

โครงการเลขที่ วศ.คพ. P008-2/2565

เรื่อง

แคลสแนป : ระบบจัดการการซื้อขายในร้านค้าปลีกอัตโนมัติด้วยตนเองโดยใช้ปัญญา  
ประดิษฐ์

โดย

นายพงศกร รัตนพันธ์ รหัส 630610749  
นางสาวศุภริภา ศิลปสิทธิ์ รหัส 630610765

โครงการนี้

เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่  
ปีการศึกษา 2565

**PROJECT No. CPE P008-2/2565**

**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  
2022**

หัวข้อโครงการ : แอปสแนป : 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 : 2565

---

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์  
(รศ.ดร. สันติ พัทธกิจนุกูร)

คณะกรรมการสอบโครงการ

..... ประธานกรรมการ  
(รศ.ดร. ศันสนีย์ เอื้อพันธ์วิริยะกุล)

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

..... กรรมการ  
(รศ.ดร. นิพนธ์ อีร์อำพน)

หัวข้อโครงการ : แคมเปญ : 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 : 2565

---

### บทคัดย่อ

แคมเปญ(CapSnap) เป็นระบบจัดการการซื้อ-ขายสินค้าในร้านค้าปลีก เพื่ออำนวยความสะดวก และลดความยุ่งยากในการ ซื้อ-ขายสินค้าทั้งทางฝั่งลูกค้า และฝั่งร้านค้า โดยลูกค้าสามารถจัดการการซื้อสินค้าได้ด้วยตนเองผ่านทางแอปพลิเคชัน ในโทรศัพท์มือถือ โดยแอปพลิเคชันดังกล่าวจะทำการเชื่อมต่อกับร้านค้าเมื่อลูกค้าเดินเข้าร้านค้า จากนั้นลูกค้าสามารถสแกนสินค้าที่ ต้องการแบบเรียลไทม์ ซึ่งระบบจะใช้หลักการทาง computational intelligence เพื่อบอกรายละเอียดและราคาของสินค้าแต่ละชนิดที่ถูกสแกน ซึ่งจะเป็นประโยชน์ต่อคนที่มีปัญหาในการมองเห็นในการเลือกซื้อสินค้า ลูกค้าสามารถระบุรายการสินค้าได้ด้วยตนเองผ่านการเพิ่ม หรือลดสินค้าในตะกร้าของแอปพลิเคชัน เมื่อลูกค้าเดินออกจากร้าน ระบบจะทำการชำระเงินให้โดยอัตโนมัติ ซึ่งระบบรองรับการใช้งานแอปพลิเคชันกับร้านค้าปลีกหลาย ๆ ร้านที่มีสินค้าแตกต่างกัน นอกจากนี้ระบบยังมี Website dashboard ให้บริการสำหรับฝั่งร้านค้า เพื่อช่วยให้ร้านค้าปลีกสามารถจัดการร้านได้อย่างมีประสิทธิภาพมากขึ้นโดยสามารถจัดการ และดูข้อมูลคลังสินค้า รวมถึงข้อมูลการขายสินค้าได้ในเว็บไซต์เดียว

## กิตติกรรมประกาศ

Your acknowledgments go here. Make sure it sits inside the acknowledgment environment.

นายพงศกร รัตนพันธ์  
นางสาวศุภริญา ศิลปสิทธิ์  
25 พฤษภาคม 2563

## สารบัญ

บทคัดย่อ . . . . .	ข
กิตติกรรมประกาศ . . . . .	ค
สารบัญ . . . . .	ง
สารบัญรูป . . . . .	ฉ
สารบัญตาราง . . . . .	ช
<b>1 บทนำ</b>	<b>1</b>
1.1 ที่มาของโครงการ . . . . .	1
1.2 วัตถุประสงค์ของโครงการ . . . . .	1
1.3 ขอบเขตของโครงการ . . . . .	1
1.4 ประโยชน์ที่ได้รับ . . . . .	2
1.5 เทคโนโลยีและเครื่องมือที่ใช้ . . . . .	2
1.6 แผนการดำเนินงาน . . . . .	2
1.7 บทบาทและความรับผิดชอบ . . . . .	3
1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม . . . . .	3
<b>2 ทฤษฎีที่เกี่ยวข้อง</b>	<b>4</b>
2.1 WebRTC for Streaming image . . . . .	4
2.2 Artificial neural networks . . . . .	4
2.2.1 Multilayer perceptron . . . . .	5
2.2.2 Classification . . . . .	6
2.2.3 Training . . . . .	6
2.3 Convolutional neural network . . . . .	7
2.4 Transfer learning . . . . .	8
2.4.1 transfer learning - Xception . . . . .	8
<b>3 โครงสร้างและขั้นตอนการทำงาน</b>	<b>10</b>
3.1 โครงสร้างของระบบ . . . . .	10
3.2 เตรียมชุดข้อมูลฝึกสอน . . . . .	10
3.3 การเพิ่ม traing data . . . . .	11
3.3.1 Model Architecture . . . . .	12
3.4 classification products . . . . .	12
3.5 การพัฒนา application . . . . .	13
3.5.1 Requirement Specification . . . . .	13
3.6 การเตรียมฐานข้อมูล . . . . .	13
3.6.1 ฐานข้อมูลของระบบ . . . . .	13
3.6.2 ฐานข้อมูลในแต่ละร้านค้า . . . . .	13
3.7 การพัฒนาweb dashboard . . . . .	13
3.8 การทดสอบการทำงานของซอฟต์แวร์ . . . . .	14
3.8.1 Unit testing . . . . .	14
3.8.2 Integration testing . . . . .	14
3.8.3 System testing . . . . .	15
3.8.4 Acceptance testing . . . . .	15
3.9 แผนภาพกระแสข้อมูล (Data Flow Diagram) . . . . .	15

<b>4</b>	<b>การทดลองและผลลัพธ์</b>	<b>17</b>
4.1	Application UX/UI . . . . .	17
4.2	Web Dashboard UX/UI . . . . .	17
<b>5</b>	<b>บทสรุปและข้อเสนอแนะ</b>	<b>20</b>
5.1	สรุปผล . . . . .	20
5.2	ปัญหาที่พบและแนวทางการแก้ไข . . . . .	20
5.3	ข้อเสนอแนะและแนวทางการพัฒนาต่อ . . . . .	20
<b>ก</b>	<b>The first appendix</b>	<b>22</b>
	ประวัติผู้เขียน	24

## สารบัญรูป

2.1	webrtc structure . . . . .	4
2.2	Multilayer perceptron . . . . .	5
2.3	Convolution . . . . .	7
2.4	Convolutional neural network . . . . .	7
2.5	The concept of transfer learning . . . . .	8
3.1	Overall project structure . . . . .	10
3.2	Dataset generator . . . . .	11
3.3	Train results . . . . .	12
3.4	Mobile Application Database Schema . . . . .	14
3.5	Store Website Dashboard Database Schema . . . . .	15
3.6	Data Flow Diagram . . . . .	16
4.1	Application wire frame . . . . .	17



## สารบัญตาราง

1.1 Planning . . . . .	2
------------------------	---

# บทที่ 1

## บทนำ

### 1.1 ที่มาของโครงการ

จากระบบการขายสินค้าในร้านค้าปลีกส่วนใหญ่ของประเทศไทย ไม่ว่าจะเป็นห้างสรรพสินค้า ซูเปอร์มาเก็ต หรือ ร้านค้าปลีกรายย่อยต่าง ๆ พบว่าวิธีการที่ใช้ในการชำระสินค้า คือ การชำระสินค้าที่เคาน์เตอร์ชำระเงินโดยมีพนักงานบริการ ซึ่งข้อเสียแรกของวิธีการชำระสินค้านี้คือ การรอชำระสินค้าที่เคาน์เตอร์นั้นเสียเวลา และไม่สะดวกรวดเร็ว ยิ่งหากต้องมีการต่อแถวรอชำระเงิน ก็จะทำให้ผู้ใช้บริการเสียเวลามากขึ้น และเสียความพึงพอใจในการใช้บริการ รวมถึงต้องมีการจัดพื้นที่สำหรับการต่อแถวอีกด้วย ด้วยเหตุนี้ จึงควรพัฒนาเทคโนโลยีที่มาช่วยการแก้ปัญหาอย่างตรงจุด โดยให้ลูกค้าสามารถชำระสินค้าได้ด้วยตนเอง self-service หลังการเลือกซื้อสินค้าได้อย่างสะดวกสบายผ่านอุปกรณ์โทรศัพท์มือถือ

