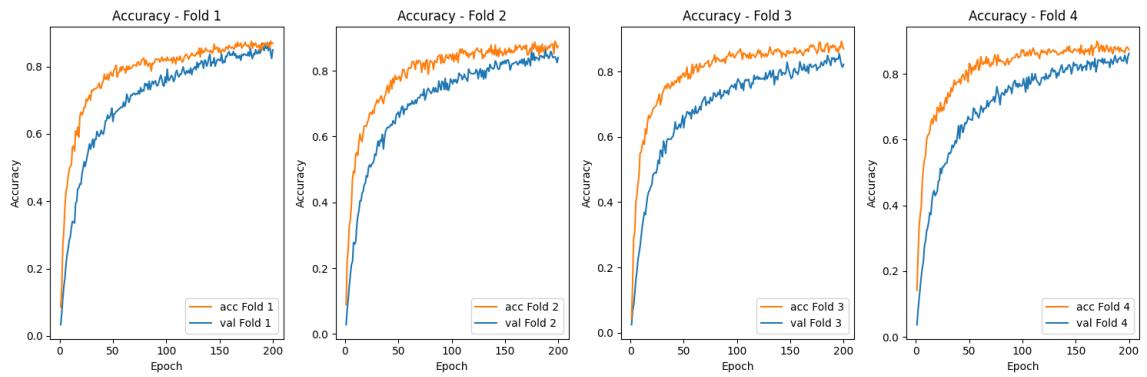
บทที่ 4

การทดลองและผลลัพธ์

4.1 Classification model

ผลลัพธ์จากการ train และ validation Classification model ด้วย 3546 sample เป็นจำนวน 200 Epoch ทั้งหมด 4 Fold

ตัวอย่าง ผลการทดลองของการ validation ด้วย 3546 sample ในแต่ละ Fold



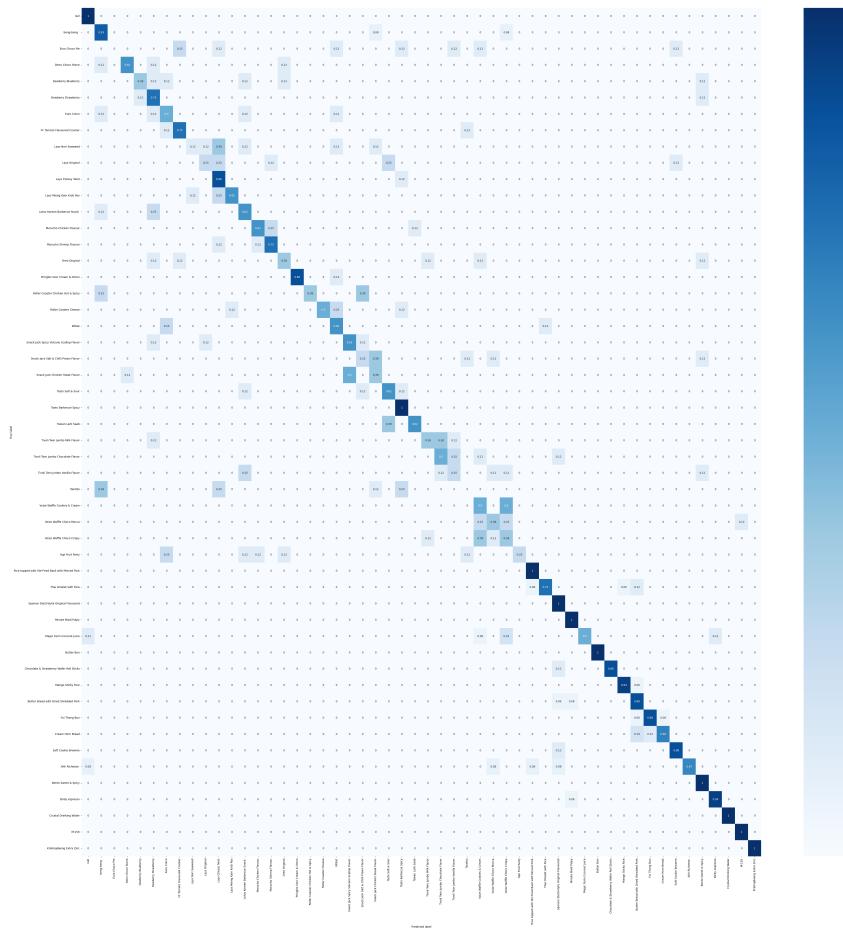
รูปที่ 4.1: Train fold

จากทั้ง 4 Fold ทำการเลือก model ที่มีค่า Validation Score สูงสุดจากใน 4 model โดยนำ model นี้ไปทำการ Evaluate ด้วย Dataset ที่ไม่เคยเห็น (Blind Test)

$$\text{cross_entropy} = 1.175 \quad (4.1)$$

$$\text{accuracy} = 0.72 \quad (4.2)$$

โดยผลการทำนายของ model จาก Blind Test ได้เป็น Confusion matrix ดังรูป 4.2



รูปที่ 4.2: Confusion matrix

4.2 ผลลัพธ์จากการฝึกสอน

การทดสอบ แบบชุดข้อมูล (Batch Detection) สำหรับการทดสอบ Model โดยได้ทำการทดลอง ฝึกสอน Model

โดยเป็นผลลัพธ์จาก Model ที่ดีที่สุดจากทั้ง 4 Fold ที่ฝึกสอนด้วย จำนวนรอบ(epoch) 100 และ 200 , batch size เป็น 32 และ 64 , และมีการใช้ Dropout Layer ที่ dropout rate เป็น 0 และ 0.5 ได้ผลลัพธ์ ดังตาราง 4.1 และผลลัพธ์จากการทดสอบด้วย Blind Dataset ดังตาราง 4.2

สำหรับการวัดผลได้แก่ Accuracy, Precision, Recall, และ F1-Score โดยโมเดลส่วนใหญ่ในการ Validation model ให้ผลลัพธ์ Accuracy ที่ประมาณ 70-80% ดังตาราง 4.1 แต่ในส่วนของการ Blind Test จะได้ Accuracy อยู่ที่ประมาณ 50-60% เท่านั้น ดังตาราง 4.2

ตารางที่ 4.1: ผลลัพธ์จากการทดสอบ Model โดยใช้ Train Dataset

Model name	Batch Size	Epoch	Dropout Rate	Metrics			
				Accuracy	Precision	Recall	F1-Score
64-100	64	100	-	0.605	0.483	0.651	0.555
64-100-5	64	100	0.5	0.673	0.391	0.823	0.530
64-200	64	200	-	0.674	0.546	0.767	0.638
64-200-5	64	200	0.5	0.698	0.481	0.837	0.611
32-100	32	100	-	0.734	0.572	0.797	0.666
32-100-5	32	100	0.5	0.641	0.469	0.766	0.582
32-200	32	200	-	0.859	0.625	0.806	0.704
32-200-5	32	200	0.5	0.766	0.559	0.875	0.682

ตารางที่ 4.2: ผลลัพธ์จากการทดสอบ Model โดยใช้ Blind Dataset

Model name	Batch Size	Epoch	Dropout Rate	Metrics			
				Accuracy	Precision	Recall	F1-Score
64-100	64	100	-	0.551	0.318	0.412	0.359
64-100-5	64	100	0.5	0.492	0.267	0.254	0.260
64-200	64	200	-	0.521	0.365	0.591	0.451
64-200-5	64	200	0.5	0.621	0.402	0.342	0.370
32-100	32	100	-	0.647	0.423	0.293	0.346
32-100-5	32	100	0.5	0.572	0.277	0.369	0.316
32-200	32	200	-	0.628	0.459	0.343	0.393
32-200-5	32	200	0.5	0.638	0.382	0.381	0.381

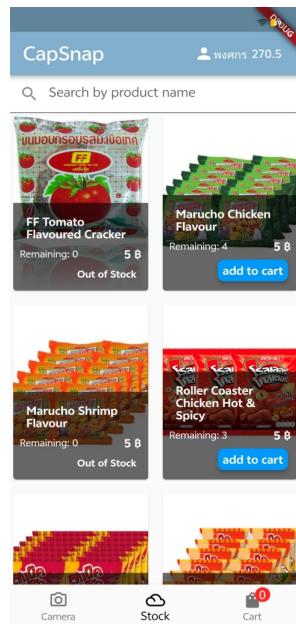
4.3 ผลการทำงานของระบบแอปพลิเคชันบนโทรศัพท์มือถือ

จากการปฏิบัติงาน สามารถพัฒนา Application สำหรับ ลูกค้า ได้เสร็จสมบูรณ์ โดยสามารถแสดงผลการทำงานของหน้า Application แบบตามแต่ละหน้า ได้ดังนี้

