

ระบบสแกนใบหน้าสำหรับแคชเชียร์
Recognition for cashier

นางสาวภัทราภรณ์ อินทา

โครงงานนี้เป็นส่วนหนึ่งของการศึกษา
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ภาควิชาคณิตศาสตร์ สถิติ และคอมพิวเตอร์ คณะวิทยาศาสตร์
มหาวิทยาลัยอุบลราชธานี
ปีการศึกษา 2562
ลิขสิทธิ์ของมหาวิทยาลัยอุบลราชธานี

โครงการ : ระบบสแกนใบหน้าสำหรับแคชเชียร์
Recognition for cashier
โดย : นางสาวภัทราภรณ์ อินทา
อาจารย์ที่ปรึกษา : ดร.ไพชยนต์ คงไชย
ระดับการศึกษา : วิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์
ปีการศึกษา : 2562

ได้รับการพิจารณาให้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์

คณะกรรมการสอบประเมินความรู้โครงการคอมพิวเตอร์

..... อาจารย์ที่ปรึกษา
(ดร.ไพชยนต์ คงไชย)

..... กรรมการ
(ดร. วราวุฒิ ผ้าเจริญ)

..... กรรมการ
(ดร. ทศพร จูณิม)

..... หัวหน้าภาควิชา
(ผศ.ดร. สุพจน์ สืบบุตร)

วันที่ ... / ... / ...

กิตติกรรมประกาศ

การพัฒนาโครงการระบบ สแกนใบหน้าสำหรับแคชเชียร์ สำเร็จลุล่วงได้ด้วยความกรุณาและความช่วยเหลือจากหลายๆ ท่าน ข้าพเจ้าขอขอพระคุณทุกท่าน ที่มีส่วนร่วมในการพัฒนาโครงการนี้

ขอขอบพระคุณอาจารย์ที่ปรึกษา ดร.ไพชยนต์ คงไชย อาจารย์ที่ปรึกษาโครงการที่ได้แนะนำทฤษฎีและแนวทางในแก้ปัญหาต่าง ๆ ที่เกิดขึ้นระหว่างการพัฒนา ระบบ อีกครั้งยังคอยตรวจสอบความก้าวหน้าของการทำงานเป็นระยะ ๆ รวมทั้งสร้างกำลังใจให้ผู้พัฒนาอยู่เสมอ

ขอขอบพระคุณเจ้าหน้าที่ภาควิชาคณิตศาสตร์ สถิติและคอมพิวเตอร์ ที่คอยเอื้ออำนวยความสะดวกทั้งเรื่องอุปกรณ์และสถานที่ต่อการปฏิบัติงานของผู้พัฒนา

ขอกราบขอบพระคุณบิดา มารดา ที่คอยให้กำลังใจ คอยให้ความรักและความห่วงใยเสมอมา ตลอดจนคอยช่วยเหลือทุนทรัพย์ทางด้านการศึกษาและอุปกรณ์ในการพัฒนาโครงการ

ขอบคุณเพื่อน ๆ สาขาวิทยาการคอมพิวเตอร์ชั้นปีที่ 4 ที่ได้คอยช่วยแก้ไขปัญหาและให้คำปรึกษาในการพัฒนาโครงการครั้งนี้จนเสร็จสิ้น

นางสาวภัทราภรณ์ อินทา

วันที่ 8 มิถุนายน 2563

โครงการ	:	ระบบสแกนใบหน้าสำหรับแคชเชียร์
โดย	:	นางสาวภัทราภรณ์ อินทา
อาจารย์ที่ปรึกษา	:	ดร.ไพชยนต์ คงไชย
ระดับการศึกษา	:	วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ปีการศึกษา	:	2562

บทคัดย่อ

แอปพลิเคชันสแกนใบหน้าสำหรับแคชเชียร์ เป็นเว็บแอปพลิเคชันสำหรับการชำระเงินของลูกค้า ที่ทำงานบนเว็บเบราว์เซอร์ (Web browser) ใช้ Sqlite3 ซึ่งเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ ในการจัดเก็บข้อมูล เว็บแอปพลิเคชันถูกพัฒนาด้วย Django และใช้ภาษาโปรแกรม Python ผู้ใช้งานจะแบ่งออกเป็น 3 กลุ่ม ได้แก่ ลูกค้า พนักงานแคชเชียร์ และเจ้าของร้าน โดยเจ้าของร้าน จะมีการจัดเก็บข้อมูลพนักงาน จำกัดสิทธิ์การใช้งานของพนักงาน เพื่อให้พนักงานทำการใช้งานเว็บแอปพลิเคชันเพื่อทำการสั่งซื้อสินค้า และจัดการข้อมูล ในส่วนลูกค้าจะมีการสมัครสมาชิก เพื่อเก็บข้อมูลที่มาใช้บริการสำหรับลูกค้าที่ต้องการเป็นสมาชิกในระบบ และการสแกนใบหน้านี้จะใช้แทนบัตรสมาชิกสำหรับลูกค้าที่เป็นสมาชิกแล้ว เมื่อสแกนใบหน้าแล้วจะมีการแสดงข้อมูลในระบบ จะประกอบด้วย ประวัติลูกค้า รายการสั่งซื้อ สำหรับลูกค้าที่เป็นสมาชิกเท่านั้น ส่วนที่ไม่เป็นสมาชิกจะไม่มีการเก็บข้อมูลลูกค้า จะสามารถสั่งซื้อสินค้าได้เพียงเท่านั้น จุดเด่นของแอปพลิเคชันจะมีเนื้อหาชี้ให้เห็นได้ชัดเจนถึงการสแกนใบหน้าเพื่อเก็บข้อมูลของลูกค้า และบันทึกข้อมูล ประวัติการสั่งซื้อสินค้าของลูกค้าทุกครั้ง และใช้แทนบัตรสมาชิกในการระบุตัวตนว่าเป็นสมาชิกในระบบแล้ว

คำสำคัญ: เว็บแอปพลิเคชัน ระบบการรู้จำใบหน้า ไพธอน ดีจังโก้เฟรมเวิร์ค

Topic : Recognition for cashier
Author : PATTARAPORN INTA
Advisor : PAICHAYON KONGCHAI , Ph.D.
Degree : Bachelor of Science (Computer Science)
Academic Year : 2019

Abstract

Face scanning application for Kashy Is a web application for payment of customers That runs on a web browser use Sqlite3, a relational database management system To store data The web application is developed by Django and uses Python programming language. Users are classified into 3 groups: customers, employees, employees. And the shopkeeper By the shopkeeper There will be staff storage. Restrict employee privileges For employees Make an order And manage information for customers There will be a subscription to collect information of customers who come to use the service for customers who want to be members in the system. And this face scan will be used as a membership card for customers that are members When the face is scanned then the information is displayed in the system, consisting of customer history Order list For customers that are members only Non-members do not collect customer information. Will be able to order products only The highlight of the application is the content that clearly shows the face scan to collect customer information and save data. This purchase history of every customer's product. And use the membership card to identify yourself as a member in the system
Keywords: Web Application Facerecognition

สารบัญ

	หน้า
กิตติกรรมประกาศ	ค
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
สารบัญ	ฉ
สารบัญตาราง	ณ
สารบัญภาพ	ญ
บทที่	
1 บทนำ	1
1.1 ที่มาและเหตุผล	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 เครื่องมือที่ใช้ในการพัฒนา (Development tools)	2
1.5.1 ฮาร์ดแวร์	2
1.5.2 ซอฟต์แวร์ (Software)	3
1.5.3 แผนการดำเนินการ	4
2 ทฤษฎีที่เกี่ยวข้อง	5
2.1 ความรู้พื้นฐานเกี่ยวกับ Python	5
2.2 ความรู้พื้นฐาน Django Framework	5
2.3 ความรู้พื้นฐานเกี่ยวกับ Bootstrap	7
2.4 ความรู้พื้นฐานเกี่ยวกับ OpenCV(OpenSourceComputerVision)	7
2.4.1 การประมวลผลภาพ (Image Processing)	7
2.5 ความรู้เกี่ยวกับ SQLite	10
2.6 ความรู้พื้นฐานเกี่ยวกับ Pillow	12
2.7 ความรู้พื้นฐานเกี่ยวกับปัญญาประดิษฐ์ (Artificial Intelligence)	13
2.7.1 โครงข่ายประสาทเทียม (Neural Network)	14
2.7.2 ระบบรู้จำใบหน้า(Face Recognition Algorithm)	15

2.7.3	การเรียนรู้แบบต้นไม้ตัดสินใจ (decision tree learning)	16
2.7.4	Random Forest	17
2.8	งานวิจัยที่เกี่ยวข้อง	19
3	การวิเคราะห์และออกแบบระบบ	20
3.1	โครงสร้างภาพรวมของระบบ	21
3.2	System Requirements	22
3.2.1	Functional Requirements	22
3.2.2	Non-functional Requirements	22
3.3	User Interface Design	23
3.3.1	ส่วนของผู้ดูแลระบบ	25
3.4	Use Case Diagram	26
3.5	Class Diagram	31
3.6	Sequence Diagram	35
3.7	โครงสร้างฐานข้อมูล SQLite3	40
4	การพัฒนาระบบ	44
4.1	การพัฒนาในส่วนการสมัครสมาชิกหรือเพิ่มข้อมูล	44
4.2	การพัฒนาในส่วนการเข้าสู่ระบบโดยการสแกนใบหน้า	47
4.3	การพัฒนาในส่วนการเข้าสู่ระบบของพนักงาน	48
4.4	การพัฒนาในส่วนการแสดงผลข้อมูลลูกค้า	49
4.5	การพัฒนาในส่วนการสั่งซื้อสินค้า	50
4.6	การพัฒนาในส่วนของระบบรู้จำใบหน้า	51
5	การทดสอบระบบ	57
5.1	การทดสอบในส่วนฟังก์ชันของระบบ	57
5.1.1	ผลการทดสอบการสมัครสมาชิกสำหรับลูกค้า	57
5.1.2	ผลการทดสอบการเข้าสู่ระบบผู้ใช้งาน	58
5.1.3	ผลการทดสอบการแก้ไขข้อมูลลูกค้า	58
5.1.4	ผลการทดสอบหน้าสั่งซื้อสินค้า	59
6	สรุปและข้อเสนอแนะ	60
6.1	สรุปความสามารถของระบบ	60

6.2 ปัญหาและอุปสรรคในการพัฒนา	60
6.3 แนวทางการพัฒนาต่อ	61
บรรณานุกรม	62
ภาคผนวก	64
ภาคผนวก ก การติดตั้งเครื่องมือที่ใช้พัฒนาโปรแกรม	64
ก.1 การติดตั้ง Visual Studio Code	64
ก.2 การติดตั้ง Django Framework	67
ภาคผนวก ข คู่มือการติดตั้งระบบ	69
ภาคผนวก ค คู่มือการใช้งานระบบ	71
ค.1 ส่วนผู้ใช้งาน	71
ค.2 ส่วนผู้ดูแลระบบ	76
ประวัติผู้พัฒนา	80

สารบัญตาราง

ตารางที่	หน้า
1.1 ขั้นตอนการดำเนินงาน	4
3.1 สัญลักษณ์ของ Use case Diagram	26
3.2 อธิบาย Use Case หน้าที่ของระบบ ในภาพที่ 3.9	28
3.3 Use Caseสมัครสมาชิก	28
3.4 Use Case สแกนใบหน้าลูกค้า	29
3.5 Use Case แสดงข้อมูลลูกค้า	29
3.6 Use Case สั่งซื้อสินค้า	30
3.7 สัญลักษณ์ของ Class Diagram	31
3.8 อธิบาย Class Diagram ของคลาสพื้นฐานของระบบ	33
3.9 อธิบาย Class Diagram ของคลาสพื้นฐานของระบบ	34
3.10 สัญลักษณ์ของ Sequence Diagram	35
3.11 อธิบายโครงสร้างฐานข้อมูลของ datamodel ของ Order	41
3.12 อธิบายโครงสร้างฐานข้อมูลของ datamodel ของ Incustomer	41
3.13 อธิบายโครงสร้างฐานข้อมูลของ datamodel ของ Employees	42
3.14 อธิบายโครงสร้างฐานข้อมูลของ datamodel ของ user	42
3.15 อธิบายโครงสร้างฐานข้อมูลของ datamodel ของ Product	43
5.1 ผลการทดสอบการสมัครสมาชิกสำหรับลูกค้า	57
5.2 ผลการทดสอบการเข้าสู่ระบบโดยการสแกนใบหน้าลูกค้า	58
5.3 ผลการทดสอบการแก้ไขข้อมูลลูกค้า	58
5.4 ผลการทดสอบหน้าสั่งซื้อสินค้า	59

สารบัญภาพ

รูปที่	หน้า
2.1	รูปแบบ MVC ของ Django 6
2.2	ภาพแบบ Binary 8
2.3	ภาพแบบ Grayscale Image 8
2.4	ภาพแบบ RGB Image 9
2.5	การขยายรูปโดยใช้ Pillow 13
2.6	ขั้นตอนของโครงข่ายประสาทเทียมแบบคอนโวลูชันนัล 14
2.7	ภาพแสดงกระบวนการทำงานของระบบรู้จำใบหน้า 16
2.8	หลักการทำ Random Forest 18
3.1	ภาพรวมและโครงสร้างการทำงานของระบบ 21
3.2	หน้าเข้าสู่ระบบสำหรับพนักงาน 23
3.3	หน้าเข้าสู่ระบบสำหรับพนักงาน 23
3.4	หน้าจอหลัก 24
3.5	หน้าจอแสดงข้อมูลลูกค้า 24
3.6	หน้าจอสั่งซื้อสินค้า 24
3.7	หน้าจอเข้าสู่ระบบผู้ดูแลระบบ 25
3.8	หน้าจอฐานข้อมูลทั้งหมดในระบบ 25
3.9	Use Case Diagram ระบบสแกนใบหน้าสำหรับแคชเชียร์ 27
3.10	Class Diagram ของแอปพลิเคชันระบบสแกนใบหน้าสำหรับแคชเชียร์ 32
3.11	Sequence Diagram การสมัครสมาชิก 36
3.12	Sequence Diagram ล็อกอินลูกค้าโดยการสแกนใบหน้า 37
3.13	Sequence Diagram การแก้ไขข้อมูลลูกค้าที่เป็นสมาชิก 38
3.14	Sequence Diagram การสั่งซื้อสินค้า 39
3.15	ตารางฐานข้อมูลรายการสั่งซื้อสินค้า 40
3.16	ตารางฐานข้อมูลสินค้า 43
4.1	การทำงานของระบบเมื่อผู้ใช้สมัครสมาชิกหรือเพิ่มข้อมูล 44
4.2	การทำงานของฟังก์ชันaddFace() 45
4.3	การทำงานของระบบเมื่อผู้ใช้สมัครสมาชิกหรือเพิ่มข้อมูล (ต่อ) 46
4.4	การทำงานของระบบเมื่อผู้ใช้เข้าสู่ระบบโดยการสแกนใบหน้า 47
4.5	การทำงานของระบบเมื่อผู้ใช้เข้าสู่ระบบโดยการสแกนใบหน้า 48
4.6	การทำงานของระบบเมื่อผู้ใช้กดดูรายการเสื้อผ้าทั้งหมด 49
4.7	การทำงานของระบบเมื่อผู้ใช้กดดูรายละเอียดสินค้า 50
4.8	การทำงานของระบบเมื่อผู้ใช้กดแก้ไขสินค้า 51
4.9	การทำงานของฟังก์ชัน recognizeFace 53
4.10	การทำงานของฟังก์ชัน recognizeFace() 55
ก.1	หน้าเว็บดาวน์โหลด Visual Studio Code 64

ก.2	หน้าต่างต้อนรับของ Visual Studio Code	65
ก.3	หน้าต่างข้อตกลงการใช้งาน Visual Studio Code	65
ก.4	หน้าต่างที่จัดเก็บไฟล์ต่างๆ ของ Visual Studio Code	66
ก.5	หน้าต่างติดตั้งโปรแกรม Visual Studio Code	66
ก.6	หน้าต่างผลการติดตั้ง Visual Studio Code	67
ก.7	การติดตั้ง Django Framework	67
ก.8	การติดตั้งส่วนของโปรเจค	67
ก.9	การติดตั้ง Django Framework	68
ก.10	การรัน Server เพื่อใช้งานระบบ	68
ข.1	หน้าเว็บ Repository ของโปรเจ็ค	69
ข.2	คำสั่ง migrate สร้างตาราง	69
ข.3	คำสั่ง migrate สร้างตาราง	70
ข.4	คำสั่ง migrate สร้างตาราง	70
ข.5	คำสั่ง migrate สร้างตาราง	70
ค.1	หน้าจอเข้าสู่ระบบสำหรับพนักงาน	71
ค.2	หน้าจอสมัครสมาชิก	72
ค.3	หน้าจอหลักของ	73
ค.4	หน้าจอแสดงข้อมูลลูกค้า	74
ค.5	หน้าจอสั่งซื้อสินค้า	75
ค.6	หน้าจอเข้าสู่ระบบสำหรับผู้ดูแลระบบในฐานข้อมูล	76
ค.7	หน้าฐานข้อมูลทั้งหมดในระบบ	77
ค.8	หน้าจอแสดงข้อมูลร้านค้าทั้งหมด	78
ค.9	หน้าจอแสดงรายการสั่งซื้อ โดยผู้ดูแลระบบสามารถเพิ่ม ลบ แก้ไขได้	78
ค.10	หน้าจอแสดงข้อมูลรายชื่อลูกค้า	79
ค.11	หน้าจอแสดงรายชื่อลูกค้าที่เป็นสมาชิก โดยผู้ดูแลระบบสามารถเพิ่ม ลบ แก้ไขได้	79

บทที่ 1

บทนำ

1.1 ที่มาและเหตุผล

การตรวจสอบเพื่อยืนยันตัวตนของลูกค้าที่มาชำระเงินนั้นมีหลากหลายวิธี เช่น การตรวจสอบจากข้อมูลที่ลงทะเบียนโดยการบันทึกข้อมูลหรือประวัติส่วนตัวไว้ แต่ละวิธีก็มีระยะเวลาในการประมวลผลแตกต่างกันและใช้เวลานาน บางวิธีจะต้องมีอุปกรณ์เพื่อช่วยให้เข้าถึงข้อมูล ในบางครั้งลูกค้าลืมบัตรสมาชิกบ้าง จำเบอร์โทรศัพท์เพื่อดูข้อมูลไม่ได้บ้าง ซึ่งทำให้เกิดความยุ่งยากมากยิ่งขึ้น

ปัจจุบันเทคโนโลยีเกี่ยวกับการประมวลผลภาพได้ถูกนำไปประยุกต์ใช้มากมาย เช่น การสแกนใบหน้าเพื่อเข้าใช้งานโทรศัพท์มือถือ การสแกนลายนิ้วมือ อีกทั้งการนำไปจดจำใบหน้าเพื่อเก็บข้อมูล และนำข้อมูลมาใช้ในครั้งต่อไป และเทคโนโลยีการประมวลผลภาพนี้ยังเป็นที่น่าสนใจในสังคมตอนนี้ เพราะมันง่ายต่อการใช้งาน ไม่ต้องมีอุปกรณ์ที่ช่วยในการเข้าถึงข้อมูล ใช้แค่ใบหน้าในการตรวจสอบ อีกทั้งยังมีการนำ Machine learning มาใช้ในการช่วยตัดสินใจในการสั่งสินค้าของลูกค้า

ด้วยเหตุนี้จึงมีความประสงค์ที่จะนำเทคโนโลยีการประมวลผลภาพมาใช้ เพื่อที่จะบันทึกใบหน้าของลูกค้าที่มาชำระเงิน เพื่อนำข้อมูลมาใช้ในครั้งต่อไป แทนการใช้อุปกรณ์ที่ช่วยในการเข้าถึงข้อมูลลูกค้าที่ยุ่งยาก และสะดวกต่อลูกค้าเพื่อช่วยแก้ไขการลืมบัตรสมาชิก สะกดชื่อไม่ถูก ลืมเบอร์โทรศัพท์ที่ใช้ในการสมัคร และช่วยในการตัดสินใจสั่งสินค้าได้รวดเร็วขึ้น โดยใช้ Decision Tree มาช่วยในการแนะนำเมนู

1.2 วัตถุประสงค์

1. เพื่อออกแบบและพัฒนาแอปพลิเคชัน จดจำใบหน้าในการชำระเงิน
2. เพื่อดูข้อมูล และประวัติการสั่งซื้อที่ผ่านมาของลูกค้าได้ โดยผ่านการสแกนใบหน้าลูกค้า
3. ใช้การสแกนใบหน้าแทนการใช้บัตรสมาชิก
4. เพื่อช่วยลดทรัพยากรที่ใช้ในการผลิตบัตรสมาชิก และอุปกรณ์อื่น ๆ ที่ต้องใช้ร่วมกัน

1.3 ขอบเขตของโครงการ

1. ใช้สำหรับพนักงานแคชเชียร์เท่านั้น
2. เพื่อดูข้อมูล และประวัติการสั่งซื้อที่ผ่านมาของลูกค้าได้อย่างรวดเร็ว โดยผ่านการสแกนใบหน้าลูกค้าเท่านั้น
3. ไม่ต้องมีบัตร ก็สามารถดูข้อมูลต่างๆของลูกค้าได้ จากการสแกนใบหน้า
4. พนักงานแคชเชียร์สามารถส่งสินค้าให้ลูกค้าได้เท่านั้น

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ช่วยลดปริมาณทรัพยากรที่ใช้ในการผลิตบัตร
2. ช่วยให้พนักงานแคชเชียร์ทำงานได้ถูกต้องและแม่นยำ
3. ช่วยลดปัญหาลูกค้าลืมบัตรสมาชิก ลืมชื่อที่ใช้สมัครสมาชิก

1.5 เครื่องมือที่ใช้ในการพัฒนา (Development tools)

1.5.1 ฮาร์ดแวร์

1. สมาร์ทโฟน (Smart phone)

- ทำงานบนระบบปฏิบัติการแอนดรอยด์เวอร์ชัน 5.0 หรือ API Level 21
- หน่วยประมวลผลกลาง Mediatek MT6753 Octa-core ความเร็ว 1.3 กิกะเฮิร์ตซ์ (Gigahertz, GHz)
- หน่วยประมวลผลกราฟฟิกลำดับที่น้อย Mali-T720MP3
- หน่วยความจำหลักอย่างน้อย 2 กิกะไบต์ (Gigabyte, GB)
- หน่วยความจำสำรองอย่างน้อย 16 กิกะไบต์ (Gigabyte, GB)
- หน้าจอแสดงผลความละเอียดอย่างน้อย 1080 x 1920 พิกเซล (Pixel)
- หน้าจอแสดงผลขนาดอย่างน้อย 5 นิ้ว
- กล้องถ่ายภาพความละเอียดอย่างน้อย 13 เมกะพิกเซล (Megapixel)

2. เครื่องคอมพิวเตอร์ส่วนบุคคล (Personal computer)

- ทำงานบนระบบปฏิบัติการ Elementary os พื้นฐานการทำงานบน Linux
- หน่วยประมวลผลกลาง Intel Core i3-3217U ความเร็ว 1.80 กิกะเฮิร์ตซ์ (Gigahertz, GHz)

- หน่วยประมวลผลกราฟฟิก NVIDIA GeForce GT 720M ความจำ 2 กิกะไบต์ (Gigabyte, GB)
- หน่วยความจำหลัก 4 กิกะไบต์ (Gigabyte, GB)
- หน่วยความจำสำรอง 120 กิกะไบต์ (Gigabyte, GB)