นอกจากนี้สำหรับร้านค้าปลีกย่อยที่มีรูปแบบการขายเป็นแบบบริการตนเอง หรือแม้กระทั่งร้านค้าปลีกที่มีพนักงานชำระเงิน แต่ไม่มีระบบช่วยจัดการยอดขาย ก็สามารถพบปัญหาในการจัดการยอดขาย และคลังสินค้าได้ เนื่องจากพนักงาน หรือเจ้าของร้านต้องติดตามการขายสินค้าด้วยตนเองทั้งหมด โดยจะต้องคอยนับจำนวนสินค้าที่เหลืออยู่ภายในร้านเพื่อตรวจสอบว่าได้ขายอะไรไปแล้ว ทำให้เกิดความยากลำบาก และผิดพลาดได้ง่าย

ผู้จัดทำจึงพัฒนาระบบ ‘CapSnap’ self-service เพื่อลดปัญหาที่เกิดจากการรอชำระเงินที่เคาน์เตอร์ โดยพัฒนาแอปพลิเคชันสำหรับผู้ใช้งานทั่วไปที่สามารถเชื่อมต่อกับร้านค้าที่เข้าร่วมเพื่อใช้บริการการซื้อสินค้าแบบ self-service ที่ร้านค้านั้น ๆ โดยลูกค้าสามารถเข้าใช้งานแอปพลิเคชันในโทรศัพท์มือถือแล้วทำการเลือกซื้อสินค้าผ่านฟังก์ชันการสตรึมมิงภาพสินค้าเพื่อ ทราบถึงรายละเอียดของสินค้า และเพิ่มสินค้าเข้าตะกร้า โดยระบบจะใช้หลักการทาง Computational Intelligence ในการแยกแยะสินค้า จากการสตรึมมิง ลูกค้าสามารถชำระเงินได้ด้วยตนเองพร้อมนำสินค้าที่ซื้อออกจากร้านค้าได้เลย ซึ่งระบบจะเก็บบันทึกประวัติการขายสินค้าลง ในฐานข้อมูลเพื่อแสดงให้กับฝั่ง Web dashboard ของร้านค้าซึ่งสามารถบริหารจัดการคลังสินค้า และข้อมูลการขายได้ในทีเดียว เพิ่มประสิทธิภาพการให้บริการที่ดีให้ลูกค้า และเพิ่มประสิทธิภาพในการบริหารจัดการสินค้าในร้านค้าสำหรับร้านค้าที่ต้องการให้บริการแบบ self-service ลดการว่างงานพนักงาน และต้องการเครื่องมือในการบริหารร้านค้า

### 1.2 วัตถุประสงค์ของโครงการ

1. เพื่อช่วยให้ลูกค้าสามารถบริการตนเองในร้านค้าได้ และให้ร้านค้าสามารถตรวจสอบสินค้าที่ขายได้
2. เพื่อพัฒนาระบบ แยกแยะชนิดสินค้าโดยใช้ Computational Intelligence ได้
3. เพื่อสร้างเว็บไซต์และแอปโทรศัพท์เพื่อให้ผู้ใช้งานสามารถใช้งานระบบได้

### 1.3 ขอบเขตของโครงการ

ข้อมูลที่ใช้ในการฝึกระบบ Computational Intelligence ในการแยกแยะชนิดสินค้า มาจาก ร้านค้าห้อง 422 ตึก 30 ปี คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเชียงใหม่ โดยมีสินค้าประมาณ 100 ชนิด

#### 1.4 ประโยชน์ที่ได้รับ

1. แอปพลิเคชันที่ลูกค้าสามารถแยกแยะชนิดสินค้าเพื่อให้ลูกค้าสามารถจ่ายเงินให้กับสินค้านั้นๆได้ด้วยตัวเอง โดยไม่ต้องคอยเลือกชนิดของสินค้านั้นๆ
2. เว็บไซต์ที่แสดงจำนวนการขายของสินค้าแต่ละชนิดเพื่อให้ร้านค้าสามารถจัดการสินค้าได้

#### 1.5 เทคโนโลยีและเครื่องมือที่ใช้

1. Python + aiortc : สำหรับพัฒนาในส่วน Backend , การรับข้อมูล streaming จาก Application , การฝึกสอนโมเดล , และการ classify Product โดยไม่ต้องคอยเลือกชนิดของสินค้านั้นๆ
2. Flask และ Next.js: สำหรับการพัฒนา Frontend ในส่วนเว็บไซต์ของร้านค้า (Web dashboard)
3. Flutter + webrtc (Mobile Application Framework) : สำหรับพัฒนา Frontend ในส่วน Application ในโทรศัพท์ของลูกค้า และใช้ webrtc streaming ภาพจาก camera ไปยัง Backend
4. Supabase: สำหรับเก็บฐานข้อมูลของสินค้า ข้อมูลร้านค้า และ ร้านค้า
5. MongoDB : สำหรับเก็บข้อมูลชนิดของสินค้าที่เอาไว้ใช้ฝึกสอนโมเดล , ใช้ในการแสดงราคาบน Frontend และเก็บจำนวนของสินค้าที่ขายไปแล้ว

#### 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	Progress
Planning																100%
Document																100%
Collect data																100%
Back-end development																80%
App development																20%
Dashboard development																10%
Payment development																0%
Testing																0%

ตารางที่ 1.1: Planning

## 1.7 บทบาทและความรับผิดชอบ

นายพงศกร รัตนพันธ์ รหัส 630610749 รับผิดชอบในส่วนของ Backend ได้แก่

1. การรับข้อมูลสตรีมมิ่งจากระหว่าง Mobile application และ Classifier ผ่าน WebRTC
2. การฝึกสอนโมเดลเพื่อทำ Product classification
3. การจัดเก็บฐานข้อมูลข้อมูลรูปภาพของสินค้าเพื่อทำการฝึกสอนโมเดล

นางสาวศุภริภา ศิลปสิทธิ์ รหัส 630610765 รับผิดชอบการพัฒนาส่วน Frontend และ ฐานข้อมูลบน Supabase ดังนี้

1. ทั้หน้าเว็บไซต์ของร้าน (website dashboard)
2. Mobile Application
3. จัดการการเก็บฐานข้อมูลทั้งหมดที่ใช้ในการแสดงผล

## 1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม

โครงการนี้ลดความซับซ้อนและเวลาที่ลูกค้าจะต้องรอต่อแถวเพื่อจ่ายเงินของสินค้า และ ช่วยให้สังคมสามารถเข้าถึงเทคโนโลยีที่ก้าวหน้าไปจากเดิมในการใช้กิจวัตรประจำวันอย่างการซื้อสินค้า โดยเปลี่ยนมาใช้บริการตนเองผ่านแอปพลิเคชันที่อำนวยความสะดวกผ่านอุปกรณ์มือถือที่ใช้งานกันอย่างแพร่หลาย ทำให้สังคมก้าวสู่ความทันสมัย และความสะดวกสบายมากขึ้น ตอบโจทย์ความต้องการ และรูปแบบการใช้ชีวิตของผู้คนในยุคสมัยใหม่ ช่วยหลีกเลี่ยงปัญหาทางสุขภาพกาย ที่อาจเกิดจากการยืนรอชำระสินค้า หรือการใช้สายตาในการหาข้อมูลสินค้า และพัฒนาสุขภาพจิตจากประสบการณ์การซื้อสินค้าที่ดีขึ้น

อีกทั้งยังเป็นอีกทางเลือกหนึ่งในการที่ร้านค้าจะมาใช้ระบบ Self-Service ที่มีการจัดการที่ดี เจ้าของกิจการร้านค้าปลีกสามารถจัดการบริหารร้านค้าได้สะดวก และมีประสิทธิภาพมากขึ้น ทำให้เกิดความคุ้นเคยกับวัฒนธรรมการซื้อของแบบบริการตนเองในสังคมประเทศไทยมากขึ้น รวมถึงเป็นต้นแบบในการพัฒนาโครงการในลักษณะเดียวกันเพื่อความก้าวหน้าทางเทคโนโลยีในประเทศต่อไป