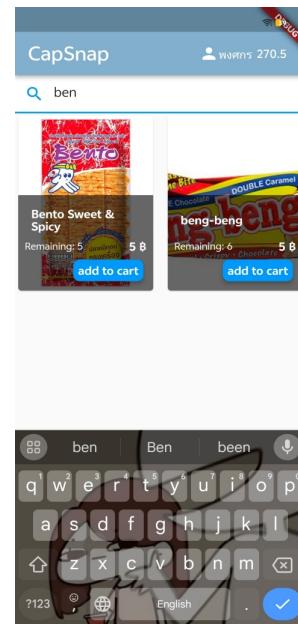
1. หน้าแสดงสินค้า และ รายละเอียดของสินค้า (Stock page)
2. หน้าแสดงสินค้า หรือ อ่าน barcode โดยใช้กล้อง (Streaming page)
3. หน้าลงทะเบียนเข้าใช้งาน และ ลงทะเบียนเข้าใช้งาน (Login / Register page)
4. หน้าการตั้งค่าผู้ใช้ (Profile / Edit profile page)
5. หน้าแสดงตะกร้าสินค้าที่ผู้ใช้เลือก (cart page)
6. หน้าแสดงประวัติการซื้อสินค้าของลูกค้า (History page)

4.3.1 หน้าแสดงสินค้า และ รายละเอียดของสินค้า (Stock page)

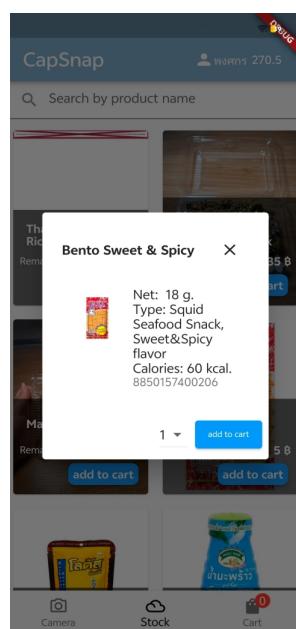
สามารถกดดูข้อมูลสินค้าที่เลือก ดูจำนวนคงเหลือ ราคา คำอธิบายสินค้า และ สามารถค้นหาสินค้าได้ และสามารถเพิ่มสินค้าเข้าไปยัง ตระกร้าสินค้า (cart) ได้



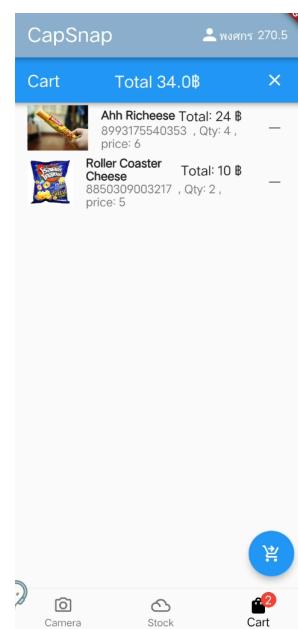
Stock



Search products



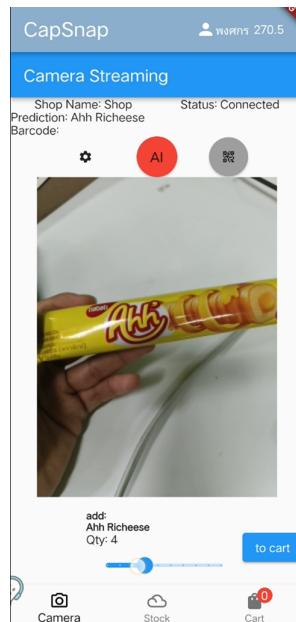
Product detail



Add to cart

4.3.2 หน้าแสดงสินค้า หรือ อ่าน barcode โดยใช้กล้อง (Streaming page)

ผู้ใช้สามารถใช้ กล้องของ smart phone ส่องไปที่สินค้าเพื่อทำการ Streaming รูปของสินค้า ไปยัง Classification Server หรือ สามารถ เลือกที่จะแสดง barcode ของสินค้า เพื่อทำการเพิ่มสินค้าเข้าตะกร้า (cart)ได้



Camera streaming



Camera barcode

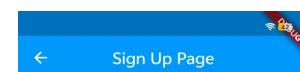
4.3.3 หน้าลงชื่อเข้าใช้งาน และ ลงทะเบียนเข้าใช้งาน (Login / Register page)

สำหรับหน้านี้ โดยสามารถลงชื่อเข้าใช้งาน หรือ สามารถสมัครสมาชิกได้ โดยการใช้ Email และ Password



Login

New User? [Create Account](#)



Sign Up

Already a User? [Login](#)

Login

Signup

4.3.4 หน้าการตั้งค่าผู้ใช้ (Profile page / Edit profile page)

สำหรับหน้านี้ เมื่อผู้ใช้ทำการเข้าสู่ระบบ สามารถทำการแก้ไขข้อมูลส่วนตัวของผู้ใช้ได้ และ สามารถเติมเงิน เสมือน (Mock Credit) เพื่อใช้ในการซื้อสินค้าได้

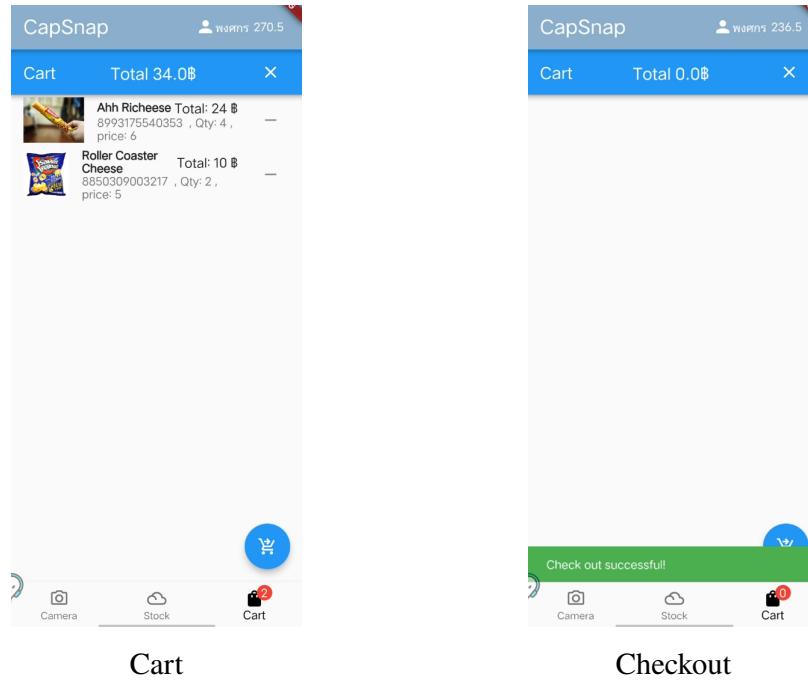


Profile

Edit profile

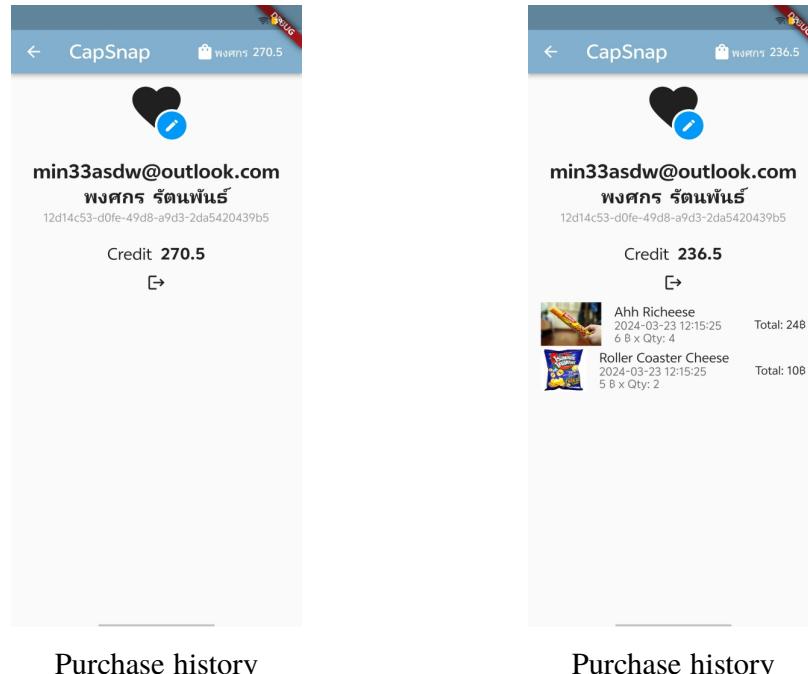
4.3.5 หน้าแสดงตะกร้าสินค้าที่ผู้ใช้เลือก (cart page)

สินค้าที่ผู้ใช้ได้ทำการเลือกเพิ่มเข้ามาในตะกร้า ไม่ว่าจะมาจาก Stock page, Streaming page จะถูกสรุป เป็นรายการและจำนวนเงินที่ผู้ใช้ต้องทำการจ่าย โดยผู้ใช้สามารถกด checkout และตัดเงินจาก Credit



4.3.6 หน้าแสดงประวัติการซื้อสินค้าของลูกค้า (History page)

โดยเมื่อทำการ checkout เสร็จแล้ว Credit ก็จะถูกลบติด รวมถึง Stock ของสินค้าภายในร้าน โดยประวัติการซื้อของผู้ใช้ก็จะสามารถดูได้ภายใน Application



4.4 ผลการ checkout

จากการทดลองและบันทึกผลของเรา ทำการบันทึกข้อมูลและค่าเฉลี่ยได้ดังนี้ (ขึ้นอยู่กับผู้ใช้งานแต่ละคน)