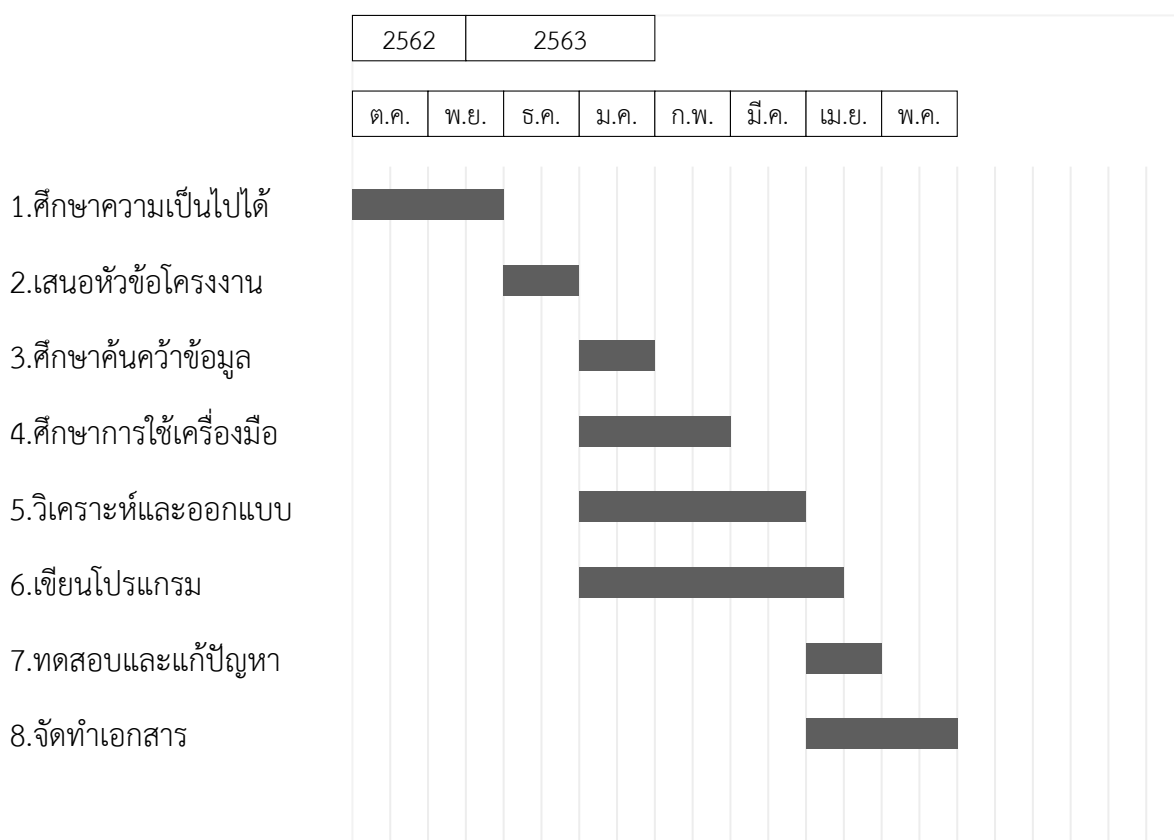
1.5.2 ซอฟต์แวร์ (Software)

1. Visual Studio Code เป็นโปรแกรมสำหรับพัฒนาเว็บแอปพลิเคชัน พัฒนาด้วยภาษา HTML, JavaScript, Python และ Java
2. Django เป็น Framework ที่ใช้สำหรับสร้าง UI สำหรับระบบ หรือเรียกอีกอย่างว่า Backend Framework ใช้สำหรับเป็นเว็บเซิร์ฟเวอร์ (Web Server)
3. OpenCV เป็นไลบรารีฟังก์ชันการเขียนโปรแกรมที่มุ่งเน้นไปทาง การแสดงผลแบบเรียลไทม์ ซึ่งนำมาใช้ในส่วนของการสแกนใบหน้า
4. Machine Learning ในส่วนของการเรียนรู้ของอัลกอริทึม Association Rule
5. Sqlite3 เป็นตัวจัดการฐานข้อมูล

1.5.3 แผนการดำเนินการ

ในการสร้างระบบสแกนใบหน้าสำหรับแคชเชียร์ ผู้พัฒนาได้แบ่งขั้นตอนการดำเนินงานไว้ด้วยกัน 7 ขั้นตอน ดังต่อไปนี้

ตารางที่ 1.1: ขั้นตอนการดำเนินงาน



บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

บทนี้จะเป็นรายละเอียดเกี่ยวกับทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการพัฒนาโปรแกรมในครั้งนี้ โดยที่แต่ละหัวข้อจะมีความสัมพันธ์กันเป็นลำดับ โดยหัวข้อที่หนึ่งจะแนะนำความรู้เรื่อง Python เพื่อให้เข้าใจพื้นฐานเบื้องต้นเกี่ยวกับที่มาของโครงการ หัวข้อที่สองที่สามจะช่วยเตรียมให้อ่านเข้าใจเทคโนโลยีที่ใช้ในการออกแบบและพัฒนา ส่วน ...

2.1 ความรู้พื้นฐานเกี่ยวกับ Python

Python [1] [2] คือ ภาษาเขียนโปรแกรมระดับสูงที่ใช้กันอย่างกว้างขวางในการเขียนโปรแกรมสำหรับวัตถุประสงค์ทั่วไป ภาษา Python นั้นเป็นภาษาแบบ Interpreter ที่ถูกออกแบบให้โค้ดอ่านได้ง่ายขึ้น และโครงสร้างของภาษานั้นจะทำให้โปรแกรมเมอร์สามารถเข้าใจแนวคิดการเขียนโค้ด โดยใช้บรรทัดที่น้อยลงกว่าภาษาอย่าง C++ และ Java ซึ่งภาษานั้นถูกกำหนดให้มีโครงสร้างที่ตั้งใจให้การเขียนโค้ดเข้าใจง่าย ทั้งในโปรแกรมเล็กไปจนถึงโปรแกรมขนาดใหญ่ คุณสมบัติเด่นของภาษา Python

1. สนับสนุนแนวแบบคิดออบเจกต์โอเรียนเตด หรือ OOP (Object Oriented Programming)
2. เป็น Open Source
3. โค้ดที่เขียนด้วย Python สามารถนำไปรันบนระบบปฏิบัติการได้หลากหลาย
4. สามารถประมวลผลทางด้านวิทยาศาสตร์ และวิศวกรรมศาสตร์ได้อย่างมีประสิทธิภาพ
5. มีฟังก์ชันสนับสนุนฐานข้อมูล เช่น MySQL, Sybase, Oracle , Informix, ODBC และอื่นๆ
6. มี Library สนับสนุนด้านการสร้างภาพกราฟฟิก ตลอดจนบันทึกไฟล์ในรูปแบบต่างๆ ได้อย่างสะดวกและมีประสิทธิภาพ
7. มี Library สนับสนุนด้านปัญญาประดิษฐ์

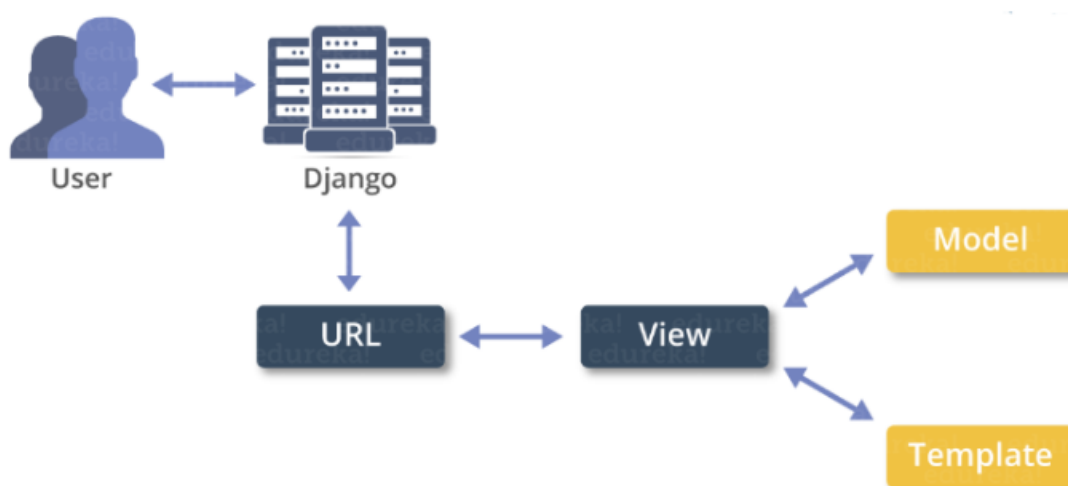
2.2 ความรู้พื้นฐาน Django Framework

Django Framework [3] คือ ชุดเครื่องมือ Framework สำหรับ การพัฒนาเว็บไซต์ด้วยภาษา Python ซึ่งความเป็นจริงแล้วทุกวันนี้มี Framework สำหรับการเขียนเว็บไซต์ด้วยภาษา Python ค่อนข้างเยอะ ซึ่ง Django Framework เป็นหนึ่งใน Framework สำหรับการพัฒนา

เว็บไซต์ และทำเว็บไซต์ด้วยภาษา Python ด้วยเช่นกัน โดยปัจจุบันภาษา Python นั้นค่อนข้างได้
 ได้รับความนิยมเพิ่มมากขึ้น Django Framework [?]] มีคุณสมบัติดังนี้

1. Object-relational mapper คือ การกำหนด Data Model ในภาษา Python เพื่อการทำงานด้านข้อมูล และสนับสนุน Dynamic database-access API
2. Automatic admin interface คือ ส่วนของการสร้าง Interface อัตโนมัติสำหรับการ add, edit , delete และ search ด้วย Django Framework
3. Elegant URL design คือ การทำให้ URL มีความสวยงาม สั้น กระชับ และสื่อความหมายของหน้านั้น ๆ ได้อย่างชัดเจน เหมาะสมกับการทำ SEO ในปัจจุบัน
4. Template system คือ Django นั้นมีการออกแบบ Template Language เพื่อการเขียนแยกส่วนระหว่าง Design และ Business Logic
5. Cache system คือ ส่วนของการบันทึก หรือจัดการข้อมูลที่มีการดาวน์โหลดไปแล้ว เพื่อเพิ่มประสิทธิภาพการทำงานของเว็บไซต์ด้านความเร็ว และด้านอื่น ๆ
6. Internationalization คือ Django สนับสนุน Application ที่มีความหลากหลายด้านภาษาในการแสดงผล

รูปแบบ MVC ของ Django เรียกอีกอย่างว่า MTV คือ Model Template View ดังรูปที่ 2.1



รูปที่ 2.1: รูปแบบ MVC ของ Django

ที่มา: <https://www.edureka.co/blog/django-tutorial/>

2.3 ความรู้พื้นฐานเกี่ยวกับ Bootstrap

Bootstrap คือ Front-end Framework ที่ช่วยให้สามารถสร้างเว็บแอปพลิเคชันได้อย่างสวยงาม ตัว Bootstrap มีทั้ง CSS Component และ JavaScript Plugin ให้เรียกใช้งานได้ และยังถูกออกแบบมาให้รองรับการทำงานแบบ Responsive Web ซึ่งทำให้ การพัฒนาเว็บแค่ครั้งเดียวสามารถนำไปใช้งานบนเบราว์เซอร์ใน มือถือ แท็บเล็ต และคอมพิวเตอร์ ได้ และผู้พัฒนาไม่ต้องมีความรู้ด้าน CSS มาก สามารถสร้างเว็บที่สวยงามได้เช่นกัน

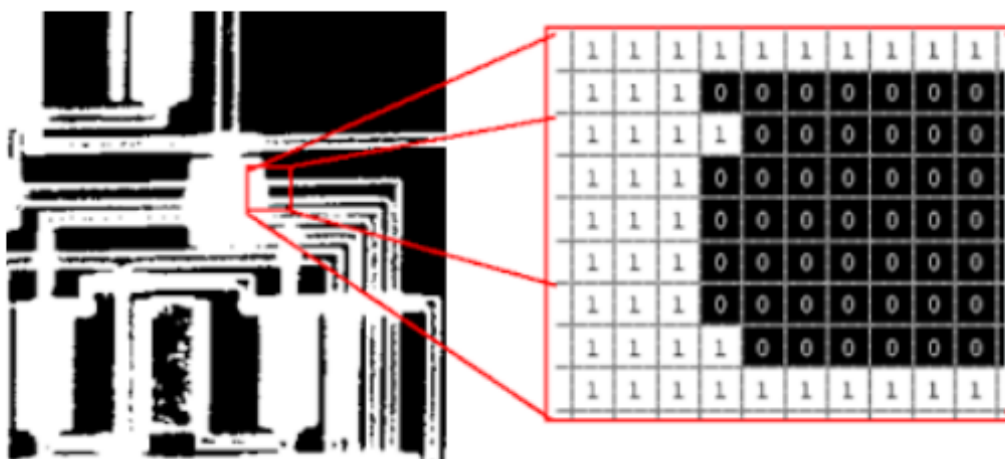
2.4 ความรู้พื้นฐานเกี่ยวกับ OpenCV(OpenSourceComputerVision)

OpenCV [4] เป็น Library ที่รวบรวมฟังก์ชันสำหรับการประมวลผลภาพและคอมพิวเตอร์วิทัศน์ศาสตร์ ไว้เป็นจำนวนมากอยู่ภายใต้ใบอนุญาต BSD ซึ่งสามารถใช้ได้ฟรีทั้งทางด้าน การศึกษา และทางการค้า เนื่องจากนี้ยังมีการประยุกต์ใช้งาน OpenCV ในแบบต่างๆ ได้แก่ ระบบตรวจจับใบหน้า (Face Detection) การจดจำใบหน้า (Face Recognition) การติดตามวัตถุ (Object Tracking) การเรียนรู้ของเครื่อง (Machine Learning) เป็นต้น

2.4.1 การประมวลผลภาพ (Image Processing)

Image Processing [5] การประมวลผลภาพคือการนำรูปที่มีอยู่แล้ว หรือรูปที่รับเข้ามาจากอุปกรณ์ต่างๆ หรือรูป ที่มีอยู่มาประมวลผลเพื่อหาลักษณะเด่นบางประการของรูปที่มีอยู่ หรืออาจจะเป็นการตีความ หมายของภาพรวมถึงการปรับคุณลักษณะของภาพให้เป็นไปตามต้องการโดยใช้กระบวนการทาง คณิตศาสตร์ การประมวลผลภาพแนวคิดทางคณิตศาสตร์ที่ใช้ในการประมวลผล Signalprocessing มาทำการประยุกต์ใช้กับสัญญาณภาพ และภาพจะเก็บอยู่ในรูปของอาร์เรย์ (array) โดยกลุ่มของ ของอาร์เรย์กลุ่มหนึ่งจะเป็นค่าของภาพหนึ่งพิกเซล เช่น ภาพแบบ RGB ใช้อาร์เรย์ 3 ช่องเพื่อเก็บ ค่าสีของ RGB ในหนึ่งพิกเซล รูปแบบการจัดเก็บภาพแต่ละชนิดจะแตกต่างกัน ขึ้นอยู่กับระบบสี ของภาพโดยแบ่งชนิดของภาพได้ดังนี้

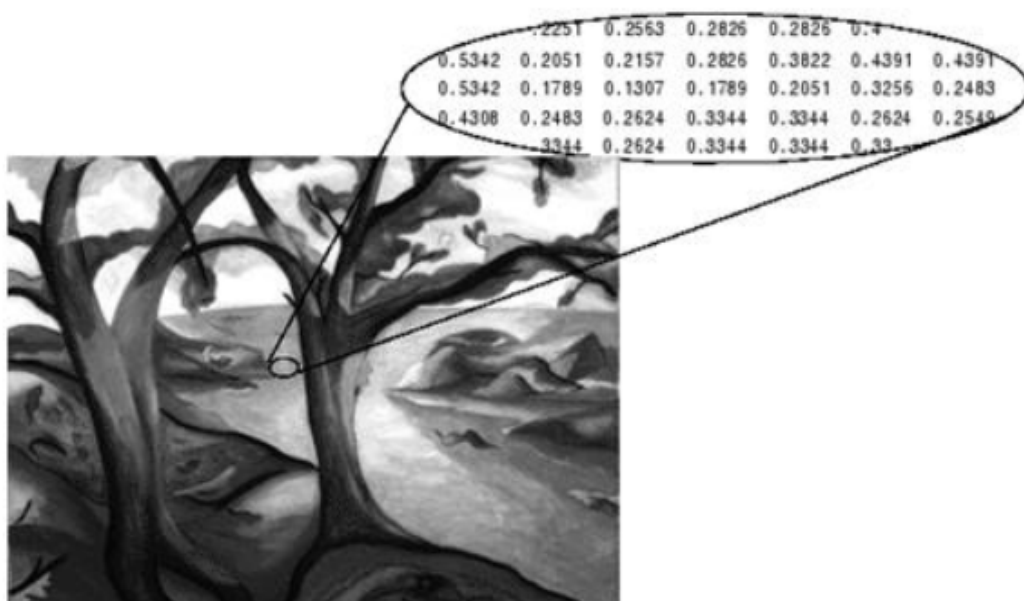
- Binary imageหรือ ภาพขาว-ดำ เป็นรูปที่ใช้เนื้อที่เพียง 1 บิต ต่อ พิกเซล โดยค่าสีจะมีแค่สองค่าคือ 0 หรือสีดำ และ 1 หรือสีขาว ดังรูปที่ 2.2



รูปที่ 2.2: ภาพแบบ Binary

ที่มา: <https://nextsoftwares.wordpress.com/2014/05/22/>

- Grayscale Image เป็นรูปที่เก็บโดยใช้รูปแบบของอาร์เรย์ 2 มิติ โดยค่าที่เก็บจะมีค่าอยู่ในช่วงๆหนึ่ง ซึ่งระดับของสีขึ้นอยู่กับขนาดของบิตที่ใช้เก็บค่าสี ดังรูปที่ 2.3



รูปที่ 2.3: ภาพแบบ Grayscale Image

ที่มา: <https://nextsoftwares.wordpress.com/2014/05/22/>

- RGB Image หรือ Tricolor Image เป็นรูปที่เก็บโดยใช้อาร์เรย์ 3 มิติ ขนาด $m \times n \times 3$ โดยที่ m คือความยาว และ n คือความกว้างของภาพในหน่วยพิกเซล ส่วนมิติสุดท้ายนั้น ในแต่ละมิติจะเก็บค่าสีแยกกัน คือสีแดง (Red) สีเขียว (Green) และสีน้ำเงิน (Blue) ดังรูปที่ 2.4



รูปที่ 2.4: ภาพแบบ RGB Image

ที่มา: <https://nextsoftwares.wordpress.com/2014/05/22/>

2.5 ความรู้เกี่ยวกับ SQLite

เอสคิวแอลไลต์ (SQLite) [6] เป็นห้องสมุดที่นำทิ้งที่ฝังตัวอยู่ในแอปพลิเคชันที่ใช้งานในฐานะที่เป็นฐานข้อมูลแบบไฟล์ในตัว SQLite มีชุดเครื่องมือที่ยอดเยี่ยมในการจัดการข้อมูลทุกประเภทโดยมีข้อ จำกัด และความสะดวกลดลงกว่ามากเมื่อเปรียบเทียบกับฐานข้อมูลเชิงสัมพันธ์แบบโฮสต์กระบวนการ (เซิร์ฟเวอร์)

เมื่อแอปพลิเคชันใช้ SQLite ภาพรวมจะทำงานกับการเรียกใช้งานและการโทรโดยตรงกับไฟล์ที่เก็บข้อมูล (เช่นฐานข้อมูล SQLite) แทนการสื่อสารผ่านอินเทอร์เฟซประเภท (พอร์ตพอร์ตที่ออกเกิด) สิ่งนี้ทำให้ SQLite รวดเร็วและมีประสิทธิภาพเป็นอย่างยิ่งและยังมีเทคโนโลยีพื้นฐาน

ประเภทข้อมูลที่สนับสนุนของ SQLite

- โหมด:

ค่า NULL

- จำนวนเต็ม:

เลขจำนวนเต็มที่ลงนามซึ่งเก็บใน 1, 2, 3, 4, 6 หรือ 8 ไบต์ขึ้นอยู่กับขนาดของค่า

- จริง:

ค่าจุดลอยตัวเก็บไว้เป็นหมายเลขทศนิยม IEEE 8 ไบต์

- ข้อความ:

สตริงข้อความที่จัดเก็บโดยใช้การเข้ารหัสฐานข้อมูล (UTF-8, UTF-16BE หรือ UTF-16LE)

- หยด:

หยดของข้อมูลที่จัดเก็บเหมือนกับข้อมูลที่ป้อนเข้า

หมายเหตุ: หากต้องการเรียนรู้เพิ่มเติมเกี่ยวกับประเภทข้อมูลของ SQLite และความสัมพันธ์ประเภท SQLite ให้ตรวจสอบเอกสารอย่างเป็นทางการเกี่ยวกับเรื่องนี้

ข้อดีของ SQLite

- ไฟล์ที่ใช้:

ฐานข้อมูลทั้งหมดประกอบด้วยไฟล์เดียวบนดิสก์ซึ่งทำให้พกพาได้อย่างมาก

- มาตรฐานการตระหนักถึง:

แม้ว่ามันอาจดูเหมือนการใช้งานฐานข้อมูล“แบบง่าย” แต่ SQLite ก็ใช้ SQL มันมีคุณสมบัติบางอย่างที่ถูกละเว้น (เข้าร่วมขวาหรือเพื่อแต่ละแบบ) แต่บางส่วนเพิ่มเติมจะถูกบดบัง

- เหมาะสำหรับการพัฒนาและการทดสอบ:

ในระหว่างขั้นตอนการพัฒนาแอปพลิเคชันส่วนใหญ่สำหรับคนส่วนใหญ่มีแนวโน้มที่จะต้องการโซลูชันที่สามารถปรับขนาดสำหรับการทำงานพร้อมกัน SQLite ซึ่งมีฐานคุณลักษณะที่หลากหลายสามารถให้มากกว่าสิ่งที่จำเป็นสำหรับการพัฒนาด้วยความเรียบง่ายในการทำงานกับไฟล์เดียวและไลบรารีบนฐาน C ที่เชื่อมโยง

ข้อเสียของ SQLite

- ไม่มีการจัดการผู้ใช้:

ฐานข้อมูลขั้นสูงมาพร้อมกับการสนับสนุนสำหรับผู้ใช้เช่นการเชื่อมต่อที่มีการจัดการที่มีสิทธิ์ตั้งค่าการเข้าถึงฐานข้อมูลและตาราง เมื่อพิจารณาถึงวัตถุประสงค์และลักษณะของ SQLite (ไม่มีระดับที่สูงขึ้นของการทำงานพร้อมกันของไคลเอนต์หลายรายการ) คุณลักษณะนี้ไม่มีอยู่

- ขาดความเป็นไปได้ที่จะจัดด้วยสำหรับประสิทธิภาพเพิ่มเติม

อีกครั้งโดยการออกแบบ SQLite เป็นไปไม่ได้ที่จะได้รับประสิทธิภาพที่เพิ่มขึ้นอย่างมาก ไลบรารีนั้นง่ายต่อการปรับแต่งและใช้งานง่าย เนื่องจากมันไม่ซับซ้อนจึงเป็นไปได้ในทางเทคนิคที่จะทำให้มันมีประสิทธิภาพมากกว่ามันแล้วน่าประหลาดใจคือ

เมื่อต้องการใช้ SQLite

- แอปพลิเคชันแบบฝัง:

แอปพลิเคชันทั้งหมดที่ต้องการความสะดวกในการพกพาซึ่งไม่จำเป็นต้องมีการขยายเช่น แอปพลิเคชันในท้องถิ่นแอปพลิเคชันมือถือหรือเกม

- การแทนที่การเข้าถึงดิสก์:

ในหลายกรณีแอปพลิเคชันที่จำเป็นต้องอ่าน / เขียนไฟล์ไปยังดิสก์โดยตรงจะได้ประโยชน์จากการสลับไปใช้ SQLite สำหรับการทำงานเพิ่มเติมและความเรียบง่ายที่มาจากการใช้ภาษาที่มีโครงสร้างของแบบสอบถาม (SQL)

- การทดสอบ:

มันเกินความเป็นจริงสำหรับแอปพลิเคชันขนาดใหญ่เพื่อใช้กระบวนการเพิ่มเติมสำหรับการทดสอบตรรกะทางธุรกิจ (เช่นวัตถุประสงค์หลักของแอปพลิเคชัน: ฟังก์ชันการทำงาน)