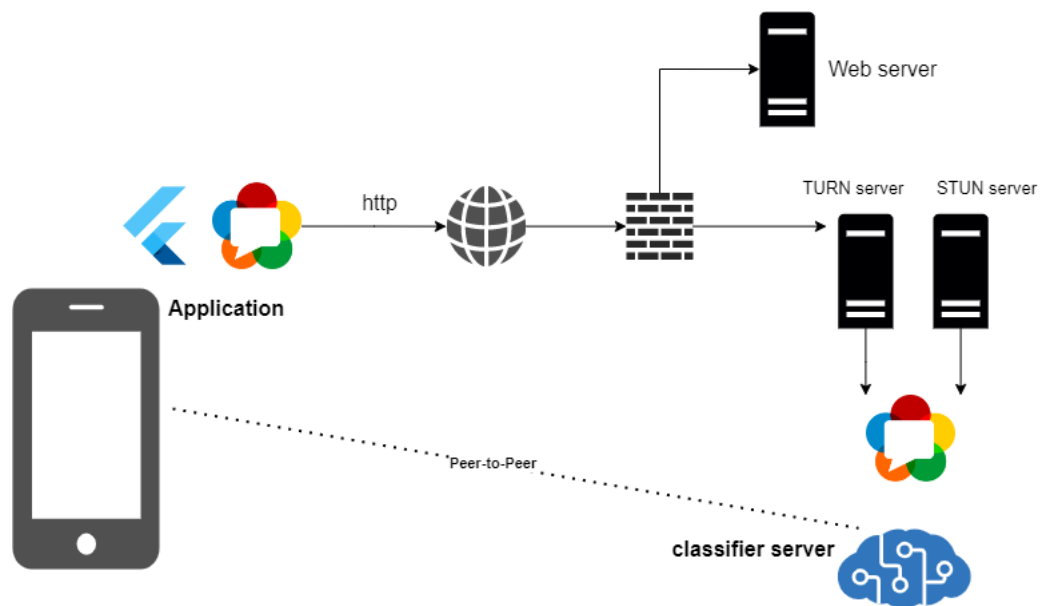
## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

โครงการนี้ได้นำองค์ความรู้ในด้านของ Computational Intelligence และ การ streaming แบบ Peer-to-peer ของรูปภาพจาก application ไปยัง backend ผ่าน WebRTC (Web Real Time Communications) เพื่อให้ backend ที่เป็น Computational Intelligence ทำการ classification products

#### 2.1 WebRTC for Streaming image

WebRTC (Web Real-Time Communication) เป็น open-source ที่ให้บริการ web browsers และ mobile applications ด้วยการสื่อสารแบบเรียลไทม์ (RTC) ผ่าน (API) ทำให้การ Communication ด้วยเสียงและวิดีโอได้ผ่าน Peer-to-peer โดยตรงตามรูป 2.1



รูปที่ 2.1: webrtc structure

#### 2.2 Artificial neural networks

เป็นแขนงหนึ่งของ Computational Intelligence ซึ่งได้รับแรงบันดาลใจมาจากการทำงานของสมองของมนุษย์ ซึ่งประกอบด้วยหน่วยเล็กๆ เรียกว่า cell Neuron แต่ละ Neuron ก็จะเชื่อมต่อโยงใยกันด้วยเส้นประสาทเรียกว่า ซิแนปส์ (Synapse) เพื่อส่งสัญญาณไฟฟ้า ที่เกิดจากสิ่งเร้าต่างๆ ว่าจะตอบสนองต่อสิ่งเร้านั้นอย่างไร โดยแต่ละ Neuron จะได้รับ Input หลาย ๆ อัน จากกิ่งก้านสาขาของ Dendrite แล้วนำมาประมวลผล ออกมาเป็น 1 Output ออกไปที่ Axon เพื่อส่งต่อไปให้ Dendrite ของ Neuron อื่น ๆ ใช้เป็น Input ต่อไป

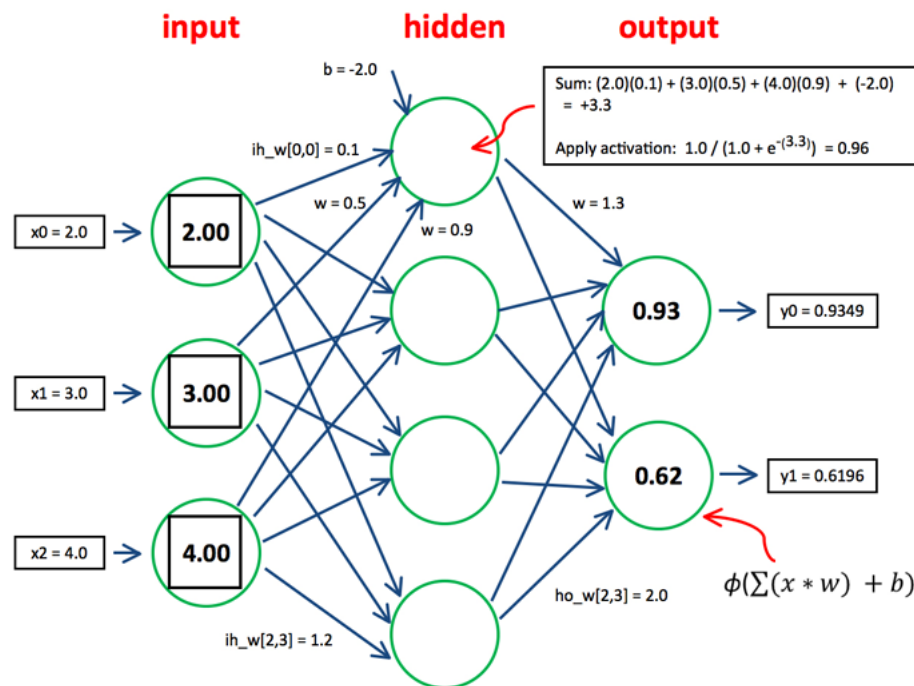
เมื่อมนุษย์เติบโตขึ้น หาก Neuron ไหนตอบสนองต่อสิ่งเร้าประเภทไหนได้ดี ก็จะสามารถส่งสัญญาณไฟฟ้าได้แรง มากกว่า Neuron อื่นๆ เมื่อ Neuron หลายๆอันต่อกันหลายๆ layer ก็จะกลายเป็น Neuron network ของมนุษย์

โดยในทาง Computational Intelligence จะใช้ node เป็นตัวแทนของ Neuron โดยจะเรียงเป็นชั้นๆ (layer) โดยสัญญาณที่ส่งออกจากแต่ละ node จะมี weight ที่กำหนดความแรงของสัญญาณนั้นๆ เมื่อมีหลายๆ node และต่อกันหลายๆ layers ก็จะถูกกลายเป็น neural network ขนาดใหญ่ เรียกว่า deep learning ซึ่งเลียนแบบการทำงานของสมองมนุษย์ ทำให้ computers สามารถ process ข้อมูลในลักษณะเดียวกับที่ สมองของมนุษย์ทำการประมวลผลข้อมูล

### 2.2.1 Multilayer perceptron

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 สำหรับรับข้อมูล (สิ่งเร้า) , hidden layer ในการตัดสินใจ , output layer ในการเลือกการกระทำกับสิ่งเร้านั้นๆ



รูปที่ 2.2: Multilayer perceptron

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

โดยสมการ Input

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

Activation function คือ ฟังก์ชันที่รับ ผลรวมจากทุก Input ( $v_j$ ) แล้วคำนวณว่าจะส่งต่อเป็น Output

เท่าไร ซึ่งมี function มากมายไม่ว่าจะเป็น Hyperbolic tangent ,Sigmoid , ReLU

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

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

โดยที่

$$y(v_i) = \begin{cases} \tanh(v_i) & , \text{Hyperbolic Activation function} \\ (1 + e^{-v_i})^{-1} & , \text{Sigmoid Activation function} \\ \max(0, v_i) & , \text{Sigmoid Activation function} \\ \text{etc} & , \text{etc} \end{cases} \quad (2.3)$$

### 2.2.2 Classification

จาก Input layer ข้อมูลจะถูกส่งต่อไปยัง layer ถัดๆไป เรียกว่า Passed 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 ใดให้ค่า output node เยอะที่สุด input data ก็ถูก classify เป็น class ของ node นั้นๆ

### 2.2.3 Training

คือการ train โดยใช้ dataset เปลี่ยน weight ในแต่ละเส้นของ MLP เพื่อให้ตอบสนองต่อ Input ให้ใกล้เคียงกับสิ่งที่ควรจะเป็นมากขึ้น โดยหนึ่งในวิธีการเปลี่ยน weight คือ backpropagation

โดยค่า error ที่ได้มาจาก output layer กับ ค่าที่ควรได้จาก Input ที่ใส่เข้าไป 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