ตารางที่ 4.3: Camera classification process

Camera Classification	ชื่อของ 1 ชิ้น (วินาที)	ชื่อของ 5 ชิ้น (วินาที)
Camera streaming	2-5 s	10-25 s
Image processing	1-2 s	5-10 s
Payment transaction	2 s	2 s
Total Time Usage	5-9 s	17-37 s

ตารางที่ 4.4: Camera barcode process

Barcode payment	ชื่อของ 1 ชิ้น (วินาที)	ชื่อของ 5 ชิ้น (วินาที)
Camera capture	2-5 s	10-25 s
Barcode processing	0.5 s	2.5 s
Payment transaction	2 s	2 s
Total Time Usage	2.5-7 s	14.5-29.5 s

ตารางที่ 4.5: Line Checkout process

Line Checkout Cashier	ชื่อของ 1 ชิ้น(seconds)	ชื่อของ 5 ชิ้น (seconds)
Waiting in line	30-120 s	30-120 s
Cashier processing	10-20 s	20-40 s
Payment transaction	5-10 s	5-10 s
Total Time Usage	45-150 s	55-170 s

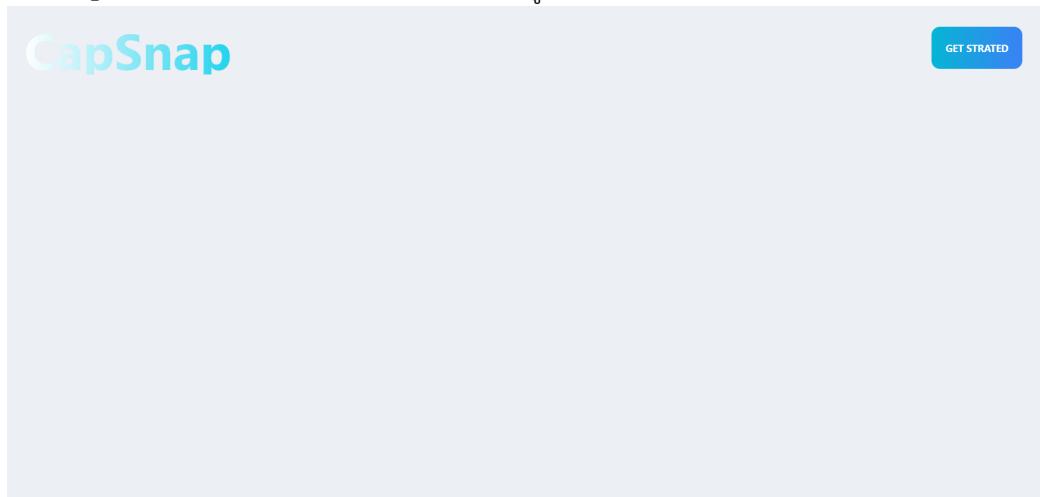
4.5 ผลการทำงานของระบบเว็บไซต์สำหรับร้านค้า

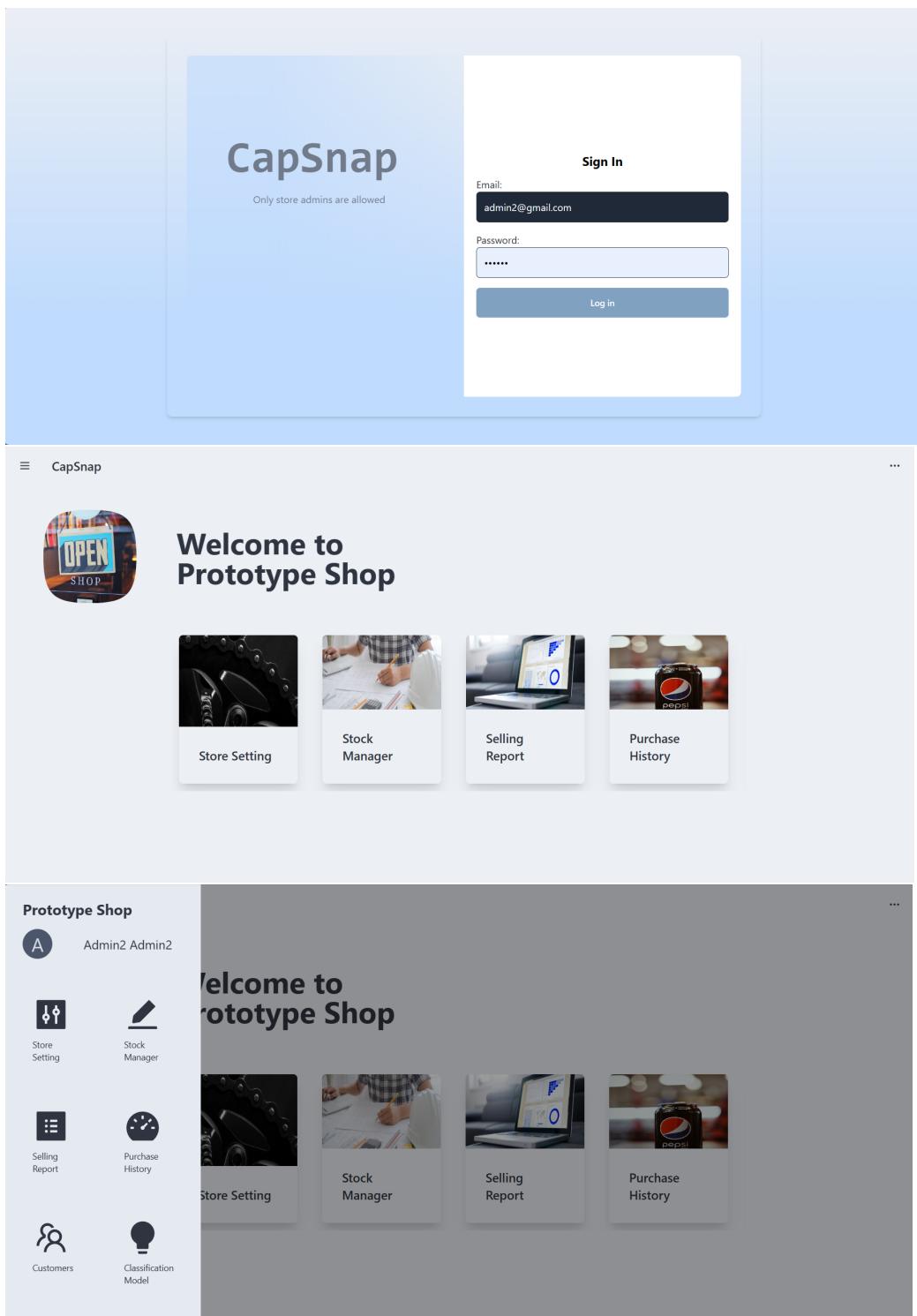
จากการปฏิบัติงาน สามารถพัฒนาระบบระบบเว็บไซต์สำหรับร้านค้าได้เสร็จสมบูรณ์ โดยสามารถแสดงผลการทำงานของหน้าเว็บไซต์แบ่งตามแต่ละหน้าของเว็บไซต์ได้ดังนี้

1. หน้าลงชื่อเข้าใช้งาน (Login)
2. หน้าการตั้งค่าร้านค้า และผู้ดูแลร้านค้า (Store Settings)
3. หน้าการจัดการสินค้า (Stock Manager)
4. หน้าแสดงรายงานยอดขายสินค้า (Selling Report)
5. หน้าแสดงข้อมูลลูกค้า (Customers)
6. หน้าแสดงประวัติการซื้อสินค้าของลูกค้า (Purchase History)

4.5.1 หน้าลงชื่อเข้าใช้งาน (Login)

สำหรับหน้านี้ จะเป็นหน้าแรกของการเข้าใช้งานระบบ โดยสามารถลงชื่อเข้าใช้งานได้เท่านั้น แต่จะไม่สามารถลงทะเบียนได้ เนื่องจากผู้ใช้งานระบบ ต้องเป็นผู้ดูแลระบบร้านค้า หรือพนักงานของร้าน ซึ่งจะถูกเพิ่มโดย Super Administer ซึ่งอาจเป็นเจ้าของร้าน หรือผู้จัดการร้าน โดยผู้ใช้งานสามารถขอข้อมูลรหัสผ่าน จาก Super Administer โดยตรงเพื่อนำมาใช้เข้าสู่ระบบ





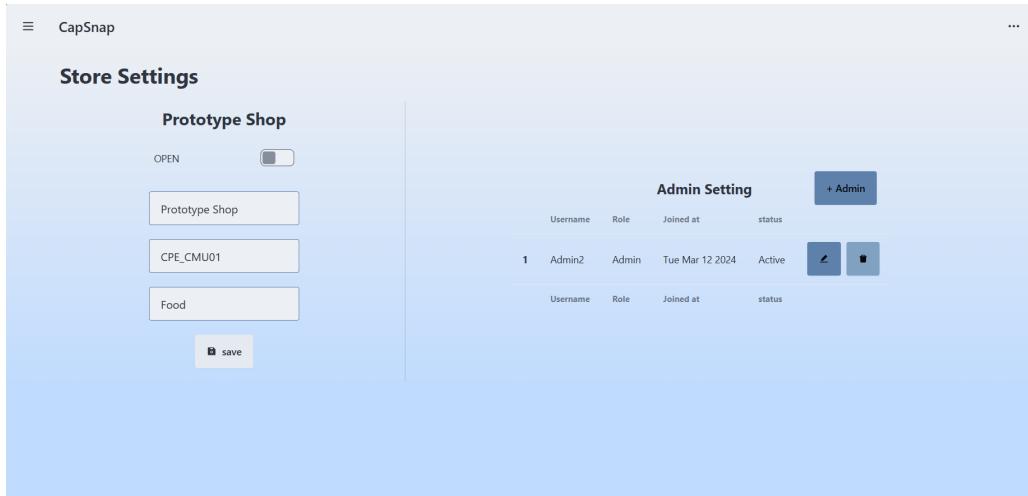
4.5.2 หน้าการตั้งค่าร้านค้า และผู้ดูแลร้านค้า (Store Settings)