เมื่อไม่ต้องการใช้ SQLite

- แอปพลิเคชันผู้ใช้หลายคน:

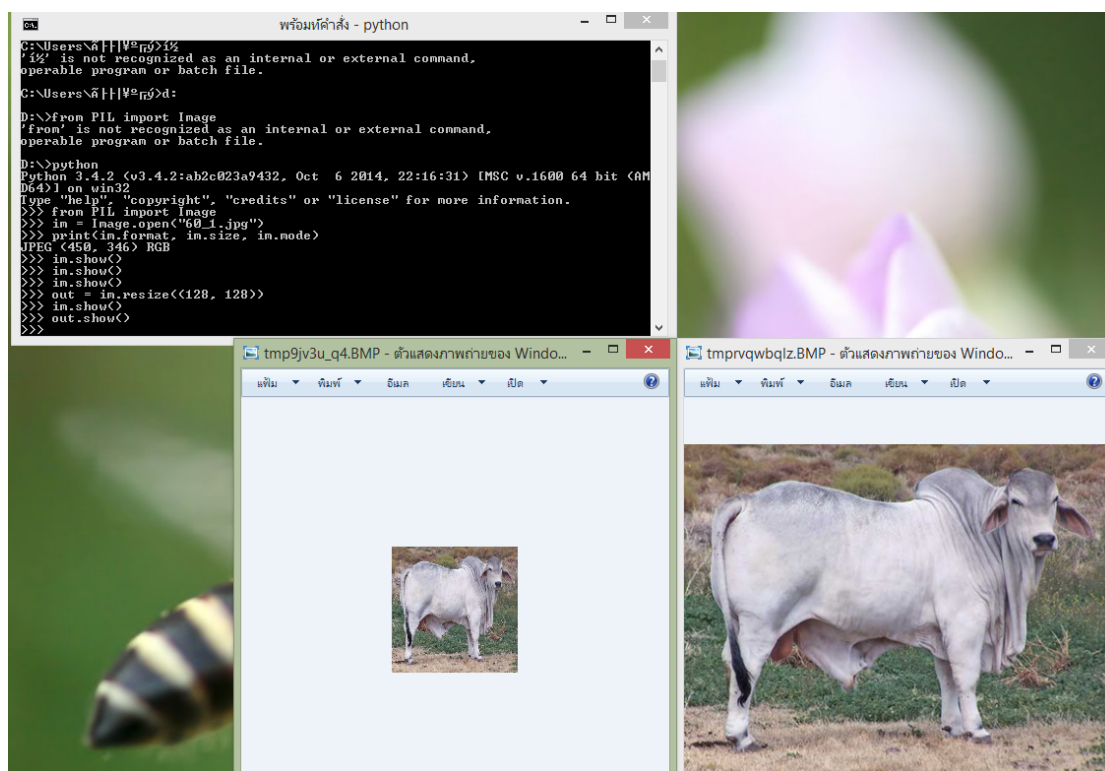
หากคุณกำลังทำงานกับแอปพลิเคชันที่ลูกค้าหลายรายต้องการเข้าถึงและใช้ฐานข้อมูลเดียวกัน RDBM ที่มีคุณสมบัติครบถ้วน (เช่น MySQL) น่าจะเลือก SQLite ได้ดีกว่า

- แอปพลิเคชันที่ต้องการปริมาณการเขียนสูง:

หนึ่งในข้อ จำกัด ของ SQLite คือการดำเนินการเขียน DBMS นี้อนุญาตให้ใช้งานการเขียน * เพียงครั้งเดียวในเวลาที่กำหนดดังนั้นจึงอนุญาตให้มีปริมาณงานที่ จำกัด

2.6 ความรู้พื้นฐานเกี่ยวกับ Pillow

Pillow เป็น Library ด้านความสามารถในการประมวลผลภาพและกราฟฟิก หรือโมดูลจัดการและประมวลผลรูปภาพบน Python ใน Python มีโมดูลด้านที่ชื่อว่า Python Imaging Library (PIL) ซึ่งรองรับแต่ Python 2 จึงมีคนมาพัฒนาเป็นโมดูล Pillow ที่รองรับทั้ง Python 2 และ Python 3 โมดูล Pillow สามารถจัดการจัดเก็บรูปภาพ (Image Archives) แสดงรูปภาพ (Image Display) ประมวลผลรูปภาพ เช่น ปรับขนาดรูปภาพ แปลงไฟล์รูปภาพ ใส่ตัวอักษรลงในภาพ และอื่น ๆ เป็นต้น ดังรูปที่ 2.5



รูปที่ 2.5: การขยายรูปโดยใช้ Pillow

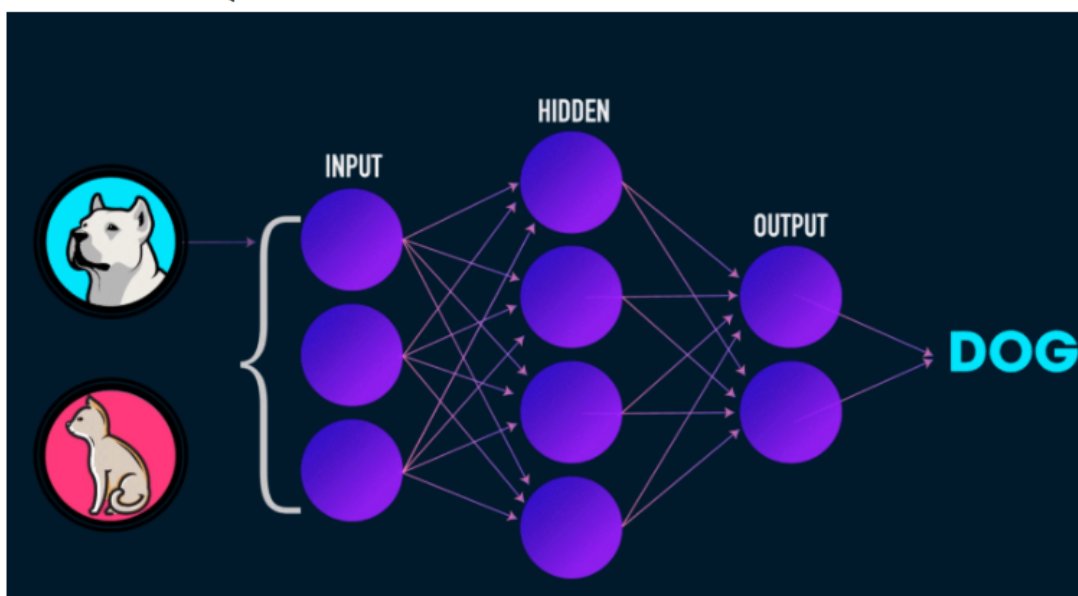
ที่มา: <https://python3.wannaphong.com/2014/11/image-processing-python.html>

2.7 ความรู้พื้นฐานเกี่ยวกับปัญญาประดิษฐ์ (Artificial Intelligence)

ปัญญาประดิษฐ์ (AI) [7] เป็นศาสตร์แขนงหนึ่งของวิทยาศาสตร์ คอมพิวเตอร์ ที่เกี่ยวข้องกับวิธีการทำให้คอมพิวเตอร์มีความสามารถคล้ายมนุษย์ หรือเลียนแบบพฤติกรรมมนุษย์ โดยเฉพาะความสามารถในการคิดเองได้ หรือมีปัญญานั้นเอง ปัญหานี้มนุษย์เป็นผู้สร้างให้คอมพิวเตอร์ จึงเรียกว่าปัญญาประดิษฐ์

2.7.1 โครงข่ายประสาทเทียม (Neural Network)

โครงข่ายประสาทเทียม คือ โครงข่ายประสาทเทียมที่เป็นการจำลองมาจากสมองของมนุษย์ เพื่อจำลองการทำงานของเครือข่ายประสาทในสมองมนุษย์ ด้วยวัตถุประสงค์ที่จะสร้างเครื่องมือซึ่งมีความสามารถในการเรียนรู้การจดจำรูปแบบ (Pattern Recognition) และการสร้างความรู้ใหม่ (Knowledge Extraction) เช่นเดียวกับความสามารถที่มีในสมองมนุษย์นั่นเอง หลักการของ Neural Network ในคอมพิวเตอร์ neurons ประกอบด้วย input และ output เหมือนกัน โดยจำลองให้ input แต่ละอันมี weight เป็นตัวกำหนดน้ำหนักของ input โดย neuron แต่ละหน่วยจะมีค่า threshold เป็นตัวกำหนดว่าน้ำหนักรวมของ input ต้องมากขนาดไหนจึงจะสามารถส่ง output ไปยัง neurons ตัวอื่นได้ เมื่อนำ neuron แต่ละหน่วยมาต่อกันให้ทำงานร่วมกัน การทำงานนี้ในทางตรรกแล้ว จะเหมือนกับปฏิกิริยาเคมีที่เกิดในสมอง เพียงแต่ในคอมพิวเตอร์ทุกอย่างเป็นตัวเลขเท่านั้นเอง ดังรูปที่ 2.6



รูปที่ 2.6: ขั้นตอนของโครงข่ายประสาทเทียมแบบคอนโวลูชัน

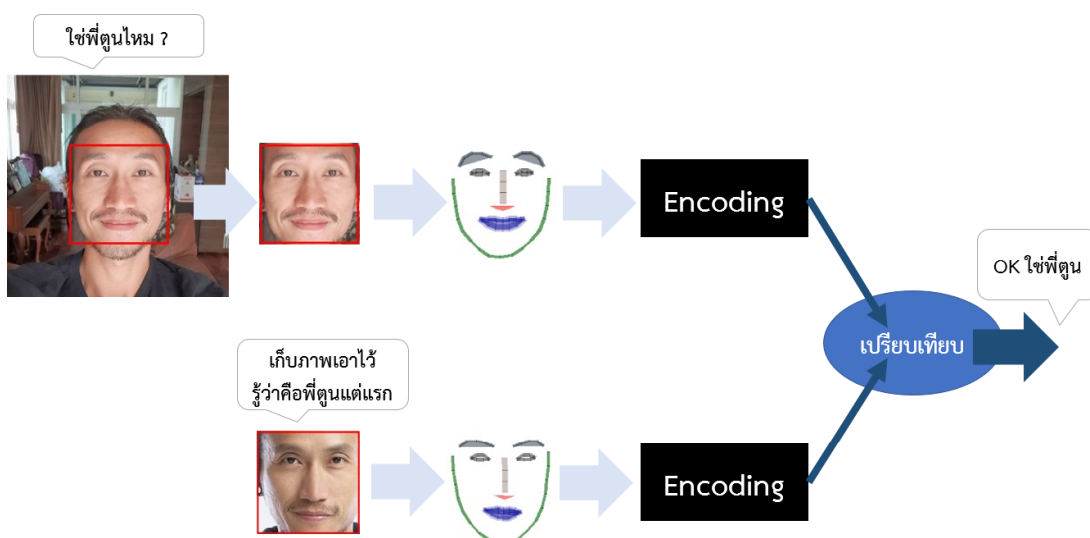
ที่มา: <https://analyticsindiamag.com/most-common-activation-functions-in-neural-networks-and-rationale-behind-it/>

2.7.2 ระบบรู้จำใบหน้า(Face Recognition Algorithm)

Face Recognition เป็นระบบที่ช่วยเพิ่มประสิทธิภาพการระบุ และยืนยันอัตลักษณ์บุคคล ด้วยความรวดเร็ว แม่นยำ และสามารถเพิ่มโอกาสในการต่อยอดความสำเร็จให้หลากหลายแวดวงธุรกิจได้ด้วย เทคโนโลยีตรวจสอบและจดจำใบหน้า คือ ความก้าวหน้าของการพัฒนาศักยภาพการทำงานเอไอด้วยระบบ แมชชีน เลิร์นนิง (Machine Learning) สร้างเป็นระบบประมวลผลที่สามารถตรวจจับใบหน้า (detect) และระบุตัวตน (identify) ได้ทั้งลักษณะภาพนิ่ง และภาพเคลื่อนไหวที่ถูกบันทึกไว้ ตลอดจนการเคลื่อนไหวแบบเรียลไทม์ (real time) หลักการทำงานของ Face Recognition คือ การสร้างโมเดลการอ้างอิง ที่เรียกว่า “faceprint” ขึ้นมา โดยระบบจะวิเคราะห์จากลักษณะเฉพาะต่างๆ บนใบหน้า เช่น โครงหน้า ความกว้างของจมูก ระยะห่างระหว่างตาทั้งสองข้าง ขนาดของโหนกแก้ม ความลึกของเบ้าตา รวมถึงพื้นผิวบนใบหน้า (facial texture) เป็นต้น จากนั้น ระบบจะทำการสร้างจุดเชื่อมโยงบนใบหน้า (nodal points) เพื่อเปรียบเทียบกับรูปภาพที่ถูกเก็บไว้ในฐานข้อมูล (data base) ทั้งในลักษณะภาพนิ่งและภาพเคลื่อนไหว เพื่อความแม่นยำในการระบุตัวตนของผู้ที่ต้องเข้าสู่กระบวนการตรวจสอบ โดยทั่วไประบบรู้จำใบหน้าจะประกอบไปด้วย 2 ขั้นตอนหลักคือ

การตรวจจับใบหน้า (Face Detection) กระบวนการค้นหาใบหน้าของบุคคลจากภาพหรือวิดีโอ หลังจากนั้นก็ทำการประมวลผลภาพใบหน้าที่ได้สำหรับนำไปใช้ในขั้นตอนถัดไป

การรู้จำใบหน้า (Face Recognition) กระบวนการที่นำภาพไปตรวจจับประมวลผลแล้ว จากนั้นขั้นตอนการตรวจจับใบหน้า แล้วนำมาเปรียบเทียบกับฐานข้อมูลของใบหน้า เพื่อระบุว่าใบหน้านั้นตรงกับบุคคลใด ซึ่งการระบุใบหน้าที่มีหลากหลายวิธีเช่น Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Elastic Bunch Graph Matching (EBGM) ดังรูปที่ 2.7



รูปที่ 2.7: ภาพแสดงกระบวนการทำงานของระบบรู้จำใบหน้า

ที่มา: <http://asd.co.th/เทคโนโลยีระบบตรวจจับใบหน้า/>

2.7.3 การเรียนรู้แบบต้นไม้ตัดสินใจ (decision tree learning)

decision tree learning เป็นหนึ่งในวิธีการเรียนรู้ซึ่งใช้ในสถิติ, การเรียนรู้ของเครื่อง และการทำเหมืองข้อมูล โดยพิจารณาการสังเกตการแบ่งแยกข้อมูลโดยพิจารณาข้อมูล ในการเรียนรู้ของเครื่อง (machine learning) ต้นไม้ตัดสินใจ เป็นโมเดลทางคณิตศาสตร์ที่ใช้ทำนายประเภทของวัตถุโดยพิจารณาจากลักษณะของวัตถุ บัพภายใน (inner node) ของต้นไม้จะแสดงตัวแปรส่วนกิ่งจะแสดงค่าที่เป็นไปได้ของตัวแปร ส่วนบัพใบ (leaf node) จะแสดงประเภทของวัตถุ ต้นไม้ตัดสินใจที่บัพใบแสดงถึงข้อมูลที่เป็นข้อมูลไม่ต่อเนื่อง (discrete values) จะเรียกว่าต้นไม้ตัดสินใจแบบจำแนก (classification trees) และต้นไม้ตัดสินใจที่บัพใบเป็นข้อมูลต่อเนื่อง (continuous values) จะเรียกว่าต้นไม้ตัดสินใจแบบถดถอย (regression trees) ต้นไม้การตัดสินใจในการบริหารธุรกิจ เป็นแผนผังต้นไม้ช่วยในการตัดสินใจ โดยแสดงถึงมูลค่าของทรัพยากรที่จะใช้ ความเสี่ยงในการลงทุนและผลลัพธ์ที่มีโอกาสเกิดขึ้น ต้นไม้ตัดสินใจสร้างขึ้นเพื่อช่วยการตัดสินใจเพื่อใช้ในการสร้างแผนงาน นิยมใช้มากในการบริหารความเสี่ยง (risk management) ต้นไม้ตัดสินใจเป็นส่วนหนึ่งของทฤษฎีการตัดสินใจ (decision theory) และ ทฤษฎีกราฟ ต้นไม้ตัดสินใจเป็นวิธีการพื้นฐานอย่างหนึ่งสำหรับการทำเหมืองข้อมูล

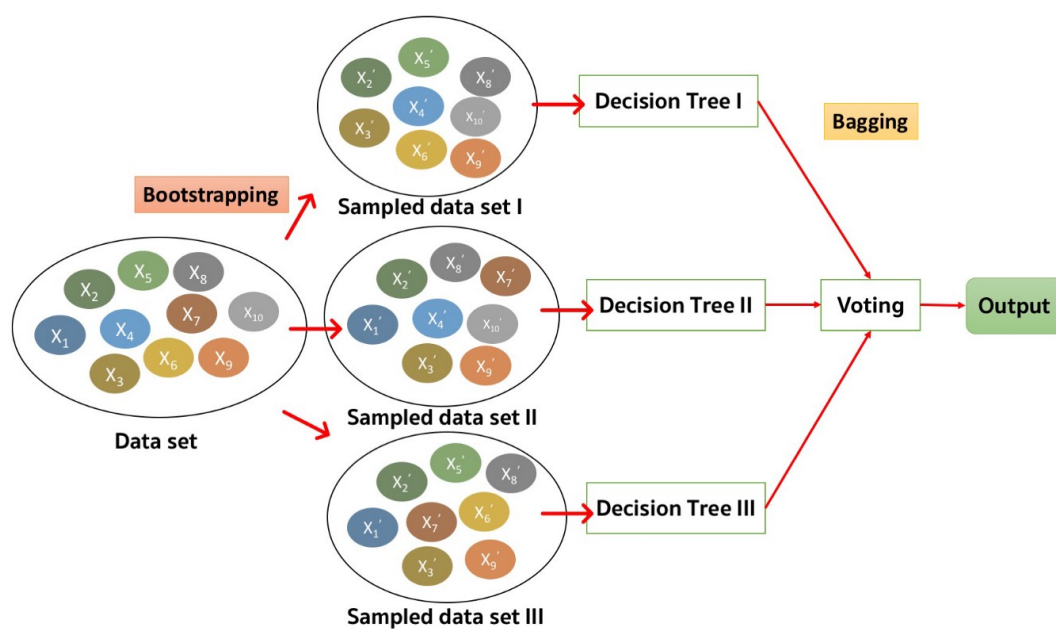
2.7.4 Random Forest

หลักการของ Random Forest คือ สร้าง model จาก Decision Tree หลายๆ model ย่อยๆ (ตั้งแต่ 10 model ถึง มากกว่า 1000 model) โดยแต่ละ model จะได้รับ data set ไม่เหมือนกัน ซึ่งเป็น subset ของ data set ทั้งหมด ตอนทำ prediction ก็ให้แต่ละ Decision Tree ทำ prediction ของใครของมัน และคำนวณผล prediction ด้วยการ vote output ที่ ถูกเลือก โดย Decision Tree มากที่สุด (กรณี classification) หรือ หาค่า mean จาก output ของแต่ละ Decision Tree (กรณี regression) Decision Tree แต่ละ model ใน Random Forest ถือว่าเป็น weak learner — ประมาณว่าเป็น model ที่ไม่เก่งเท่าไร แต่พอนำเอาแต่ละ Decision Tree มาทำ prediction ร่วมกัน ก็จะได้ model รวมที่มีความเก่ง และแม่นยำมากกว่า Decision Tree ที่ทำ prediction แบบเดียวๆ

sample ข้อมูล (bootstrapping) จาก data set ทั้งหมด ให้ได้ข้อมูลออกมา n ชุด ที่ไม่เหมือนกัน ตามจำนวน Decision Tree ใน Random Forest เช่น data set ตั้งต้นมีอยู่ 10 feature (X_1, X_2, \dots, X_{10}) แต่ละ Decision Tree จะได้ feature ไปไม่เหมือนกัน และ จะได้ข้อมูลไม่ครบทุก row ด้วยจาก data set ทั้งหมดด้วย ($X_1 \rightarrow X_1', X_2 \rightarrow X_2', \dots$)

สร้าง model Decision Tree สำหรับแต่ละชุดข้อมูล

ทำ aggregation ผลลัพธ์ จากแต่ละ model (bagging) เช่น voting ในกรณี classification หรือ หาค่า mean ในกรณี regression ดังรูปที่ 2.8



รูปที่ 2.8: หลักการทำ Random Forest

ที่มา: <https://medium.com/@witchapongdaroontham/random-forest-part-2-of-decision-tree-random-forest>

2.8 งานวิจัยที่เกี่ยวข้อง

นางสาวสุภาภรณ์ จินดาวงษ์ ได้นำเสนอวิจัยเกี่ยวกับ การศึกษาพฤติกรรมการเลือกใช้บริการร้านกาแฟสดของผู้บริโภค ผู้ศึกษาจึงสนใจเกี่ยวกับ พฤติกรรมการเลือกใช้บริการร้านกาแฟสดของผู้บริโภคในอำเภอเมือง จังหวัด นครปฐม กรณีศึกษา ร้านบ้านไร่กาแฟ สาขาที่29 เพื่อทราบถึงเหตุผลการเข้าใช้บริการร้านกาแฟสดบ้านไร่กาแฟ สาขาที่29 ผลการวิจัยทำให้ สามารถใช้เป็นแนวทางในการปรับปรุงธุรกิจให้ตรงกับความต้องการผู้บริโภคและเป็นแนวทาง ดำเนินกิจการแก่ผู้สนใจธุรกิจร้านกาแฟ

นางสาวกานดา เสือจำศีล ได้นำเสนอวิจัยเกี่ยวกับ พฤติกรรมการเข้าใช้บริการร้านกาแฟสด อเมซอน ของผู้บริโภค เพื่อศึกษาพฤติกรรมการเข้าใช้บริการร้านกาแฟสด อเมซอน ปัจจัยส่วนบุคคลของผู้บริโภคที่เข้าใช้บริการร้านกาแฟสด อเมซอน และปัจจัยส่วนผสมทางการตลาดต่อการเข้าใช้บริการร้านกาแฟสด อเมซอน กลุ่มตัวอย่างที่ใช้ คือ ผู้บริโภคในจังหวัดปทุมธานีที่เข้าใช้บริการร้านกาแฟสด อเมซอน จำนวน 400 คน

นางสาวอัจจิมา มณฑาพันธุ์ ได้นำเสนอวิจัย การวิเคราะห์เปรียบเทียบการระบุตัวตนโดยใช้ลักษณะเฉพาะทางกายภาพของร่างกาย ผู้ศึกษาได้ศึกษาเกี่ยวกับการระบุตัวตนโดยใช้คุณลักษณะเฉพาะทางกายภาพของร่างกายในแต่ละประเภท ได้แก่ รูปแบบใบหน้า เสียง ลายนิ้วมือ ม่านตา และลายเซ็น มาวิเคราะห์เปรียบเทียบเพื่อหาว่าประเภทของการแยกแยะบุคคลที่เป็นที่นิยมในด้านต่างๆมีอะไรบ้าง เช่น ความถูกต้อง จำนวนผู้ใช้ การยอมรับ ซึ่งผลที่ได้จะนำไปประยุกต์ใช้กับระบบความมั่นคงของเครื่องจักรหรือเครื่องคอมพิวเตอร์ในอนาคต

นางสาวพิชญา จตุรวัฒน์, นางสาวภาสินี พงศ์มานะวุฒิ และ นายมานพ พันธโคกกรวด ได้นำเสนอวิจัยเกี่ยวกับ การพัฒนาระบบบันทึกเวลาเรียนด้วยการตรวจจับ และรู้จำใบหน้า ผู้ศึกษาจึงสนใจเกี่ยวกับการระบุตัวตนของนักเรียน ซึ่งหากนักเรียนมีจำนวนมากจะทำให้การเรียกชื่อเสียเวลา โดยเริ่มจากกระบวนการสร้างฐานข้อมูลรูปภาพ และพัฒนาระบบในลักษณะเว็บแอปพลิเคชันเพื่อนำไปใช้งาน โดยใช้ Haar-Like Feature ในการตรวจจับใบหน้า และใช้ Local Binary Patterns Histogram ในการรู้จำใบหน้า โดยเบื้องต้นมีความแม่นยำของการรู้จำใบหน้า 48 เปอร์เซ็นต์