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

เนื่องจาก train dataset นั้นมีขนาดใหญ่เกินไป จึงมีการแบ่ง dataset ให้เล็กลง ซึ่งจะมีคำศัพท์ ดังนี้

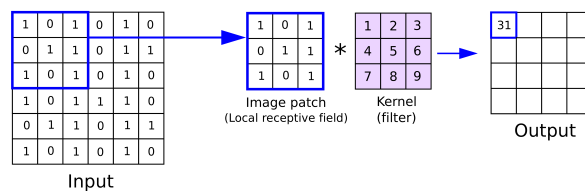
1. Epoch : โดยจาก train dataset 1 Epoch คือการที่ dataset ทั้งหมด Passed forward และ back-propagation
2. Batch Size : แบ่ง dataset เป็น set ขนาดๆ ย่อยๆ
3. Iterations : จำนวน Batch ที่ต้องใช้เพื่อที่จะครบ 1 Epoch

### 2.3 Convolutional neural network

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

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

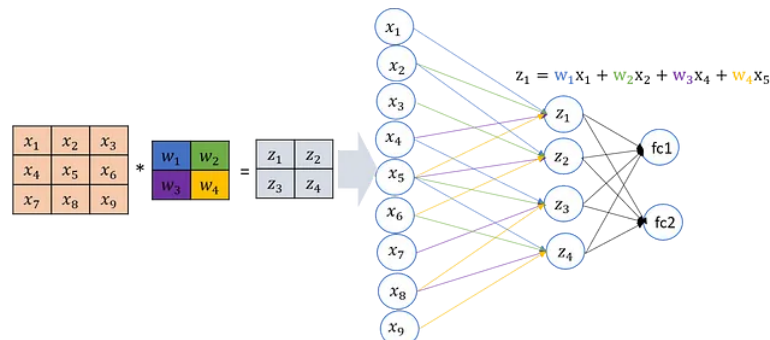
โดยจะแบ่งส่วนของภาพเป็นส่วนย่อยๆ ไปทั่วทั้งรูปภาพ แล้วนำไป convolution กับ "filters" หรือที่เรียกว่า "kernels" เพื่อหา patterns ของรูปภาพ ที่เรียกว่า feature map



รูปที่ 2.3: Convolution

CNNs ต่างจาก neural networks อื่นๆ ตรงที่ shared-weight ร่วมกัน ซึ่งทำให้มีความสามารถในการแยกแยะ patterns ได้ดี

หากเขียนในรูปแบบ Neural network

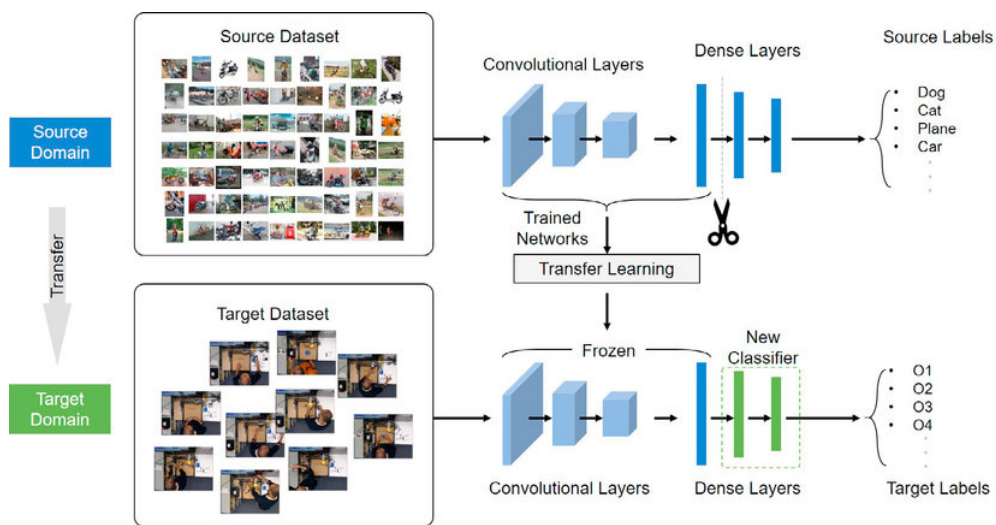


รูปที่ 2.4: Convolutional neural network



## 2.4 Transfer learning

เป็นเทคนิคที่นำ model ที่ผ่านการฝึกฝนจนแก้ ปัญหาในงานอื่นๆที่มีความคล้ายคลึงกัน นำมาเป็น model ตั้งต้น สำหรับ model ในการแก้ปัญหาใหม่ๆ ตัวอย่างเช่น โมเดลที่ได้รับการฝึกฝนให้จดจำวัตถุในภาพสามารถใช้เพื่อระบุวัตถุที่คล้ายกันในภาพต่างๆ ได้ แม้ว่าภาพใหม่จะมีสภาพแสงหรือพื้นหลังต่างก็ตาม กฎเกณฑ์สำคัญคือการระบุคุณสมบัติทั่วไปหรือการเป็นตัวแทนที่ใช้ร่วมกันระหว่างโดเมนต้นทางและโดเมนเป้าหมาย วิธี Transfer Learning ที่ได้รับความนิยมมากที่สุดวิธีหนึ่งคือ fine-tuning คือการใช้โมเดลที่ผ่านการ pre-trained มาแล้ว นำมา train ต่อบน ชุดข้อมูลใหม่ และใช้ learning rate น้อยๆ เพื่อป้องกัน weight ที่เคยผ่านการ train จนมีความแม่นยำเปลี่ยนแปลงไปมาก จนไม่มีความแม่นยำ อีกวิธีหนึ่งคือ feature extraction ซึ่งเกี่ยวข้องกับการใช้ โมเดลที่ผ่านการ pre-trained เป็นตัวแยกคุณลักษณะของ ข้อมูล และ สร้าง model ใหม่เพื่อ train จากคุณลักษณะเหล่านี้ที่ model ตั้งต้นแบ่งแยกออกมาได้

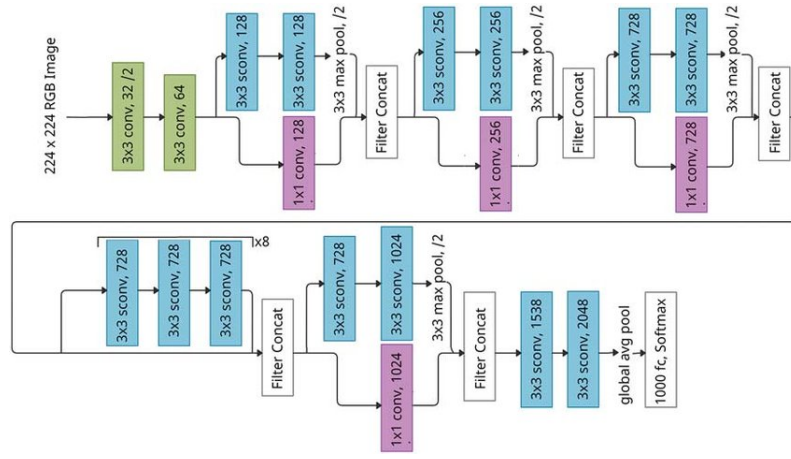


รูปที่ 2.5: The concept of transfer learning

### 2.4.1 transfer learning - Xception

โดย model pre-train 'Xception' เป็น convolutional neural network ซึ่งมีความลึก 71 layers. เป็นโมเดลที่ผ่านการจากรูปภาพต่างๆ มากกว่าล้านรูปภาพจาก ImageNet database [1].