สำหรับหน้านี้ผู้ใช้งานสามารถตั้งค่าชื่อร้าน สถานที่ตั้ง และประเภทร้านค้าได้ รวมถึงเป็นหน้าการจัดการผู้ดูแลระบบ โดยการแสดงผลสามารถแบ่งได้เป็น 2 กรณีตามสถานะของผู้ใช้งาน ได้แก่

1. Super Administer: สามารถดูรายชื่อ เพิ่ม หรือลบผู้ดูแลระบบบ้านค้า (Administer) ได้ และ

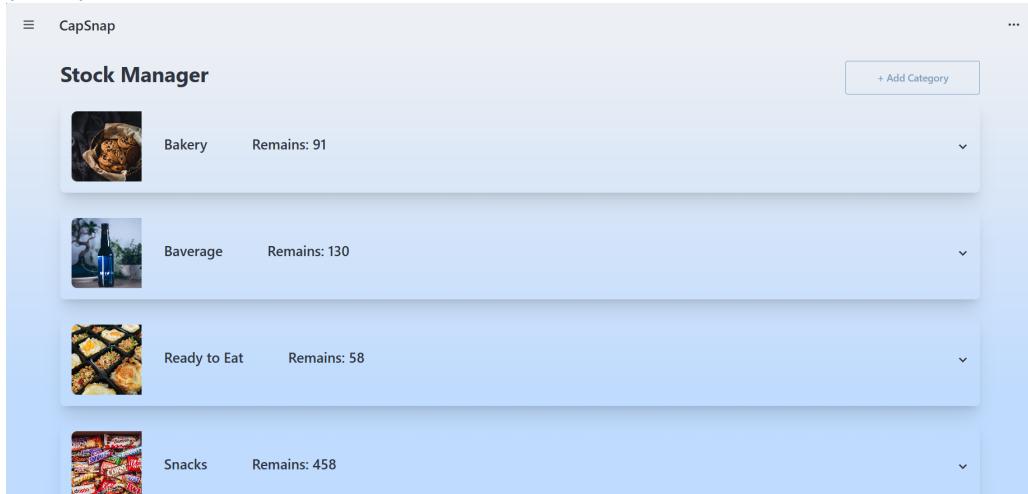
สามารถแก้ไขข้อมูลสถานะ และบทบาทของผู้ดูแลระบบร้านค้า และแก้ไขข้อมูลของตนเองได้

2. Administer: สามารถดู และแก้ไขข้อมูลของตนเองได้เท่านั้น



4.5.3 หน้าการจัดการสินค้า (Stock Manager)

สำหรับหน้านี้จะแสดงรายการสินค้าทั้งหมดแยกตามหมวดหมู่ พร้อมข้อมูลทั้งหมด ได้แก่ ชื่อสินค้า รูปสินค้า ข้อมูลสินค้า ราคา จำนวนในคลัง และประวัติคลังสินค้า โดยสามารถทำการเพิ่ม ลบ และแก้ไขข้อมูลทั้งหมดได้ ดังนี้



☰ CapSnap

Stock Manager

+ Add Category

	Bakery	Remains: 91
	Beverage	Remains: 130
	Ready to Eat	Remains: 58
	Snacks	Remains: 458

Add New Category

Name:

Image url:

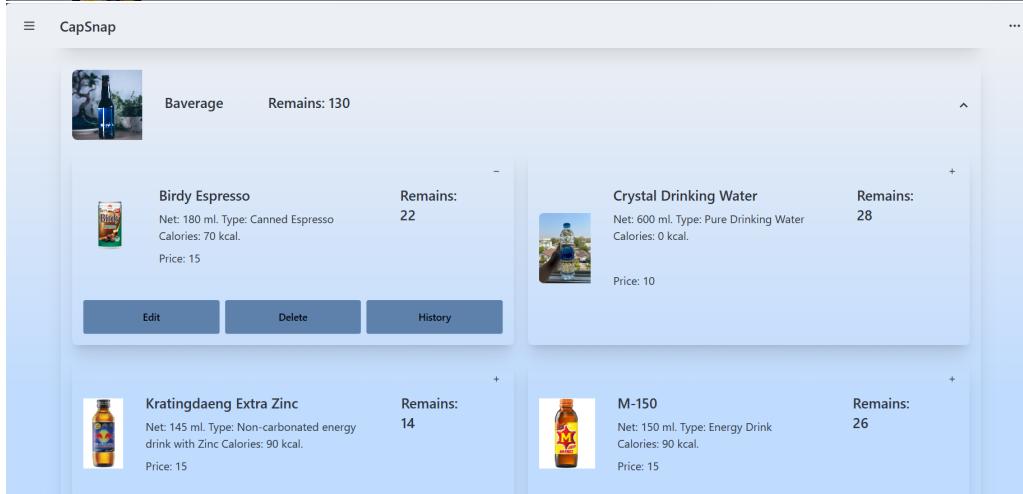
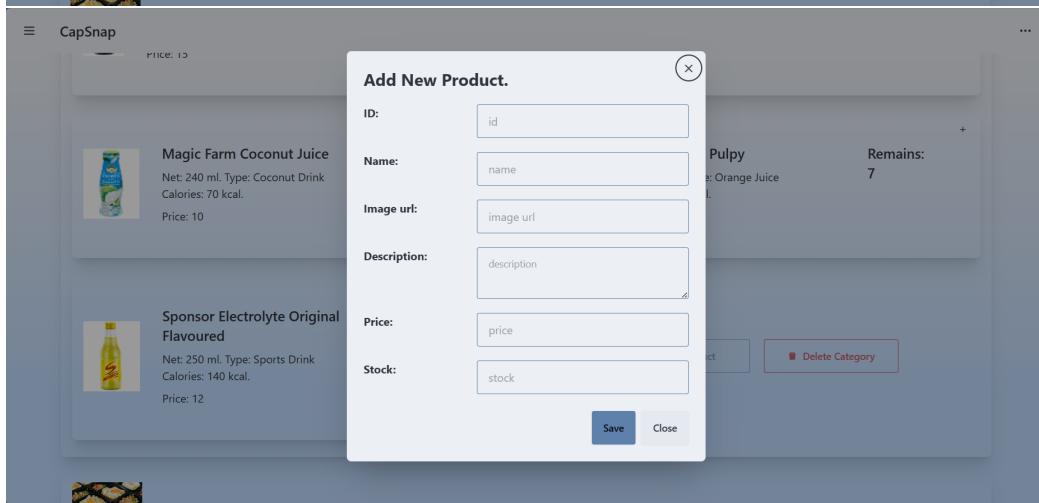
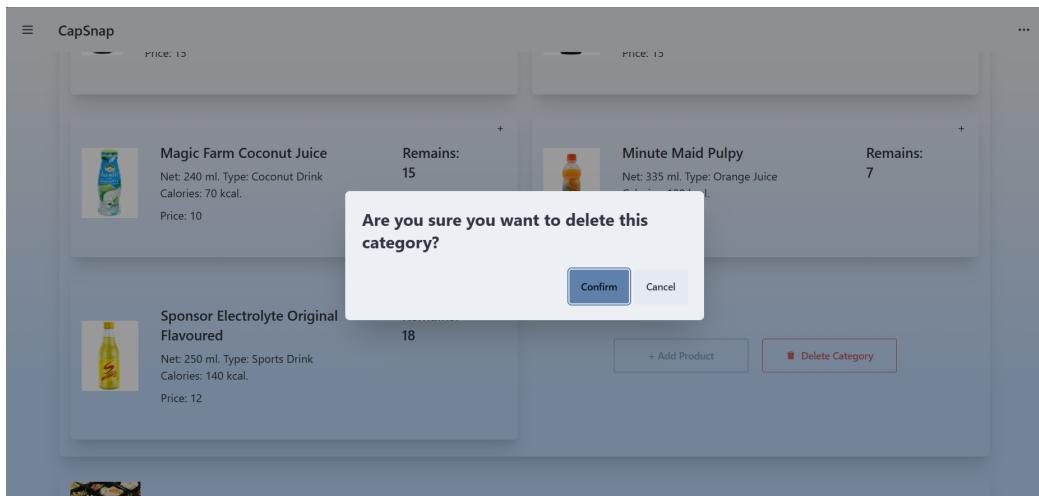
Save Close