บทที่ 3

การวิเคราะห์และออกแบบระบบ

การวิเคราะห์และออกแบบระบบก่อนดำเนินการจริงเป็นอีกหนึ่งขั้นตอนที่มีความสำคัญมาก เพราะการวิเคราะห์และออกแบบระบบนั้นเป็นการกระทำที่ทำให้ผู้พัฒนาเห็นรายละเอียดส่วนย่อยของงานทั้งหมด เพิ่มประสิทธิภาพในการวางแผน การทำงาน และยังช่วยลดปัญหาที่อาจเกิดขึ้นในระหว่างพัฒนา เพื่อให้ระบบมีความสมบูรณ์มากยิ่งขึ้น เนื่องจากการวิเคราะห์และออกแบบระบบนั้นจะช่วยให้ให้บริการ จัดการทรัพยากรได้อย่างคุ้มค่าและตรงตามความต้องการของระบบ

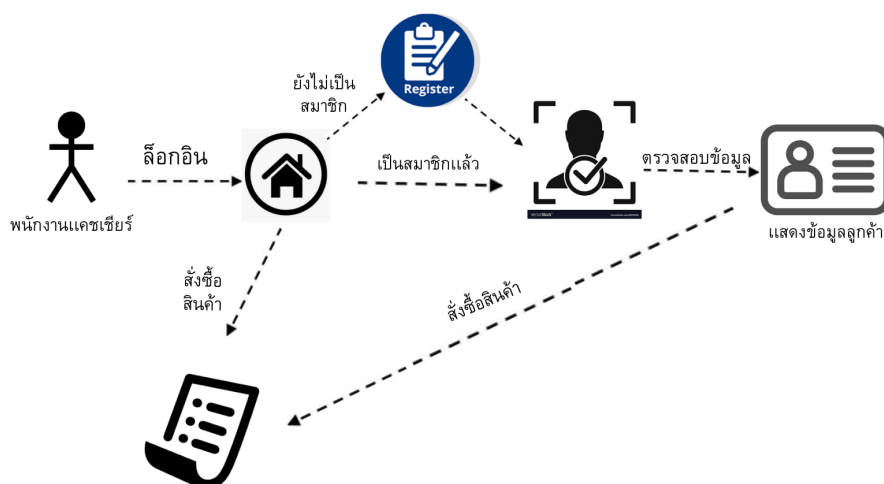
การวิเคราะห์และออกแบบระบบสแกนใบหน้าสำหรับแคชเชียร์ ในบทนี้จะแบ่งออกเป็น 6 ขั้นตอนเพื่อให้เห็นการดำเนินงานอย่างมีระบบ ในหัวข้อแรกจะนำเสนอภาพรวมของระบบ ก่อนจะนำเสนอเอกสารแสดงความต้องการของระบบซึ่งจะทำให้เห็นที่มาของเพจต่าง ๆ ในขั้นตอนของการออกแบบในหัวข้อที่สาม ส่วนหัวข้อที่เหลือจะแสดงแผนภาพการทำงานของระบบโดยใช้ UML diagram ซึ่งประกอบไปด้วย Use Case, Class และ Sequence Diagram เพื่อแสดงรายละเอียดของระบบก่อนนำไปเขียนคำสั่งด้วยภาษาโปรแกรมในบทต่อไป

- 3.1 โครงสร้างภาพรวมของระบบ (System Architecture) เป็นการออกแบบภาพรวมและเทคโนโลยีของระบบ
- 3.2 System Requirements คือ ความต้องการหรือสิ่งที่ระบบควรจะทำ หรือหน้าที่หลักของระบบที่จะต้องทำ
- 3.3 User Interface Design เป็นการออกแบบส่วนต่อประสานกับผู้ใช้
- 3.4 Use Case Diagram เป็นแผนภาพที่ใช้แสดงให้ทราบว่าระบบทำงานหรือมีหน้าที่ใดบ้าง
- 3.5 Class Diagram เป็นแผนภาพที่ใช้แสดง Class และความสัมพันธ์ระหว่าง Class
- 3.6 Sequence Diagram เป็นแผนภาพที่ใช้แสดงให้เห็นถึงการตอบโต้ข้อมูลระหว่างคลาส เรียงตามลำดับของเวลาที่เกิดเหตุการณ์จากน้อยไปมาก

3.1 โครงสร้างภาพรวมของระบบ

ความหมายของ System Architecture หมายถึง กรอบโครงสร้างของระบบที่อธิบายความสัมพันธ์ขององค์ประกอบต่าง ๆ ไปจนถึงขั้นการเชื่อมต่อกันของระบบย่อยต่าง ๆ โดยจัดกลุ่มองค์ประกอบไว้ในหลาย ๆ ลักษณะเพื่อให้ผู้เกี่ยวข้อง (Stakeholder) จากพื้นฐานสาขาอาชีพที่แตกต่าง กันสามารถทำความเข้าใจได้ง่าย เช่น การจัดแบ่งองค์ประกอบตามลักษณะการทำงานของระบบ (functional components) เป็นต้น

การออกแบบ System architecture แสดงภาพรวมและเทคโนโลยีของระบบสแกนใบหน้า สำหรับแคชเชียร์ มีรายละเอียดดังรูปที่ 3.1



รูปที่ 3.1: ภาพรวมและโครงสร้างการทำงานของระบบ

จากรูปที่ 3.1 เมื่อพนักงานแคชเชียร์ทำการล็อกอินแล้วจะเข้าหน้าหลัก ยังไม่เป็นสมาชิก จะทำการสมัครก่อน ถ้าเป็นสมาชิกแล้วจะเข้าสู่ระบบสแกนใบหน้าสำหรับลูกค้า โดยระบบจะทำการตรวจสอบรูปภาพที่สแกนว่าตรงกับรูปภาพของในระบบ แล้วแสดงข้อมูลของลูกค้าให้กับพนักงานแคชเชียร์แล้วยังหน้าส่งสินค้า

3.2 System Requirements

3.2.1 Functional Requirements

ระบบสแกนใบหน้าสำหรับแคชเชียร์ แบ่งความสามารถของระบบตามประเภทของผู้ใช้งาน ดังนี้

1. พนักงานแคชเชียร์

- สามารถทำการเข้าสู่ระบบได้
- สแกนใบหน้าของลูกค้าได้
- สมัครงานให้ลูกค้าได้
- แก้ไขข้อมูลลูกค้าเมื่อลูกค้าต้องการได้
- สามารถสั่งสินค้าให้ลูกค้าได้

2. ลูกค้า

- สแกนใบหน้าเพื่อทำการแสดงข้อมูล
- สั่งสินค้ากับพนักงาน

3.2.2 Non-functional Requirements

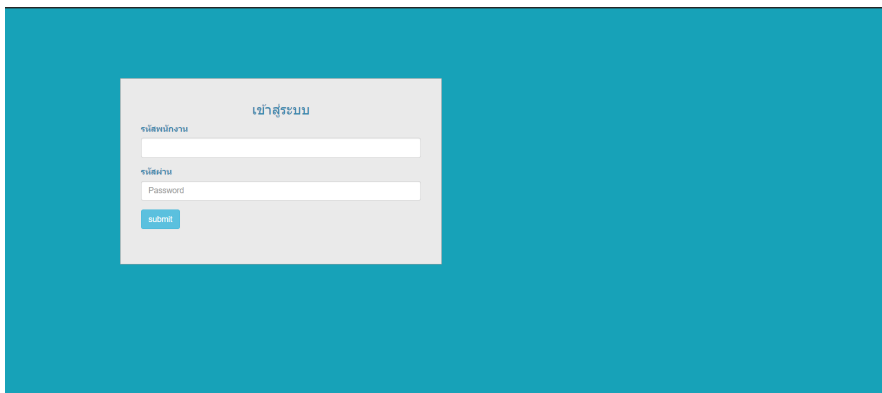
1. เว็บแอปพลิเคชัน

- ใช้โปรโตคอล (Protocol) แบบ HTTPS (Hypertext Transfer Protocol Secure) ในการสื่อสารที่ช่วยรักษาความสมบูรณ์ถูกต้องของข้อมูลผู้ใช้และเก็บข้อมูลไว้เป็นความลับระหว่างคอมพิวเตอร์ของผู้ใช้กับเว็บไซต์
- ใช้ดีลิบ (dlib) ซึ่งมี Machine learning algorithms ในการจดจำหน้าของลูกค้าเพื่อระบุตัวตน
- ใช้ต้นไม้ตัดสินใจ (Decision Trees) ในการทำโมเดล (Model) ในการทำระบบแนะนำเมนู

3.3 User Interface Design

ในการออกแบบ User Interface Design ของระบบสแกนใบหน้าสำหรับแคชเชียร์

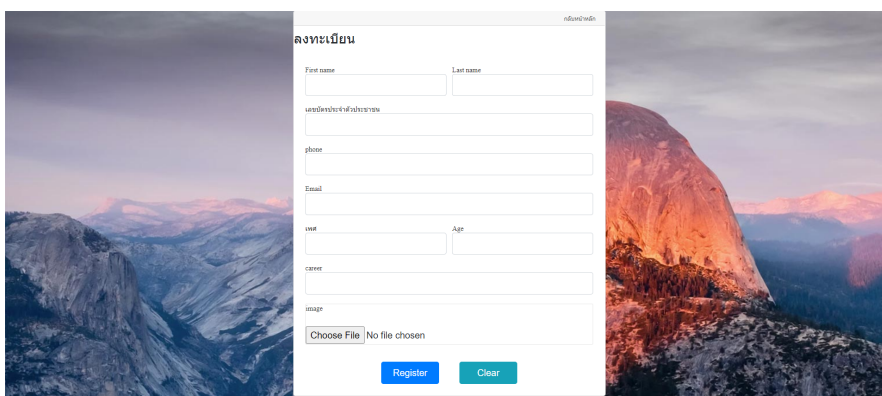
- การออกแบบหน้าจอเข้าสู่ระบบ



รูปที่ 3.2: หน้าเข้าสู่ระบบสำหรับพนักงาน

จากภาพที่ 3.2 แสดงหน้าจอการเข้าสู่ระบบของพนักงานแคชเชียร์ จำเป็นต้องกรอกข้อมูลรหัสพนักงานและรหัสผ่านเพื่อเข้าใช้งานระบบ

- การออกแบบหน้าจอเข้าสู่ระบบ



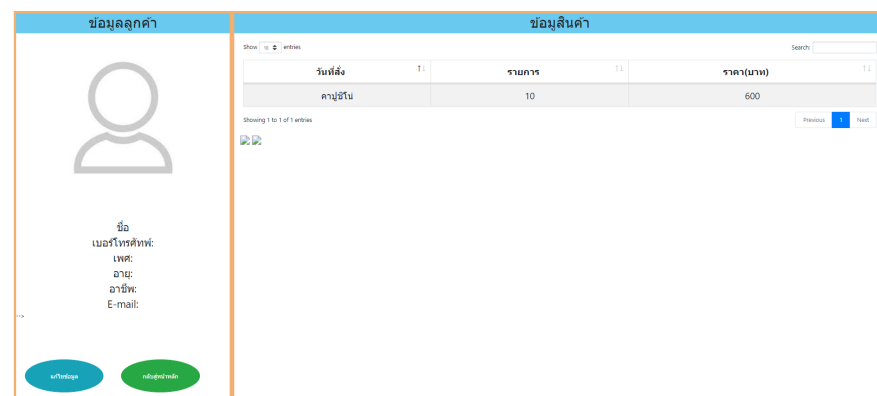
รูปที่ 3.3: หน้าเข้าสู่ระบบสำหรับพนักงาน

จากภาพที่ 3.3 แสดงหน้าจอสมัครสมาชิกสำหรับลูกค้าที่ต้องการเป็นสมาชิกในระบบ

- การออกแบบหน้าจอหลัก

รูปที่ 3.4: หน้าจอหลัก

จากภาพที่ 3.4 แสดงหน้าจอหลัก ให้พนักงานเลือก ไฮยังหน้า สมัครสมาชิก เข้าสู่ระบบ โดยการสแกนใบหน้า ดาเวสู่ระบบโดยล็อกอินผ่านเบอร์โทรศัพท์



รูปที่ 3.5: หน้าจอแสดงข้อมูลลูกค้า

จากภาพที่ 3.5 แสดงหน้าจอ ข้อมูลลูกค้า หลังจากที่ผ่านมาการสแกนใบหน้า จะแสดง ข้อมูลลูกค้าที่สมัครสมาชิก แสดงประวัติการสั่งซื้อสินค้า แต่มสละสม และมีการแนะนำเมนูแก่ลูกค้า ทั้งนี้จะแสดงเฉพาะลูกค้าที่เป็นสมาชิกเท่านั้น

- การออกแบบหน้าจอสั่งซื้อสินค้า



รูปที่ 3.6: หน้าจอสั่งซื้อสินค้า

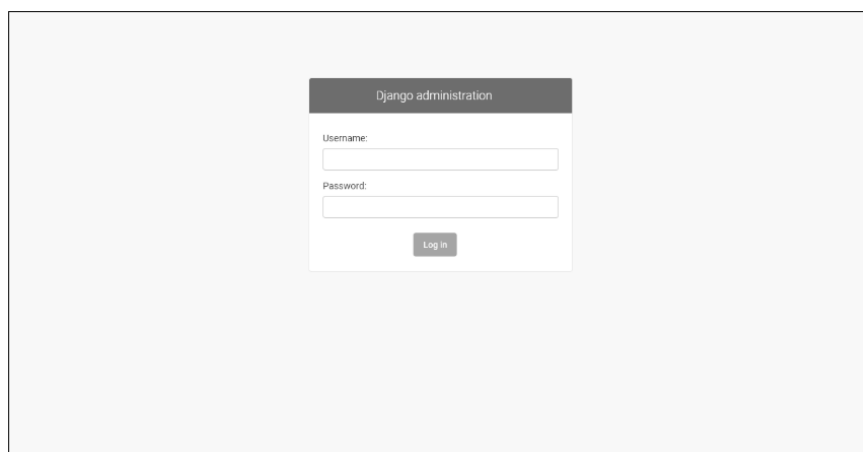
จากภาพที่ ค.8 แสดงหน้าจอรายการเมนูที่มีในร้าน โดยจะมีอยู่3ประเภทหลัก คือ

เมนูร้อน เมนูเย็น เมนูปั่น เมื่อลูกค้าสั่งจะมีการแสดงรายการที่สั่ง ราคา จำนวนที่ลูกค้าสั่ง

3.3.1 ส่วนของผู้ดูแลระบบ

โดยประกอบด้วยส่วนการออกแบบหน้าจอทำงาน ดังนี้

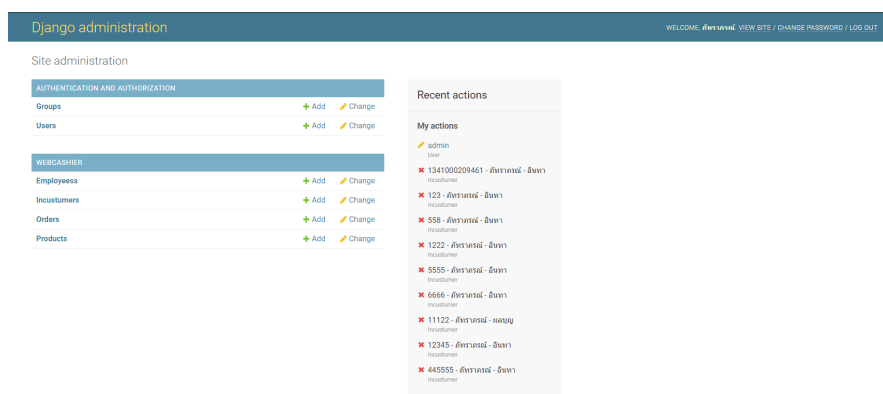
1. การออกแบบหน้าจอเข้าสู่ระบบ



รูปที่ 3.7: หน้าจอเข้าสู่ระบบผู้ดูแลระบบ

จากภาพที่ 3.7 แสดงหน้าจอเข้าสู่ระบบ โดยต้องกรอกชื่อผู้ใช้และรหัสผ่านก่อนกดปุ่ม Login

2. การออกแบบหน้าจอฐานข้อมูลทั้งหมดในระบบ





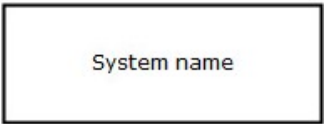
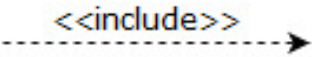

รูปที่ 3.8: หน้าจอฐานข้อมูลทั้งหมดในระบบ

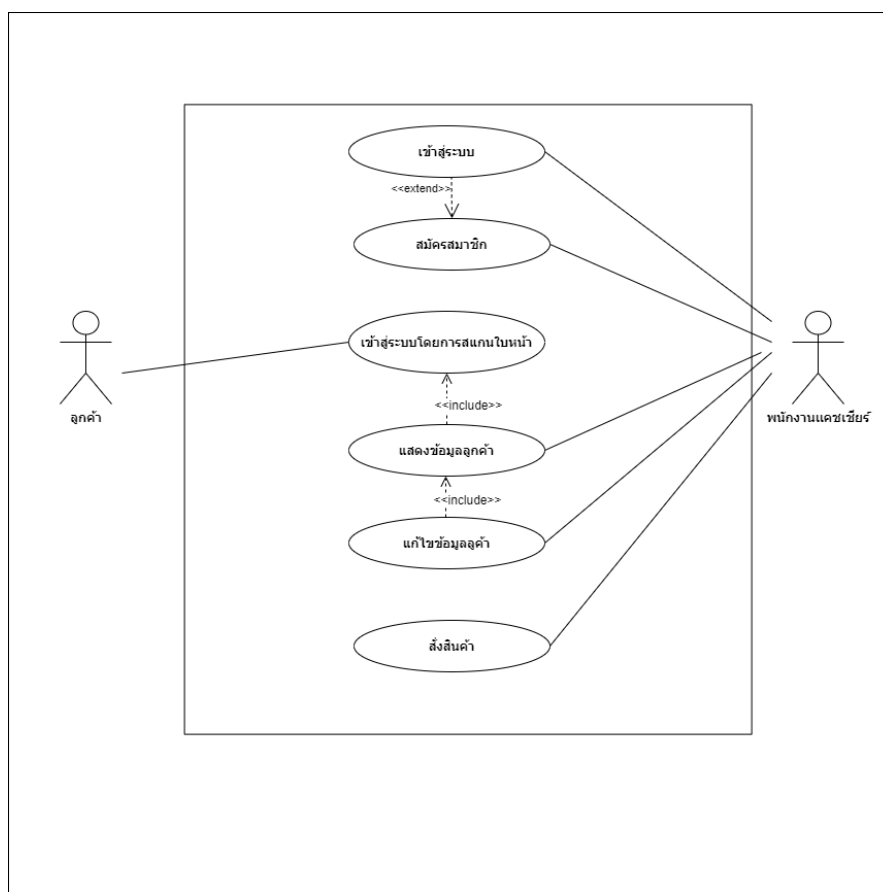
จากภาพที่ ค.7 แสดงหน้าจอฐานข้อมูลทั้งหมดในระบบ ยกตัวอย่างเช่น ข้อมูลผู้ใช้งาน ข้อมูลร้านค้า ข้อมูลเสื้อผ้า เป็นต้น

3.4 Use Case Diagram

Use Case Diagram เป็นแผนผังเพื่อแสดงฟังก์ชันการทำงานของระบบโดยรวม แสดงส่วนประกอบในระบบและกิจกรรมที่เกิดขึ้นในระบบ สัญลักษณ์ที่ใช้ในการเขียน Use Case Diagram แสดงในตารางที่ 3.1

ตารางที่ 3.1: สัญลักษณ์ของ Use case Diagram

สัญลักษณ์	การใช้งาน
Use case	Use case คือส่วนย่อยของระบบงาน แทนด้วยวงรีและชื่อของ Use case ภายในวงรี
	Actor คือบุคคลหรือระบบงานอื่นที่ใช้งานระบบหรือได้รับประโยชน์จากระบบซึ่งอยู่ภายนอกระบบ แทนด้วยรูปคนและมีชื่อบทบาทการใช้งานระบบ
	เส้นตรงที่แสดงถึงการใช้งาน Use case ของผู้กระทำ
	กรอบสี่เหลี่ยม แสดง ถึง ขอบเขต ของ ระบบ โดย แสดง ชื่อ ระบบ ภายในหรือด้านบนกรอบสี่เหลี่ยม Use case อยู่ภายในกรอบสี่เหลี่ยม และ actor อยู่ภายนอกกรอบสี่เหลี่ยม
	ความสัมพันธ์แบบ «includes» แสดงว่า Use case หนึ่งดำเนินการตามขั้นตอนของ Use case อื่น โดยแทนด้วยสัญลักษณ์ลูกศรเส้นประ ซึ่ง Use case ที่หางลูกศรเรียกใช้งาน Use case ที่หัวลูกศรทุกครั้งที่มีการทำงาน
	ความสัมพันธ์แบบ «extend» แสดงว่า Use case หนึ่งดำเนินการตามขั้นตอนของ Use case อื่น โดยแทนด้วยสัญลักษณ์ลูกศรเส้นประ ซึ่ง Use case ที่หัวลูกศรเรียกใช้งาน Use case ที่หางลูกศร แต่การใช้งานไม่จำเป็นต้องเกิดขึ้นทุกครั้งขึ้นอยู่กับเงื่อนไขระหว่างการทำงาน



รูปที่ 3.9: Use Case Diagram ระบบสแกนใบหน้าสำหรับแคชเชียร์

ตารางที่ 3.2: อธิบาย Use Case หน้าที่ของระบบ ในภาพที่ ??