ซึ่งสามารถ

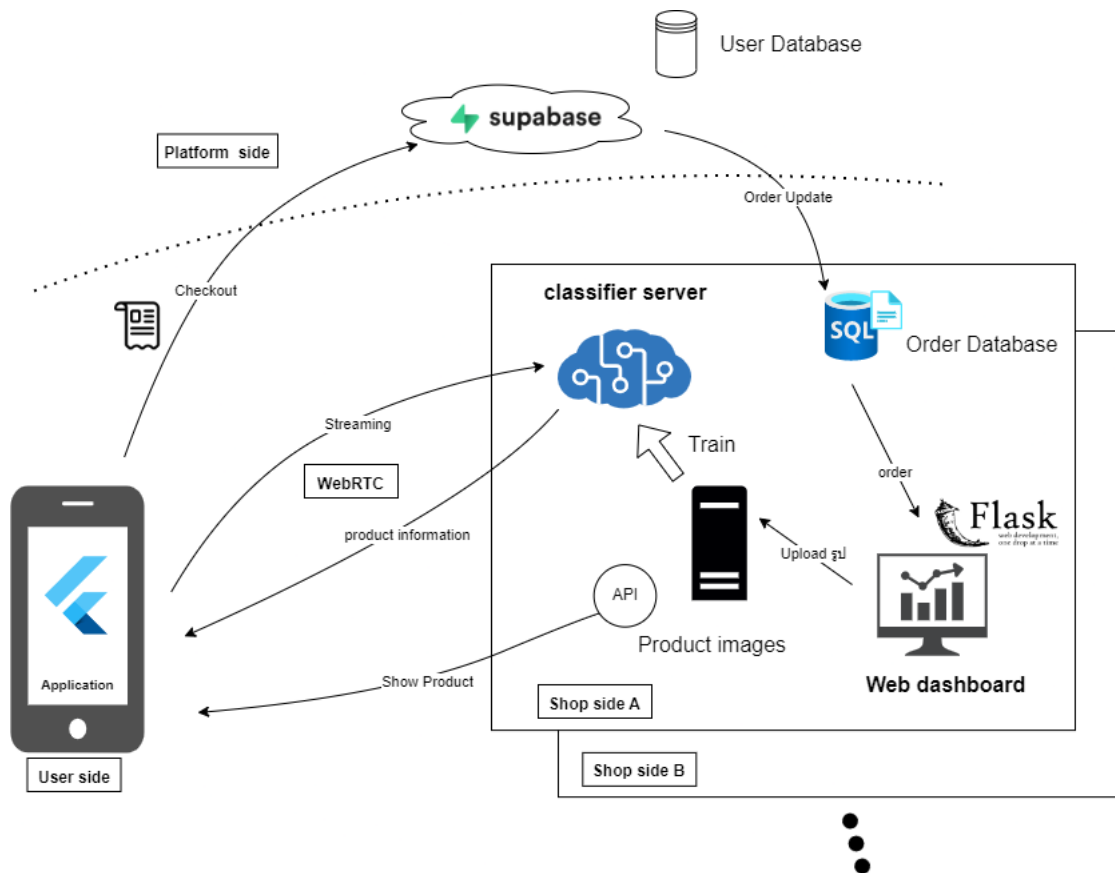


### บทที่ 3

#### โครงสร้างและขั้นตอนการทำงาน

ในบทนี้ จะกล่าวถึงหลักการ, การนำทฤษฎีที่เกี่ยวข้องมาประยุกต์ใช้ และการออกแบบของระบบ

#### 3.1 โครงสร้างของระบบ



รูปที่ 3.1: Overall project structure

#### 3.2 เตรียมชุดข้อมูลฝึกสอน

ข้อมูลที่ใช้ในการ train mode โดยมีสินค้าประมาณ 100 ชนิด โดยจัดเก็บข้อมูลใช้กล้องมือถือ ในการถ่ายภาพในมุมต่างๆ ของสินค้าชนิดนั้นๆ ตามมุมต่างๆ จำนวนชนิดละ N รูป โดยจัดเก็บใน และทำการดึงข้อมูลมา train ผ่าน Google Colab โดยโครงสร้างการเก็บข้อมูลจะเป็นดังรูป

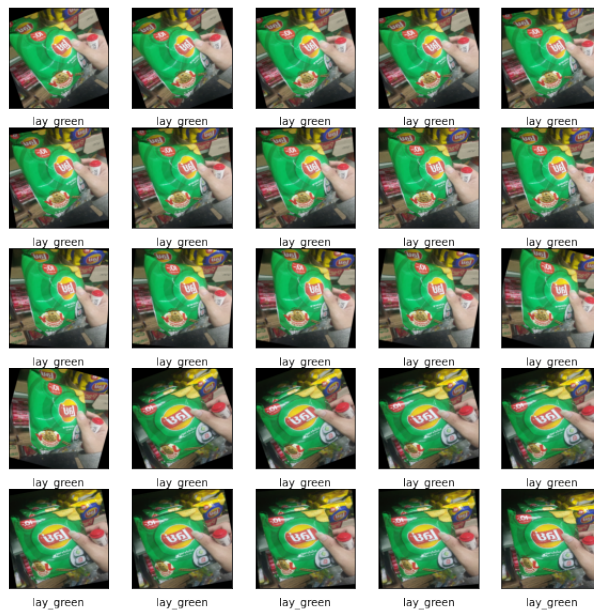
```

directory/
  class_label_1/
    image1
    image2
    .
  class_label_2/
    .
    .

```

### 3.3 การเพิ่ม traing data

โดยสินค้า 1 ชนิด ทำการถ่ายภาพ 6 รูป ในมุมที่ต่างกัน โดยในแต่ละ 1 รูปภาพที่ถ่าย จะแปลงเป็นรูปภาพ RGB ขนาด 224x224 pixel และในแต่ละรูปเพื่อให้มี train dataset จำนวนมาก นำไปหมุนและกลับด้าน ด้วยมุม -20,-15,-10,-5,0,5,10,15,20 องศา โดย 1 รูปภาพผ่านการ generate datasets จะกลายเป็น 18 รูปภาพซึ่งมีความแตกต่างกันเล็กน้อย



รูปที่ 3.2: Dataset generator

### 3.3.1 Model Architecture

โดยโมเดลในโครงงานนี้จะใช้ Xception pre-trained มาใช้ในการแยกคุณลักษณะเด่นของรูปภาพ และสร้างโมเดลมาต่อท้ายเพื่อ เรียนรู้ลักษณะเด่นจากที่ Xception ทำการแยกออกมาได้

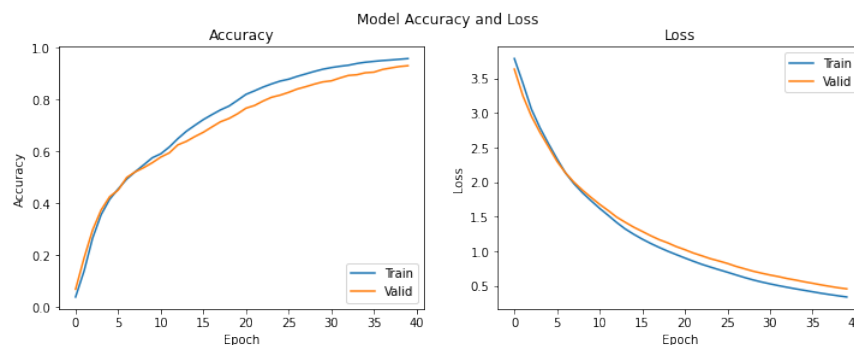
Model: "model_1"		
Layer (type)	Output Shape	Param #
-----		
input_4 (InputLayer)	[(None, 224, 224, 3)]	0
xception (Functional)	(None, 7, 7, 2048)	20861488
flatten_1 (Flatten)	(None, 100352)	0
dense_3 (Dense)	(None, 32)	3211296
dense_4 (Dense)	(None, 16)	528
dense_5 (Dense)	(None, 48)	816
-----		
Total params: 24,074,120		
Trainable params: 3,212,640		
Non-trainable params: 20,861,480		

โดยต่อท้ายด้วย fully connected node ที่เรียกว่า Dense layer ซึ่งทำหน้าที่เป็น classifier โดยมี output layer ที่มีจำนวน node เท่ากับจำนวนสินค้า สำหรับการ classify ชนิดของ products จาก รูปภาพ Subsection 1 text

### 3.4 classification products

จากรูปภาพใน 48 class ผ่านการ generate datasets จะมี dataset ทั้งหมด 4432 sample ทำการแบ่งเป็น train 3546 sample และ 886 sample สำหรับการ evaluate โดยจาก train 3546 แบ่ง 50% สำหรับการ validation ในระหว่างการ train model

ผลลัพธ์ จากการ train & validation ด้วย 3546 sample เป็นจำนวน 40 Epoch



รูปที่ 3.3: Train results

และทำการ save model ที่มีความแม่นยำระดับนี้ สำหรับเป็น service ในการ classify products ของ application ผ่าน aiortc และเว็บ WebRTC

### 3.5 การพัฒนา application

ใช้ Flutter ในการสร้าง application ในส่วนของผู้ใช้โดยใช้ service ของ WebRTC ติดต่อกับ classification server ที่ได้ทำการ train ไว้แล้ว โดยจะ application จะทำการ streaming ภาพสินค้าที่ลูกค้าถ่าย ผ่าน WebRTC ไปยัง server ของทางร้าน ที่มี model classifier อยู่ และ server ของทางร้านก็ classification ว่าเป็นสินค้าชนิดใด และตอบกลับมายัง application