☰ CapSnap

	Beverage	Remains: 130
	Birdy Espresso Net: 180 ml. Type: Canned Espresso Calories: 70 kcal. Price: 15	Remains: 22
	Crystal Drinking Water Net: 600 ml. Type: Pure Drinking Water Calories: 0 kcal. Price: 10	Remains: 28
	Kratingdaeng Extra Zinc Net: 145 ml. Type: Non-carbonated energy drink with Zinc Calories: 90 kcal. Price: 15	Remains: 14
	M-150 Net: 150 ml. Type: Energy Drink Calories: 90 kcal. Price: 15	Remains: 26

☰ CapSnap

	Beverage	Remains: 130
	Birdy Espresso Net: 180 ml. Type: Canned Espresso Calories: 70 kcal. Price: 15	Remains: 22
	Crystal Drinking Water Net: 600 ml. Type: Pure Drinking Water Calories: 0 kcal. Price: 10	Remains: 28
	Kratingdaeng Extra Zinc Net: 145 ml. Type: Non-carbonated energy drink with Zinc Calories: 90 kcal. Price: 15	Remains: 14
	M-150 Net: 150 ml. Type: Energy Drink Calories: 90 kcal. Price: 15	Remains: 26



The screenshots illustrate the CapSnap application's product management features:

- Screenshot 1: Birdy Espresso Stock History**
A modal window titled "Birdy Espresso Stock History" displays a list of stock transactions for the product. The transactions are as follows:
 - radcliff loma: - 1 (3/10/2024, 9:43:09 PM)
 - เจส นาม: - 4 (3/10/2024, 9:44:29 PM)
 - แคนท หอริน: - 5 (3/10/2024, 9:48:41 PM)
 - Admin2: + 1 (3/24/2024, 4:03:47 PM)
 The modal has a "Close" button at the bottom right.
- Screenshot 2: Edit Birdy Espresso information**
A modal window titled "Edit Birdy Espresso information" allows editing of product details. The fields are:
 - Name: Birdy Espresso
 - Image url: https://www.ajinomotofoodservicethai...
 - Description: Net: 180 ml. Type: Canned Espresso
Calories: 70 kcal.
 - Price: 15
 - Stock: There are 22 in stock.
Buttons: Increase, update unit
 The modal has "Save" and "Close" buttons at the bottom right.
- Screenshot 3: Are you sure you want to delete this product?**
A confirmation dialog box asks "Are you sure you want to delete this product?". It contains "Confirm" and "Cancel" buttons.

4.5.4 หน้าแสดงรายงานยอดขายสินค้า (Selling Report)

เมื่อโหลดหน้าใหม่จะแสดงข้อมูลยอดขายของวันที่ปัจจุบันผ่านการ์ดข้อมูล แผนภูมิ และตาราง ในโหมดรายวัน ตามสินค้าทั้งหมด สามารถเลือกวันที่ต้องการ และเปลี่ยนการแสดงผลตามรายวัน รายสัปดาห์ รายเดือน และรายปี ตามหมวดหมู่ของสินค้า หรือตามสินค้าทั้งหมดได้

☰ CapSnap ⋮

Selling Report

MM/DD/YYYY Daily Weekly Monthly Yearly Category Product

No data available this day

Table +

0 All product

0 Sold

0 Received

☰ CapSnap ⋮

Selling Report

MM/DD/YYYY Daily Weekly Monthly Yearly Category Product

No data available this day

No data available this day

Table +

0 All product

0 Sold

0 Received

☰ CapSnap ⋮

Selling Report

MM/DD/YYYY Daily Weekly Monthly Yearly Category Product

March 2024 ⏪ ⏩

S	M	T	W	T	F	S
				1	2	
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

No data available this day

No data available this day

Table +

0 All product

0 Sold

0 Received

☰ CapSnap ⋮

Selling Report

02/09/2024 Daily Weekly Monthly Yearly Category Product

Bar Chart ⌂ ⌃ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌊

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

8 All product

20 Sold

168 Received

Donut Chart

- Butter Bun: 14.9%
- Lotus Korean Barbecue Snack: 17.9%
- Magic Farm Coconut Juice: 1.9%
- Mango Sticky Rice: 26.8%
- Manucho Chicken Flavour: 20.8%
- Roller Coaster Chicken Hot & Spicy: 3.9%
- Sponsor Electrolyte Original Flavoured: 6.0%

Table +

Selling Report

8
All product

20
Sold

168
Received

02/09/2024
Daily
Weekly
Monthly
Yearly
Category
Product

Table
+-

Table
+

	Product	Product(s)	Sold unit	Total amount
1	Snacks	Marucho Chicken Flavour Roller Coaster Chicken Hot & Spicy	16	95
2	Ready to Eat	Mango Sticky Rice	1	45
3	Beverage	Magic Farm Coconut Juice	2	22
4	Bakery	Butter Bun	1	6

-

Table
+

	Product	Price	Sold unit	Total amount
1	Mango Sticky Rice	45	1	45
2	Marucho Chicken Flavour	5	7	35
3	Lotus Korean Barbecue Snack	10	3	30
4	Tivoli Twin Jumbo Vanilla Flavor	5	5	25
5	Sponsor Electrolyte Original Flavoured	12	1	12
6	Magic Farm Coconut Juice	10	1	10
7	Butter Bun	6	1	6
8	Roller Coaster Chicken Hot & Spicy	5	1	5

-

42



4.5.5 หน้าแสดงประวัติการซื้อสินค้าของลูกค้า (Purchase History)

สำหรับหน้านี้จะแสดงรายการการซื้อของลูกค้าแต่ละครั้งในรูปแบบการ์ด ประกอบไปด้วยชื่อ วัน และเวลาที่ซื้อ และจำนวนเงินที่จ่าย เมื่อกดที่การ์ดจะแสดงรายการสินค้าที่ซื้อ จำนวนที่ซื้อ และราคาของสินค้าแต่ละชิ้น



4.5.6 หน้าแสดงข้อมูลลูกค้า (Customers)

สำหรับหน้านี้จะแสดงข้อมูลของลูกค้าแต่ละคนผ่านตาราง ได้แก่ ชื่อลูกค้า วันเกิด วันที่ลงทะเบียน และอีเมล และฝ่ายการ์ดข้อมูล ได้แก่ จำนวนลูกค้าทั้งหมด จำนวนลูกค้าที่ลงทะเบียนใหม่ใน 7 วันที่ผ่านมา

The screenshot shows a dashboard titled "Customers". On the left, there are two blue boxes: one showing "19 Overall customers" and another showing "1 New customers in last 7 days". To the right is a table with 19 rows of customer data, each with a small triangle icon for sorting.

	Name	Surname	Email	Birth date	Joined at
1	Stella	nao	stella@outlook.com	3/10/2024	3/10/2024, 9:41:00 PM
2	เจ	แจน	jayant@outlook.com	3/10/2024	3/10/2024, 9:55:44 PM
3	radcliff	loma	radcliff@outlook.com	3/10/2024	3/10/2024, 9:42:30 PM
4	ain	ley	ainsley@outlook.com	3/10/2024	3/10/2024, 9:56:40 PM
5	โรส	นาน	rosabel@outlook.com	3/10/2024	3/10/2024, 9:43:46 PM
6	ริคกี้	เมอร์รี่	ricky@outlook.com	3/10/2024	3/10/2024, 9:45:33 PM
7	ลิน	ดาล	dylson@outlook.com	3/10/2024	3/10/2024, 9:46:50 PM
8	พงศกร	รัตนพันธ์	min23asd@outlook.com	2/12/2024	2/12/2024, 2:56:55 PM
9	เรนเดร์	นาเดิม	rebeccanne@outlook.com	3/10/2024	3/10/2024, 9:57:43 PM
...

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผล

โดยพบว่าการใช้ Computational Intelligence มาเพื่อช่วยในการ Classification และการ Checkout ด้วยตัวเองของ ลูกค้าทั่วไปสามารถติดตั้งแอปพลิเคชันบนโทรศัพท์มือถือ ทำให้เวลาที่ใช้ในการซื้อของ ลดลง และน้อยกว่าการไปต่อแคลที่ Cashier เนื่องจากไม่ต้องเข้าแคร์อคิด สามารถที่จะจ่ายเงินได้ด้วยตัวเองได้เลย

แต่ก็ยังมีข้อผิดพลาดในเรื่องของความแม่นยำ และความเร็วในการ Classification ที่น้อยกว่าการใช้ barcode ที่แบ่งอยู่ตามสินค้า แต่ละชนิด โดยจากในส่วนของการ Blind Test จะเห็นว่าหาก Model มี มุมที่ไม่เคยเห็นก็อาจจะทำให้ไม่สามารถ ที่จะนำযานยนต์ของสินค้าได้ถูกต้อง ดังตาราง 4.2

5.2 Web Dashboard

จากการทดลองการทำงานของเว็บไซต์สำหรับร้านค้า สามารถสรุปผลการใช้งานเมื่อเทียบกับการไม่มีระบบเว็บไซต์ดังกล่าวในการช่วยจัดการร้านค้าได้ คือ สามารถติดตามการขายสินค้าได้ตลอดเวลา ไม่ว่าจะเป็นผู้ซื้อ สินค้าที่ถูกซื้อ จำนวนเงินที่ได้ รวมถึงสามารถติดตามรายงานยอดขายได้อีกด้วย นอกจากนี้ยังสามารถแก้ไขข้อมูล สินค้าทั้งหมดบนเว็บไซต์ ทำให้ร้านค้ามีความสะดวกสบายมากขึ้นในการขาย และจัดการสินค้า โดยสามารถ สรุปผลลัพธ์การใช้งานได้ดังนี้