Use Case	คำอธิบาย
สมัครสมาชิก	พนักงานทำการสมัครสมาชิกแก่ลูกค้าที่ต้องการเป็นสมาชิกโดยผ่านการล็อกอินของพนักงานก่อน
สแกนใบหน้าลูกค้า	พนักงานสแกนใบหน้าลูกค้าที่เป็นสมาชิกในระบบโดยผ่านการล็อกอินก่อน
แสดงข้อมูลลูกค้า	แสดงข้อมูลลูกค้าที่เป็นสมาชิกในระบบ โดยต้องผ่านการสแกนใบหน้าก่อนหรือพนักงานที่เป็นสมาชิก
แก้ไขข้อมูลลูกค้า	พนักงานทำการแก้ไขข้อมูลลูกค้าที่เป็นสมาชิก โดยต้องการการแสดงข้อมูลของลูกค้าก่อน
สั่งซื้อสินค้า	พนักงานสั่งซื้อสินค้าตามที่ลูกค้าต้องการ โดยไม่ต้องสมัครเป็นสมาชิกก็สามารถสั่งซื้อสินค้าได้

ตารางที่ 3.3: Use Caseสมัครสมาชิก

Use Case Title : สมัครสมาชิก	Use case Id : 1
Primary Actor : พนักงานแคชเชียร์	
Stakeholder Actor : ลูกค้า	
Main Flow : พนักงานแคชเชียร์สมัครสมาชิกให้ลูกค้าที่ต้องการเป็นสมาชิก โดยพนักงานแคชเชียร์ต้องผ่านการล็อกอิน	
Exceptional Flow ที่ 1 : หากผู้ใช้งานไม่ได้เชื่อมต่ออินเทอร์เน็ตจะไม่สามารถสมัครสมาชิกให้แก่ลูกค้าได้	
Exceptional Flow ที่ 2 : พนักงานแคชเชียร์จะไม่สามารถสมัครสมาชิกให้แก่ลูกค้าได้ถ้าไม่ทำการล็อกอินก่อน	

ตารางที่ 3.4: Use Case สแกนใบหน้าลูกค้า

Use Case Title : สแกนใบหน้าลูกค้า	Use case Id : 2
Primary Actor : พนักงานแคชเชียร์	
Stakeholder Actor : ลูกค้า	
Main Flow : พนักงานแคชเชียร์ทำการสแกนหน้าลูกค้าที่เป็นสมาชิกแล้ว โดยพนักงานจะต้องผ่านล็อกอินเข้าสู่ระบบก่อน	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถสแกนใบหน้าลูกค้าได้	
Exceptional Flow ที่ 2 : พนักงานแคชเชียร์จะไม่สามารถสแกนใบหน้าลูกค้าได้ ถ้าไม่ทำการล็อกอินเข้าสู่ระบบ	
Exceptional Flow ที่ 3 : พนักงานแคชเชียร์จะไม่สามารถสแกนใบหน้าลูกค้าได้ ถ้าไม่สมัครสมาชิกแก่ลูกค้า	

ตารางที่ 3.5: Use Case แสดงข้อมูลลูกค้า

Use Case Title : แสดงข้อมูลลูกค้า	Use case Id : 3
Primary Actor : พนักงานแคชเชียร์	
Stakeholder Actor : ลูกค้า	
Main Flow : ระบบจะแสดงข้อมูลลูกค้าที่เป็นสมัครสมาชิก โดยผ่านการสแกนใบหน้า	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถสแกนใบหน้าลูกค้าได้	
Exceptional Flow ที่ 2 : ระบบจะไม่สามารถแสดงข้อมูลลูกค้าได้ถ้าไม่ผ่านการสแกนใบหน้า	

ตารางที่ 3.6: Use Case สั่งซื้อสินค้า

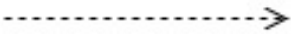
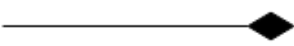
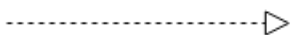

Use Case Title : สม	Use case Id : 5
Primary Actor : พนักงานแคชเชียร์	
Stakeholder Actor : ลูกค้า	
Main Flow เมื่อพนักงานแคชเชียร์เข้าสู่ระบบแล้ว จะทำการสั่งสินค้าให้ ลูกค้าจะเป็นสมาชิก หรือไม่เป็นสมาชิก	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถสั่งสินค้าให้ลูกค้าได้	
Exceptional Flow ที่ 2 : พนักงานแคชเชียร์จะไม่สามารถสั่งสินค้าให้ลูกค้าได้ถ้าไม่ล็อกอินเข้าสู่ระบบ	

3.5 Class Diagram

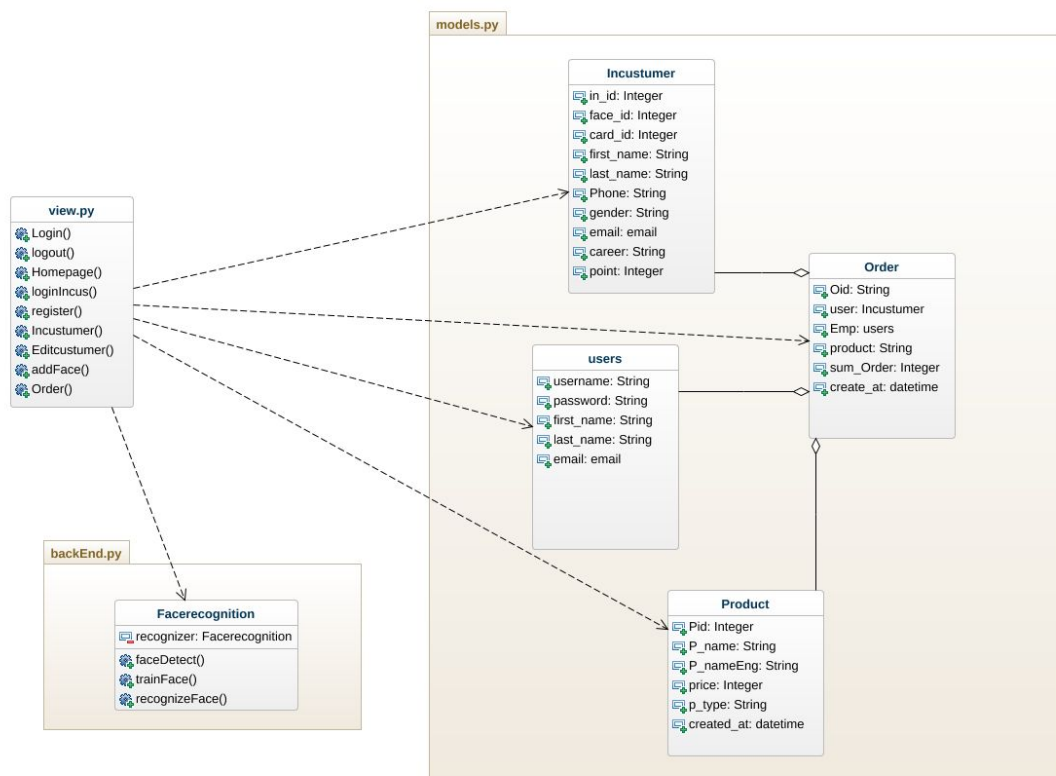
Class Diagram คือแผนภาพที่ใช้แสดงคลาสและความสัมพันธ์ในแบบต่างๆ ระหว่างคลาส

สัญลักษณ์ที่ใช้ในการเขียน Class Diagram แสดงในตารางที่ 3.7

ตารางที่ 3.7: สัญลักษณ์ของ Class Diagram

สัญลักษณ์	การใช้งาน
<div style="border: 1px solid black; padding: 5px; margin: 5px;"> <div style="border: 1px solid black; padding: 2px; text-align: center;">Class Name</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">Attribute Name</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">Operation Name()</div> </div>	<p>คลาส สัญลักษณ์แทนด้วยสี่เหลี่ยมแบ่งเป็น 3 ส่วน ส่วนบน เป็นชื่อของ class ส่วนกลาง เป็นชื่อ Attribute และส่วนล่างเป็น Operation Name หรือ Method ใช้สำหรับเขียนฟังก์ชันในการทำงานของคลาสนั้น ๆ ชนิดของ Visibility ของ Method และ Attribute แบ่งเป็น 3 ชนิด ได้แก่</p> <p>(a) Public แทนสัญลักษณ์ด้วยเครื่องหมายบวก (+)</p> <p>(b) Private แทนสัญลักษณ์ด้วยเครื่องหมายลบ (-)</p> <p>(c) Protected แทนสัญลักษณ์ด้วยเครื่องหมายชาร์ป ()</p>
	Dependency Relationship หมายความว่า คลาสที่อยู่ฝั่งต้นลูกศรสามารถเรียกใช้คลาสที่อยู่ฝั่งหัวลูกศร
	Composition Relationship เป็นความสัมพันธ์ระหว่างออบเจกต์หรือคลาสแบบขึ้นต่อกันและมีความเกี่ยวข้องกันเสมอ
	Realization Relationship เป็นความสัมพันธ์ระหว่าง Object หรือ Class ในลักษณะของการสืบทอดคุณสมบัติจาก Class หนึ่ง (Super class) ไปยังอีก Class หนึ่ง (Subclass)
	Connector เป็นสัญลักษณ์แทนด้วยรูปห้าเหลี่ยมและมีชื่ออยู่ตรงกลาง จะสร้างสัญลักษณ์นี้ไว้เมื่อต้องการเชื่อมต่อคลาสที่อยู่คนละหน้า

Class Diagram แสดงความสัมพันธ์ในรูปแบบต่างๆ ระหว่างคลาสของระบบสแกนใบหน้าสำหรับแคชเชียร์ อธิบายได้ตามภาพที่ 3.10 ดังต่อไปนี้



รูปที่ 3.10: Class Diagram ของแอปพลิเคชันระบบสแกนใบหน้าสำหรับแคชเชียร์

จากรูปภาพที่ 3.10 สามารถอธิบายแผนภาพ Class Diagram ได้ดังนี้

ตารางที่ 3.8: อธิบาย Class Diagram ของคลาสพื้นฐานของระบบ

Class Diagram	คำอธิบาย
Login	คลาส Login จะถูกเรียกใช้งานทุกครั้งเมื่อผู้ใช้เปิดเว็บแอปพลิเคชัน โดยวัตถุประสงค์การทำงานของคลาสนี้คือ เพื่อใช้จัดการทรัพยากรที่จำเป็นสำหรับการใช้งานในคลาสอื่น ๆ
Register	คลาส Register จะถูกใช้งานเมื่อลูกค้าต้องการเป็นสมาชิกในระบบ จากนั้นพนักงานแคชเชียร์จะทำการกรอกข้อมูลลูกค้า แล้วทำการสมัคร โดยจะไปเรียกใช้ฟังก์ชัน register() ใน views.py เมื่อเสร็จแล้วจะเรียกใช้ ฟังก์ชัน Facedtct() และ trainFace() จากคลาส facerecognition ใน backEnd.py แล้วไปเรียกใช้ addFace() views.py
loginIncus	คลาส loginIncus จะ ถูก เรียก ใช้ เมื่อ ลูกค้า ที่ เป็น สมาชิก ต้องการสั่งซื้อสินค้า ระบบจะทำการเรียกใช้ฟังก์ชันloginIncus() ใน views.pyจากนั้นจะทำการเรียกใช้ฟังก์ชัน recognizeFace() จาก คลาส facerecognition ใน backEnd.py เพื่อสแกนใบหน้าลูกค้า
Editcustomer	คลาส Editcustomer จะถูกเรียกใช้เมื่อลูกค้าที่เป็นสมาชิก ต้องการแก้ไขข้อมูล จะเรียกใช้ฟังก์ชัน Incustomer() ใน views.py ผ่าน registerForm() จะแสดงหน้าแก้ไขข้อมูลลูกค้าที่เป็นสมาชิก
Incustomer	คลาสIncustomer ถูกเรียกใช้เมื่อลูกค้าที่เป็นสมาชิกต้องการสั่งซื้อสินค้า จะเรียกใช้จาก loginIncus() เป็นฟังก์ชันที่แสดงข้อมูลลูกค้าที่เป็นสมาชิก
Order	คลาส Order เป็นคลาสสำหรับสั่งซื้อสินค้า โดยจะเรียกใช้ฟังก์ชัน Order() ใน views.py ผ่าน OrderForm() แล้วเก็บข้อมูลรายการสินค้าที่ลูกค้าสั่งซื้อ
addFace	คลาส addFace เป็นคลาส ที่ใช้เรียกฟังก์ชัน Facedtct() และ trainFace() จากคลาส facerecognition ใน backEnd.py เพื่อทำการบันทึกใบหน้าของลูกค้าที่สมัครสมาชิก

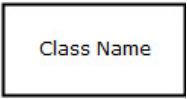


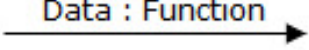



ตารางที่ 3.9: อธิบาย Class Diagram ของคลาสพื้นฐานของระบบ

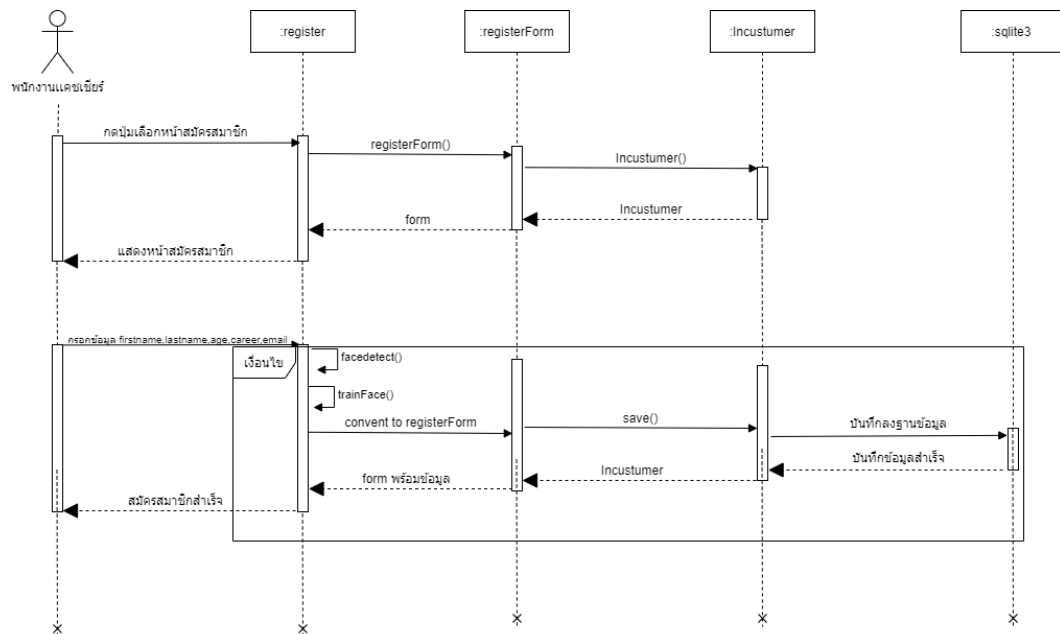
Class Diagram	คำอธิบาย
faceDetect	คลาส faceDetect เป็นคลาสที่ใช้ในการ บันทึกใบหน้าลูกค้าที่สมัครสมาชิก
trainFace	คลาส trainFace เป็นคลาสที่ ทำการtrain ใบหน้าลูกค้าหลังจากทำการfaceDetect()
recognizeFace	คลาส recognizeFace เป็น คลาส ที่ ใช้ เพื่อ เปรียบ เทียบ ใบหน้า ใน วิดีโอ ว่า ตรง กับ ใบหน้า ที่ ผ่าน การ faDetect() และ facetrain()มาแล้ว

3.6 Sequence Diagram

Sequence Diagram เป็น Diagram ที่แสดงขั้นตอนการทำงานของแต่ละ Use Case ระหว่าง Object ต่างๆ ที่ส่งข้อความถึงกันและกัน โดย Sequence Diagram จะช่วยให้มองเห็นการทำงานของภาพรวมของระบบ ส่วนประกอบสัญลักษณ์ที่ใช้ในการเขียน Sequence Diagram แสดงดังตารางที่ 3.10

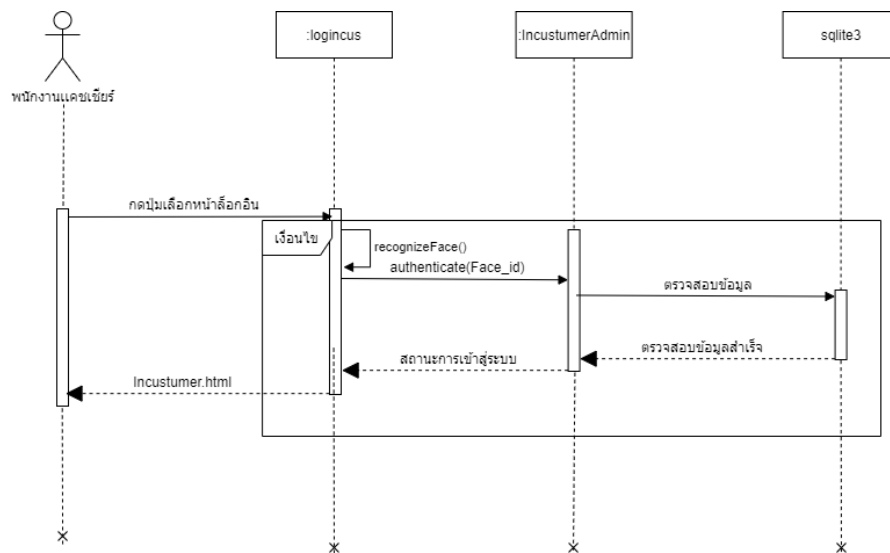
ตารางที่ 3.10: สัญลักษณ์ของ Sequence Diagram

สัญลักษณ์	การใช้งาน
	Class แสดงถึงการทำงานของ Use Case ในการส่งหรือรับข้อความ แทนด้วยสัญลักษณ์สี่เหลี่ยมมีชื่อคลาสอยู่ภายใน
	Lifeline หรือเส้นอายุขัย แสดงช่วงเวลาตั้งแต่เริ่มสร้าง object ในคลาสนั้น จนกระทั่ง object นั้นถูกทำลาย สัญลักษณ์แทนด้วยเส้นประ
	Focus of control หรือจุดควบคุม เป็นจุดควบคุมที่ object ใช้ทำการส่งหรือรับข้อความ สัญลักษณ์แทนด้วยสี่เหลี่ยม
	Message คือ ข้อความที่รับส่งระหว่าง Object สัญลักษณ์แทนด้วยลูกศรและประกอบด้วย 2 ส่วน คือ ข้อมูล (Data) และฟังก์ชัน (Function)
	Return Message เป็นข้อมูลที่ส่งกลับหลังจากทำงานเสร็จ
	Self call เป็นการเรียกฟังก์ชันการทำงานภายในตัวเอง
	สร้างกรอบการทำงานของโปรแกรม เพื่อให้รู้ขอบเขตของการทำงานเช่น ลูป(loop)



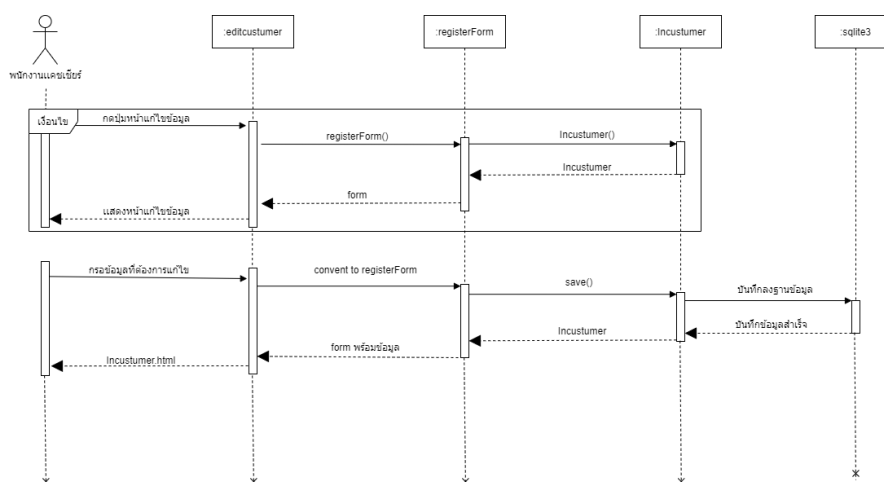
รูปที่ 3.11: Sequence Diagram การสมัครสมาชิก

จากภาพที่ 3.11สามารถอธิบายแผนภาพ Sequence Diagram ของการสมัครสมาชิก ได้ดังนี้ เมื่อพนักงานกดสมัครสมาชิกระบบจะไปเรียกใช้ฟังก์ชัน Register() โดยจะแสดงหน้าสมัครสมาชิกส่งข้อความไปเรียกข้อมูลใน registerForm() และส่งค่าข้อมูลทั้งหมดของสมาชิกมาในรูปแบบ form เมื่อพนักงานกรอกชื่อ นามสกุล เลขบัตรประจำตัวประชาชน เบอร์โทรศัพท์ อายุ เพศ อาชีพ เลือกรูปโปรไฟล์เสร็จและกดปุ่มสมัครสมาชิกระบบจะไปเรียกฟังก์ชัน facedetect() และ trainFace() ซึ่งจะทำการบันทึกใบหน้าลูกค้า นำใบหน้าไปตรวจสอบ ว่ากรอกข้อมูลครบทุกช่องหรือไม่ ถ้าครบแล้วจะส่งข้อมูลไปตรวจสอบในฐานข้อมูล SQLite3 และจะทำการสร้าง user เสร็จจะส่งสถานะการสมัครสมาชิกไปกลับไปให้คลาส Registeruser() และจะแสดงข้อมูลของผู้ใช้ถือว่าสมัครสมาชิกสำเร็จ หรือในกรณีกรอกข้อมูลไม่ครบทุกช่องตามที่กำหนดระบบจะแจ้งให้กรอกช่องที่ยังไม่ได้ทำการกรอก และทำการแจ้งสมัครสมาชิกสำเร็จ



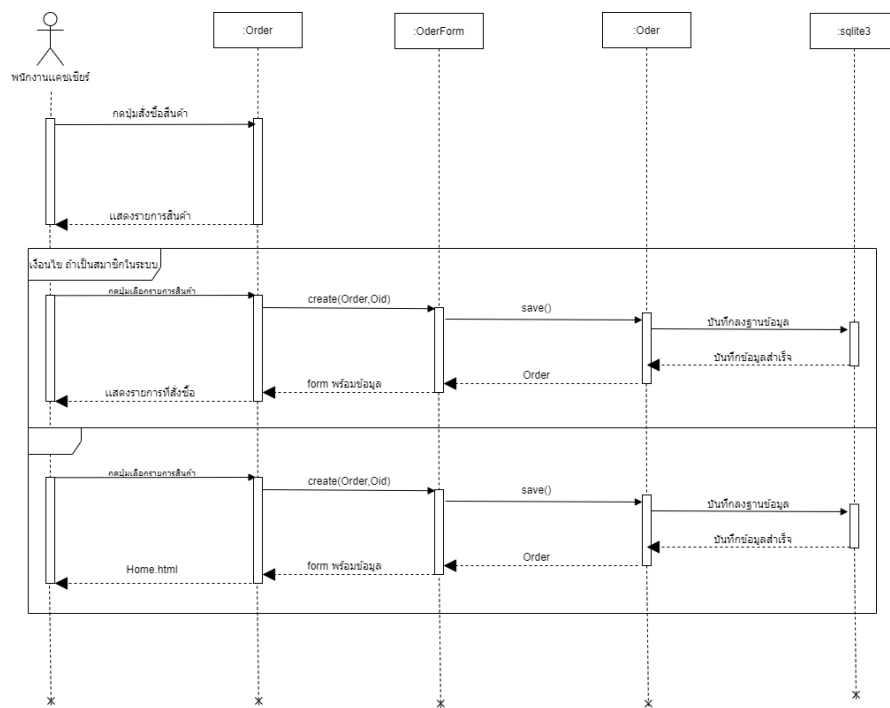
รูปที่ 3.12: Sequence Diagram ล็อกอินลูกค้าโดยการสแกนใบหน้า

จากภาพที่ 3.12 สามารถอธิบายแผนภาพ Sequence Diagram ของการเข้าสู่ระบบ ได้ดังนี้ เมื่อพนักงานแคชเชียร์กดปุ่มเข้าสู่ระบบ จากนั้นระบบจะเรียกใช้ฟังก์ชัน Login() โดยฟังก์ชัน Login() จะไปเรียกใช้ฟังก์ชัน recognizeFace() มาทำการสแกนใบหน้าลูกค้าจากนั้นจะไป เรียกใช้ฟังก์ชัน Login() โดยจะส่ง faceid ไปตรวจสอบ security และ SQLite3 ทำการค้นหาและส่งสถานะการเข้าสู่ระบบกลับไปคลาส Login ซึ่งในคลาสนี้จะทำการตรวจสอบสถานะการเข้าสู่ระบบและจะแสดงหน้าข้อมูลลูกค้า แต่ถ้าหากไม่ผ่านการสมัครสมาชิกมระบบจะแสดงหน้าหลักของระบบ



รูปที่ 3.13: Sequence Diagram การแก้ไขข้อมูลลูกค้าที่เป็นสมาชิก

จากภาพที่ 3.13 สามารถอธิบายแผนภาพ Sequence Diagram ของแก้ไขข้อมูลลูกค้า ได้ ดังนี้ เมื่อผู้ใช้กดเลือกหน้าแก้ไขข้อมูลผู้ใช้งาน ระบบจะเรียกใช้ฟังก์ชัน Editcuster() ส่งข้อความไปเรียกข้อมูลใน RegisterForm() และส่งค่าข้อมูลทั้งหมดของสมาชิกมาในรูปแบบ form จากนั้นเมื่อผู้ใช้ทำการกรอกกรอกข้อมูลหรือเปลี่ยนแปลงข้อมูลลูกค้า และทำการแก้ไขข้อมูลลูกค้า ระบบจะทำการส่งข้อมูลทั้งหมดที่ผู้ใช้กรอกครบไปยังฐานข้อมูล SQLite3 ทำการบันทึกลงฐานข้อมูล โดยมีเงื่อนไขลูกค้าต้องมีข้อมูลในระบบหรือต้องผ่านการสมัครสมาชิกมาก่อน และทำการแจ้งแก้ไขข้อมูลผู้ใช้งานสำเร็จ