#### 3.5.1 Requirement Specification

1. ผู้คนทั่วไปสามารถลงทะเบียนเข้าใช้งานได้ผ่าน Email , SSO , หมายเลขโทรศัพท์มือถือ
2. สามารถเชื่อมต่อกับข้อมูลร้านค้าได้ผ่านการสแกนคิวอาร์โค้ดเข้าใช้งานร้านค้า
3. สามารถใช้กล้องโทรศัพท์มือถือสแกนสินค้าเพื่อสตรีมภาพ Classification server เพื่อทำการดึงข้อมูลสินค้าจาก Server ของร้านค้ามาแสดงผลบนแอปพลิเคชัน
4. สามารถเพิ่มหรือลดสินค้าในตะกร้าได้
5. สามารถทำการชำระเงินได้ผ่าน Payment gateway ในตัวแอปพลิเคชัน
6. สามารถ Checkout จากร้านค้าผ่านการสแกนคิวอาร์โค้ดออกจากร้านค้า

### 3.6 การเตรียมฐานข้อมูล

ในการเก็บฐานข้อมูลจะแบ่งเป็นฐานข้อมูลสองตัว โดยใช้ Supabase ในการสร้างโครงการ และจัดฐานข้อมูลทั้งหมดแบบ SQL ได้แก่

#### 3.6.1 ฐานข้อมูลของระบบ

สำหรับเก็บข้อมูลร้านค้าที่เข้าร่วม และประวัติการซื้อสินค้าของผู้ใช้งานแอปพลิเคชันมือถือ ซึ่งมี Database schema ดังนี้

#### 3.6.2 ฐานข้อมูลในแต่ละร้านค้า

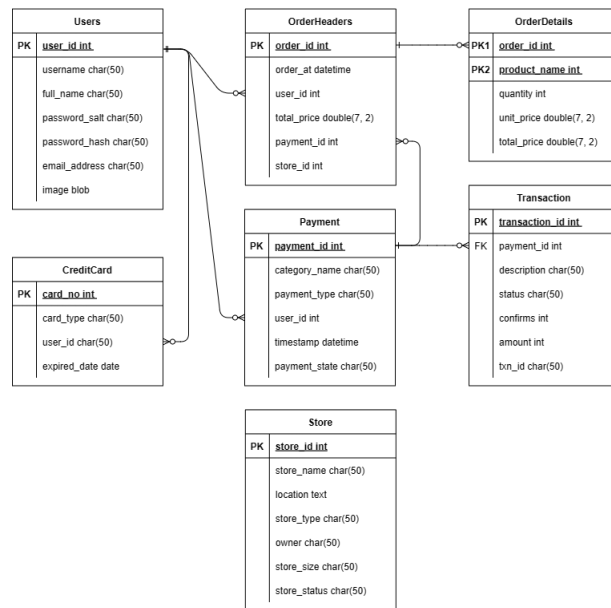
สำหรับเก็บข้อมูลสินค้า และประวัติยอดขายของร้านค้า โดยร้านค้าแต่ละร้านจะมีฐานข้อมูลเป็นของตนเองเพื่อใช้งานกับ Server ของร้านนั้น ๆ โดยตรง ซึ่งจะมี Database schema ดังนี้

### 3.7 การพัฒนา web dashboard

ออกแบบ UI/UX ของเว็บไซต์และ application ด้วย Figma หลังจากการออกแบบ และจัดการตั้งค่าฐานข้อมูลเสร็จแล้ว ก็จะพัฒนาในส่วนของเว็บไซต์ทางฝั่งร้านค้าโดยใช้ Supabase ในการโฮสต์เว็บไซต์ และเชื่อมต่อกับฐานข้อมูลที่ได้ตั้งค่าไว้ โดยใช้ Flask framework ในการจัดการ API และ Python ในการจัดการระบบ Backend ของเว็บไซต์โดยใช้ SQLAlchemy ในการดึงข้อมูลจากฐานข้อมูล Requirement Specification ดังนี้

1. สามารถเข้าใช้งานได้ผ่านการยืนยันตัวตนเป็น Administers เท่านั้น โดยสามารถมีได้ 1-5 คน

## Mobile Application Database Schema



รูปที่ 3.4: Mobile Application Database Schema

2. สามารถดูคลังสินค้า และแก้ไขข้อมูลสินค้าในแต่ละชนิดได้แบบเรียลไทม์
3. สามารถดูสถิติยอดขายสินค้าได้ทั้งแบบรายวัน รายเดือน และรายปี โดยแบ่งได้ 2 แบบ คือตามชนิดสินค้า และประเภทสินค้า
4. สามารถดูประวัติการขายตามออเดอร์ของลูกค้าแต่ละคนได้ และดูข้อมูลของแต่ละออเดอร์ได้

### 3.8 การทดสอบการทำงานของซอฟต์แวร์

การทดสอบการทำงานของระบบ สามารถแบ่งเป็นขั้นตอนดังนี้

#### 3.8.1 Unit testing

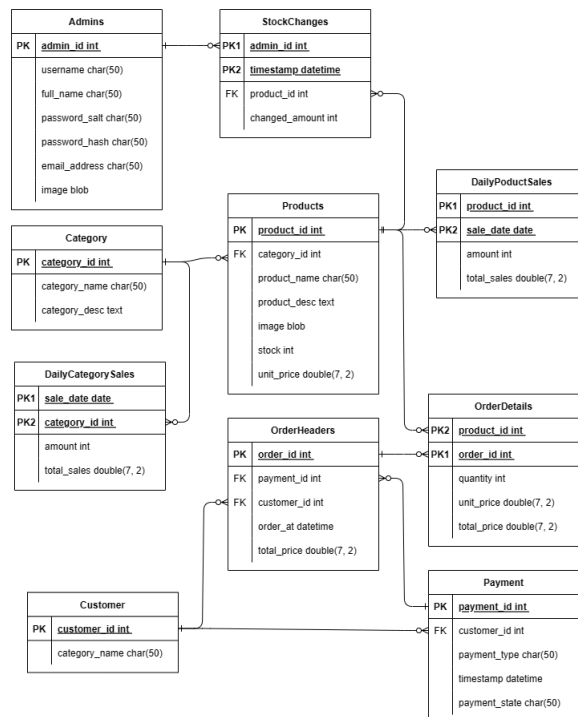
การทดสอบความถูกต้องของการทำงานในแต่ละฟังก์ชันหลักของระบบแยกกัน โดยยังไม่รวมแต่ละ Component เข้าด้วยกัน ซึ่งได้แก่

1. Classification system
2. Website dashboard
3. Mobile Application

#### 3.8.2 Integration testing

การทดสอบการทำงานเมื่อรวมระบบย่อยทั้งหมดเข้าด้วยกัน โดยหลัก ๆ จะทดสอบในเรื่อง API ว่ามีการรับส่งข้อมูลถูกต้องหรือไม่ และทำงานโดยรวมได้ถูกต้องทั้งหมดหรือไม่

### Store Website Dashboard Database Schema



รูปที่ 3.5: Store Website Dashboard Database Schema

### 3.8.3 System testing

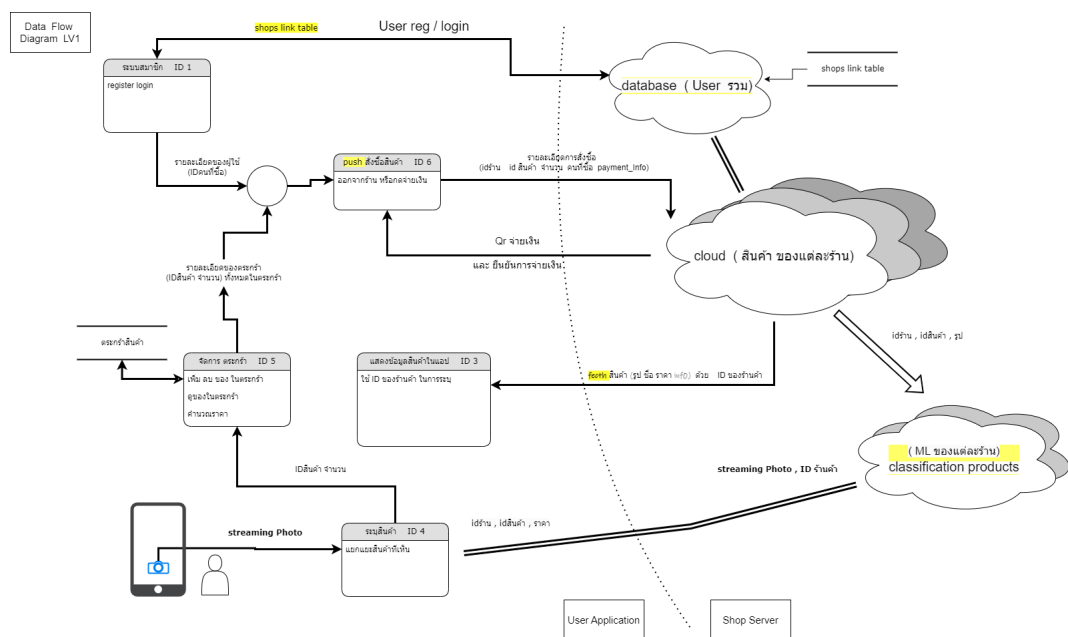
การทดสอบระบบซึ่งแต่ละโมดูลข้างต้นจะถูกรวม และทดสอบเป็นกลุ่ม เพื่อประเมินความสอดคล้องของระบบว่าทำงานได้ตามที่กำหนดไว้หรือไม่