ตารางที่ 5.1: เปรียบเทียบการจัดการร้านค้าเมื่อไม่มีและมีระบบเว็บไซต์สำหรับร้านค้า

	Without Website Dashboard	With Website Dashboard
การดูข้อมูลสินค้า	ดูบนไฟล์ Excel	มีการแสดงผลบนเว็บไซต์
การเปลี่ยนแก้ไขข้อมูลสินค้า	แก้ไขบนไฟล์ Excel	แก้ไขบนเว็บไซต์
การติดตามสินค้าที่ขาย	นับ สินค้า ที่ขาย ไป จาก จำนวน ที่ เหลือ และติดตามยอดเงินของ	มีหน้าสรุปว่าลูกท่านใดซื้ออะไรไป บ้าง
การติดตามยอดขาย	เขียนสรุป และบันทึกยอดขายของ	มีหน้าสรุปยอดขายในทุกวัน
การเพิ่มลดคลังสินค้า	ไม่สามารถติดตามผู้เพิ่มลดได้	มีหน้าสรุปผู้เพิ่มลดคลังสินค้า
การติดตามลูกค้า	ไม่สามารถติดตามได้	ติดตามได้บนเว็บไซต์

5.3 ปัญหาที่พบและแนวทางการแก้ไข

ในการทำโครงการนี้ พบร่วมกับปัญหาหลักๆ ดังนี้

5.3.1 Classification model

การ Classification ของ product สามารถเกิดความผิดพลาดได้ เมื่อมีการเคลื่อนที่ของกล้อง หรือการขยับ ของกล้องเร็วเกินไป

แนวทางการแก้ไขปัญหา: การใช้โมเดลที่มีความยืดหยุ่นมากขึ้น (Robust Models): เลือกใช้โมเดลที่มีความสามารถในการรับมือกับภาพที่เคลื่อนที่ หรือโมเดลที่ได้รับการฝึกสอนด้วยข้อมูลที่มีการเคลื่อนที่ภายใน

- การปรับแต่งพารามิเตอร์ (Parameter Tuning): ปรับค่าพารามิเตอร์ของโมเดล เช่น การปรับค่าความสำคัญของชั้นคอนโวลูชัน (convolutional layer) หรือการเปลี่ยนแปลงการตั้งค่าการฝึกสอนเพื่อให้โมเดลมีประสิทธิภาพต่อการเคลื่อนที่ของกล้อง
- การแสดงผลลัพธ์ที่มีความมั่นใจมากเท่านั้น (confidence): โดยหาก model ทำนายชนิดสินค้าโดยมีความน่าเชื่อถือ (confidence ไม่สูง) อาจจะทำให้ได้ผลลัพธ์ที่ผิดพลาดได้ง่าย
- การใช้ผลการ predict หลายครั้งมาเพิ่มความแม่นยำ (Voting): โดยใช้การส่างรูปภาพไปหลายภาพ และนำผลลัพธ์การ predict ในแต่ละครั้งมาหาชนิดสินค้าที่ได้รับการ predict ออกมายةจะสูงที่สุด
- การใช้เทคโนโลยีการตรวจจับการเคลื่อนที่ (Motion Detection Technology): การใช้เทคโนโลยีเชิงภาพเพื่อตรวจจับการเคลื่อนที่ของกล้องและปรับปรุงการทำงานคลาสตามไปด้วย เช่น การใช้เซ็นเซอร์การเคลื่อนที่หรือการตรวจจับการเคลื่อนที่จากภาพ

5.3.2 Machine learning scalability

ในโครงการนี้ Classification server โดย Model ของเรามีขนาดอยู่ที่ 33 ล้านพารามิเตอร์ (128 MB) ทำงานบนคอมพิวเตอร์ที่มี GPU GTX 1660Ti (6GB VRAM) computational power อยู่ที่ FP32 (float) 5.437 TFLOPS. ซึ่งการทำนาย ของรูปภาพ 1 รูป model จะใช้เวลาอย่างนานที่สุด 25 ms 5.437 TFLOPS (computational power 5.437 TFLOPS หากใช้ GPU ที่มีพลังคำนวนมากกว่าจะทำให้ใช้เวลาน้อยลง)

โดยเฉลี่ยแล้ว User 1 คน ใช้เวลาในการใช้งาน Camera classification อยู่ที่ 3-10 วินาที โดย Application จะส่งรูปมา.yang เซิร์ฟเวอร์ทุก 500 มิลลิวินาที (1วินาที ได้รับ 2 รูปภาพ)

ซึ่งสามารถคาดได้ว่า ใน 1 วินาที จะส่วน slot ที่เท่ากับ 50 มิลลิวินาที เป็นของผู้ใช้ 1 คน ระบบนี้ก็สามารถรองรับได้ถึง 20 ผู้ใช้ต่อ 1 วินาที

$$\left(\frac{1,000ms}{25ms/image}\right)/2 \text{ image / User} = 20\text{User} \quad (5.1)$$

ดังนั้น ใน 1 วินาทีซึ่งแบ่งได้ทั้งหมด 20 slot สามารถรองรับผู้ใช้ได้พร้อมกัน 20 คน และสมมุติว่า User แต่ละคนใช้เวลามากถึง 10 วินาที (20 รูปภาพ) ระบบก็สามารถรองรับได้ถึง 120 ผู้ใช้ต่อ 1 นาที

$$\left(\frac{60,000ms}{25ms/image}\right)/20 \text{ image / User} = 120\text{User} \quad (5.2)$$

แนวทางการแก้ไขปัญหา:

- เพิ่ม computational power ของ Classification server ให้มากขึ้น
- ลดจำนวนรูปภาพที่ Application จะส่งมา.yang Classification server ลง
- ลด Input Dimension ของรูปภาพ

5.3.3 Web Dashboard

ปัญหาที่พบในการใช้งานส่วนของเว็บไซต์สำหรับร้านค้า คือเวลาที่ใช้ในการแสดงผลข้อมูลรายงานยอดขายในกรณีที่มีข้อมูลจำนวนมากนั้นใช้เวลาประมาณ 5 วินาที ซึ่งเป็นการแสดงผลช้ากว่าค่อนข้างช้า และให้ประสบการณ์ที่ได้รับรื่นในการใช้บริการเว็บไซต์ แนวทางในการแก้ไขคือการปรับปรุงอัลกอริทึมในการประมวลผลโดยอาศัยพิจารณาทฤษฎีของ Big O น้อยที่สุดที่เป็นไปได้ เพื่อให้การประมวลผลเร็วขึ้น

5.4 ข้อเสนอแนะและแนวทางการพัฒนาต่อ

สำหรับข้อแนะนำสำหรับการพัฒนาต่อของระบบโดยรวม สามารถจำแนกได้ดังนี้

- **ปรับปรุงระบบคลาวด์:** พัฒนาระบบคลาวด์โดยการอัปเกรด Supabase เพื่อเพิ่มประสิทธิภาพในการทำงานและประหยัดค่าใช้จ่าย เช่น การเพิ่ม Scalability Reliability และ Security ให้กับระบบ
- **ปรับปรุงโมดูลความสามารถของ Classification Model:** พัฒนาโมดูล CI เพื่อให้สามารถจำแนกสินค้าจากรูปภาพได้ที่แม่นยำมากยิ่งขึ้น โดยอาจเพิ่มข้อมูล หรือปรับปรุงโมเดลใหม่
- **เพิ่มฟีเจอร์ใน Mobile Application:** เพิ่มฟีเจอร์ใหม่บนแอปพลิเคชัน เพื่อให้มีการทำงานที่สมบูรณ์ยิ่งขึ้น เช่น เพิ่มตัวเลือกการลงทะเบียนเข้าใช้งาน เพิ่มการตั้งค่าข้อมูลผู้ใช้งานมากขึ้น ปรับปรุงการแสดงผลประวัติการซื้อสินค้า ปรับปรุงเรื่องความเสถียรของการสแกนสินค้า และเพิ่มการชำระเงินหลากหลายรูปแบบ
- **พัฒนาฟีเจอร์ใน Web Dashboard:** เพิ่มฟีเจอร์การวิเคราะห์ข้อมูลเพิ่มเติม เช่น การวิเคราะห์ยอดขายที่ซับซ้อนมากขึ้น การติดตามแนวโน้มการซื้อของลูกค้า รวมถึงปรับปรุงโปรแกรมให้เร็ว และเสียรุ่นมากขึ้น
- **ปรับปรุงฐานข้อมูล:** ปรับปรุงโครงสร้างของฐานข้อมูลเพื่อเพิ่มประสิทธิภาพในการจัดการข้อมูล และการสืบค้น
- **ทดสอบและปรับปรุงความปลอดภัย:** ทำการทดสอบความปลอดภัยของระบบอย่างสม่ำเสมอ และปรับปรุงเพื่อป้องกันการโจงแจงข้อมูลหรือการโจมตีจากภัยคุกคามต่าง ๆ
- **การพัฒนา UX/UI:** ปรับปรุงประสิทธิภาพ และประสบการณ์ผู้ใช้ในทั้ง Mobile Application และ Web Dashboard เพื่อให้การใช้งานเป็นไปอย่างราบรื่นและสะดวกสบายยิ่งขึ้น
- **ใช้ Machine Learning ในภาคเดาความต้องการของลูกค้า:** ใช้ Machine Learning เพื่อวิเคราะห์และคาดเดาความต้องการของลูกค้าเพื่อเสนอสินค้าที่เข้ากับความต้องการของลูกค้าได้ดียิ่งขึ้น
- **เพิ่มระบบการสนับสนุนลูกค้า:** พัฒนาระบบการสนับสนุนลูกค้าเพิ่มเติม เช่น ระบบแชทบอท หรือระบบช่วยเหลืออื่น ๆ เพื่อสนับสนุนลูกค้าในการใช้บริการและแก้ไขปัญหาที่เกิดขึ้น
- **การใช้ Big Data Analytics:** ใช้เทคโนโลยี Big Data Analytics เพื่อวิเคราะห์และสร้างความเข้าใจในพฤติกรรมของลูกค้า เพื่อช่วยในการตัดสินใจในการพัฒนาและปรับปรุงระบบให้เหมาะสมยิ่งขึ้น