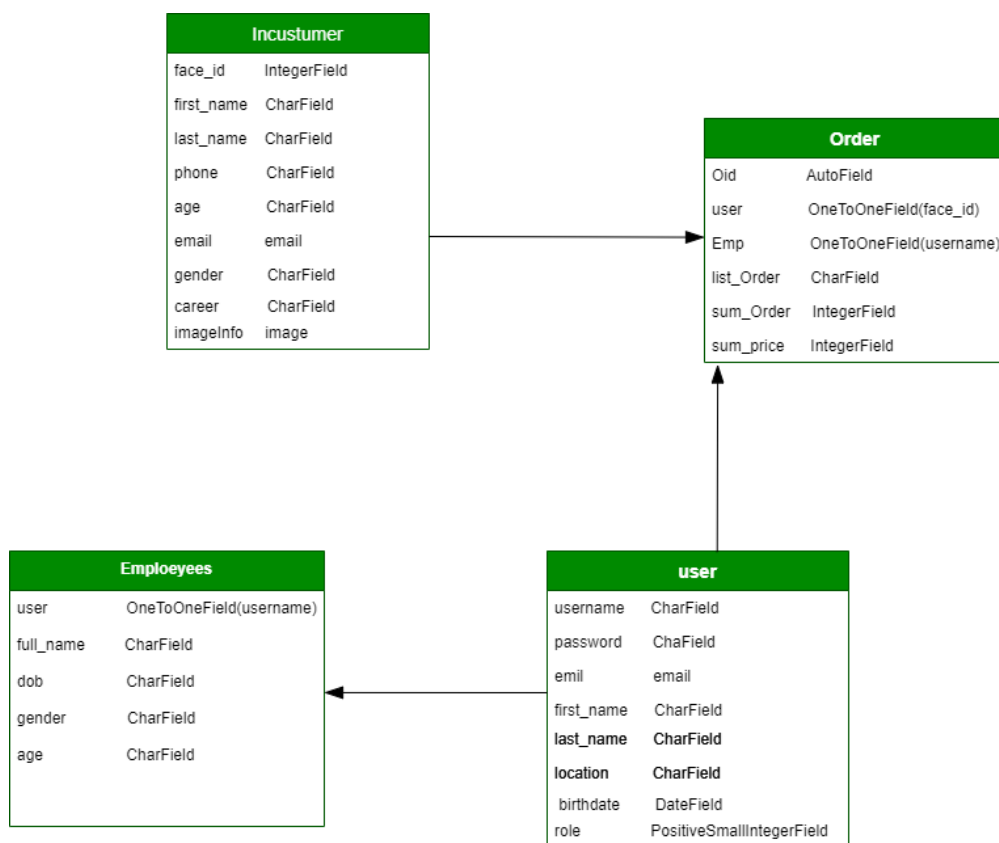


รูปที่ 3.14: Sequence Diagram การสั่งซื้อสินค้า

จากภาพที่ 3.14 สามารถอธิบายแผนภาพ Sequence Diagram แสดงการสั่งซื้อสินค้า ได้ดังนี้ เมื่อลูกค้าต้องการสั่งซื้อสินค้า พนักงานแคชเชียร์จะกดปุ่มสั่งซื้อสินค้า ระบบจะเรียกใช้ฟังก์ชัน `Order()` ระบบจะแสดงหน้ารายการเมนูแล้วการเลือกเมนูที่ลูกค้าต้องการสั่งซื้อ เมนูกดปุ่มสั่งซื้อสินค้าระบบจะไปเรียกใช้ `OrderFom()` และทำการสร้าง รายการสั่งซื้อขึ้นมาโดยใช้เมธอด `Create()` ที่คลาส `Order()` จากนั้นจะทำการบันทึกที่ข้อมูลผ่าน `OederFrom()` แล้วทำการบันทึกไปยัง `sqlite3` จากนั้นจะทำการแสดงรายการสินค้าไปที่หน้าข้อมูลลูกค้า โดยมีเงื่อนไขถ้าลูกค้าไม่เป็นสมาชิกในระบบจะไม่สามารถดูรายการสั่งซื้อได้

3.7 โครงสร้างฐานข้อมูล SQLite3

ฐานข้อมูล SQLite เป็นโปรแกรมฐานข้อมูลที่มีขนาดเล็กมาก (ไม่ถึง 1MB) เก็บฐานข้อมูลเป็นไฟล์โดยไม่จำเป็นต้องมีเซิร์ฟเวอร์ ทำให้ถูกใช้ในหลายๆ โปรแกรมหรือถูกติดตั้งลงในอุปกรณ์พกพาหลายชนิดๆ เช่น iPhone, Android เพื่อใช้ในการเก็บข้อมูล



รูปที่ 3.15: ตารางฐานข้อมูลรายการสั่งซื้อสินค้า

ตารางฐานข้อมูลสมาชิกหรือผู้ใช้งาน Order จะประกอบไปด้วย Incustomer, user, Employees

ตารางที่ 3.11: อธิบายโครงสร้างฐานข้อมูลของ datamodel ของ Order

Key	คำอธิบาย
Oid	สำหรับเก็บรหัสรายการสั่งซื้อ
user	สำหรับเก็บข้อมูลชื่อลูกค้าที่สั่งซื้อ โดยดึงข้อมูลมาจากตาราง Incustomer
Emp	สำหรับเก็บข้อมูลชื่อพนักงานแคชเชียร์ โดยดึงข้อมูลมาจากตาราง user
list Order	สำหรับเก็บข้อมูลรายการสินค้า
sum Order	สำหรับเก็บข้อมูลจำนวนสินค้า
sum price	สำหรับเก็บข้อมูลราคารวมสินค้า

ตารางที่ 3.12: อธิบายโครงสร้างฐานข้อมูลของ datamodel ของ Incustomer

Key	คำอธิบาย
face id	สำหรับเก็บรหัสสมาชิกลูกค้า
first name	สำหรับเก็บชื่อลูกค้า
last name	สำหรับเก็บข้อมูลนามสกุลลูกค้า
phone	สำหรับเก็บข้อมูลเบอร์โทรศัพท์ลูกค้า
age	สำหรับเก็บข้อมูลอายุลูกค้า
email	สำหรับเก็บข้อมูลอีเมลลูกค้า
gender	สำหรับเก็บข้อมูลเพศลูกค้า
career	สำหรับเก็บข้อมูลอาชีพลูกค้า
imageInfo	สำหรับเก็บรูปภาพลูกค้า

ตารางที่ 3.13: อธิบายโครงสร้างฐานข้อมูลของ datamodel ของ Employees

Key	คำอธิบาย
user	สำหรับเก็บชื่อผู้ใช้
full name	สำหรับเก็บข้อมูลชื่อและนามสกุลพนักงาน
dob	สำหรับเก็บข้อมูลวันเดือนปีเกิดพนักงาน
gender	สำหรับเก็บข้อมูลเพศพนักงาน
age	สำหรับเก็บข้อมูลอายุพนักงาน

ตารางที่ 3.14: อธิบายโครงสร้างฐานข้อมูลของ datamodel ของ user

Key	คำอธิบาย
username	สำหรับเก็บข้อมูลชื่อผู้ใช้งาน
password	สำหรับเก็บรหัสผ่าน
email	สำหรับเก็บข้อมูลอีเมลผู้ใช้งาน
first name	สำหรับเก็บข้อมูลชื่อผู้ใช้งาน
last name	สำหรับเก็บข้อมูลนามสกุลผู้ใช้งาน
location	สำหรับเก็บข้อมูลที่อยู่ผู้ใช้งาน
birthdate	สำหรับเก็บข้อมูลวันเกิดผู้ใช้งาน
role	สำหรับเก็บข้อมูลประเภทของผู้ใช้งาน

Product	
Pid	CharField
P_name	CharField
P_nameEng	CharField
Price	IntegerField
p_type	CharField
Create_at	DateTimField

รูปที่ 3.16: ตารางฐานข้อมูลสินค้า

ตารางฐานข้อมูลเมนูสินค้า

ตารางที่ 3.15: อธิบายโครงสร้างฐานข้อมูลของ datamodel ของ Product

Key	คำอธิบาย
Pod	สำหรับเก็บรหัสเมนูเครื่องดื่ม
P name	สำหรับเก็บชื่อสินค้าที่เป็นภาษาไทย
P nameEng	สำหรับเก็บชื่อสินค้าที่เขียนภาษาอังกฤษ
Price	สำหรับเก็บราคาสินค้า
p type	สำหรับเก็บประเภทของเมนู
Cteate at	สำหรับเก็บเวลาที่เพิ่มเมนู

บทที่ 4

การพัฒนาระบบ

ในบทนี้จะกล่าวถึงการสร้างระบบงานของระบบเว็บแนะนำการแต่งกายสุภาพบุรุษและสุภาพสตรี โดยนำผลที่ได้จากการวิเคราะห์และออกแบบระบบมาสร้างเป็นระบบงานซึ่งจะอธิบายถึงตัวอย่างการเขียนโปรแกรมการทำงานของระบบในส่วนต่างๆ ดังต่อไปนี้

4.1 การพัฒนาในส่วนการสมัครสมาชิกหรือเพิ่มข้อมูล

เมื่อพนักงานต้องการสมัครสมาชิกเพื่อเพิ่มข้อมูล โดยผู้พัฒนาจะยกตัวอย่างการทำงานแค่การสมัครสมาชิก เนื่องจากการทำงานของทั้งการสมัครสมาชิก มีรายละเอียดการทำงานดังนี้

```
1 def registercus(request):
2     if request.POST:
3         form = RegisterForm(request.POST or
4                               None)
5         print(form)
6         if form.is_valid():
7             instance = form.save(commit=False)
8             instance.save()
9             messages.success(request, '
10                Successfully Registered!')
11             addFace(request.POST['face_id'])
12             return redirect('/Home/')
13         else:
14             messages.error(request, "Account
15                Register Failed!")
16
17         form = RegisterForm()
18
19         context = {
20             'title' : 'Register Form',
21             'form' : form,
22         }
23         return render(request, 'Webcashier/
24             register.html', context )
```

รูปที่ 4.1: การทำงานของระบบเมื่อผู้ใช้สมัครสมาชิกหรือเพิ่มข้อมูล

จากภาพที่ 4.1 โครงสร้างของไฟล์ views.py อธิบายการทำงานได้ดังนี้ สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เมื่อผู้ใช้กดปุ่มสมัครสมาชิกระบบจะเข้ามาทำงานในฟังก์ชัน registeruser
- บรรทัดที่ 2-12 เป็นการตรวจสอบว่าข้อมูลที่กรอกเข้ามาถูกต้องตรงตามที่กำหนดในไฟล์ forms.py ถ้าครบและถูกต้องให้ทำการบันทึกข้อมูลได้ และทำการเรียกใช้ฟังก์ชัน addFace() แต่ถ้าไม่ถูกต้องให้แสดง message "Account REgister Failed" ในส่วนที่ผิฉะนั้นออกมา
- บรรทัดที่ 13 เป็นการสร้าง form เพื่อเก็บข้อมูลของลูกค้าโดยจะสร้างแยกออกไปยัง form.py
- บรรทัดที่ 13 เป็นการส่งค่าไปแสดงยังหน้า registeruser.html
- บรรทัดที่ 16-19 เป็นการกำหนดค่าตัวแปรที่จะให้แสดงในหน้า register.html
- บรรทัดที่ 20 เป็นการส่งค่าไปยังหน้า register.html

```

1      def addFace(faceid):
2          faceid = faceid
3          facerecognition.faceDetect(faceid)
4          facerecognition.trainFace()
5          return redirect('/')

```

รูปที่ 4.2: การทำงานฟังก์ชันaddFace()

จากภาพที่ 4.2 โครงสร้างของไฟล์ views.py อธิบายการทำงานได้ดังนี้ สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เมื่อมีการสมัครสมาชิกฟังก์ชัน addFace() จะทำงาน
- บรรทัดที่ 2 เป็นการกำหนดค่าตรงกันกับค่าในฐานข้อมูล
- บรรทัดที่ 3 เป็นการเรียกใช้งานของฟังก์ชัน faceDetect() หลังจากกดปุ่มสมัครสมัคร
- บรรทัดที่ 4 เป็นการเรียกใช้งานฟังก์ชัน trainFace() เมื่อฟังก์ชัน faceDetect() ทำงานเสร็จ


```

1 class RegisterForm(forms.ModelForm):
2
3     class Meta:
4         model = Incustomer
5         fields = (
6             'face id',
7             'first name',
8             'last name',
9             'phone',
10            'gender',
11            'age',
12            'email',
13            'career',
14            'imageInfo',
15        )

```

รูปที่ 4.3: การทำงานของระบบเมื่อผู้ใช้สมัครสมาชิกหรือเพิ่มข้อมูล (ต่อ)

จากภาพที่ 4.3 โครงสร้างของไฟล์ forms.py สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-14 เป็นการสร้าง form ข้อมูลแยกออกมาเพื่อทำการส่งค่าไปยังไฟล์ views.py เพื่อให้ได้ข้อมูลที่ครบและถูกต้องตามที่กำหนด

4.2 การพัฒนาในส่วนการเข้าสู่ระบบโดยการสแกนใบหน้า

เมื่อผู้ใช้กดปุ่มเข้าสู่ระบบ ระบบจะมีรายละเอียดการทำงานดังนี้

```

1      def loginIncus(request):
2          if request.method=='POST'
3              face_id = facerecognition.
                  recognizeFace()
4              print(face_id)
5              return redirect('/home/Incustumer
                  /)
6              if face_id is not None:
7                  authlogin(request,face_id)
8                  return redirect('/Home/')
9              else:
10                 pass
11             return render(request, 'Webcashier/Home.
                  html')
```

รูปที่ 4.4: การทำงานของระบบเมื่อผู้ใช้เข้าสู่ระบบโดยการสแกนใบหน้า

จากภาพที่ 4.5 โครงสร้างของไฟล์ views.py สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เมื่อผู้ใช้กดเข้าสู่ระบบจะเข้ามาทำงานในฟังก์ชัน loginIncus()
- บรรทัดที่ 2-3 เป็นการยิงข้อมูล face id ไปตรวจสอบกับข้อมูลในฐานข้อมูล
- บรรทัดที่ 5 ระบบจะแสดงหน้าข้อมูลลูกค้า
- บรรทัดที่ 6-8 หากสแกนใบหน้าแล้ว ข้อมูลที่ส่งมาไม่ตรงในระบบ ระบบจะแสดงหน้าเข้าสู่ระบบเหมือนเดิม
- บรรทัดที่ 7-9 หากไม่มีการเข้าสู่ระบบ ระบบจะแสดงหน้าหลัก

4.3 การพัฒนาในส่วนการเข้าสู่ระบบของพนักงาน

เมื่อผู้ใช้กดปุ่มเข้าสู่ระบบ ระบบจะมีรายละเอียดการทำงานดังนี้

```

1 def loginpage(request):
2     print(request)
3     if request.method == 'POST':
4         print("request.POST")
5         user = authenticate(username=request.
6                               POST['username'], password=request.
7                               POST['password'])
8         print('USER =', user)
9         return redirect('/Home/')
10        if user is not None:
11            print(user)
12            authlogin(request, user)
13            return redirect('/Home/')
14        else :
15            printเขายังไม่ได้กรอก(' login/password ครั้งแรกที่เข้าหน้า
16              นี้()')
17            return render(request, 'Webcashier/login.html')

```

รูปที่ 4.5: การทำงานของระบบเมื่อผู้ใช้เข้าสู่ระบบโดยการสแกนใบหน้า

จากภาพที่ 4.5 โครงสร้างของไฟล์ views.py สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เมื่อผู้ใช้กดเข้าสู่ระบบจะเข้ามาทำงานในฟังก์ชัน loginpage()
- บรรทัดที่ 3-7 เป็นการยิงข้อมูล username และ password ที่ผู้ใช้กรอกเข้ามาเพื่อนำไปตรวจสอบกับข้อมูลในฐานข้อมูล ถ้าตรงกันให้ไปยังหน้าหลัก
- บรรทัดที่ 8-11 หากผู้ใช้กรอก username หรือ password ไม่ตรงกับข้อมูลในฐานข้อมูล และกดปุ่มเข้าสู่ระบบ ระบบจะแสดงหน้าเข้าสู่ระบบเหมือนเดิม
- บรรทัดที่ 12-13 หากผู้ใช้ยังไม่ได้กรอก username และ password แล้วกดปุ่มเข้าสู่ระบบ ระบบจะแสดงหน้าเข้าสู่ระบบเหมือนเดิม

4.4 การพัฒนาในส่วนการแสดงผลข้อมูลลูกค้า

เมื่อพนักงานแคชเชียร์ทำการสแกนใบหน้า เมื่อมีข้อมูลในระบบจะไปยังหน้าแสดงผลข้อมูลลูกค้า ระบบจะมีรายละเอียดการทำงานดังนี้

```

1 def Incustomerpage(request, faceid, Oid ):
2     faceid = int(faceid)
3     data = {
4         'custosmer' : Incustomer.objects.get(
5             face_id= face_id),
6         'Orders' : Order.objects.filter(customer=
7             customer)
8     }
9     return render(request, 'Webcashier/Incustomer.
10         html', data))

```

รูปที่ 4.6: การทำงานของระบบเมื่อผู้ใช้กดดูรายการเสื้อผ้าทั้งหมด

จากภาพที่ 4.6 โครงสร้างของไฟล์ views.py สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เมื่อทำการสแกนใบหน้าแล้วพบว่ามีข้อมูลในระบบ จะเข้ามาทำงานในฟังก์ชัน Incustomerpage()
- บรรทัดที่ 2 เป็นการเรียกดูข้อมูลตรงกับ face id ในตาราง ที่เท่ากับสแกนหน้าใบหน้า และรายการสินค้าในตารางที่ตรงกับ face
- บรรทัดที่ 3-6 เป็นการดึงข้อมูลในตาราง Incustomer และ Order ที่มี Face id ตรงกับที่ส่งค่ามา
- บรรทัดที่ 8 เป็นการส่งค่าไปแสดงยังหน้า Incustomer.html และส่งค่าตัวแปรที่กำหนดไปแสดง

4.5 การพัฒนาในส่วนการสั่งซื้อสินค้า

เมื่อผู้ใช้กดปุ่มสั่งซื้อสินค้า จะแสดงหน้าสั่งซื้อ และเมื่อกดปุ่มสั่งซื้อ ระบบจะมีรายละเอียดการทำงานดังนี้

```

1 def Order(request):
2     context = {
3         'order' : Product.objects.all()
4     }
5     if request.POST:
6         form = OrderForm(request.POST)
7         if form.is_valid():
8             form.instance.Incustomer =
                Incustomer.objects.get(face_id=
                    face_id.Incustomer.request)
9             form.save()
10        else:
11            form.save()
12        return render(request, 'Webcashier/Order.html
            ', context)

```

รูปที่ 4.7: การทำงานของระบบเมื่อผู้ใช้กดดูรายละเอียดสินค้า

จากภาพที่ 4.7 โครงสร้างของไฟล์ backEnd.py สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เมื่อผู้ใช้กดปุ่มสั่งซื้อสินค้าระบบจะเข้ามาทำงานในฟังก์ชัน Order()
- บรรทัดที่ 2-4 เป็นการดึงข้อมูลชื่อเมนูในตาราง Product มาแสดงค่า
- บรรทัดที่ 5-11 เป็นการตรวจสอบว่าลูกค้าเป็นสมาชิกไหม ถ้ามีเก็บข้อมูลรายการสั่งซื้อและข้อมูลลูกค้าผ่าน Orderform() ถ้าลูกค้าไม่เป็นสมาชิกจะทำการเก็บข้อมูลลงในฐานข้อมูลโดยไม่มีการเก็บชื่อลูกค้า
- บรรทัดที่ 12 เป็นการส่งค่าและกำหนดค่าไปแสดงยังหน้า clothedetail.html

4.6 การพัฒนาในส่วนของระบบรู้จำใบหน้า

เมื่อผู้ใช้ต้องการสมัครสมาชิก และตรวจสอบข้อมูลลูกค้าโดยผ่านการสแกนใบหน้า ระบบจะมีรายละเอียดการทำงานดังนี้

```

1 def faceDetect(self, Entry1,):
2     face_id = Entry1
3     cam = cv2.VideoCapture(0)
4     count = 0
5
6     while(True):
7         ret, img = cam.read()
8         gray = cv2.cvtColor(img, cv2.
9             COLOR_BGR2GRAY)
10        faces = detector.detectMultiScale(gray,
11            scaleFactor=1.5,minNeighbors=5)
12
13        for (x,y,w,h) in faces:
14            cv2.rectangle(img, (x,y), (x+w,y+h
15                ), (255,0,0), 2) count += 1
16            cv2.imwrite(BASE_DIR+'Webcashier/
17                dataset/Incustumer.' + str(
18                    face_id) + '.' + str(count) +
19                    ".jpg", gray[y:y+h,x:x+w])
20
21            cv2.imshow('Register Face', img)
22
23            k = cv2.waitKey(100) & 0xff # Press 'ESC'
24            for exiting video
25            if k == 27:
26                break
27            elif count >= 15: # Take 30 face sample
28                and stop video
29                break
30
31        cam.release()
32        cv2.destroyAllWindows()

```

รูปที่ 4.8: การทำงานของระบบเมื่อผู้ใช้กดแก้ไขสินค้า

จากภาพที่ 4.8 โครงสร้างของไฟล์ backEnd.py สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เมื่อเรียกใช้ฟังก์ชัน faceDetect() และมีการประกาศค่า Entry1
- บรรทัดที่ 2 กำหนดให้ Entry1 เท่ากับ faceid
- บรรทัดที่ 6-9 ถ้ามีการทำงานจะทำการตรวจสอบเงื่อนไขว่า ขณะที่กล้องเปิดจะต้องเป็นแนวตั้ง พร้อมรูปภาพ โดยภาพจะเป็นสีขาว-ดำ จะกำหนดขนาดที่จะdetect รูปภาพ
- บรรทัดที่ 11-15 เป็นบันทึกรูปภาพไปไว้ในpathที่กำหนด
- บรรทัดที่ 17-24 เป็นกำหนดจำนวนรูปภาพที่จะบันทึก เมื่อครบตามที่กำหนด จะหยุดทำการบันทึกแล้วหยุดวิดีโอ

```

1 def trainFace(self):
2     path = BASE_DIR+'/Webcashier/dataset'
3
4     def getImagesAndLabels(path):
5         imagePaths = [os.path.join(path,f) for f
6             in os.listdir(path)]
7         faceSamples=[]
8         ids = []
9
10        for imagePath in imagePaths:
11            PIL_img = Image.open(imagePath).
12                convert('L') # convert it to
13                grayscale
14            img_numpy = np.array(PIL_img,'
15                uint8')
16
17            face_id = int(os.path.split(
18                imagePath)[-1].split(".")[1])
19            faces = detector.detectMultiScale(
20                img_numpy)
21
22            for (x,y,w,h) in faces:
23                faceSamples.append(img_numpy
24                    [y:y+h,x:x+w])
25                ids.append(face_id)
26
27        return faceSamples,ids
28
29    print ("\n Training faces. It will take a few
30        seconds. Wait ...")
31    faces,ids = getImagesAndLabels(path)
32    self.__recognizer.train(faces, np.array(ids))
33    self.__recognizer.save(BASE_DIR+'/Webcashier/
34        trainer/trainer.yml') # recognizer.save()
35        worked on Mac, but not on Pi
36    print("\n {0} faces trained. Exiting Program".
37        format(len(np.unique(ids))))