### 3.8.4 Acceptance testing

การทดสอบระบบโดยดูภาพรวมของการทำงาน ว่ามีการตอบสนองความต้องการของผู้ใช้ทั้งในส่วนของฟังก์ชันการทำงาน และประสิทธิภาพการทำงาน ว่าสอดคล้องกับลักษณะของความต้องการของซอฟต์แวร์หรือไม่ โดยการใช้การทดสอบแบบ Functional testing (Black box testing)

## 3.9 แผนภาพกระแสข้อมูล (Data Flow Diagram)



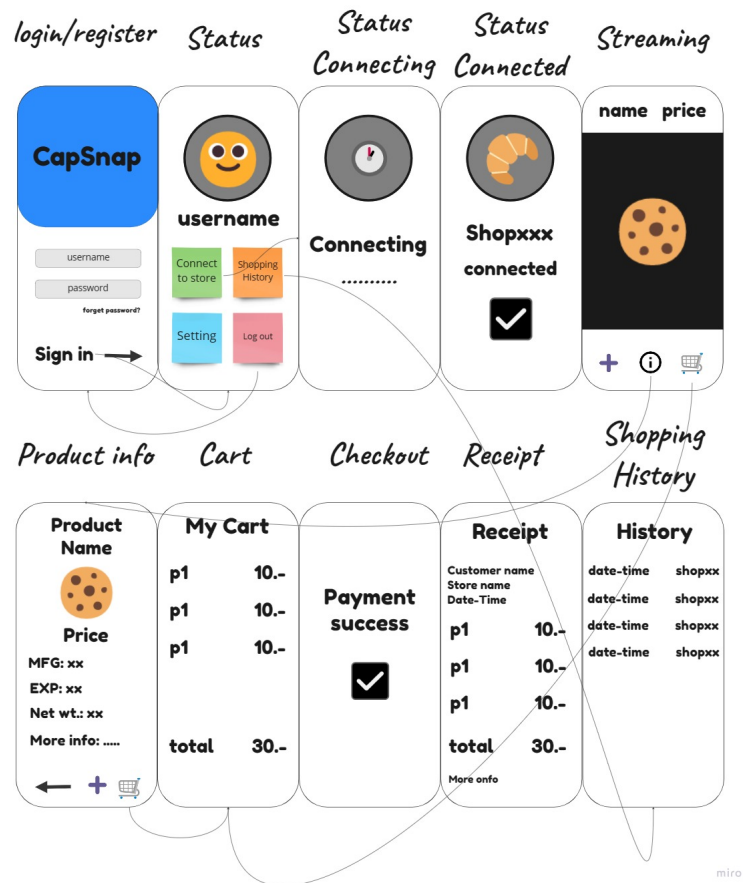


รูปที่ 3.6: Data Flow Diagram

## บทที่ 4

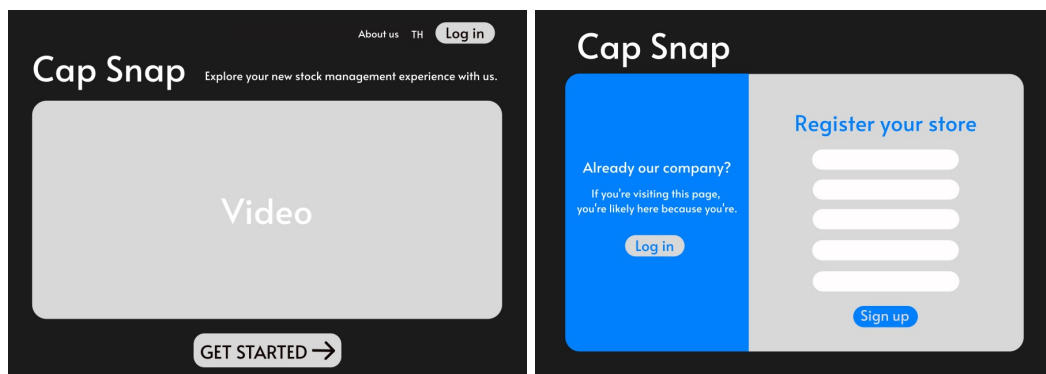
### การทดลองและผลลัพธ์

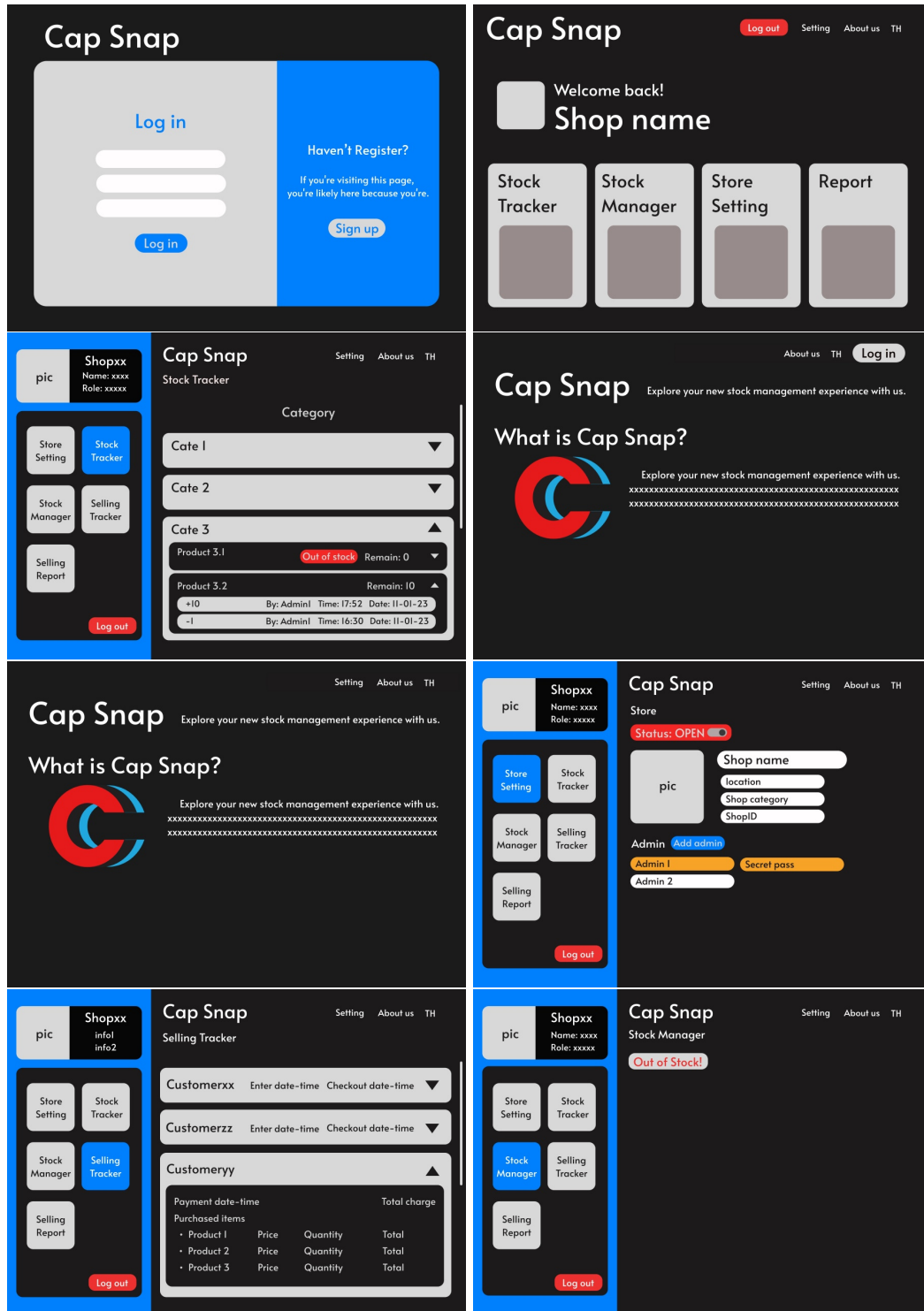
#### 4.1 Application UX/UI

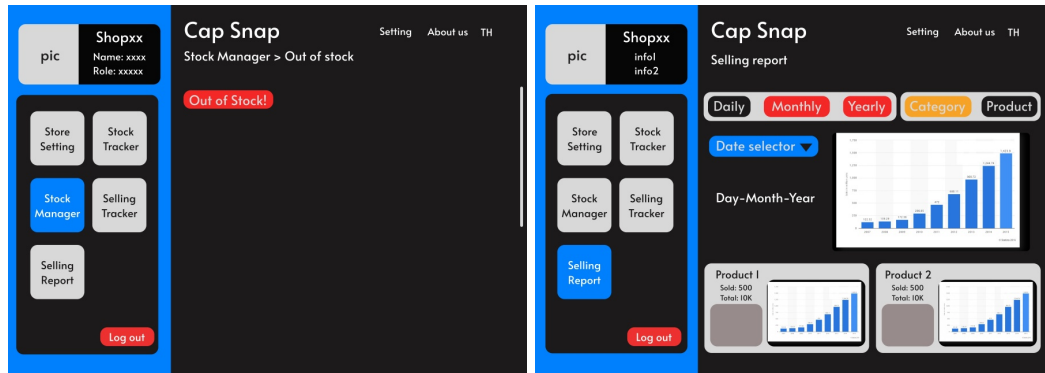


รูปที่ 4.1: Application wire frame

#### 4.2 Web Dashboard UX/UI







## บทที่ 5

### บทสรุปและข้อเสนอแนะ

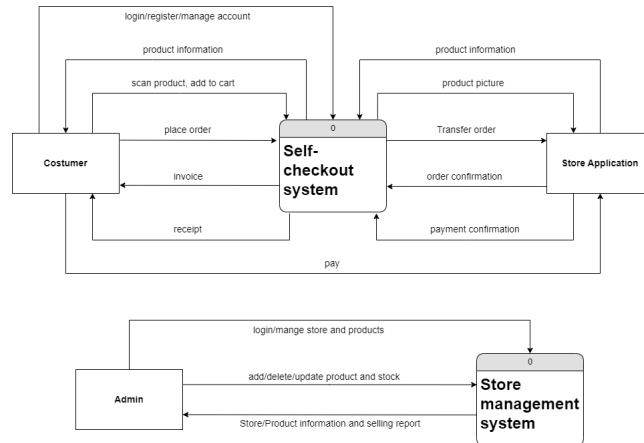