บรรณานุกรม

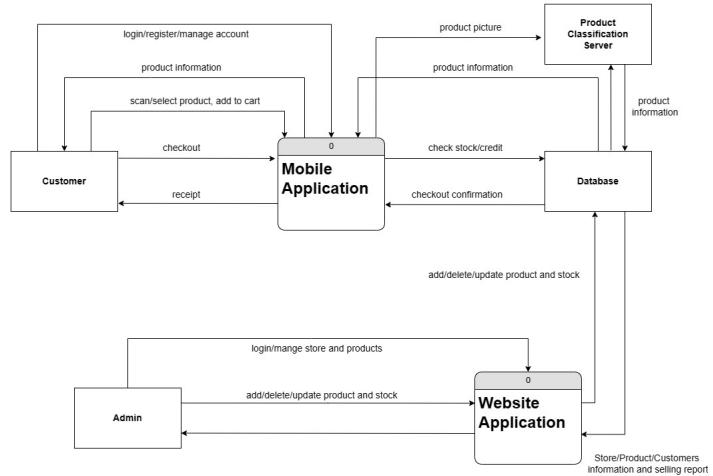
- [1] jittagornp. Websocket គីអោឡិវេទ នាំងយ៉ាង។ jittagornp.me, 2021. URL <https://www.jittagornp.me/blog/what-is-websocket>.
- [2] Artificial neural network. wikipedia.org, 2013. URL https://en.wikipedia.org/wiki/artificial_neural_network.
- [3] Multilayer perceptron. wikipedia.org, 2023. URL https://en.wikipedia.org/wiki/multilayer_perceptron.
- [4] Feedforward neural network. wikipedia.org, 2023. URL https://en.wikipedia.org/wiki/activation_function.
- [5] Dr. James McCaffrey. The neural network input-process-output mechanism. visualstudiomagazine.com, 2013. URL <https://visualstudiomagazine.com/miscs/2013/05/01/neural-network-feed-forward.aspx>.
- [6] Activation function. wikipedia.org, 2022. URL https://en.wikipedia.org/wiki/feedforward_neural_network.
- [7] HD COE. Convolutional neural network គីអោឡិវេទ. medium.com, 2020. URL <https://medium.com/@haddee2531earvesdrop/convolutional-neural-network-គីអោឡិវេទ-42c45f7ec16b>.
- [8] Wenjin Tao, Md. Al-Amin, Haodong Chen, and Ming C. Leu. Real-time assembly operation recognition with fog computing and transfer learning for human-centered intelligent manufacturing. sciencedirect.net, 2020. URL <https://www.sciencedirect.com/science/article/pii/S2351978920315833>.
- [9] Cross-validation. mathworks.com, 2023. URL <https://www.mathworks.com/discovery/cross-validation.html>.
- [10] Kathiravan Srinivasan, Lalit Garg, Debajit Datta, and Abdulellah Abdullah Alaboudi. Performance comparison of deep cnn models for detecting driver distraction. techscience.com, 2021. URL <https://www.techscience.com/cmc/v68n3/42484/html>.

ภาคผนวก

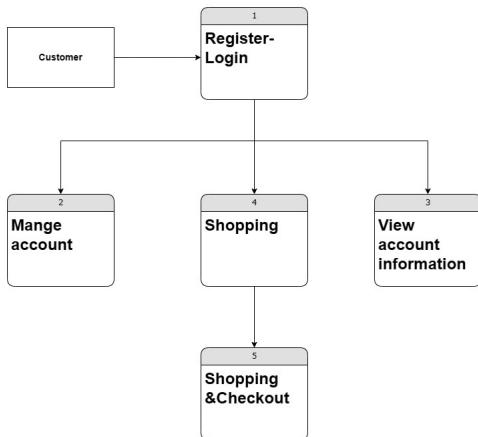
ການຜົນວັນ ກ

Data Flow

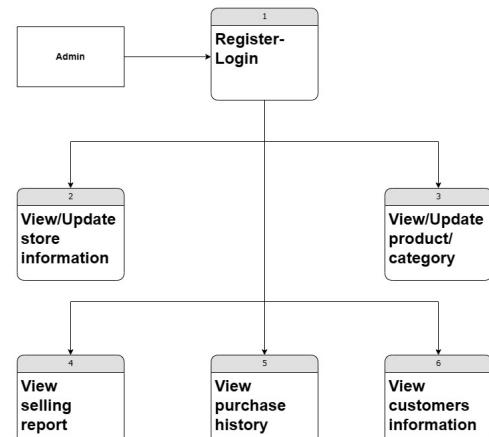
DFD level 0: Self-checkout system



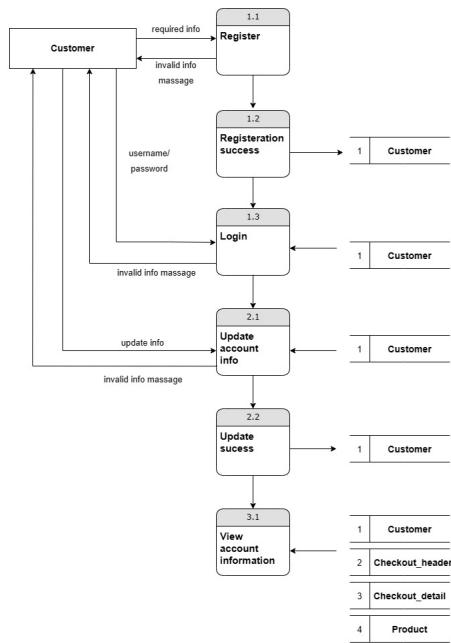
DFD level 1: Mobile Application



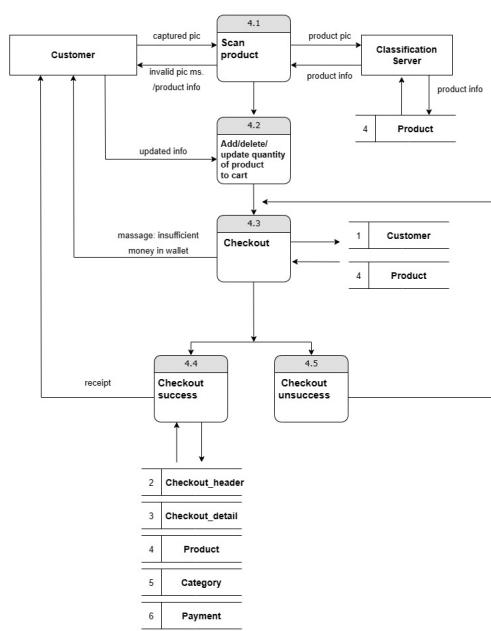
DFD level 1: Website Application



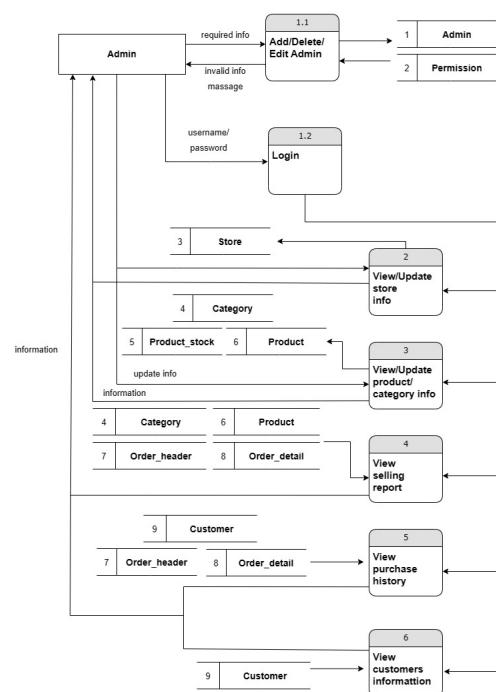
DFD level 2: Mobile Application (Process 1-3)