```

รูปที่ 4.9: การทำงานของฟังก์ชัน recognizeFace

จากภาพที่ ?? โครงสร้างของไฟล์ backEnd.py สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เมื่อมีการเรียกใช้ฟังก์ชัน trainFace() หลังจากฟังก์ชัน faceDetect() ทำงานเสร็จ
- บรรทัดที่ 2 การกำหนดเส้นทางสำหรับเก็บรูปภาพ
- บรรทัดที่ 3-20 เป็นการใช้งานฟังก์ชันเรียกเก็บรูปภาพและรหัสภาพ แล้วส่งค่าออกมาเก็บเป็นอาร์เรย์
- บรรทัดที่ 22-26 เป็นการนำรูปภาพและรหัสภาพมาเก็บไว้ในตัวแปรชื่อ faces,ids จากนั้นทำการtrain แล้วบันทึกโมเดลไปยังpath ที่กำหนดไว้
- บรรทัดที่ 26 ทำการปรี้นจำนวนหน้าที่train เข้าไปยังโมเดลแล้วจบการทำงาน

```

1 def recognizeFace(self):
2     self.__recognizer.read(BASE_DIR+'/trainer/
      trainer.yml')
3     cascadePath = BASE_DIR+'/
      haarcascade_frontalface_default.xml'
4     faceCascade = cv2.CascadeClassifier(cascadePath
      )
5     font = cv2.FONT_HERSHEY_SIMPLEX
6     confidence = 0
7     cam = cv2.VideoCapture(0)
8     minW = 0.1*cam.get(3)
9     minH = 0.1*cam.get(4)
10    while True:
11        ret, img =cam.read()
12        gray = cv2.cvtColor(img,cv2.
          COLOR_BGR2GRAY)
13        faces = faceCascade.detectMultiScale(
14            gray,
15            scaleFactor = 1.2,
16            minNeighbors = 5,
17            minSize = (int(minW), int(minH)),
18            )
19        for(x,y,w,h) in faces:
20            cv2.rectangle(img, (x,y), (x+w,y+h
              ), (0,255,0), 2)
21            face_id, confidence = self.
              __recognizer.predict(gray[y:y+h
                ,x:x+w])
22            if (confidence < 100):
23                name = 'Detected'
24            else:
25                name = "Unknown"
26            cv2.putText(img, str(name), (x+5,y
              -5), font, 1, (255,255,255), 2)
27            cv2.putText(img, str(confidence), (
              x+5,y+h-5), font, 1,
              (255,255,0), 1)
28            cv2.imshow('Detect Face',img)
29            k = cv2.waitKey(10) & 0xff # Press 'ESC'
              for exiting video
30            if k == 27:
31                break
32            if confidence > 50:
33                break
34            print("\n Exiting Program")
35            cam.release()
36            cv2.destroyAllWindows()
37            return face_id

```

รูปที่ 4.10: การทำงานของฟังก์ชัน recognizeFace()

จากภาพที่ 4.10 โครงสร้างของไฟล์ backEnd.py สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เมื่อมีการเรียกใช้ฟังก์ชัน recognizeFace() หลังจากเรียกใช้ฟังก์ชัน LoginIncus() ใน view.py
- บรรทัดที่ 2-9 เป็นการอ่านไฟล์ trainer.yml และตรวจสอบที่อยู่ของไฟล์ haarcascadefrontalface default.xml แล้วกลัองเริ่มทำงาน
- บรรทัดที่ 10-35 เมื่อมีการทำงานจะทำการเปรียบเทียบใบหน้าว่า ตรงกับส่วนที่ detect มาหรือไม่ ถ้าตรงกัน กลัองจะหยุดทำงาน
- บรรทัดที่ 36 เป็นส่งค่าตัวแปรที่ชื่อ face id ออกมา

บทที่ 5

การทดสอบระบบ

หลังจากที่ได้ผ่านขั้นตอนสำหรับการพัฒนาระบบทั้งหมดแล้ว ขั้นตอนต่อไปคือการทดสอบระบบเพื่อตรวจสอบการทำงานของแต่ละฟังก์ชันกันรวมถึงตรวจสอบความถูกต้องของการทำงาน ทั้งระบบเพื่อให้การพัฒนาโปรแกรมเป็นไปอย่างมีประสิทธิภาพ ทดสอบระบบเพื่อหาข้อผิดพลาด และแก้ไขข้อบกพร่องนั้นก่อนการใช้งานจริง

5.1 การทดสอบในส่วนฟังก์ชันก์ของระบบ

5.1.1 ผลการทดสอบการสมัครสมาชิกสำหรับลูกค้า

ตารางที่ 5.1: ผลการทดสอบการสมัครสมาชิกสำหรับลูกค้า

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบการสมัครสมาชิก	ผู้ใช้กดปุ่มสมัครสมาชิก	ระบบแสดงหน้าสมัครสมาชิก ซึ่งข้อมูลที่แสดงในหน้านั้นจะประกอบไปด้วย faceid ชื่อ นามสกุล เบอร์ เพศ อายุ อาชีพ รูปภาพ และกดปุ่มสมัครสมาชิก
	ผู้ใช้กด ปุ่ม สมัคร สมาชิก โดยกรอก ข้อมูล ไม่ ครบ ทุก ช่อง ตาม ที่ กำหนด และ กด ปุ่มสมัครสมาชิก	ระบบแสดง error ในช่องที่ผู้ใช้ยังไม่ได้ทำการกรอก
	ผู้ใช้กรอกข้อมูลครบทุกช่อง และกดปุ่มสมัครสมาชิก	ระบบจะทำการ detect ใบหน้า แล้วกลับไปหน้าหลัก

5.1.2 ผลการทดสอบการเข้าสู่ระบบผู้ใช้งาน

ตารางที่ 5.2: ผลการทดสอบการเข้าสู่ระบบโดยการสแกนใบหน้าลูกค้า

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบ การ เข้า สู่ ระบบ โดย การสแกนใบหน้าลูกค้า	ผู้ใช้เข้ามาในหน้าเข้าสู่ระบบ	ระบบแสดงหน้าหลัก
	ผู้ใช้กดปุ่มเข้าสู่ระบบโดยการสแกนใบหน้า	ทำการเปิดกล้องเพื่อตรวจจับใบหน้า
	เข้าสู่ระบบ โดย ไม่ได้ สมัครสมาชิก	ระบบ จะ กลับ ไป แสดง หน้าหลัก
	เข้าสู่ระบบ โดย มี การ สมัครสมาชิกแล้ว	ระบบ จะแสดง ไป ยัง หน้าแสดงข้อมูลลูกค้า

5.1.3 ผลการทดสอบการแก้ไขข้อมูลลูกค้า

ตารางที่ 5.3: ผลการทดสอบการแก้ไขข้อมูลลูกค้า

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบหน้าแก้ไขข้อมูลลูกค้า	ผู้ใช้กดหน้าแก้ไขข้อมูลลูกค้า	ระบบ จะแสดง หน้าแก้ไขข้อมูลลูกค้าพร้อมข้อมูลลูกค้า
	ผู้ใช้มี การ เปลี่ยนแปลง แก้ไขข้อมูลลูกค้าและกดปุ่มบันทึกข้อมูล	ระบบจะ แสดง หน้าข้อมูลลูกค้าพร้อมข้อมูลที่เปลี่ยนแปลงแก้ไข
	ผู้ใช้ไม่เปลี่ยนแปลงแก้ไขข้อมูลลูกค้าและกดปุ่มบันทึกข้อมูล	ระบบจะแสดง หน้าข้อมูลลูกค้าที่ไม่มีการเปลี่ยนแปลง

5.1.4 ผลการทดสอบหน้าสั่งซื้อสินค้า

ตารางที่ 5.4: ผลการทดสอบหน้าสั่งซื้อสินค้า

การทดสอบ	เงื่อนไขการทดสอบ	ผลการทดสอบ
ทดสอบรายการสั่งซื้อสินค้า	ผู้ใช้กดปุ่มสั่งซื้อสินค้า	ระบบแสดงหน้าเมนูสินค้าที่มีในระบบ
	ผู้ใช้กดปุ่มเลือกเมนูที่ต้องการ	ระบบ จะ ทำการ เพิ่ม สินค้า และและแสดง รายการ สินค้า ที่เลือก
	ไม่มีกดปุ่มเลือกเมนู	ระบบ จะจะไม่ มี การ แสดง รายการเมนู
	ผู้ใช้กดปุ่มยืนยันการสั่งซื้อที่มีรายการเมนูแสดง	ระบบ จะ ทำ การ แสดง ราคา รวมสินค้า เงินที่รับมา และแสดงเงินทอน
	ผู้ใช้ กด ปุ่ม ยืนยัน การ สั่ง ซื้อ โดยที่ไม่มีรายการเมนูแสดง	ระบบจะแสดงหน้าสั่งซื้อสินค้าเหมือนเดิม

บทที่ 6

สรุปและข้อเสนอแนะ

การดำเนินโครงการเพื่อพัฒนาระบบเว็บไซต์แบบใหม่สำหรับแคชเชียร์นี้ พบว่าระบบสามารถทำงานได้ตามที่วิเคราะห์และออกแบบไว้ แต่ก็พบปัญหาและอุปสรรคระหว่างการพัฒนา ในบทนี้ผู้พัฒนาจึงขอสรุปความสามารถของระบบชี้แจงปัญหาและอุปสรรค พร้อมเสนอ แนวทางในการพัฒนาระบบเว็บไซต์แบบใหม่สำหรับแคชเชียร์ต่อตามลำดับ

6.1 สรุปความสามารถของระบบ

1. ผู้ใช้งานหรือพนักงานแคชเชียร์
 - สามารถสมัครสมาชิกลูกค้าได้
 - สามารถเข้าสู่ระบบโดยการสแกนใบหน้าได้
 - สามารถแก้ไขข้อมูลลูกค้าได้
 - สามารถดูข้อมูลลูกค้าได้แต่ต้องผ่านการสแกนใบหน้าลูกค้าเท่านั้น
 - สามารถสั่งสินค้าแล้วบันทึกข้อมูลได้
 - สามารถแสดงรายการสินค้าที่ลูกค้าสั่งได้

6.2 ปัญหาและอุปสรรคในการพัฒนา

1. ตัวอย่างการใช้ djangoframework ในการพัฒนา มีในส่วนที่เป็นภาษาไทยน้อย ทำให้ผู้ศึกษาต้องใช้เวลาในการทำความเข้าใจ
2. การพัฒนาระบบรู้จำใบหน้า มีเนื้อหาในการศึกษาค่อนข้างน้อย และมีการอัปเดตหรือเปลี่ยนแปลงเวอร์ชันของเครื่องมือที่ใช้พัฒนาสูงขึ้นเรื่อยๆ ทำให้รูปแบบการเขียนเปลี่ยนแปลงไปเรื่อยๆ และค่อนข้างแก้ไขได้ยาก เลยทำให้เกิดความล่าช้าในการพัฒนาในส่วนนี้

6.3 แนวทางการพัฒนาต่อ

1. เพิ่มฟังก์ชันแนะนำเมนูเพื่อช่วยในการตัดสินใจในการสั่งซื้อสินค้า
2. เพิ่มในส่วนของฟังก์ชันสั่งซื้อสินค้า มีประเภทเมนูมากขึ้น
3. เพิ่มในส่วนการเก็บสะสมแต้มของลูกค้าที่เป็นสมาชิก สามารถทำการใช้แต้มเพื่อเป็นส่วนลด
4. เพิ่มในส่วนแสดงโปรโมชั่นของร้านที่มีในขณะนั้นได้

บรรณานุกรม

- [1] Sarayut Nonsiri, P. (). ภาษาโปรแกรม python คืออะไร [ออนไลน์]. สืบค้นเมื่อ มีนาคม 2563. จาก <https://www.9experttraining.com/articles/python-คืออะไร>.
- [2] AmplySoft (2556). เริ่มต้น การเขียนเว็บไซต์ ทำเว็บไซต์ ด้วย ภาษา python กับ django framework [ออนไลน์]. สืบค้นเมื่อ มีนาคม 2563. จาก <http://www.amplysoft.com/knowledge/what-is-django-framework-python.html>.
- [3] Thiengkunakrit, W. (2560). เริ่มพัฒนา web application กับ ภาษา python ด้วย django framework [ออนไลน์]. สืบค้นเมื่อ มีนาคม 2563. จาก <https://www.codeburst.io/เริ่มพัฒนา-web-application-กับภาษา-python-ด้วย-django-framework-38ce132ac706>.
- [4] ()Opencv python tutorial 4 for face recognition and identification [ออนไลน์]. สืบค้นเมื่อ ธันวาคม 2562. จาก <https://www.youtube.com/watch?v=PmZ29Vta7Vc>.
- [5] ()รู้จักเทคโนโลยี “จดจำใบหน้า” [ออนไลน์]. สืบค้นเมื่อ กันยายน 2562. จาก <http://cctv.co.th/บทความกล้องวงจรปิด/รู้จักเทคโนโลยี-จดจำ ใบหน้า/176>.
- [6] bccrwp.org (2562). Sqlite เทียบ กับ mysql เทียบ กับ postgresql: การเปรียบเทียบระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ [ออนไลน์]. สืบค้นเมื่อ มีนาคม 2563. จาก <https://th.bccrwp.org/compare/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems-8f2704/>.
- [7] NEWS, M. T. (). ปัญญาประดิษฐ์ คืออะไร (ai : Artificial intelligence) [ออนไลน์]. สืบค้นเมื่อ ธันวาคม 2562. จาก <https://www.modify.in.th/17128>.

ภาคผนวก

ภาคผนวก ก

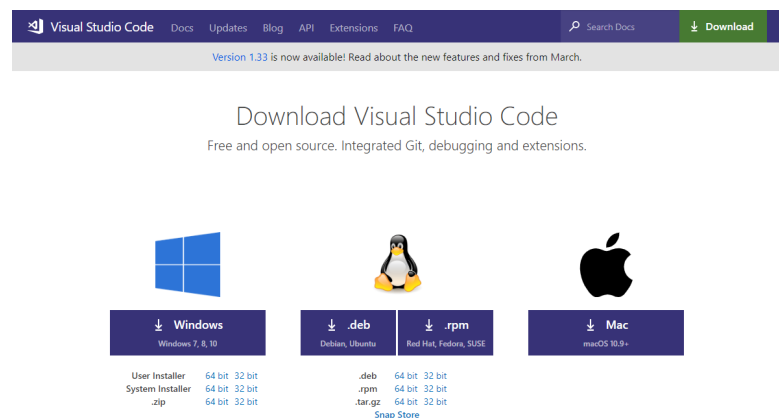
การติดตั้งเครื่องมือที่ใช้พัฒนาโปรแกรม

การติดตั้งเครื่องมือที่ใช้ในการพัฒนาเว็บแอปพลิเคชันระบบแพลตฟอร์มระบบติดตามงาน มีโปรแกรมที่จำเป็นในการพัฒนาระบบดังต่อไปนี้

- การติดตั้ง Visual Studio Code

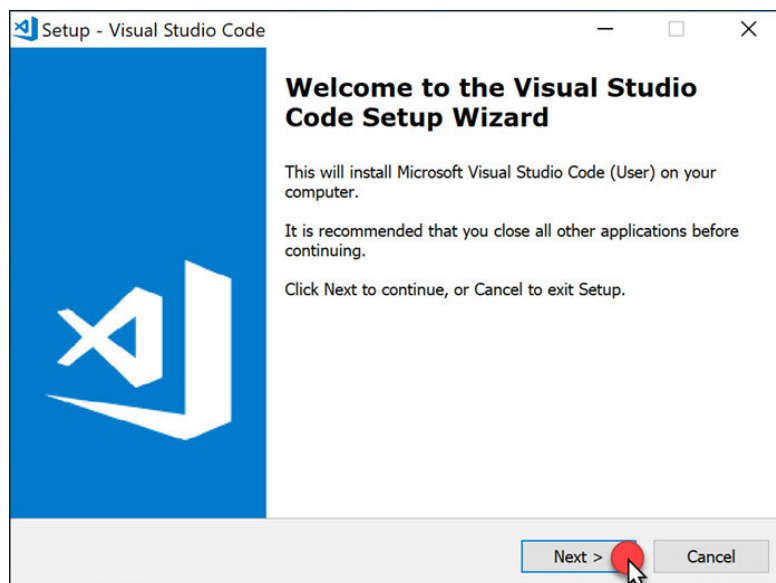
ก.1 การติดตั้ง Visual Studio Code

1. สามารถดาวน์โหลด Visual Studio Code ได้ที่ <https://code.visualstudio.com/download> ดังแสดงในรูปที่ ก.1



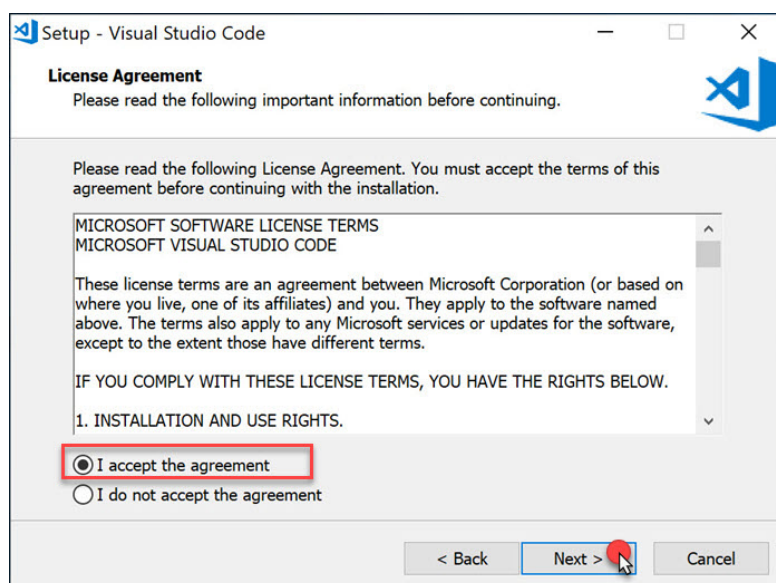
รูปที่ ก.1: หน้าเว็บดาวน์โหลด Visual Studio Code

2. แสดงหน้าต่างต้อนรับของ Visual Studio Code ทำการกด Next เพื่อเริ่มกระบวนการติดตั้ง ดังแสดงในรูปที่ ก.2



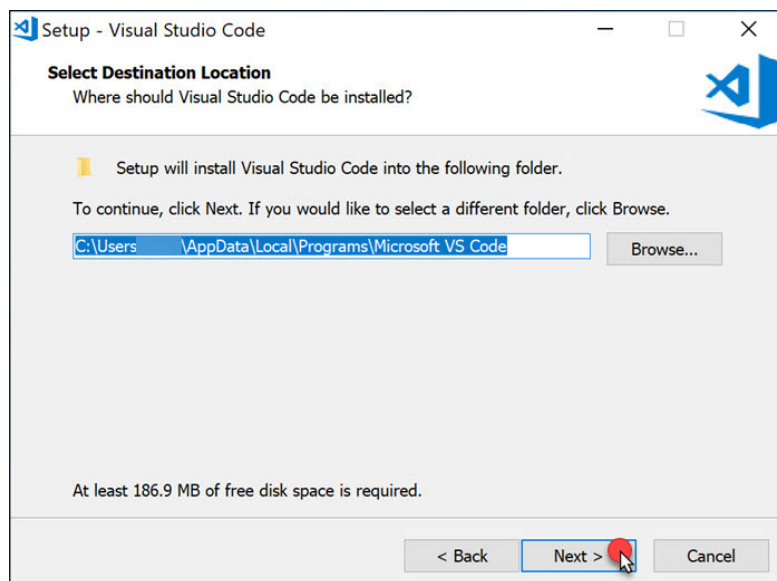
รูปที่ ก.2: หน้าต่างต้อนรับของ Visual Studio Code

3. แสดงหน้าต่างข้อตกลงการใช้งาน Visual Studio Code ทำการกด I Agree ดังแสดงในรูปที่ ก.3



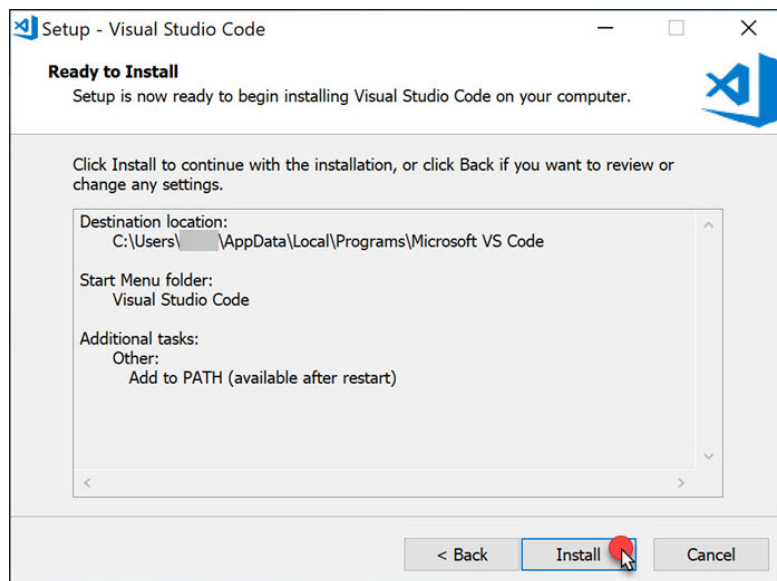
รูปที่ ก.3: หน้าต่างข้อตกลงการใช้งาน Visual Studio Code

4. แสดงหน้าต่างที่จัดเก็บไฟล์ต่างๆ ของ Visual Studio Code ทำการกด Next ดังแสดงในรูปที่ ก.4



รูปที่ ก.4: หน้าต่างที่จัดเก็บไฟล์ต่างๆ ของ Visual Studio Code

5. แสดงหน้าต่างเริ่มทำการติดตั้งทำการกด Install ดังแสดงในรูปที่ ก.5



รูปที่ ก.5: หน้าต่างติดตั้งโปรแกรม Visual Studio Code

6. แสดงหน้าต่างผลการติดตั้ง Visual Studio Code ดังแสดงในรูปที่ ก.6



รูปที่ ก.6: หน้าต่างผลการติดตั้ง Visual Studio Code

ก.2 การติดตั้ง Django Framework

การติดตั้ง Django Framework ผ่าน command line ดังนี้

1 การติดตั้ง library Django Framework ดังแสดงในรูปที่ ก.7

```
1 pip install django
```

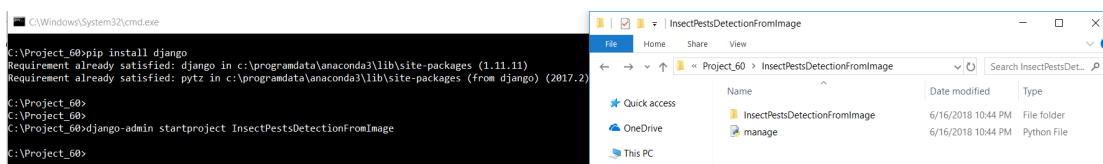
รูปที่ ก.7: การติดตั้ง Django Framework

2 การติดตั้งการสร้างโปรเจค ดังแสดงในรูปที่ ??

```
1 django-admin startproject
  InsectPestsDetectionFromImage
```

รูปที่ ก.8: การติดตั้งส่วนของโปรเจค

รูปภาพประกอบในส่วนการ การติดตั้ง library และการสร้างโปรเจคของ Django Framework และการติดตั้งการสร้างโปรเจค ดังรูปที่??