- 5.1 สรุปผล
- 5.2 ปัญหาที่พบและแนวทางการแก้ไข
- 5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ

ภาคผนวก

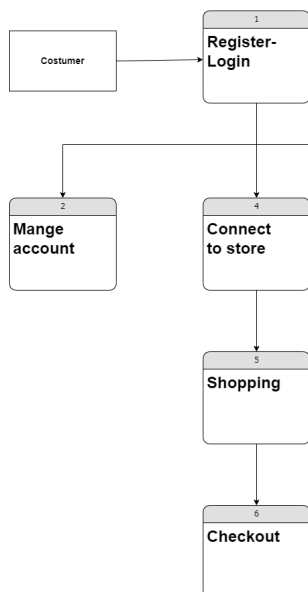
## ภาคผนวก ก

### The first appendix

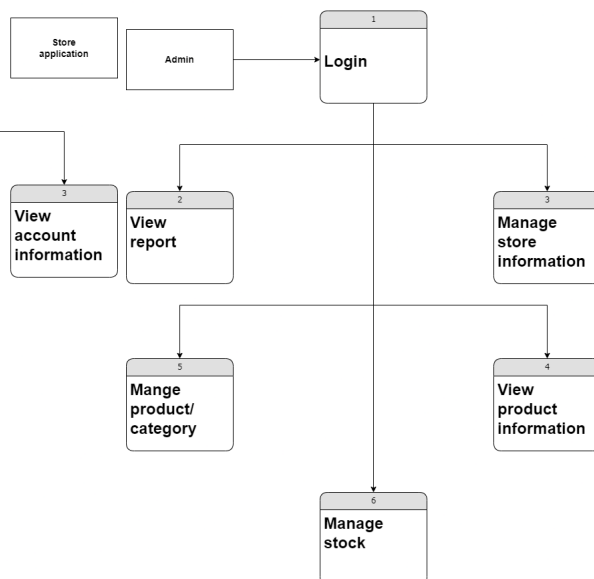
DFD level 0: Self-checkout system & Store management system



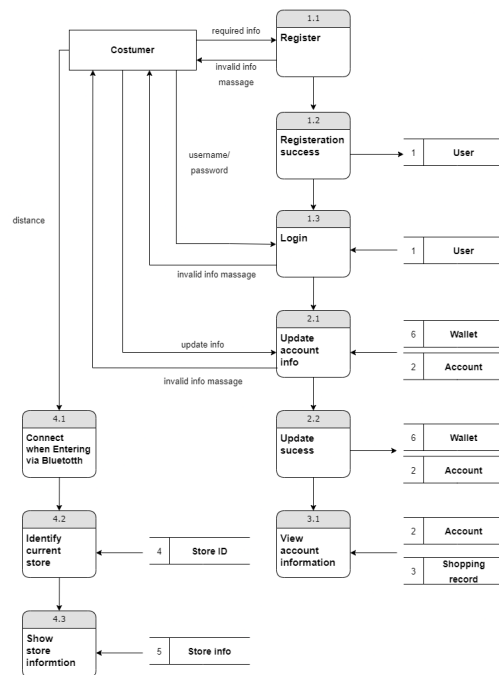
DFD level 1: Self-checkout system



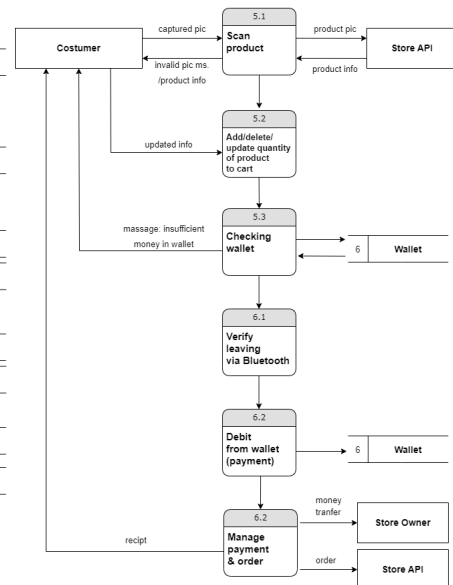
DFD level 1: Store management system



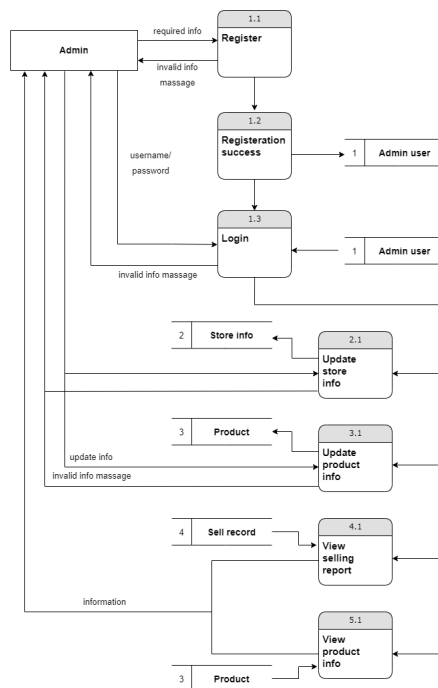
DFD level 2: Self-checkout system (Process 1-4)



DFD level 2: Self-checkout system (Process 5-6)



DFD level 2: Store management system





ประวัติผู้เขียน