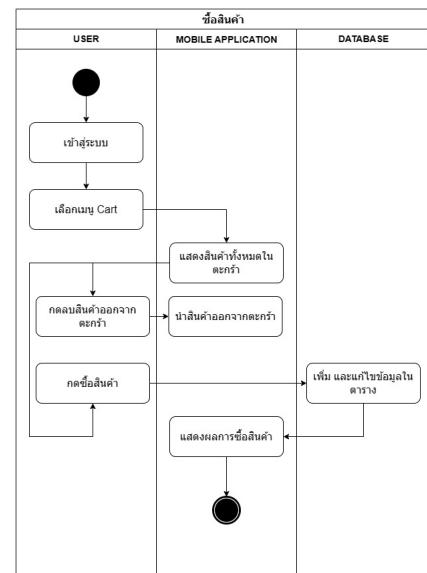
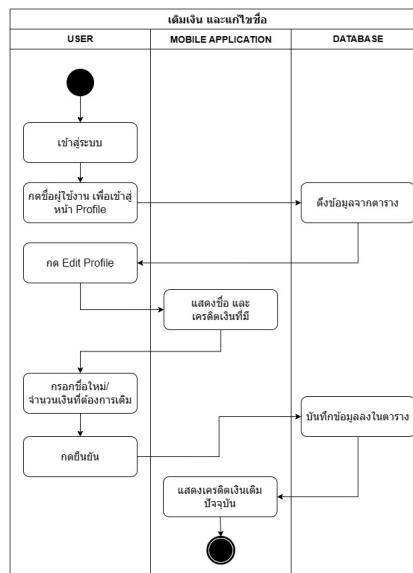
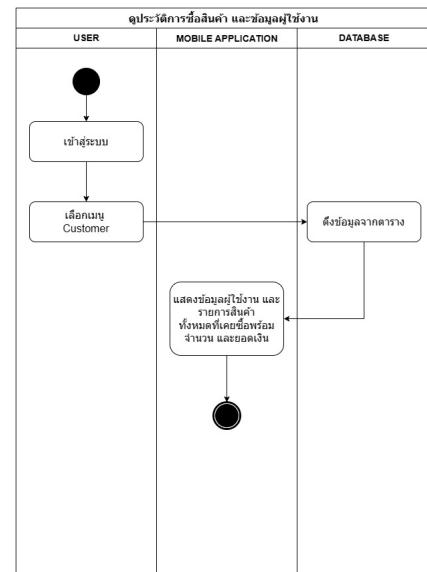
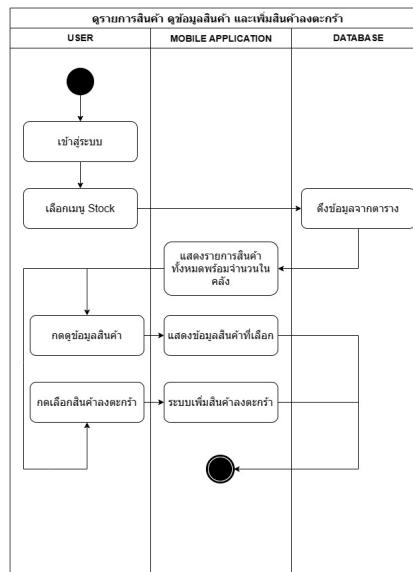
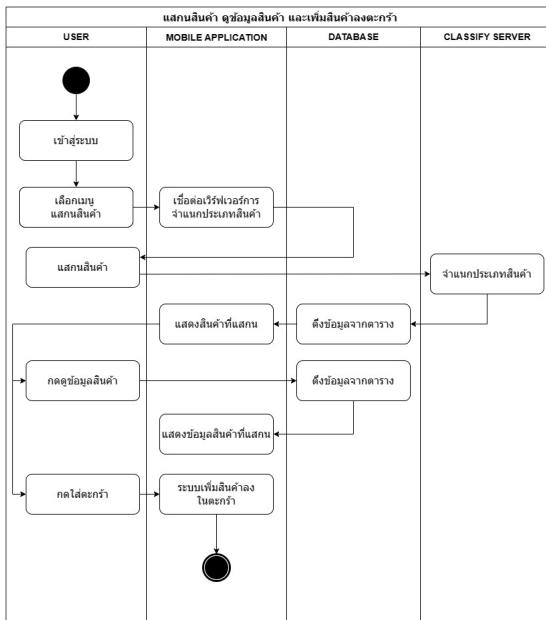


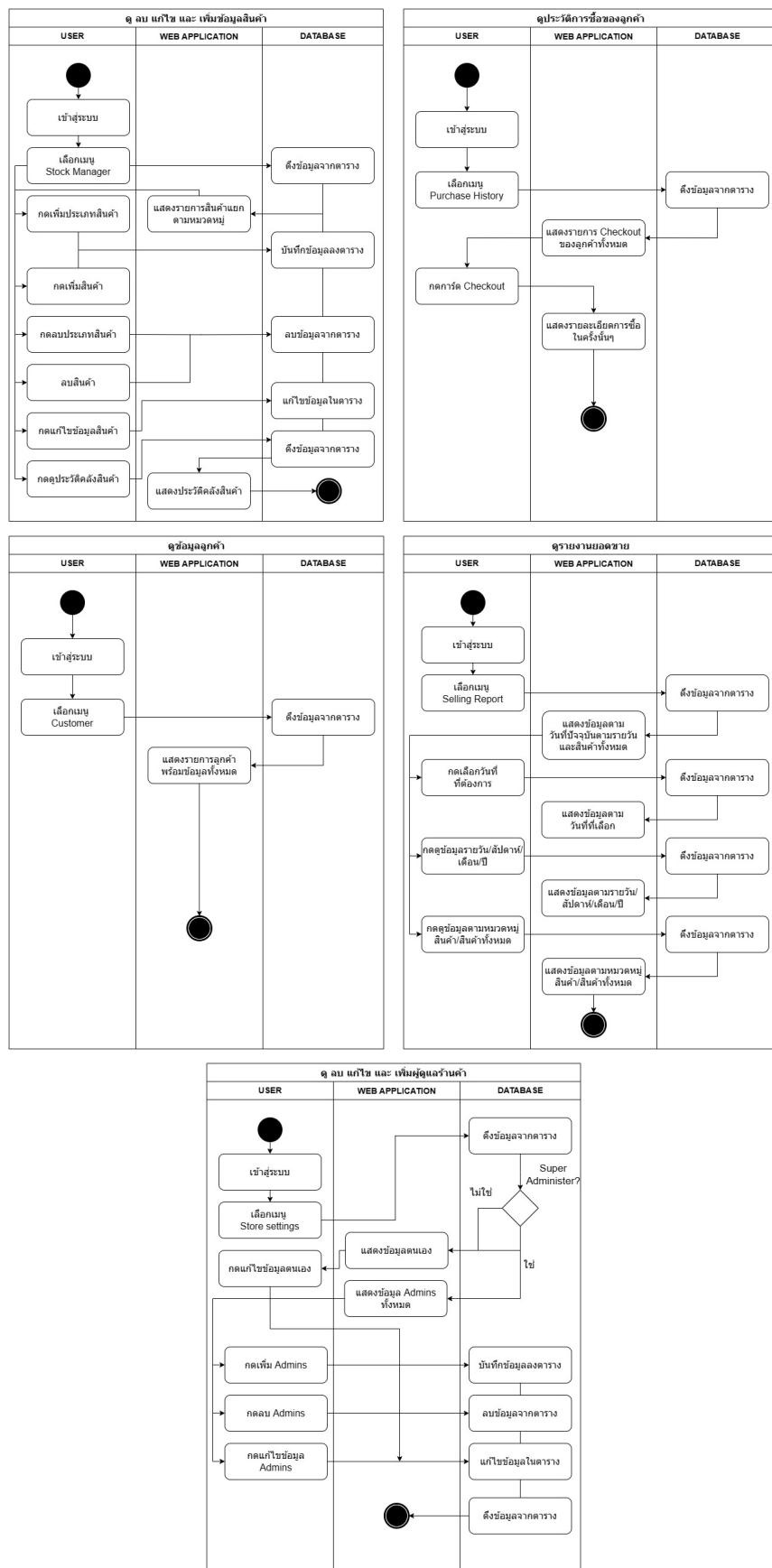
DFD level 2: Mobile Application (Process 4)



DFD level 2: Website Application







week	work
4 Dec - 10 Dec 2022	Init Github repositories for Back-end , Website and Application
11 Dec - 17 Mar 2022	Collect products data from 422 shop
18 Dec - 24 Dec 2022	Implementing Google colab for pre processing product data
25 Dec - 21 Dec 2022	Coding mechanism for Training model
1 Jan - 7 Jan 2023	Training model and model validation
8 Jan - 14 Jan 2023	Designing Mobile Application's UI/UX
15 Jan - 21 Jan 2023	Designing Website dashboard's UI/UX
22 Jan - 28 Jan 2023	Prepare for midterm presentation.
29 Jan - 4 Feb 2023	Designing architecture neural network model for classification
5 Feb - 11 Feb 2023	Feasibility study Website dashboard's architecture
12 Feb - 18 Feb 2023	Feasibility study Application's architecture
19 Feb - 25 Feb 2023	
26 Feb - 4 Mar 2023	Implementing supabase database for store data and products data
5 Mar - 11 Mar 2023	Refactor most of traing code
12 Mar - 17 Mar 2023	Write and fix the report

ຮູບທີ່ ໜ.1: Work table