รูปที่ ก.9: การติดตั้ง Django Framework

3 การรัน Server เพื่อใช้งาน Django Framework ด้วยคำสั่ง `py manage.py runserver`

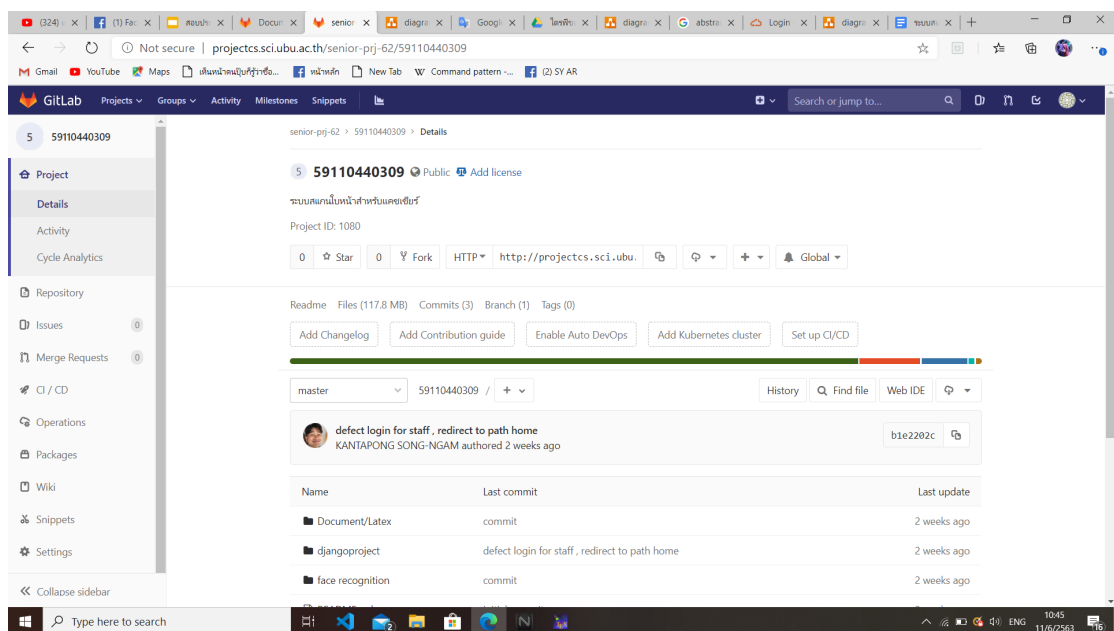
รูปที่ ก.10: การรัน Server เพื่อใช้งานระบบ

ภาคผนวก ข

คู่มือการติดตั้งระบบ

คู่มือการติดตั้งระบบ ในการติดตั้งเพื่อใช้งานเว็บแอปพลิเคชันระบบแนะนำการแต่งกายรูปภาพ
บุรุษและสุภาพสตรี สามารถทำได้โดยมีขั้นตอนดังนี้

1. สามารถ Clone Project ได้ที่ <http://projectcs.sci.ubu.ac.th/senior-prj-62/59110440309.git>



รูปที่ ข.1: หน้าเว็บ Repository ของโปรเจ็ค

2. เข้าไปใน 59110440204 แล้วใช้คำสั่ง migrate สร้างตารางตามลำดับดังต่อไปนี้

```
1 manage.py makemigrations  
   lookinggreat  
2 python manage.py makemigrations  
3 python manage.py migrate  
   lookinggreat  
4 python manage.py migrate
```

รูปที่ ข.2: คำสั่ง migrate สร้างตาราง

3. จากนั้นทำการโหลดข้อมูลลงในตารางดังต่อไปนี้

```
1 python manage.py loaddata Category
    Gender Shape Size SkinColor
    Member Store Color Cloth
    FavoriteCloth FavoriteStore
    Recommend
```

รูปที่ ข.3: คำสั่ง migrate สร้างตาราง

4. จากนั้นทำการสร้าง admin ดังต่อไปนี้

```
1 python manage.py createsuperuser
```

รูปที่ ข.4: คำสั่ง migrate สร้างตาราง

5. จากนั้นทำการ run โปรแกรมโดยใช้คำสั่งดังรูปที่ ข.4 website <http://127.0.0.1:8000> บนเว็บ server <http://lookinggreat.pythonanywhere.com/> ดังต่อไปนี้

```
1 python manage.py runserver
```

รูปที่ ข.5: คำสั่ง migrate สร้างตาราง

ภาคผนวก ค

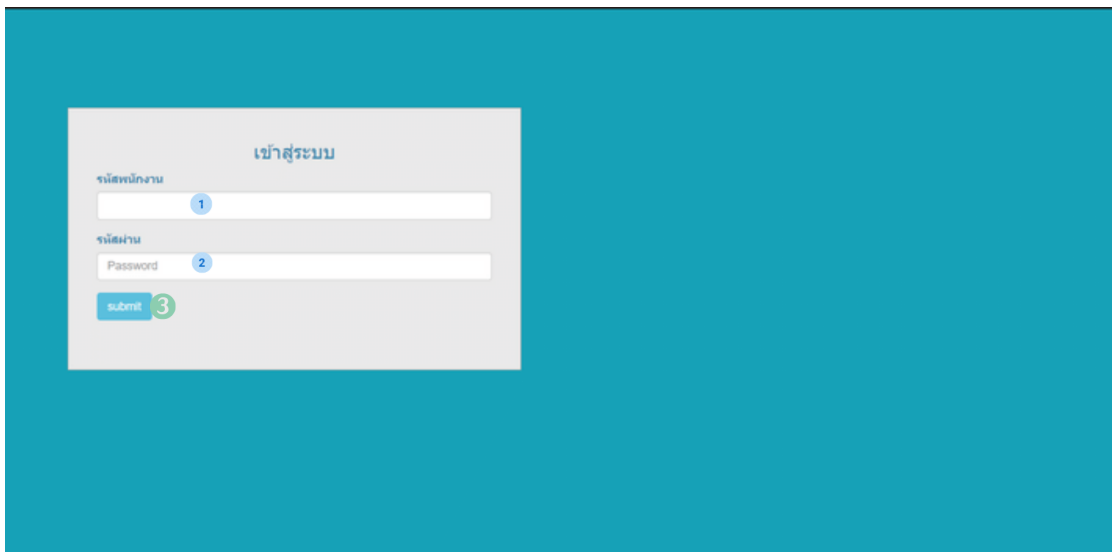
คู่มือการใช้งานระบบ

คู่มือการใช้งานทั้งหมดของระบบ สามารถแบ่งออกเป็น 2 ส่วน ดังนี้

- 1 ส่วนผู้ใช้งาน
- 2 ส่วนของหน้าตาฐานข้อมูลทั้งหมด

ค.1 ส่วนผู้ใช้งาน

1. ส่วนหน้าเข้าสู่ระบบสำหรับพนักงาน
 - เมื่อผู้ใช้เข้าเว็บระบบจะแสดงหน้าหลักดังต่อไปนี้



รูปที่ ค.1: หน้าจอเข้าสู่ระบบสำหรับพนักงาน

จากรูปที่ ค.1 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลขที่ 1 ช่องกรอกชื่อผู้ใช้
 - หมายเลขที่ 2 ช่องกรอกรหัสผ่าน
 - หมายเลขที่ 3 ปุ่มกดเข้าสู่ระบบ
2. ส่วนของหน้าสมัครสมาชิก
 - เมื่อผู้ใช้กดปุ่มสมัครสมาชิกระบบจะแสดงหน้าให้กรอกข้อมูลสมัครสมาชิกดัง

ต่อไปนี้เป็น

The screenshot shows a web form titled 'ลงทะเบียน' (Register). The form includes the following elements marked with numbers:

- 1: First name input field
- 2: Last name input field
- 3: Thai ID number input field (เลขบัตรประชาชน)
- 4: Phone number input field (phone)
- 5: Email input field
- 6: Gender input field (เพศ)
- 7: Age input field (Age)
- 8: Career input field (career)
- 9: Register button
- 10: Clear button
- 11: File upload button (Choose file) and 'No file chosen' text

รูปที่ ค.2: หน้าจอสมัครสมาชิก

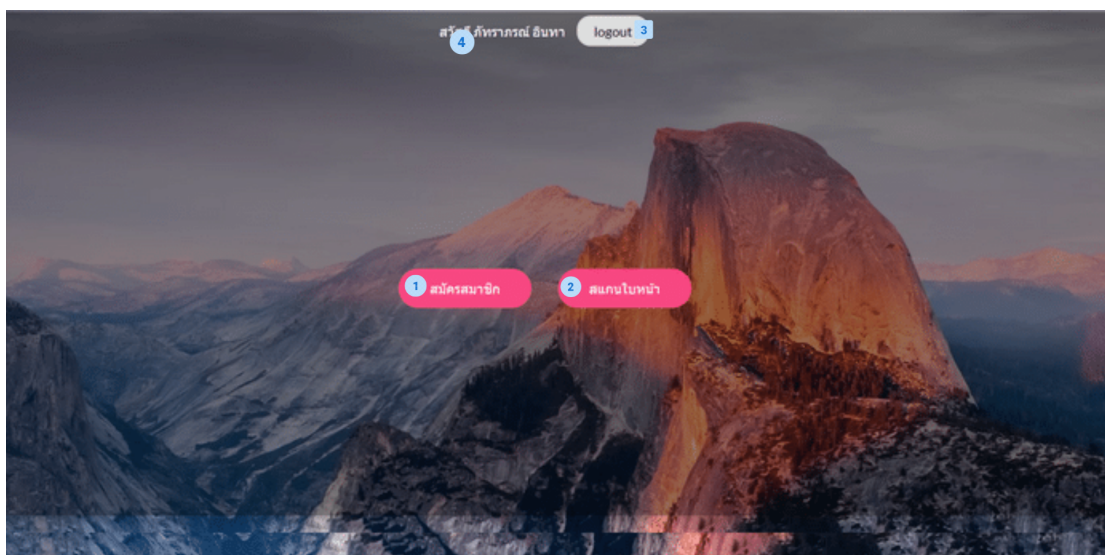
จากรูปที่ ค.2 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลขที่ 1 ช่องกรอกชื่อ
- หมายเลขที่ 2 ช่องกรอกนามสกุล
- หมายเลขที่ 3 ช่องกรอกเลขบัตรประชาชน
- หมายเลขที่ 4 ช่องกรอกเบอร์โทรศัพท์
- หมายเลขที่ 5 ช่องกรอก E-mail
- หมายเลขที่ 6 ช่องเพศ
- หมายเลขที่ 7 ช่องกรอกอายุ
- หมายเลขที่ 8 ช่องเลือกกรุปโปรไฟล์
- หมายเลขที่ 9 ปุ่มกดสมัครสมาชิก
- หมายเลขที่ 10 ปุ่มกดยกเลิกข้อมูลที่กรอก
- หมายเลขที่ 11 ปุ่มกลับหน้าหลักของระบบ

3. ส่วนของหน้าหลักของระบบ

- เมื่อผู้ใช้กดปุ่มเข้าสู่ระบบ ระบบจะทำการสแกนใบหน้าลูก จากนั้นจะแสดงหน้า

จอดังต่อไปนี้



รูปที่ ค.3: หน้าจอหลักของ

จากรูปที่ ค.3 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลขที่ 1 ปุ่มกดสมัครสมาชิกลูกค้า
- หมายเลขที่ 2 ปุ่มกดสแกนใบหน้าเพื่อแสดงข้อมูลลูกค้า
- หมายเลขที่ 3 ปุ่มกดออกจากระบบพนักงาน
- หมายเลขที่ 4 ส่วนแสดงชื่อลูกค้าที่เข้าสู่ระบบ

4. ส่วนของหน้าแสดงข้อมูลลูกค้า

- เมื่อผู้ใช้กดปุ่มเข้าสู่ในหน้าจอหลักระบบจะแสดงหน้าจอต่อไปนี้

The screenshot displays a user interface with two main sections: 'ข้อมูลลูกค้า' (Customer Information) on the left and 'ข้อมูลสินค้า' (Product Information) on the right.

Customer Information Section:

- 1: Login button (ปุ่มกดแก้ไขข้อมูลลูกค้า)
- 2: Logout button (ปุ่มกดดูประวัติ)
- 3: Profile icon (Placeholder for user profile picture)
- Fields for: ชื่อ (Name), เบอร์โทรศัพท์ (Phone Number), ที่อยู่ (Address), อีเมล (E-mail)

Product Information Section:

- 4: Order history table
- 5: Logout button (ปุ่มกดดูประวัติ)

รหัสสินค้า	รายการ	ราคา(บาท)
10	10	600

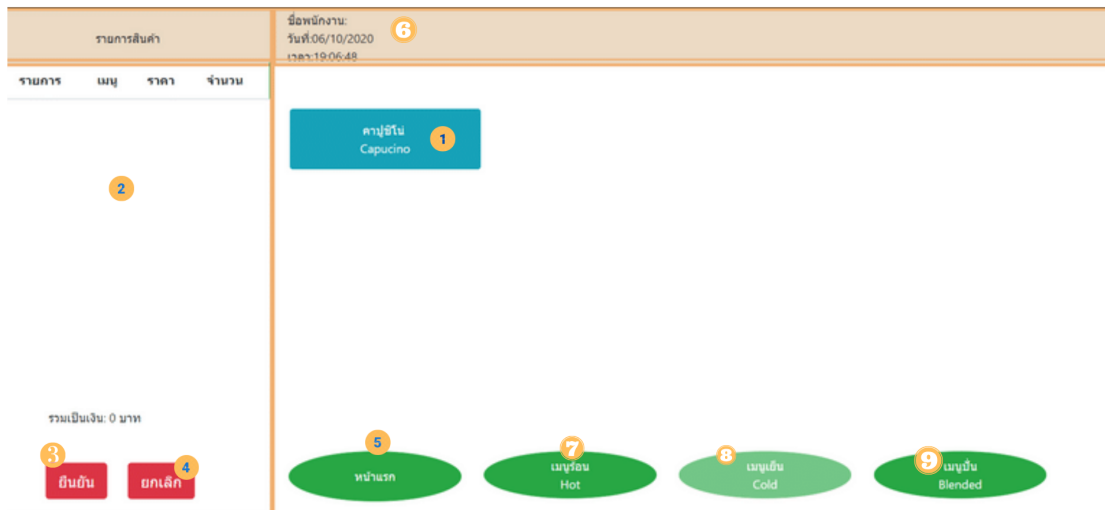
รูปที่ ค.4: หน้าจอแสดงข้อมูลลูกค้า

จากรูปที่ ค.4 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลขที่ 1 ปุ่มกดแก้ไขข้อมูลลูกค้า
- หมายเลขที่ 2 ปุ่มกดดูประวัติ
- หมายเลขที่ 3 ส่วนแสดงข้อมูลลูกค้าทั้งหมด
- หมายเลขที่ 4 ส่วนแสดงรายการสินค้าที่ลูกค้าสั่งซื้อ
- หมายเลขที่ 5 ปุ่มกดดูประวัติ

5. ส่วนของหน้าสั่งซื้อสินค้า

- เมื่อผู้ใช้กดปุ่มสั่งซื้อสินค้าระบบจะแสดงหน้าจอต่อไปนี้



รูปที่ ค.5: หน้าจอสั่งซื้อสินค้า

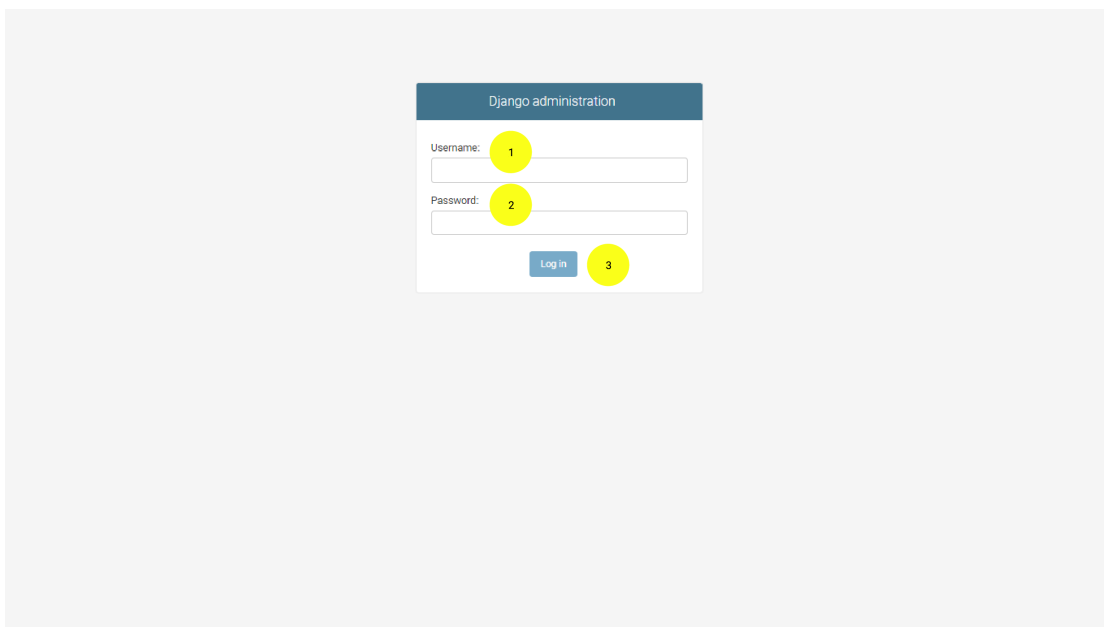
จากรูปที่ ค.5 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลขที่ 1 ปุ่มกดสำหรับรายการเมนูที่ต้องการสั่ง
- หมายเลขที่ 2 ส่วนแสดงรายละเอียดสินค้าที่สั่ง
- หมายเลขที่ 3 ปุ่มกดยืนยันการสั่งซื้อ
- หมายเลขที่ 4 ปุ่มกดยกเลิกการสั่งซื้อ
- หมายเลขที่ 5 ปุ่มกดย้อนกลับไปยังหน้าแสดงข้อมูลลูกค้า
- หมายเลขที่ 6 ส่วนแสดงชื่อพนักงาน วันที่ และเวลาในขณะนั้น
- หมายเลขที่ 7-9 ปุ่มกดสำหรับประเภทเมนู

ค.2 ส่วนผู้ดูแลระบบ

1. ส่วนของหน้าเข้าสู่ระบบสำหรับผู้ดูแลระบบในฐานข้อมูล

- เมื่อผู้ดูแลทำการสมัครสมาชิกเสร็จเรียบร้อยแล้วทำการเข้าสู่ระบบ ระบบจะแสดงหน้าจอดังต่อไปนี้



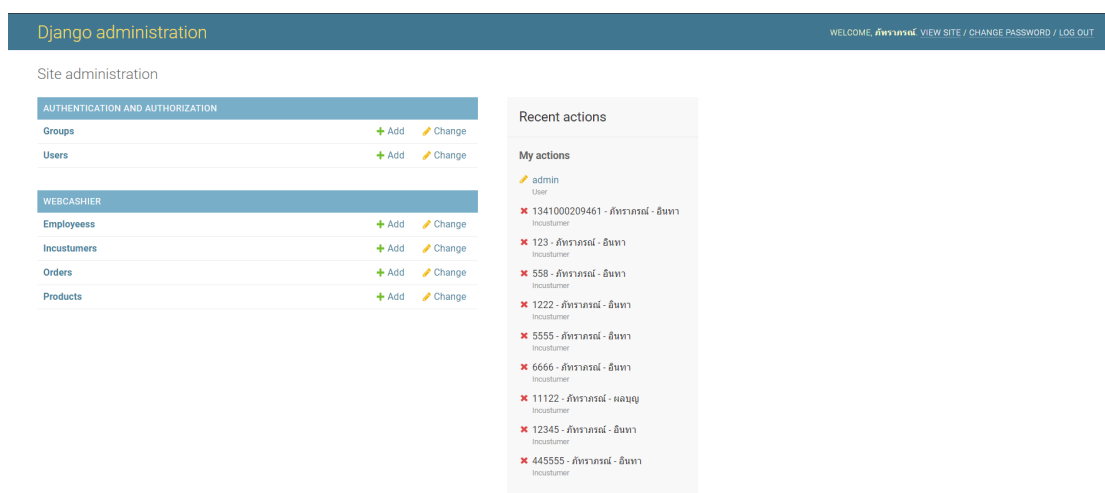
รูปที่ ค.6: หน้าจอเข้าสู่ระบบสำหรับผู้ดูแลระบบในฐานข้อมูล

จากรูปที่ ค.6 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลขที่ 1 ช่องกรอกชื่อผู้ใช้
- หมายเลขที่ 2 ช่องกรอกรหัสผ่าน
- หมายเลขที่ 3 ปุ่มกดเข้าสู่ระบบ

2. ส่วนของหน้าตารางฐานข้อมูลทั้งหมด

- เมื่อผู้ดูแลระบบเข้าสู่ระบบเรียบร้อยแล้ว โดยในที่นี่จะยกตัวอย่างข้อมูลเพียงแคในส่วนของผู้ใช้งาน รายการสั่งซื้อ และตารางข้อมูลลูกค้า ระบบจะแสดงหน้าจอดังต่อไปนี้



รูปที่ ค.7: หน้าฐานข้อมูลทั้งหมดในระบบ

3. ส่วนของหน้าจอตารางข้อมูลรายการสั่งซื้อ

- เมื่อผู้ดูแลระบบกดปุ่ม Order ระบบจะแสดงหน้าจอดังต่อไปนี้

Django administration

WELCOME, [admin](#) / [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Webcashier > Orders

Select order to change

ADD ORDER +

Action: Go 0 of 1 selected

ORDER
<input type="checkbox"/> 1-admin-2020-06-11 12:18:12+00:00

1 order

รูปที่ ค.8: หน้าจอแสดงข้อมูลร้านค้าทั้งหมด

- เมื่อผู้ใช้กดที่รายการสั่งซื้อระบบจะแสดงหน้าจอดังต่อไปนี้

Django administration

WELCOME, [admin](#) / [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Webcashier > Orders > Add order

Add order

Emp: [✎](#) [+](#)

User: [✎](#) [+](#)

List Order:

Sum Order:

Point:

Sum price:

Created at: Date: Today [📅](#)
Time: Now [🕒](#)
Note: You are 7 hours ahead of server time.

[Save and add another](#) [Save and continue editing](#) [SAVE](#)

รูปที่ ค.9: หน้าจอแสดงรายการสั่งซื้อ โดยผู้ดูแลระบบสามารถเพิ่ม ลบ แก้ไขได้

4. ส่วนของหน้าจอตารางข้อมูลข้อมูลลูกค้า

- เมื่อผู้ดูแลระบบกดปุ่ม Incustomer ระบบจะแสดงหน้าจอดังต่อไปนี้

Django administration

WELCOME, [เปลี่ยนพาสเวิร์ด](#) / [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Webcashier > Incustomers

Select incustomer to change [ADD INCUSTOMER +](#)

Action: 0 of 2 selected

<input type="checkbox"/>	FACE ID	FIRST NAME	LAST NAME
<input type="checkbox"/>	1341000209461	ปัทธพรณ์	อินหา
<input type="checkbox"/>	1339900532321	อินหา	ปัทธพรณ์

2 incustomers

รูปที่ ค.10: หน้าจอแสดงข้อมูลรายชื่อลูกค้า

- เมื่อผู้ใช้กดที่รายชื่อลูกค้า

Django administration

WELCOME, [เปลี่ยนพาสเวิร์ด](#) / [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Webcashier > Incustomers > 1341000209461 - ปัทธพรณ์ - อินหา

Change incustomer [HISTORY](#)

Face id:

First name:

Last name:

Phone:

Gender:

Age:

Email:

Career:

ImageInfo: No file chosen

รูปที่ ค.11: หน้าจอแสดงรายชื่อลูกค้าที่เป็นสมาชิก โดยผู้ดูแลระบบสามารถเพิ่ม ลบ แก้ไขได้

ประวัติผู้พัฒนา

ชื่อ-สกุล: นางสาวภัทราภรณ์ อินทา

รหัสประจำตัวนักศึกษา: 59110440309

วันเกิด: 19 ธันวาคม 2540

ที่อยู่ที่สามารถติดต่อได้: 152 ม.6 ต.โพนงาม อ.บุณฑริก จ.อุบลราชธานี 34230

เบอร์โทรศัพท์: (+66) 95 659 3170

อีเมลล์: pattaraporn.in.59.59@ubu.ac.th

ระดับมัธยมศึกษาต้น: โรงเรียน บุณฑริกวิทยาการ จังหวัด อุบลราชธานี

ระดับมัธยมศึกษาปลาย: โรงเรียน บุณฑริกวิทยาการ จังหวัด อุบลราชธานี

ระดับอุดมศึกษา: ภาควิชาคณิตศาสตร์ สถิติ และคอมพิวเตอร์ สาขาวิทยาการ คอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี