

กลอนประตูรู้จำใบหน้า

FACIAL RECOGNITION CONTROLLED DOOR



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

กลอนประตูรู้จำใบหน้า

FACIAL RECOGNITION CONTROLLED DOOR



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ประจำปี การศึกษา 2552

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง กลอนประตูรู้จำใบหน้า

FACIAL RECOGNITION CONTROLLED DOOR

ผู้จัดทำ

1. นางสาวทันยา ศรีสมบัติ รหัสนักศึกษา 57011090

2. นางสาวพัชรดา อารีย์สมาน รหัสนักศึกษา 57011567



mark ศรีบุญ

(รศ.กฤตวัน ศรีบุญ)

อาจารย์ที่ปรึกษา

บุญศรี เครือตรา

(รศ. ดร. บุญศรี เครือตรา)

อาจารย์ที่ปรึกษาร่วม

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลอนประตูล็อกจำใบหน้า

| | |
|---------------------------|----------------------|
| นางสาวทันทยา ศรีสมบัติ | 57011090 |
| นางสาวพัชรดา อารีย์สมาน | 57011567 |
| รศ.กฤตวัน ศิริบูรณ์ | อาจารย์ที่ปรึกษา |
| รศ. ดร. บุญธีร์ เครือตราฐ | อาจารย์ที่ปรึกษาร่วม |
| ปีการศึกษา 2560 | |

บทคัดย่อ

ระบบการรู้จำใบหน้า(Face Recognition) มีการใช้งานอย่างแพร่หลายมากขึ้นในยุคปัจจุบัน ทั้งบนโลกอินเทอร์เน็ต สังคมออนไลน์ ในระบบรักษาความปลอดภัย และอื่นๆอีกมากมาย โดยโครงการนี้จะประยุกต์ใช้ระบบจดจำใบหน้าในการจำลองกลอนประตูแบบสแกนหน้า โดยเมื่อสแกนใบหน้าที่ประตูแล้วเจ้าของบ้านสามารถควบคุมการเปิด - ปิดประตูระยะไกลได้ โดยเชื่อมต่อผ่านแอปพลิเคชัน ซึ่งจะมีการจดจำใบหน้าเจ้าของบ้าน แจ้งเตือนคนแปลกหน้า และสามารถบันทึกเวลาการเข้าออกได้อีกด้วย ซึ่งจะช่วยให้ความสะดวก รวดเร็วและปลอดภัย ให้กับคนในบ้าน ซึ่งระบบการรู้จำใบหน้านั้นมีซึ่งหลายวิธี ในแต่ละวิธีก็จะมีข้อจำกัดที่แตกต่างกัน เช่น สีแสง สภาพแวดล้อม คุณภาพวิดีโอ เป็นต้น ซึ่งจะทำให้ความแม่นยำแตกต่างกันออกไปเช่นกัน

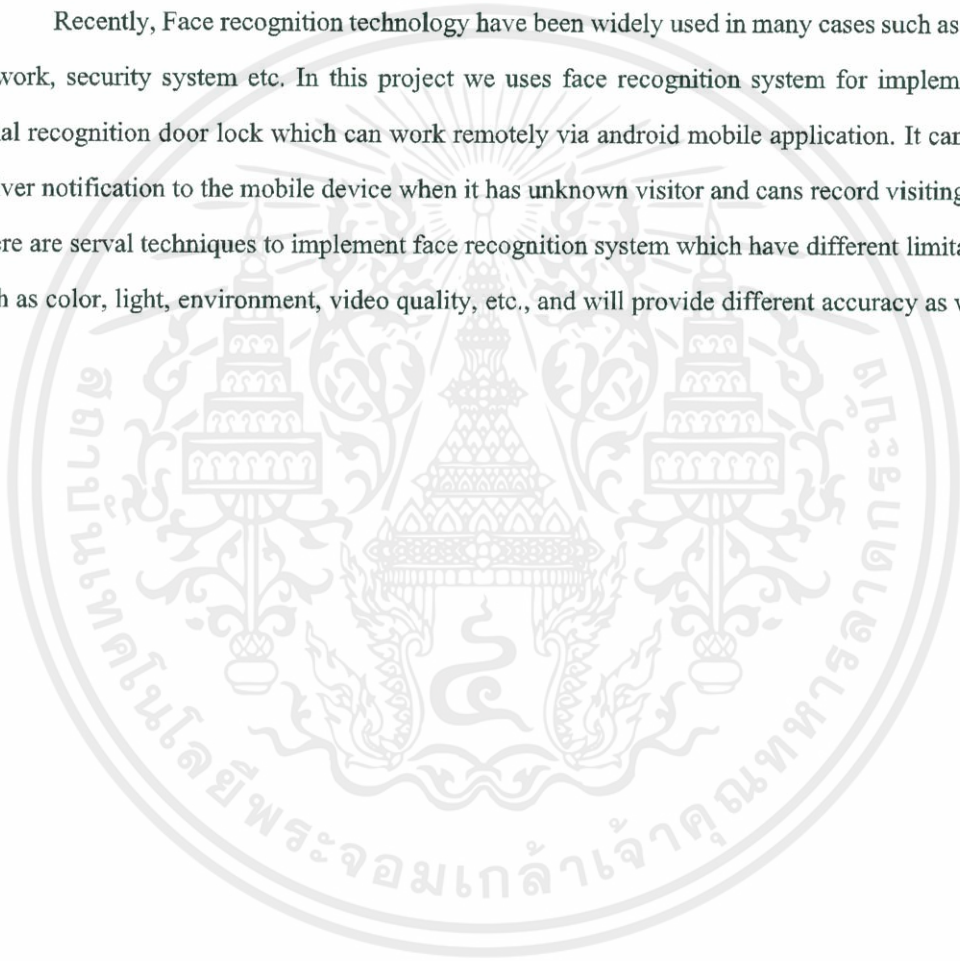
FACIAL RECOGNITION CONTROLLED DOOR

| | | |
|------------------------|-------------|------------|
| Ms. Watanya | Srisombut | 57011090 |
| Ms. Patchrida | Areesmarn | 57011567 |
| Assoc.Prof. Kritawan | Sriboon | Advisor |
| Assoc.Prof.Dr. Boontee | Kruatrachue | Co-Advisor |

Academic Year 2017

ABSTRACT

Recently, Face recognition technology have been widely used in many cases such as social network, security system etc. In this project we uses face recognition system for implementing facial recognition door lock which can work remotely via android mobile application. It cans also deliver notification to the mobile device when it has unknown visitor and cans record visiting time. There are serval techniques to implement face recognition system which have different limitations, such as color, light, environment, video quality, etc., and will provide different accuracy as well.



กิตติกรรมประกาศ

ปริญญาบัตรเล่มนี้สามารถสำเร็จลุล่วงมาได้อย่างสมบูรณ์ เนื่องด้วยจากการให้ความช่วยเหลือและให้คำปรึกษาจากอาจารย์ที่ปรึกษา รศ.กฤตวัน ศิริบูรณ์ ที่คอยชี้แนะแนวทางปฏิบัติ คอยติดตามความคืบหน้า ให้ความดูแลเอาใจใส่ และให้ความสนับสนุนที่ดีในทุกๆเรื่อง ทางคณะผู้จัดทำต้องขอขอบพระคุณอาจารย์ที่ปรึกษา รศ.กฤตวัน ศิริบูรณ์ เป็นอย่างสูง

ผู้จัดทำขอขอบพระคุณคณาจารย์ทุกท่านในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ได้ให้ความรู้ ให้คำแนะนำทั้งเรื่องการเรียนรู้ การทำงาน และการใช้ชีวิตประจำวัน ทางคณะผู้จัดทำได้รวบรวมคำสั่งสอนทุกอย่างมาบูรณาการประกอบกันจนสามารถทำปริญญาบัตรเล่มนี้ได้สำเร็จ ลุล่วงไปด้วยดี ทางคณะผู้จัดทำมีความซาบซึ้งในพระคุณของคณาจารย์ทุกท่านเป็นอย่างยิ่ง

คณะผู้จัดทำขอขอบพระคุณ เพื่อนๆที่เฝ้าระวังทุกท่านในคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้คอยให้ความช่วยเหลือ ให้คำปรึกษาในด้านต่างๆเสมอ

สุดท้ายนี้ทางคณะผู้จัดทำขอขอบพระคุณบิดา มารดา และผู้ช่วยเหลือทุกท่านที่เกี่ยวข้องทั้งที่ไม่ได้กล่าวนามไว้ ณ ที่นี้ที่คอยช่วยเหลือให้โครงการฉบับนี้สำเร็จลุล่วงได้ด้วยดี

วทันยา ศรีสมบัติ
พัชรดา อารีย์สมาน

สารบัญ

| | หน้า |
|---|------|
| บทคัดย่อภาษาไทย | I |
| บทคัดย่อภาษาอังกฤษ | II |
| สารบัญ | IV |
| สารบัญตาราง | VI |
| สารบัญภาพ | VII |
| | |
| บทที่ 1 บทนำ | 1 |
| 1.1 ความสำคัญและที่มาของโครงการ | 1 |
| 1.2 วัตถุประสงค์ของโครงการ | 1 |
| 1.3 ขอบเขตของโครงการ | 2 |
| 1.4 วิธีการดำเนินการ | 2 |
| 1.5 ประโยชน์ที่คาดว่าจะได้รับ | 2 |
| 1.6 ส่วนประกอบของปริญญานิพนธ์ | 3 |
| | |
| บทที่ 2 ทฤษฎีที่เกี่ยวข้อง | 4 |
| 2.1 ทฤษฎีที่เกี่ยวกับการทำระบบรู้จำใบหน้า | 4 |
| 2.2 ทฤษฎีที่เกี่ยวข้องด้าน Cloud | 13 |
| 2.3 ทฤษฎีที่เกี่ยวข้องด้านซอฟต์แวร์ | 19 |
| 2.4 ทฤษฎีที่เกี่ยวข้องด้านฮาร์ดแวร์ | 23 |
| | |
| บทที่ 3 การออกแบบและการพัฒนา | 25 |
| 3.1 สถาปัตยกรรมของระบบ | 25 |
| 3.2 อัลกอริทึมที่ใช้ในการประมวลผลในขั้นตอนรู้จำใบหน้า | 27 |
| 3.3 โครงสร้างของฮาร์ดแวร์ | 30 |
| 3.4 แผนภาพ UML (Unified Modeling Language Diagram) | 31 |
| 3.5 การออกแบบส่วนติดต่อผู้ใช้งาน (User Interface) | 39 |
| 3.6 การออกแบบส่วนของระบบคลาวด์ (Cloud Architecture) | 42 |

สารบัญ (ต่อ)

| | หน้า |
|---|------|
| บทที่ 4 การทดลองและผลการทดลอง..... | 44 |
| 4.1 ผลการทดลองอัลกอริทึมรู้จำใบหน้า | 44 |
| 4.2 ผลการทดลองการประมวลผลใบหน้าบนบอร์ด Raspberry pi | 48 |
| 4.3 ผลการทดลองในส่วนของประมวลผลบนคลาวด์..... | 52 |
| 4.4 การทดลองใช้แอปพลิเคชัน | 57 |
| บทที่ 5 บทสรุปและข้อเสนอแนะ..... | 65 |
| 5.1 สรุปผล..... | 65 |
| 5.2 แนวทางการพัฒนาต่อ..... | 66 |
| บรรณานุกรม..... | 67 |

สารบัญตาราง

| ตาราง | หน้า |
|--|------|
| 3.1 Use case ลงทะเบียน..... | 33 |
| 3.2 Use case เข้าสู่ระบบ/ออกจากระบบ..... | 33 |
| 3.3 Use case ดูประวัติผู้มาเยือน..... | 33 |
| 3.4 Use case ดูสมาชิก..... | 34 |
| 3.5 Use case แก้ไขข้อมูลสมาชิก..... | 34 |
| 3.6 Use case ลบสมาชิก..... | 35 |
| 3.7 Use case ปลดล็อกกลอนผ่านแอปพลิเคชัน..... | 35 |
| 3.8 Use case บันทึกใบหน้า..... | 36 |
| 3.9 Use case สแกนใบหน้า..... | 36 |
| 3.10 Use case สแกนใบหน้า..... | 36 |
| 3.11 รายละเอียด DEVICE..... | 37 |
| 3.12 MEMBER..... | 37 |
| 3.13 HISTORY..... | 37 |
| 3.14 รูปแบบของตาราง DEVICE ในฐานข้อมูล..... | 38 |
| 3.15 รูปแบบของตาราง MEMBER ในฐานข้อมูล..... | 38 |
| 3.16 รูปแบบของตาราง HISTORY ในฐานข้อมูล..... | 39 |
| 4.1 ผลการทดลองความแม่นยำของอัลกอริทึม โดยใช้ Library OpenCV..... | 44 |
| 4.2 ผลการทดลองความแม่นยำของอัลกอริทึม โดยใช้ Library OpenCV..... | 45 |

สารบัญรูป

| รูป | หน้า |
|---|------|
| 2.1 ผลที่ได้จากการใช้ทฤษฎี OpenCV Face Recognizers | 5 |
| 2.2 กระบวนการของระบบรู้จำใบหน้า..... | 5 |
| 2.3 รูปแบบของรูปเหลี่ยมสำหรับการตรวจจับ | 6 |
| 2.4 การคำนวณแบบ Integral image | 6 |
| 2.5 ขั้นตอนการรวมตัวจำแนกกลุ่มแบบต่อเรียง | 7 |
| 2.6 อัลกอริทึมต่างๆ ในขั้นตอนแยกองค์ประกอบ (Feature Extraction) | 8 |
| 2.7 การทำงานพื้นฐานของ Local Binary Pattern | 9 |
| 2.8 Local Binary Pattern ในรูปแบบที่แตกต่างกันของ (P, R) | 9 |
| 2.9 แสดงตัวอย่างใบหน้า Eigen | 11 |
| 2.10 ตัวอย่างการทำเว็บแอปพลิเคชันเชื่อมต่อกับ AWS Lambda..... | 14 |
| 2.11 ตัวอย่างการใช้งานร่วมกับ AWS S3 | 15 |
| 2.12 แนวคิดการทำงานของระบบเก็บข้อมูลแบบกระจาย | 16 |
| 2.13 หลักการทำงานของ AWS SNS | 17 |
| 2.14 การใช้งาน Amazon SNS..... | 18 |
| 2.15 โครงสร้างของระบบปฏิบัติการแอนดรอยด์..... | 19 |
| 2.16 ส่วนประกอบของแอปพลิเคชัน..... | 21 |
| 2.17 บอร์ด Raspberry Pi 3 Module B | 23 |
| 2.18 Raspberry Pi Camera Module | 24 |
| 3.1 ภาพรวมของระบบ | 25 |
| 3.2 แผนผังแสดงขั้นตอนการทำงานของระบบ | 26 |
| 3.3 ขั้นตอนการเปรียบเทียบอัลกอริทึมแบบต่างๆ | 27 |
| 3.4 ตัวอย่างของฐานข้อมูลใบหน้า AT&T Facedatabase..... | 27 |
| 3.5 ตัวอย่างของฐานข้อมูลใบหน้า Extended Yale Facedatabase B | 28 |
| 3.6 ตัวอย่างของฐานข้อมูลใบหน้า Extended Yale Facedatabase B | 28 |
| 3.7 การเตรียม Data set และ Training Set..... | 29 |
| 3.8 การทำงานของวงจรไฟฟ้า..... | 30 |
| 3.9 รายละเอียด Pin 1 ของการทำงานของวงจรไฟฟ้า..... | 31 |
| 3.10 แผนภาพ Use caseของระบบปลดล็อคกลอนแบบรู้จำใบหน้า | 32 |

สารบัญรูป (ต่อ)

| รูป | หน้า |
|--|------|
| 3.11 แผนภาพ Use case ของระบบแอปพลิเคชัน | 32 |
| 3.12 ER Diagram ของระบบปลดล็อคกลอนประตูรั้วอัตโนมัติ | 37 |
| 3.13 Flow การทำงานส่วนติดต่อผู้ใช้งาน (1) | 40 |
| 3.14 Flow การทำงานส่วนติดต่อผู้ใช้งาน (2) | 41 |
| 3.15 การออกแบบระบบคลาวด์..... | 42 |
| 3.16 การเรียกดึงข้อมูลบนฐานข้อมูลผ่านบอร์ด Raspberry pi | 42 |
| 3.17 การทำการการเรียกดึงหรือแก้ไขข้อมูลบนฐานข้อมูล | 43 |
| 3.18 การส่งข้อความผ่านเซอร์วิส AWS SNS | 43 |
| 3.19 การปลดล็อคกลอนประตูทางไกล | 43 |
| 4.1 กราฟแสดงการเปรียบเทียบของผลการทดลอง | 44 |
| 4.2 ตัวอย่างภาพที่ใช้กล้อง web-cam ของคอมพิวเตอร์โน้ตบุ๊ก..... | 45 |
| 4.3 ผลการทดลองรับอินพุทเป็นวิดีโอสตรีมบุคคลเดี่ยว | 46 |
| 4.4 ผลการทดลองรับอินพุทเป็นวิดีโอสตรีม 2 บุคคล | 46 |
| 4.5 ผลการทดลองรับอินพุทเป็นวิดีโอสตรีมที่ผิดพลาดจากการก้มเกินไป..... | 47 |
| 4.6 ผลการทดลองรับอินพุทเป็นวิดีโอสตรีมที่ผิดพลาดจากการเงยเกินไป..... | 47 |
| 4.7 ผลการทดลองรับอินพุทเป็นวิดีโอสตรีมที่ผิดพลาดจากแสงที่มีมากเกินไป | 47 |
| 4.8 ประสิทธิภาพของระบบปลดล็อคกลอนประตูรั้วอัตโนมัติ (ด้านหน้า) | 48 |
| 4.9 ประสิทธิภาพของระบบปลดล็อคกลอนประตูรั้วอัตโนมัติ (ด้านหลัง)..... | 48 |
| 4.10 ส่วนวงจรระบบปลดล็อคกลอนประตูรั้วอัตโนมัติ | 49 |
| 4.11 หน้าจอ LCD โหมดตรวจสอบใบหน้า | 49 |
| 4.12 หน้าจอแสดงเมื่อระบบทำการปลดล็อคสำเร็จ | 50 |
| 4.13 หน้าจอแสดงเมื่อระบบทำการปลดล็อคไม่สำเร็จ | 50 |
| 4.14 หน้าจอโหมดปลดล็อคด้วยรหัสผ่าน | 50 |
| 4.15 ข้อมูลในตาราง HISTORY | 50 |
| 4.16 ภาพที่ถูกรับที่กไว้นับบอร์ด Raspberry pi(ด้านขวา) และตัวอย่างภาพ(ด้านซ้าย)..... | 51 |
| 4.17 หน้าจอแสดงเมื่อระบบเมื่อทำการบันทึกใบหน้าผู้ใช้เสร็จสิ้น..... | 51 |
| 4.18 ข้อมูลในตาราง MEMBER | 51 |
| 4.19 ข้อความแจ้งเตือน | 52 |

สารบัญรูป (ต่อ)

| รูป | หน้า |
|--|------|
| 4.20 ข้อมูลในตาราง HISTORY | 52 |
| 4.21 THING ที่สร้างขึ้น | 53 |
| 4.22 ไฟล์ Certificate และ key ต่างๆ | 53 |
| 4.23 การ Publish ข้อความบนหน้า Test ของ AWS IOT | 53 |
| 4.24 ข้อความที่ได้รับบนหน้า Commands ของบอร์ด Raspberry pi | 54 |
| 4.25 functions ที่สร้างขึ้น | 54 |
| 4.26 รายละเอียดการออกแบบ Function | 54 |
| 4.27 ทดสอบการเรียกใช้ Fuction เมื่อการเรียกใช้สำเร็จ | 55 |
| 4.28 API ที่สร้างขึ้น | 55 |
| 4.29 Resources และ Methods ของแต่ละ Resource ที่สร้างขึ้น | 55 |
| 4.30 แผนผังการประมวลผลของ Method | 56 |
| 4.31 การทดสอบ API | 56 |
| 4.32 ฐานข้อมูลที่สร้างขึ้น | 56 |
| 4.33 ส่วนแสดงผลการเข้าสู่ระบบ | 58 |
| 4.34 ส่วนแสดงผลการสมัครสมาชิก | 59 |
| 4.35 ส่วนของการควบคุมทางไกล | 60 |
| 4.36 หน้าจอแสดงเมื่อระบบทำการปลดล็อคสำเร็จ | 60 |
| 4.37 ส่วนแสดงประวัติ (History) | 61 |
| 4.38 ส่วนแสดงสมาชิก | 62 |
| 4.39 ส่วนแสดงข้อมูลสมาชิกและการจัดการสมาชิก | 63 |
| 4.40 การแก้ไขข้อมูล Profile และ Logout | 64 |

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของโครงการ

ความปลอดภัยในชีวิตและทรัพย์สินจัดเป็นหนึ่งในความต้องการพื้นฐานของมนุษย์ จึงมีความพยายามในการประยุกต์ใช้เทคโนโลยีต่างๆ เข้ามาเพื่อเพิ่มประสิทธิภาพในระบบรักษาความปลอดภัย ซึ่งกลอนประตูอิเล็กทรอนิกส์ก็จัดเป็นอีกหนึ่งเทคโนโลยีอีกหนึ่งทางเลือกที่ถูกพัฒนาขึ้นมา โดยตลอดและใช้กันอย่างแพร่หลายในธุรกิจที่พักอาศัย อุตสาหกรรมโรงแรม ไปจนถึงระดับครัวเรือน ซึ่งไม่เพียงแต่ประสิทธิภาพและคุณภาพเท่านั้นที่ควรถูกคำนึงถึงในการพัฒนา ความสะดวกในการใช้งานและความสามารถนอกเหนือจากกลอนธรรมดาทั่วไปก็เช่นกัน เป็นเหตุผลให้กลอนประตูอิเล็กทรอนิกส์มีมากมายหลายรูปแบบในท้องตลาด เช่น แบบระบบเข้ารหัส ตรวจสอบลายนิ้วมือ เป็นต้น แต่นั่นก็อาจยังไม่เพียงพอต่อการรองรับความต้องการของผู้ใช้ในอนาคตอันใกล้ หรือแม้กระทั่ง ณ ตอนปัจจุบันนี้ เพราะในโลกยุคที่แนวคิดเกี่ยวกับ Internet of Things (IoT) ซึ่งถือเป็นเทคโนโลยีใหม่ กำลังเป็นเรื่องธรรมดา โดยอุปกรณ์ต่างๆ ล้วนมีโครงสร้างพื้นฐานที่สามารถเชื่อมต่อกับโลกอินเทอร์เน็ต ไม่เพียงแต่มนุษย์จะสามารถสื่อสารสั่งการควบคุมการใช้งานอุปกรณ์ผ่านทางเครือข่ายอินเทอร์เน็ต อุปกรณ์เองก็สามารถสื่อสารกันได้เช่นกัน

โครงการ “กลอนประตูรู้จำใบหน้า (Facial Recognition Controlled Door)” ถูกจัดขึ้นศึกษาและออกแบบกลอนประตูอิเล็กทรอนิกส์แบบระบบตรวจสอบใบหน้า สามารถควบคุมการปลดล็อกกลอนประตูโดยโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์ ซึ่งจะเชื่อมโยงกันผ่านบริการของคลาวด์ มีระบบแจ้งเตือนเข้าสู่โทรศัพท์มือถือพร้อมรูปภาพของบุคคลนั้นเมื่อมาเยี่ยมชม นอกจากนี้ยังสามารถตรวจประวัติสอบเวลาเข้าออกด้วยแอปพลิเคชันมือถือ

จะเห็นได้ว่าการนำเทคโนโลยี IoT เข้ามาประยุกต์ใช้กับอุปกรณ์นั้น ไม่เพียงแต่จะช่วยเพิ่มประสิทธิภาพของระบบรักษาความปลอดภัยเท่านั้น ยังช่วยเพิ่มความสะดวกและความสบายใจให้กับผู้ใช้งาน เพราะผู้ใช้งานสามารถเข้าถึงอุปกรณ์ได้มากขึ้น อีกทั้งยังมีระบบการแจ้งเตือน และยังสามารถควบคุมสั่งการได้ตลอดเวลาไม่ว่าจะอยู่ที่ไหนก็ตาม

1.2 วัตถุประสงค์ของโครงการ

- 1) เพื่อศึกษาและพัฒนาการเขียนโปรแกรมบน บอร์ด Raspberry pi
- 2) เพื่อศึกษาและพัฒนาการสร้าง Application บนระบบปฏิบัติการ Android
- 3) เพื่อสร้างระบบรักษาความปลอดภัยที่สามารถจดจำใบหน้าของเจ้าของบ้าน แจ้งเตือนคน

เปลี่ยนหน้า และสามารถบันทึกเวลาการเข้าออกได้

- 4) เพื่อสร้างระบบรักษาความปลอดภัยที่สามารถควบคุมการเปิด-ปิดประตูรวมถึงการควบคุมจากระยะไกลได้ผ่านอุปกรณ์มือถือ

1.3 ขอบเขตของโครงการ

- 1) ระบบสามารถจดจำใบหน้าของเจ้าของบ้านได้
- 2) ระบบสามารถแจ้งเตือนบุคคลเปลี่ยนหน้าทั้งข้อความอักษรและรูปภาพได้
- 3) ระบบสามารถบันทึกเวลาการเข้า-ออก ได้
- 4) ระบบสามารถควบคุมการเปิด-ปิดประตูระยะ ได้ทั้งระยะใกล้และระยะไกลโดยเชื่อมต่อผ่านแอปพลิเคชันบนมือถือแอนดรอยด์
- 5) ระบบจะรับอินพุตเป็นวิดีโอสตรีมแล้วถอดรหัส สำหรับทำการตรวจจำใบหน้า (Face Detection)
- 6) เนื่องจากแสงมีผลกระทบต่อการถอดรหัสภาพ จึงต้องการสภาพแวดล้อมในการใช้งานที่มีแสงเพียงพอ

1.4 วิธีการดำเนินการ

- 1) กำหนดขอบเขตและความสามารถของระบบ และแอปพลิเคชัน
- 2) ศึกษาค้นหาว่าข้อมูลและทศลงไลบรารีต่างๆที่เกี่ยวกับการทำระบบการรู้จำใบหน้า
- 3) ทศลงและปรับปรุงความแม่นยำของการทำระบบการรู้จำใบหน้า
- 4) ศึกษาเกี่ยวกับเครื่องมือและอุปกรณ์ต่างๆที่ใช้ในการพัฒนา แล้วนำมาออกแบบระบบ
- 5) ศึกษาการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์
- 6) ออกแบบส่วนติดต่อผู้ใช้ (User interface) ของแอปพลิเคชัน
- 7) ศึกษาวิธีการส่งข้อมูลและบันทึกข้อมูลบนคลาวด์
- 8) เริ่มพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ไปที่ละส่วนย่อยๆ
- 9) ทศสอบระบบและแก้ไขปัญหที่เกิดขึ้น

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้มีความรู้ความเข้าใจในกระบวนการทำระบบตรวจจับและจดจำใบหน้า
- 2) ได้เรียนรู้การใช้ประโยชน์จากบริการต่างๆ ที่มอยู่อย่างหลากหลายบนระบบคลาวด์
- 3) ได้มีความรู้ในการพัฒนา Application บนระบบปฏิบัติการ Android
- 4) สามารถช่วยเพิ่มประสิทธิภาพของระบบรักษาความปลอดภัย รวมไปถึงให้มีความสะดวก

รวดเร็วมากยิ่งขึ้น

1.6 ส่วนประกอบของปฏิญานิพนธ์

ปฏิญานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็นทั้งหมด 5 บท โดยมีรายละเอียดดังนี้

บทที่ 1 บทนำ กล่าวถึงความสำคัญและที่มาของโครงการ วัตถุประสงค์ของโครงการ ขอบเขตของโครงการ วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปฏิญานิพนธ์

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง โดยจะนำเสนอเกี่ยวกับทฤษฎีที่ถูกนำมาใช้ในการทำโครงการ

บทที่ 3 การออกแบบและการพัฒนา กล่าวถึงภาพรวมของระบบ (System Diagram) , ขั้นตอนการทำงานของระบบในแต่ละส่วนของ User Interface , Use Case Diagram, การออกแบบการทำงานของอุปกรณ์ฮาร์ดแวร์, การออกแบบการทำงานในส่วนของเทคโนโลยีที่ใช้พัฒนา IoT

บทที่ 4 การทดลองและผลการทดลอง โดยจะนำเสนอเกี่ยวกับการทดลองและผลการทดลอง เพื่อทดสอบประสิทธิภาพของระบบและการทดสอบผลของการนำอัลกอริทึมมาปรับใช้

บทที่ 5 บทสรุป โดยจะนำเสนอข้อสรุปของโครงการ ข้อจำกัดของโครงการ ปัญหาและอุปสรรค แนวทางการแก้ไข แนวทางในการพัฒนา และกิตติกรรมประกาศ

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ในบทนี้กล่าวถึงทฤษฎีและเทคโนโลยีที่เกี่ยวข้องกับโครงการซึ่งประกอบด้วย ฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการพัฒนา รวมไปถึงอัลกอริทึมที่เกี่ยวกับระบบการรู้จำใบหน้า

2.1 ทฤษฎีเกี่ยวกับการทำระบบรู้จำใบหน้า

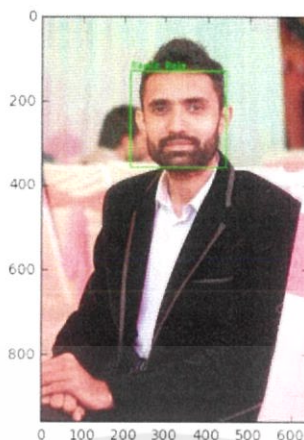
เทคโนโลยีในการระบุตัวตนสำหรับระบบปลดล็อกคลอนประตูดิจิทัลทั่วไปนั้นสามารถจำแนกออกได้เป็นสองประเภทหลักๆ คือ Biometric และ None- Biometric

- 1) Biometric คือ การผสมผสานเทคโนโลยี ทางด้านชีวภาพ และทางการแพทย์ กับเทคโนโลยีทาง คอมพิวเตอร์เข้าด้วยกัน โดยการตรวจวัด คุณลักษณะทางกายภาพ (Physical Characteristics) และ ลักษณะทาง พฤติกรรม (Behaviors) ที่เป็นลักษณะเฉพาะของแต่ละคน มาใช้ในการระบุตัวบุคคลนั้น ๆ แล้วนำสิ่งเหล่านั้น มาเปรียบเทียบกับคุณลักษณะ ที่ได้มีการบันทึกไว้ใน ฐานข้อมูล ก่อนหน้านี้ เพื่อใช้แยกแยะ บุคคลนั้นจากบุคคลอื่น ๆ โดยได้นำมาใช้ประยุกต์ใช้ในเทคโนโลยีกลอนประตูดิจิทัล เช่น ระบบรู้จำลายนิ้วมือ รู้ระบบจำใบหน้า ระบบรู้จำม่านตา เป็นต้น
- 2) None- Biometric คือ เทคโนโลยีที่อาจใช้การเข้ารหัส หรือการใช้สมาร์ทการ์ด (Smart Card) ซึ่งอาจมีประสิทธิภาพน้อยกว่าในการระบุตัวตนเมื่อเทียบกับวิธี Biometric

2.1.1 ทฤษฎี OpenCV Face Recognizers

ในปัจจุบัน OpenCV library ได้เข้ามาทำให้การพัฒนาการรู้จำใบหน้าหรือ Facial Recognition นั้นทำได้ง่ายขึ้นกว่าเดิม ซึ่งสามารถแบ่งขั้นตอนการพัฒนาการจดจำใบหน้าได้ 3 ขั้นตอนดังนี้

- 1) การรวบรวมข้อมูล (Data Gathering) รวบรวมข้อมูลใบหน้า หรือ ภาพใบหน้าของบุคคลที่ต้องการ
- 2) การฝึกฝนการจดจำ (Train the Recognizer) ป้อนข้อมูลภาพใบหน้าที่ตรงกับชื่อ ของแต่ละบุคคล เพื่อให้ recognizer สามารถเรียนรู้ได้
- 3) การจดจำ (Recognition) ป้อนภาพใบหน้าใหม่ของบุคคลเดิม และดูว่า recognizer ที่ผ่านการเรียนรู้มาแล้ว สามารถจดจำใบหน้าของแต่ละบุคคลได้หรือไม่



รูปที่ 2.1 ผลที่ได้จากการใช้ทฤษฎี OpenCV Face Recognizers

Library OpenCV มีตัวจำแนกใบหน้าอยู่ 3 ตัวดังนี้

- 1) EigenFaces – `cv2.face.createEigenFaceRecognizer()`
- 2) FisherFaces – `cv2.face.createFisherFaceRecognizer()`
- 3) Local Binary Patterns Histograms
(LBPH) – `cv2.face.createLBPHFaceRecognizer()`

ระบบรู้จำใบหน้าจะรับอินพุตซึ่งอาจอยู่ในรูปแบบภาพนิ่งหรือวิดีโอแล้วจะทำการตรวจจับใบหน้า (face detection) เพื่อให้ได้เฉพาะส่วนที่ต้องการนำมาวิเคราะห์ จากนั้นภาพที่ได้จะเข้าสู่กระบวนการรู้จำใบหน้าซึ่งจำแนกออกเป็น 3 กระบวนการหลักคือ เตรียมข้อมูล(Preprocessing), แยกคุณลักษณะของภาพใบหน้า(face extraction)และการระบุตัวตน(face identification)

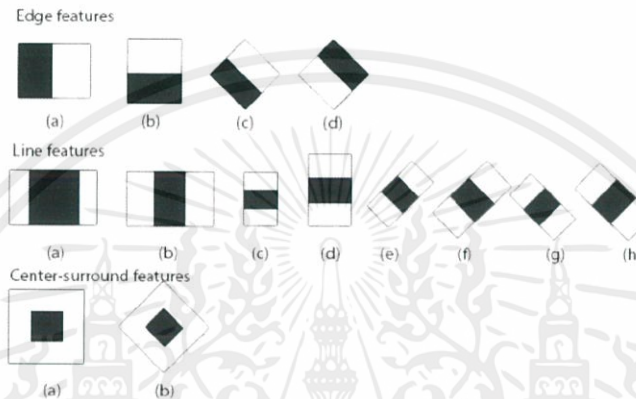


รูปที่ 2.2 กระบวนการของระบบรู้จำใบหน้า

2.1.2 การตรวจจับใบหน้า

คือกระบวนการค้นหาใบหน้าของบุคคลจากภาพหรือวิดีโอหลังจากนั้นก็ทำการประมวลผลภาพใบหน้าที่ได้สำหรับขั้นตอนถัดไปเพื่อให้ภาพใบหน้าที่ตรวจจับได้ง่ายต่อการจำแนก และ อัลกอริทึมที่ใช้ในการตรวจจับใบหน้าในปัจจุบันก็มีอยู่ด้วยกันหลายวิธีซึ่งอัลกอริทึมในการตรวจจับใบหน้าที่ดีนั้นมีส่วนช่วยในการจำแนกใบหน้าได้แม่นยำและรวดเร็วขึ้นเป็นอย่างมากซึ่ง โครงงานนี้จะใช้นี้ใช้อัลกอริทึมของ Haar-like feature

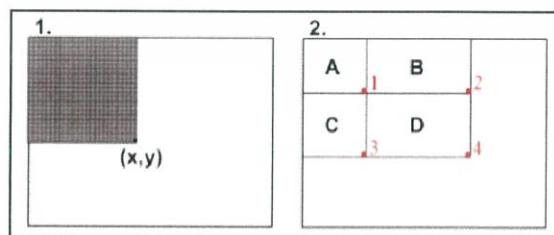
Haar-like feature เป็นวิธีการที่ใช้ตรวจจับใบหน้าที่มีความสามารถในการประมวลผลได้รวดเร็วและมีอัตราความถูกต้องสูง ของ Paul Viola และ Michael J. Jones โดยตรวจจับและตีความวัตถุภายในภาพ ด้วยวิธีการจำลองรูปแบบ Haar-like สำหรับสร้างรูปที่เหลี่ยม (Feature) ซึ่งวิธีการตรวจจับใบหน้าประกอบด้วย 3 ขั้นตอน คือ การคำนวณการจำลองรูปแบบ Haar-like ด้วยการ Integral image, การค้นหาการจำลองรูปแบบ Haar-like ด้วย Adaboost และการรวมตัวจำแนกกลุ่มแบบต่อเรียง (Cascade Classifier)



รูปที่ 2.3 รูปแบบของรูปเหลี่ยมสำหรับการตรวจจับ

จากรูปที่ 2.2 แสดงถึงผลต่างระหว่างพื้นที่ส่วนสีขาว และส่วนที่เป็นสีดำ ซึ่ง Feature สามารถเปลี่ยนแปลงขนาดและตำแหน่งได้ใช้สำหรับการตรวจจับลักษณะบนภาพแบบต่างการคำนวณค่าของ Feature นั้น ใช้หลักการคำนวณแบบ Integral image ซึ่ง Integral image คือ ผลรวมของค่าในทุก ๆ พิกเซล ที่ตำแหน่ง (x,y) ใด ๆ ซึ่งมีเวลาการทำงานรวดเร็ว ทำให้การคำนวณ Feature นั้นทำได้เร็วมาก ดังสมการที่ 2.1 และดังรูปที่ 2.4

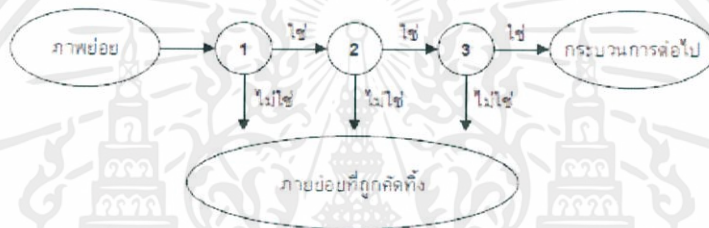
$$P(x, y) = \sum_{x' \leq x, y' \leq y} i(x' - y') \quad (2.1)$$



รูปที่ 2.4 การคำนวณแบบ Integral image

การทำ Haar-like Feature นั้น จำเป็นต้องมีภาพตัวอย่าง ซึ่งใช้ในการคัดเลือกลักษณะของรูปที่ต้องการตรวจจับและตีความหมาย ซึ่งมีสองลักษณะคือ Positive Image หรือภาพที่มีวัตถุ นั้นๆ ประกอบอยู่ในภาพ และ Negative Image หรือภาพใดๆที่ไม่มีวัตถุที่เราต้องการอยู่ภายในภาพ

ส่วน Cascade Classifiers คือ Classifier หลายตัวต่อกันเป็นลำดับดังแสดงในรูปที่ 2.5 ซึ่งเมื่อ sub-window ถูกจัดประเภทเป็น ไม่ใช่ใบหน้า (non-face) จะถูกปฏิเสธทันที แต่ในทางตรงกันข้าม ถ้า sub-window นั้น ถูกจำแนกเป็น มีโอกาสเป็น ใบหน้า (maybe-face) จะถูกส่งต่อไปยัง Classifier ตัวถัดไปตามลำดับ และกล่าวได้ว่ายังมีจำนวนชั้น ของ Classifier มากเท่าใด โอกาสที่ sub-window จะเป็นใบหน้าที่จะยังมีมากขึ้น



รูปที่ 2.5 ขั้นตอนการรวมตัวจำแนกกลุ่มแบบต่อเรียง

2.1.3 การเตรียมข้อมูล (Preprocessing)

ขั้นตอนนี้ใช้ในการเตรียมข้อมูลสำหรับการประมวลผลในขั้นตอนต่อ มีการปรับปรุงภาพสีเป็นภาพระดับสีเทา Grayscale ซึ่งคือการแปลงภาพให้มีขาว ดำ เทา โดยจะมีระดับความเข้มข้มของสีคือ 0-255 ภาพระดับสีเทา นอกจากนี้สิ่งที่จำเป็นต้องทำในขั้นตอนนี้คือลดสัญญาณรบกวนให้น้อยลง

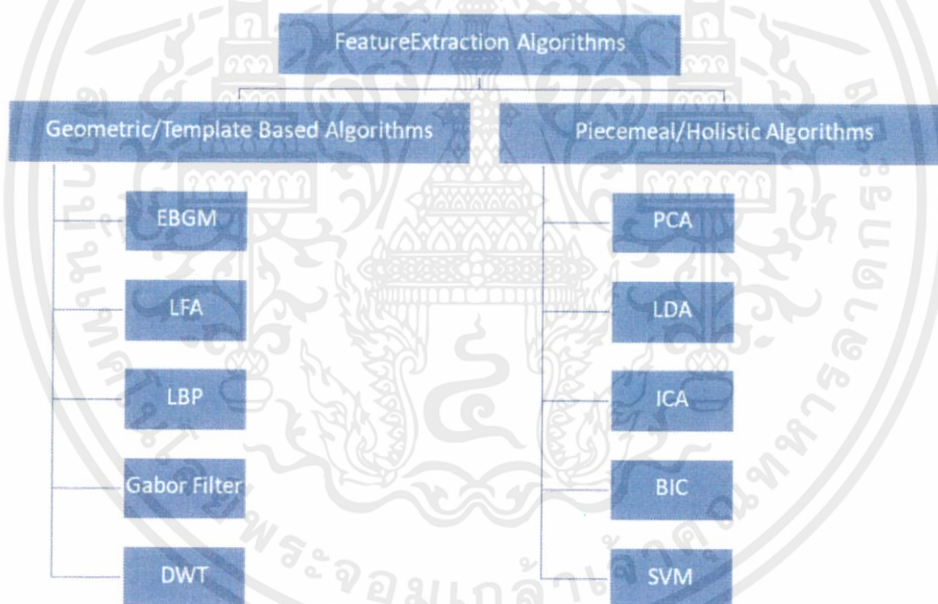
2.1.4 การดึงลักษณะเฉพาะ (feature extraction)

เป็นการดึงคุณลักษณะที่สำคัญของใบหน้า เพื่อนำไปใช้ในขั้นตอนการรู้จำ เทคนิคการดึงลักษณะเฉพาะของใบหน้าที่มีเทคนิคการทำงานที่สำคัญคือ วิธีการทางคณิตศาสตร์และแม่แบบ(Geometric/Template Based Approaches), วิธีการแบบแยกส่วนและแบบองค์รวม (Piecemeal/Holistic Approaches)

- 1) วิธีการทางคณิตศาสตร์และแม่แบบ (Geometric/Template Based Approaches) ใช้วิธีในการวิเคราะห์คุณสมบัติของใบหน้าที่ร่วมกับความสัมพันธ์ทางเรขาคณิต เช่น การใช้เทคนิค Elastic Bunch Graph Matching เป็นต้น Template-based methods การทำงานของเทคนิคนี้คือการเปรียบเทียบภาพที่ต้องการกับชุดภาพที่มีอยู่ก่อนแล้วโดย

ต้องผ่านขบวนการแปลงชุดภาพให้อยู่ในรูปของแม่แบบก่อน สามารถทำได้หลายวิธี เช่น Support Vector Machines (SVM), Principal Component Analysis (PCM), Linear Discriminant Analysis (LDA), Kernel Methods เป็นต้น

- 2) วิธีการแบบแยกส่วนและแบบองค์รวม (Piecemeal/Holistic Approaches) เกิดขึ้นจากแนวความคิดที่ว่าการระบุใบหน้าสามารถดำเนินการได้จากรายละเอียดเพียงบางส่วนหรืออีกนัยหนึ่งคือไม่มีความจำเป็นต้องเก็บคุณลักษณะของใบหน้าทุกส่วนก็สามารถระบุใบหน้าได้ จากแนวคิดดังที่ได้กล่าว นักวิจัยได้พยายามที่จะใช้การอนุมานความเกี่ยวข้องของคุณลักษณะต่างๆ บนใบหน้า เช่น บางเทคนิคพยายามที่จะใช้คุณลักษณะของดวงตาร่วมกับคุณลักษณะอื่นๆ แต่วิธีนี้ยังไม่เป็นที่นิยมมากนักเนื่องจากความจริงแล้วคุณลักษณะต่างๆบนใบหน้าล้วนสำคัญและมีความสัมพันธ์ซึ่งกันและกันนั้นเป็นเหตุที่ทำให้ในปัจจุบันหลายๆ เทคนิคยังคงใช้การรู้จำแบบองค์รวม (Holistic) มากกว่า



รูปที่ 2.6 อัลกอริทึมต่างๆ ในขั้นตอนแยกองค์ประกอบ (Feature Extraction)

ในโครงการนี้ได้ทดสอบ 3 Algorithms ในขั้นตอน เพื่อเปรียบเทียบหาวิธีที่เหมาะสมที่สุด ซึ่งได้แก่ Local Binary Pattern Histogram(LBPH), Principal Component Analysis (LBPH) หรือ EigenFaces, Linear Discriminant Analysis (LDA) หรือ Linear Discriminate (FLD)

2.1.4.1 Local Binary Pattern (LBP)

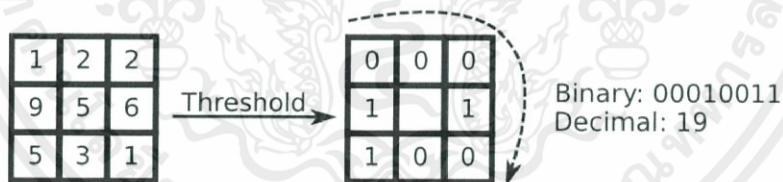
เป็นเทคนิคที่มีประสิทธิภาพมากในการในการเก็บคุณสมบัติทางด้านพื้นผิว การ

ทำงานพื้นฐานของ LBP คือคำนวณหาค่าค่าหนึ่งเพื่อใช้เป็นตัวแทนของพื้นที่ขนาด 3x3 พิกเซลโดยใช้จุดศูนย์กลางของพื้นที่เป็นค่าที่ใช้อ้างอิงในการคำนวณ ผลที่ได้จะอยู่ในรูปของแพทเทิร์นของเลขฐานสองซึ่งสามารถแปลงเป็น Histogram เพื่อแสดงคุณสมบัติทางด้านพื้นผิวของภาพได้ ต่อมา LBP ได้ถูกพัฒนาให้มีประสิทธิภาพมากขึ้นโดยวิธีปรับรูปแบบการทำงานเป็นแบบวงกลม ทำให้สามารถนำไปใช้งานได้ยืดหยุ่นมากขึ้น โดยวิธีการนี้ได้เพิ่มตัวแปรขึ้นมา 2 ตัว คือตัวแปร P แทนจำนวนตำแหน่งรอบจุดศูนย์กลางซึ่งอาจมี 8 จุด หรือ 16 จุดต่อจุดศูนย์กลาง 1 ตำแหน่งและตัวแปร R คือรัศมีระหว่างตำแหน่ง P กับจุดศูนย์กลางของวงกลม จากขบวนการทำงานข้างต้นสามารถนำมาเขียนเป็นสมการเพื่อคำนวณหาค่า LBP ของแต่ละพื้นที่ ได้ดังนี้

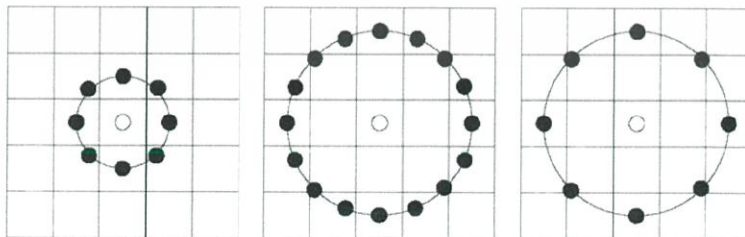
$$LBP(x_c, y_c) = \sum_{n=0}^{N-1} f(i_n - i_c) 2^n \quad (2.2)$$

เมื่อ i_c คือค่าของพิกเซลตำแหน่งศูนย์กลาง i_n คือค่าของพิกเซลรอบตำแหน่งศูนย์กลางของพื้นที่ (x_c, y_c) ใดๆ N คือจำนวนบิตในที่นี่คือ 8 และฟังก์ชัน $f(x)$ สามารถแทนค่าได้ดังนี้

$$f(x) = 1 \text{ if } x \geq 0, 0 \text{ if } x < 0 \quad (2.3)$$



รูปที่ 2.7 การทำงานพื้นฐานของ Local Binary Pattern



รูปที่ 2.8 Local Binary Pattern ในรูปแบบที่ต่างกันของ (P, R)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4.2 Principal Components Analysis (PCA) และการหาค่า Eigen face

การวิเคราะห์องค์ประกอบหลัก PCA คือ การวิเคราะห์องค์ประกอบหลักของใบหน้า ซึ่งขั้นตอนการทำงานนั้นต้องมีการเตรียมภาพเพื่อใช้ในการวิเคราะห์และทำการรู้จำใบหน้าซึ่งมีขั้นตอนการดังนี้

- 1) รับภาพอินพุท (Input images) และ ค่าเฉลี่ยของรูปภาพกำหนดให้รูปภาพใบหน้า เป็น $\tau_1, \tau_2, \dots, \tau_M$ เมื่อ M เป็นจำนวนของรูปภาพของภาพใบหน้า และ กำหนดให้ค่าเฉลี่ยของรูปภาพภาพใบหน้า มีสมการคือ

$$\varphi = \frac{1}{M} \sum_{n=1}^M \tau_n \quad (2.4)$$

เมื่อ φ คือค่าเฉลี่ยของใบหน้าฝึกสอน

M คือจำนวนภาพที่ใช้ฝึกสอน

τ_i คือเวกเตอร์ใบหน้าฝึกสอน

- 2) การดึงลักษณะเด่นออกจากภาพ ทำได้โดยหาลักษณะความแตกต่างของภาพจากภาพอินพุทของหน้าแต่ละคน กับภาพเฉลี่ยของภาพหน้าคน มีสมการคือ

$$\Phi_n = \tau_n - \varphi \quad (2.5)$$

เมื่อ Φ คือภาพผลต่างของใบหน้าเฉลี่ยกับใบหน้าฝึกสอน

- 3) การคำนวณหาค่า Covariance Matrix (Covariance Matrix Calculation) หลังจากนั้น จึงได้นำภาพความแตกต่างที่ได้ไปทำให้เป็นภาพเวกเตอร์ซึ่งมีขนาดความกว้างเท่ากับ 1 จุดภาพ คูณด้วยความกว้างของภาพ คูณด้วยความกว้างของภาพ ($1 \times (w \times h)$) จึงนำมาหา Correlation ของภาพความแตกต่าง โดยการสร้างCovariance Matrix

$$C = \frac{1}{M} [\sum_{n=1}^M \Phi_n \Phi_n^T] \quad (2.6)$$

$$C = AA^T; A = [\Phi_1 \Phi_2 \Phi_3 \dots \Phi_M] \quad (2.7)$$

เมื่อ c คือเมทริกความแปรปรวนร่วมแบบดั้งเดิม

- 4) การคำนวณหา ค่า Eigenvector (Eigenvector Calculation) เมื่อได้ Covariance Matrix มาแล้วจะทำการหาเวกเตอร์เจาะจง(Eigenvector,) และค่าเจาะจง

(Eigenvalue,) โดยวิธีการแก้ค่าตัวแปรของสมการเชิงเส้นจาก Covariance Matrix ผลที่ได้ออกมาจะเป็นภาพ eigen โดยที่ มีจำนวนเท่ากับจำนวนภาพ การสอน มีสมการคือ

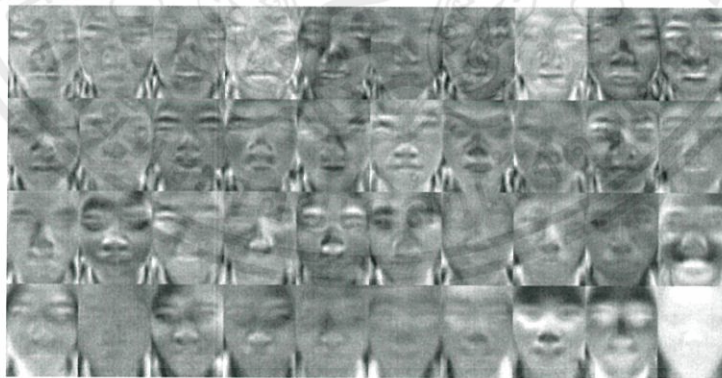
$$\Omega = [\omega_1, \omega_2, \omega_3, \dots, \omega_M] \quad (2.8)$$

เมื่อ Ω คือค่า Eigenvector (Eigenvector Calculation)

- 5) การคำนวณหาค่าเวกเตอร์น้ำหนัก (Vector of weights calculation) เมื่อภาพได้ถูกแปลงไปสู่ยังบริเวณ ของเวกเตอร์เจาะจง (Eigen space) ที่แทนลักษณะของภาพหน้าคน จากนั้นเวกเตอร์เจาะจงจะถูกนำไปใช้หาเวกเตอร์น้ำหนัก โดยสมาชิกของเวกเตอร์ของน้ำหนักหาได้จาก โดยที่ เป็นเวกเตอร์เจาะจงซึ่งจะใช้เวกเตอร์น้ำหนักนี้ ในการอธิบายลักษณะของใบหน้าแต่ละหน้าคน โดยส่วนแรกจะเก็บเวกเตอร์น้ำหนัก ของชุดข้อมูลสำหรับสอนเอาไว้ เพื่อใช้เป็นตัวเปรียบเทียบกับข้อมูลภาพที่เก็บเข้ามาเปรียบเทียบหน้าคน ซึ่งสามารถแยกแยะลักษณะของแต่ละหน้าคน

$$\omega_k = U_k^T (T_k - \varphi) \quad (2.9)$$

เมื่อ ω_k ใบหน้าไอแกนของใบหน้าฝึกสอน



รูปที่ 2.9 แสดงตัวอย่างใบหน้า Eigen

2.1.4.3 Linear Discriminant Analysis (LDA)

คือการวิเคราะห์จำแนกประเภทเชิงเส้น มีจุดประสงค์เพื่อจำแนกวัตถุต่างๆ LDA จะนิยมใช้ในการจำแนกข้อมูลและการลดมิติ LDA รู้จักกันในอีกชื่อหนึ่งคือ Fisher Linear

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Discriminate (FLD) เป้าหมายของการ LDA คือการเพิ่มปริมาณการกระจายระหว่างกลุ่มให้มากที่สุด และลดปริมาณการกระจายภายในกลุ่มให้ได้น้อยที่สุด โดย LDA แบ่งเป็นขั้นตอนการฝึกสอน และขั้นตอนการทดสอบ เช่นเดียวกับ PCA โดยมีขั้นตอนการฝึกสอนดังนี้

- 1) การกระจายตัวภายในกลุ่ม (within-class scatter matrix) แทนโดย S_W สามารถหาได้จากสมการ

$$S_W = \sum_{j=1}^C \sum_{i=1}^{N_j} (\tau_i^j - \mu_j)(\tau_i^j - \mu_j)^T \quad (2.10)$$

เมื่อ C คือ จำนวนกลุ่ม
 N_j คือ จำนวน Sample ในกลุ่มที่ j
 τ_i^j คือ Sample ที่ i ของกลุ่มที่ j
 μ_j คือ Mean ของกลุ่มที่ j

- 2) การกระจายตัวระหว่างกลุ่ม (between-class scatter matrix) แทนโดย S_B สามารถหาได้จากสมการ

$$S_B = \sum_{j=1}^C (\mu_j - \mu)(\mu_j - \mu)^T \quad (2.11)$$

เมื่อ μ คือ Mean รวม ของกลุ่มทั้งหมด

- 3) หา eigen vector ของ $S_W^{-1}S_B$ จากเมทริก W ที่มากที่สุดจากสมการ

$$J(W) = \frac{|W^T S_B W|}{|W^T S_W W|} \quad (2.12)$$

เมื่อได้ eigen vectors แล้ว จึงทำตามขั้นตอนการรู้จำใบหน้าดังกล่าวข้อที่

2.1.4.2

2.1.4.4 การระบุตัวตน(face identification)

ผลลัพธ์ของขั้นตอนการดึงลักษณะเด่น(feature extraction) จะถูกแปลงให้อยู่ในรูป เวกเตอร์ลักษณะเด่น(feature vector) ซึ่งเวกเตอร์ลักษณะเด่น(feature vector) นี้สามารถนำไปเก็บไว้ในฐานข้อมูลเพื่อนำมาเปรียบเทียบระหว่างรูปแบบของใบหน้าที่ต้องการระบุได้กับใบหน้าอื่นๆ ในฐานข้อมูลซึ่งขั้นตอนนี้เรียกว่าการระบุตัวตน(face identification) ซึ่งถ้ามีรูปแบบตรงกันก็จะอนุญาตให้ใช้ระบบตามที่ออกแบบไว้

ในขั้นตอนนี้ใช้การคำนวณระยะห่างของของเวกเตอร์ลักษณะเด่น(feature vector) ในแต่ละรูปภาพในฐานะข้อมูลว่าของภาพใดใกล้เคียงกับรูปภาพอินพุตมากที่สุด

โดยวิธีการที่จะใช้ในโครงงานนี้คือ K-nearest neighbor(K-NN) ซึ่งจะทำการทดสอบสมการการหาระยะทาง Euclidean Distance(ED)

- 1) K-nearest neighbor (K-NN) เป็นวิธีการจัดประเภท(Classification) โดยการสำรวจว่ามีวัตถุของข้อมูล(Neighbor) โดยอยู่ใกล้มากที่สุดและข้อมูลนั้นเป็นประเภทใดตามจำนวน ภายในจำนวนวัตถุ k ที่กำหนด มีประเภทของข้อมูลใดข้อมูลหนึ่งมากที่สุด อินพุตที่รับเข้ามาจะถูกจำแนกเป็นข้อมูลประเภทนั้น
- 2) Euclidean Distance(ED) คือสมการดังกล่าวเป็นการหาระยะทางในระบบพิกัดฉากสองมิติ จึงใช้ได้กับการหาระยะของข้อมูลที่มีตัวแปรเป็นจำนวน 2 ตัวเท่านั้น มีสมการดังนี้คือ

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.13)$$

2.2 ทฤษฎีที่เกี่ยวข้องด้าน Cloud

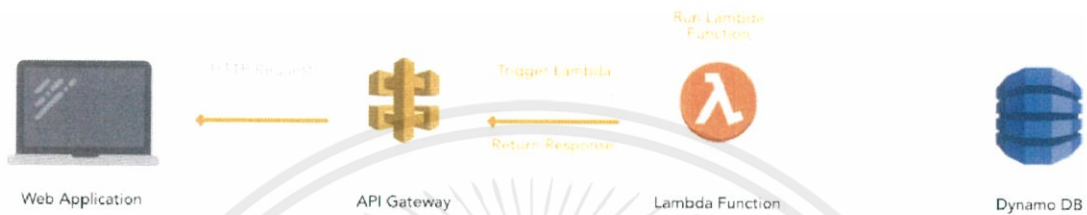
2.2.1 AWS Lambda

ในยุคปัจจุบันการสร้างเว็บใช้วิธีการรันบนเซิร์ฟเวอร์ซึ่งจำเป็นต้องรันตลอดเวลาเพื่อให้สามารถเข้าใช้งานได้ ถึงแม้จะไม่มีกรเข้าใช้งานก็ตามก็ยังคงต้องสูญเสีทรัพยากรอยู่ตลอดเวลา วิธีการทำ Serverless Application สามารถช่วยแก้ปัญหานี้ได้โดย Serverless Application คือ ไม่จำเป็นต้องมีเซิร์ฟเวอร์รันอยู่ตลอดเวลา โดยแอปพลิเคชันที่รันนั้นจะรันเฉพาะเมื่อมี Event ในการเรียกใช้งานเข้ามาเท่านั้น เช่น มีการเรียก HTTP Request เข้ามา หรือมีค่าอัปเดตทาง Data Base แล้ว Trigger เข้ามารันใช้งานแอปพลิเคชันซึ่งแน่นอนว่าวิธีการเขียนการออกแบบ และการสร้างแอปพลิเคชันก็จะแตกต่างออกไปจากเดิมค่อนข้างมาก ซึ่งทาง Amazon Web Services ก็มีเครื่องมือที่จะสามารถสร้าง Serverless Application ได้ง่ายโดยมีชื่อว่า AWS Lambda

AWS Lambda เป็นเครื่องมือที่ไว้ใช้สำหรับสร้างและรัน Serverless Application บน Amazon Web Services โดย AWS Lambda เป็นเครื่องมือกลางในการผนวก service อื่นๆของ Amazon Web Services เข้ามาใช้ในการรันแอปพลิเคชัน โดยเป็นบริการประมวลผลข้อมูลอัตโนมัติตามเหตุการณ์ (event) ที่กำหนดไว้ (เช่น ถ้าผู้ใช้ถ่ายภาพใหม่เมื่อไร ให้ Lambda นำภาพนี้จัดเก็บลงบัญชี S3)

2.2.1.1 Architecture

การรันโปรแกรมบน AWS Lambda นั้น AWS Lambda จะแปลงโค้ดที่ต้องการรัน หรือเรียกสั้นๆว่า Lambda Function เป็น Container เมื่อมีการรัน โปรแกรมผ่านช่องทางที่มีการผูก Event ไว้ ตัว lambda ก็จะทำการรันตัวโปรแกรม เมื่อทำงานจบก็จะหยุดทำงาน ถือเป็นการทำงาน 1 ครั้ง และเมื่อไม่มีการเรียกใช้งานก็จะอยู่ในสถานะ freeze คือไม่มีการรันและใช้งานทรัพยากรใดๆของตัวประมวลผล



รูปที่ 2.10 ตัวอย่างการทำเว็บแอปพลิเคชันเชื่อมต่อกับ AWS Lambda

ยกตัวอย่างการทำเว็บแอปพลิเคชันเชื่อมต่อกับ AWS Lambda ซึ่งทำตัวเป็น API ตัวหนึ่ง ก็จะเกิดการรันก็ต่อเมื่อมี Request เข้ามาที่ตัวระบบ

2.2.1.2 Lambda Function

การเขียนโปรแกรมบน AWS Lambda นั้นสามารถเขียนเป็นฟังก์ชันที่ต้องการจะรันในการเกิด Event แต่ละครั้งได้ โดยสามารถเขียนบนภาษาที่ Lambda รองรับได้ คือ Node.js, C#, Java, Python โดยเมื่อมี event เกิดขึ้น lambda จะเข้ามารันฟังก์ชันที่ได้ผูกไว้กับตัว event นั้น

2.2.1.3 Event Driven

การรัน Lambda Function ที่เขียนไว้นั้นจะเกิดขึ้นก็ต่อเมื่อมี Event มาทำการสั่งรันตัวฟังก์ชัน ซึ่งมีด้วยกันหลาย trigger โดยแบ่งเป็น 2 ชนิดหลักๆดังนี้

- 1) Streaming Event เป็นอีเวนต์ที่เข้ามาตลอดเวลาในช่วงระยะเวลาหนึ่งๆ เช่น ดึงข้อมูลจาก social network เป็นต้น โดยอีเวนต์นี้สามารถที่จะเชื่อมต่อกับเซอร์วิสของ AWS 2 ตัว คือ DynamoDB Streaming และ AWS Kinesis
- 2) Non-Streaming Event เป็นอีเวนต์ปกติที่สามารถ trigger ขึ้นมาด้วยการกระทำบางอย่าง เช่น การเรียก HTTP Request / IoT Event เป็นต้น โดยสามารถเชื่อมต่อกับ AWS Services ได้ดังเช่น API Gateway สำหรับสร้าง API ให้สามารถเรียกได้ผ่าน HTTP Request, AWS IoT, Cloudfront, Cloudwatch, S3, SNS



รูปที่ 2.11 ตัวอย่างการใช้งานร่วมกับ AWS S3

ยกตัวอย่างการใช้งานร่วมกับ S3 เมื่อมีรูปถูกอัปโหลดขึ้นมาที่ S3 ก็จะมี Event ไป trigger เพื่อรัน lambda function ไปทำการ resize รูป

2.2.2 AMAZON SIMPLE STORAGE (AMAZON S3)

Amazon S3 คือ ระบบ Object Storage หรือที่เรียกว่า Amazon Simple Storage Service (Amazon S3) ซึ่งเป็นระบบ storage ที่มีการเรียกใช้งานโดยผ่านทาง HTTP protocol ดังนั้นจะพบว่าระบบ S3 เหมาะกับการทำงานผ่านเว็บ หรือ mobile และเน้นการจัดเก็บหรือเรียกดูไฟล์ข้อมูลเป็นหลัก โดยระบบนี้จะมีความสามารถในการขยาย (highly scalable) และมีความคงทน (highly durable) ในระดับ 99.99999999 เปอร์เซ็นต์ หรือที่เรียกว่า Eleven Nine ซึ่งในการเข้าถึงข้อมูลจะใช้คอนเซ็ปต์ของ Bucket (หรือพื้นที่จัดเก็บ) และตัว Object ซึ่งก็คือ ตัวไฟล์ข้อมูล โดยแต่ละ object หรือแต่ละไฟล์จะมีขนาดใหญ่มากที่สุดได้ถึง 5 TB มีระบบรักษาความปลอดภัยและระบบ encryption ให้ผู้ใช้สามารถที่จะเลือกเปิด-ปิดได้เอง

2.2.2.1 ชนิดของข้อมูลที่เหมาะสมกับการใช้ Amazon S3

- 1) Log files ต่าง ๆ
- 2) ไฟล์เอกสาร เช่น Word, Excel, PowerPoint, PDF, Image files เป็นต้น
- 3) ไฟล์ multimedia
- 4) ไฟล์ที่ต้องการการเข้าถึงแบบ Sequential
- 5) ข้อมูล backup ที่ต้องการเก็บไว้เป็นระยะเวลานาน

2.2.2.2 ค่าใช้จ่ายของAWS S3 Storage

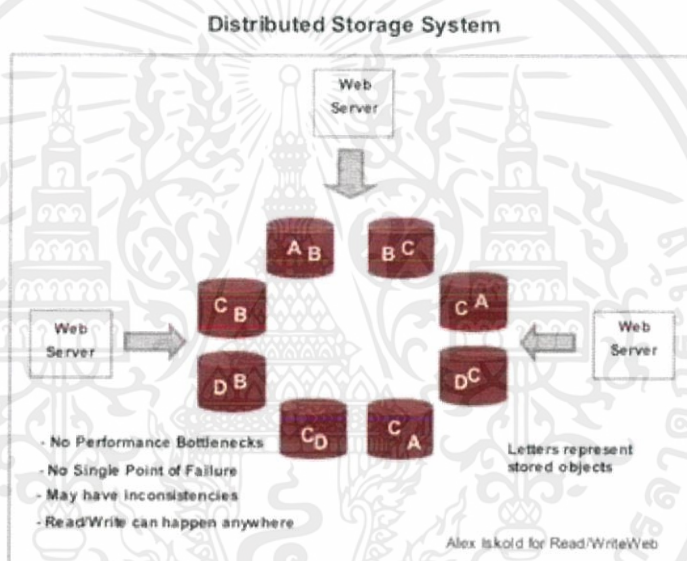
S3 มีการเก็บค่าใช้จ่ายหลักๆแบ่งเป็น 3 ส่วนคือ จำนวนครั้งในการเรียกใช้, พื้นที่ที่ใช้เก็บ และ bandwidth ที่เสียไป เช่น การ upload เข้ามา (เรียกว่า PUT) หรือ download หรือเอาไปแสดงผล (เรียกว่า GET) เหล่านี้ จะถูกคิดเงินทั้งหมด แต่สำหรับ free tier ที่ใช้งานนั้นสามารถใช้งานได้ดังนี้

- 1) 5 GB of Standard Storage
- 2) 20,000 Get Requests
- 3) 2,000 Put Requests
- 4) First 1 GB / month data transfer

คือ สามารถเก็บไฟล์ได้ 5GB ถูกโหลดได้ไม่เกิน 20000 ครั้งต่อเดือน upload/copy ไฟล์ได้ไม่เกิน 2000 ครั้งต่อเดือน และ download ผ่าน internet ได้ไม่เกิน 1GB ต่อเดือน

2.2.3 AWS Dynamo DB

AWS Dynamo DB เป็น Database service ของค่าย Amazon ที่เก็บข้อมูลแบบ Document-store เช่นเดียวกับ MongoDB และเป็นประเภทหนึ่งของ Database แบบ NoSQL โดยเป็นระบบเก็บข้อมูลที่มีความสามารถในการใช้งานสูงและสามารถขยายขนาดได้และอนุญาตให้เจ้าของบริการขยายระบบเพิ่มหรือลดลงได้ขึ้นอยู่กับภาระการใช้งานในขณะนั้น นอกจากนี้ระบบเก็บข้อมูลที่มีความถูกต้องเสมอแบบ Dynamo นั้นสามารถเป็นรากฐานที่สำคัญให้กับ application ที่มีความต้องการในการใช้งานสูงอีกด้วย



รูปที่ 2.12 แนวคิดการทำงานของระบบเก็บข้อมูลแบบกระจาย

จากรูปข้างต้นแสดงให้เห็นถึงแนวคิดการทำงานของระบบเก็บข้อมูลแบบกระจาย โดยที่ข้อมูลต่างๆจะถูกกระจายอยู่ภายในวงคอมพิวเตอร์ คอมพิวเตอร์แต่ละเครื่องนั้นจะสามารถระบุตัวตนได้ เพื่อให้สามารถรับประกันการทนต่อความผิดปกติได้ ในกรณีที่มีเครื่องใดหยุดการทำงานลง ข้อมูลจะถูกทำสำรอง ทำให้วัตถุถูกเก็บในระบบหลายครั้ง

Dynamo DB มีชื่อเรียกทางเทคนิคว่า eventually consistent storage system (ระบบเก็บข้อมูลที่ถูกต้องเสมอ) โดยเป็นการสร้างระบบเก็บข้อมูลแบบกระจายซึ่งแก้ปัญหาทั้งในเรื่องของความเร็วในการตอบสนองและการทำให้ข้อมูลตรงกันซึ่งเป็นปัญหาที่แก้ยาก จากที่เห็นในรูป หากคอมพิวเตอร์เครื่องหนึ่งทำการปรับปรุงวัตถุ A การเปลี่ยนแปลงนี้จำเป็นต้องมีการถ่ายทอดไปยังเครื่องอื่นๆ โดยจะใช้วิธีการติดต่อกันแบบไม่สอดคล้อง (Asynchronous

communication) ซึ่งเป็นที่มาของคำว่า “eventually consistent” (ถูกต้องเสมอ)

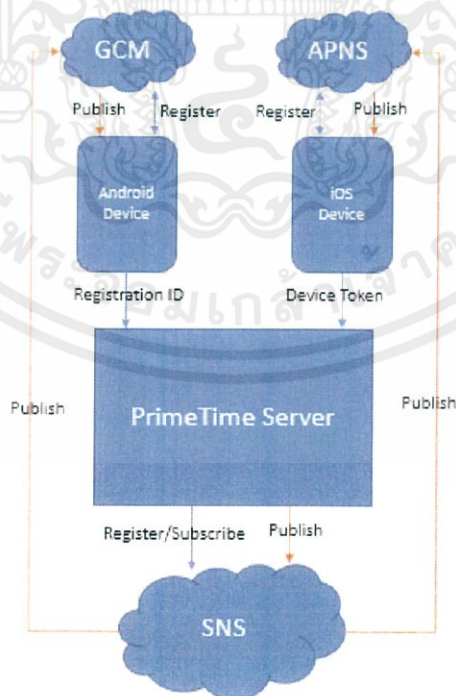
2.2.3.1 ค่าใช้จ่ายของAWS Dynamo DB

บริการ Dynamo DB คิดราคาตามโหนดที่เราต้องซื้อล่วงหน้าคิดเป็นหน่วย โดย 1 หน่วยจะหมายถึงการเขียนหรืออ่าน 1KB หรือเล็กกว่าต่อวินาที ราคาสำหรับการจองประสิทธิภาพของ Dynamo DB คือ 0.01 ดอลลาร์ต่อชั่วโมงสำหรับการเขียน หน่วยและ 0.01 ดอลลาร์ต่อชั่วโมงสำหรับการอ่าน 10 หน่วย นอกจากนี้ยังมีค่าเก็บข้อมูลอีก 1 ดอลลาร์ต่อกิกะไบต์ต่อเดือน และค่าขนถ่ายข้อมูล

2.2.4 Amazon SNS

Amazon Simple Notification Service (Amazon SNS) เป็นเว็บเซอร์วิสที่ช่วยให้ตั้งค่า ใช้งาน และส่งการแจ้งข้อมูลจากระบบคลาวด์ได้ง่ายๆ มีความสามารถที่จะรองรับการปรับขยายได้ดี มีความยืดหยุ่น และคุ้มค่าสำหรับนักพัฒนาที่จะใช้เผยแพร่ข้อความจากแอปพลิเคชันและจัดส่งไปยังสมาชิกหรือแอปพลิเคชันอื่นๆ ได้ทันที

โดยไม่ใช้ส่งแค่ push notification แต่ยังสามารถส่งไปยังช่องทางอื่นๆ เช่น email, SMS เป็นต้น หลักการของ SNS นั้นคือให้เราสร้าง Topic แล้ว subscribe แต่ละ device ไปที่ Topic นั้น จากนั้นเมื่อเราจะส่งค่าไปยัง device ต่างๆ ก็เพียงแค่ส่ง message ไปยัง Topic เดียว แล้วทุก device ที่ subscribe ไว้ก็จะได้รับข้อความกันทั้งหมดโดยไม่ต้องเสียพลังงานบน server ของเราเอง วน loop 10 device เลย



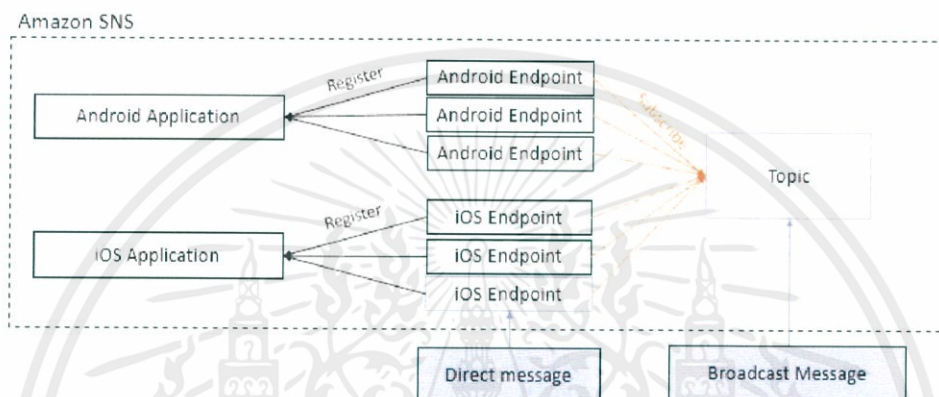
รูปที่ 2.13 หลักการทำงานของ AWS SNS

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4.2 การใช้งาน

ในฝั่งของ SNS นั้นเราต้องเข้าไปสร้างสิ่งเหล่านี้เตรียมไว้

- 1) Application แยกออกเป็น 2 ตัว คือ Android โดยใช้ API Key ของแอปพลิเคชัน และ iOS โดยใช้ P12 ไฟล์
- 2) Topic สำหรับให้ device มา subscribe ไว้ หากจะแยกกรุปก็สามารถสร้างหลายๆ Topic ได้



รูปที่ 2.14 การใช้งาน Amazon SNS

เมื่อ client ลงทะเบียนกับ Apple Push Notification Service (APNS) หรือ Google Cloud Messaging for Android (GCM) สำเร็จแล้วก็ส่งค่า Device token จาก APNS หรือ Registration ID จาก GCM มายัง server จากนั้น server จะนำ token/id มา register กับ Application บน SNS จะได้เป็น Endpoint และก็นำ Endpoint ไป subscribe กับ Topic ของ SNS

จากนั้นหากจะส่งข้อความไปยัง Device ก็สามารถส่งไปยัง Endpoint บน SNS ได้เลย ส่วนถ้าจะส่งแบบ Broadcast ไปหลายเครื่องก็ส่งไปที่ Topic ซึ่งทุก Endpoint ที่ subscribe ไว้กับ Topic นี้จะ ได้รับ message เหมือนกันทั้งหมด การส่ง Push notification ผ่าน SNS ช่วยให้เราประหยัดทรัพยากรไปได้มาก โดยไม่ต้องกังวลกับจำนวน Device ที่เพิ่มขึ้น

2.2.5 AWS API gateway

เป็น HTTP REST Service ที่สร้าง Response โดยการรวบรวมข้อมูลจาก Core Services ต่างๆ ตามที่ API Client ขอ เพื่อลดภาระและระยะเวลาในการส่งข้อมูลหลายๆครั้งของ Client

ประโยชน์อีกอย่างของ API Gateway คือ Frontend/Mobile Application Developer สามารถกำหนดและแก้ไขหน้าตาของ Response ได้เอง ไม่จำเป็นต้องรอทีม Core Services ทำให้ トラバไตที่ยังขอข้อมูลจาก Core Services ได้

เนื่องจากยังอยู่ในขั้นตอนการเปลี่ยนแปลง การส่งข้อมูลกันระหว่าง API Gateway และ

Core Services/Monolithic Web จะมี 2 แบบคือ

- 1) Simple HTTP REST— ใช้ส่งข้อมูลระหว่าง API Gateway และ Monolithic Web เดิม จะมี overhead ในการติดต่อขอข้อมูล
- 2) RPC ด้วย GRPC— ใช้ส่งข้อมูลระหว่าง API Gateway และ Core Services ใหม่ๆ ที่ ถูกแยกออกมาจาก Monolithic App เดิม เพื่อลด overhead ของแบบแรก

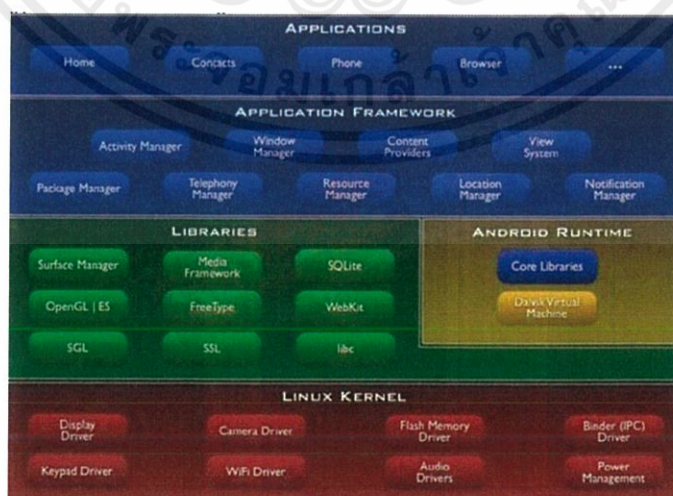
2.3 ทฤษฎีที่เกี่ยวข้องด้านซอฟต์แวร์

2.3.1 ระบบปฏิบัติการแอนดรอยด์

ระบบปฏิบัติการแอนดรอยด์ เป็นระบบปฏิบัติการที่พัฒนามาจากการนำเอา แกนกลางของระบบปฏิบัติการลินุกซ์ (Linux Kernel) ซึ่งเป็นระบบปฏิบัติการที่ออกแบบมาเพื่อทำงานเป็นเครื่องให้บริการ (Server) มาพัฒนาต่อ เพื่อให้กลายเป็นระบบปฏิบัติการบนอุปกรณ์พกพา (Mobile Operating System) โดยเป็นระบบปฏิบัติการแบบเปิดเผยแพร่ซอฟต์แวร์ต้นฉบับ (Open Source) โดยบริษัท กูเกิล (Google Inc.) ที่ได้รับความนิยมเป็นอย่างสูง เนื่องจากอุปกรณ์ที่ใช้ระบบปฏิบัติการแอนดรอยด์ มีจำนวนมาก อุปกรณ์มีหลากหลายระดับ หลากราคา รวมทั้งสามารถทำงานบนอุปกรณ์ที่มีขนาดหน้าจอ และความละเอียดแตกต่างกันได้ ทำให้ผู้บริโภคสามารถเลือกได้ตามต้องการ

2.3.1.1 โครงสร้างของแอนดรอยด์

การทำความเข้าใจโครงสร้างของระบบปฏิบัติการแอนดรอยด์ ถือว่าเป็นสิ่งสำคัญ เพราะถ้านักพัฒนาโปรแกรม สามารถมองภาพโดยรวมของระบบได้ทั้งหมด จะทำให้สามารถเข้าใจถึงกระบวนการทำงานได้ดียิ่งขึ้น และสามารถนำไปช่วยในการออกแบบโปรแกรมที่ต้องการพัฒนา เพื่อให้เกิดประสิทธิภาพในการทำงาน



รูปที่ 2.15 โครงสร้างของระบบปฏิบัติการแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโครงสร้างของระบบปฏิบัติการแอนดรอยด์ จะสังเกตได้ว่า มีการแบ่งออกมาเป็นส่วนๆ ที่มีความเกี่ยวเนื่องกัน โดยส่วนบนสุดจะเป็นส่วนที่ผู้ใช้งานทำการติดต่อโดยตรงซึ่งก็คือส่วนของ (Applications) จากนั้นก็จะลำดับลงมาเป็นองค์ประกอบอื่นๆ ตามลำดับ และสุดท้ายจะเป็นส่วนที่ติดต่อกับอุปกรณ์โดยผ่านทาง Linux Kernel

โครงสร้างของแอนดรอยด์ สามารถอธิบายเป็นส่วนๆ ได้ดังนี้

- 1) Applications ส่วน Application หรือ ส่วนของ โปรแกรมที่มีมากับระบบปฏิบัติการ หรือเป็นกลุ่มของ โปรแกรมที่ผู้ใช้งานได้ทำการติดตั้งไว้ โดยผู้ใช้งานสามารถเรียกใช้โปรแกรมต่างๆ ได้โดยตรง ซึ่งการทำงานของแต่ละโปรแกรมจะเป็นไปตามที่ผู้พัฒนาโปรแกรมได้ออกแบบและเขียนโค้ดโปรแกรมเอาไว้
- 2) Application Framework เป็นส่วนที่มีการพัฒนาขึ้นเพื่อให้นักพัฒนาสามารถพัฒนาโปรแกรมได้สะดวก และมีประสิทธิภาพมากยิ่งขึ้น โดยนักพัฒนาไม่จำเป็นต้องพัฒนาในส่วนที่มีความยุ่งยากมากๆ เพียงแค่ทำการศึกษาถึงวิธีการเรียกใช้งาน Application Framework ในส่วนที่ต้องการใช้งาน แล้วนำมาใช้งาน ซึ่งมีหลายกลุ่มด้วยกัน ตัวอย่างเช่น
 - Activities Manager เป็นกลุ่มของชุดคำสั่งที่จัดการเกี่ยวกับวงจรการทำงานของหน้าต่างโปรแกรม(Activity)
 - Content Providers เป็นกลุ่มของชุดคำสั่ง ที่ใช้ในการเข้าถึงข้อมูลของโปรแกรมอื่น และสามารถแบ่งปันข้อมูลให้โปรแกรมอื่นเข้าถึงได้
 - View System เป็นกลุ่มของชุดคำสั่งที่เกี่ยวกับการจัดการโครงสร้างของหน้าจอที่แสดงผลในส่วนที่ติดต่อกับผู้ใช้งาน (User Interface)
 - Telephony Manager เป็นกลุ่มของชุดคำสั่งที่ใช้ในการเข้าถึงข้อมูลด้านโทรศัพท์ เช่นหมายเลขโทรศัพท์ เป็นต้น
 - Resource Manager เป็นกลุ่มของชุดคำสั่ง ในการเข้าถึงข้อมูลที่เป็นข้อความ, รูปภาพ
 - Location Manager เป็นกลุ่มของชุดคำสั่งที่เกี่ยวกับตำแหน่งทางภูมิศาสตร์ที่ระบบปฏิบัติการได้รับค่าจากอุปกรณ์
 - Notification Manager เป็นกลุ่มของชุดคำสั่งที่จะถูกเรียกใช้เมื่อ โปรแกรมต้องการแสดงผลให้กับผู้ใช้งาน ผ่านทางแถบสถานะ(Status Bar) ของหน้าจอ
- 3) Libraries เป็นส่วนของชุดคำสั่งที่พัฒนาด้วย C/C++ โดยแบ่งชุดคำสั่งออกเป็น

กลุ่มตามวัตถุประสงค์ของการทำงาน เช่น Surface Manage จัดการเกี่ยวกับการแสดงผล, Media Framework จัดการเกี่ยวกับการการแสดงผลและเสียง, Open GL | ES และ SGL จัดการเกี่ยวกับภาพ 3มิติ และ 2มิติ, SQLite จัดการเกี่ยวกับระบบฐานข้อมูล เป็นต้น

- 4) Android Runtime จะมี Dalvik Virtual Machine ที่ถูกออกแบบมา เพื่อให้ทำงานบนอุปกรณ์ที่มี หน่วยความจำ(Memory), หน่วยประมวลผลกลาง (CPU) และพลังงาน(Battery)ที่จำกัด ซึ่งการทำงานของ Dalvik Virtual Machine จะทำการแปลงไฟล์ที่ต้องการทำงาน ไปเป็นไฟล์ .DEX ก่อนการทำงาน เหตุผลก็เพื่อให้มีประสิทธิภาพเพิ่มขึ้นเมื่อใช้งานกับ หน่วยประมวลผลกลางที่มีความเร็วไม่มาก ส่วนต่อมาก็คือ Core Libraries ที่เป็นส่วนรวบรวมคำสั่งและชุดคำสั่งสำคัญ โดยถูกเขียนด้วยภาษาจาวา (Java Language)
- 5) Linux Kernel เป็นส่วนที่ทำหน้าที่หัวใจสำคัญ ในจัดการกับบริการหลักของระบบปฏิบัติการ เช่น เรื่องหน่วยความจำ พลังงาน ติดต่อกับอุปกรณ์ต่างๆ ความปลอดภัย เครือข่าย โดยแอนดรอยด์ได้นำเอาส่วนนี้มาจาก ระบบปฏิบัติการลินุกซ์ รุ่น 2.6 (Linux 2.6. Kernel) ซึ่งได้มีการออกแบบมาเป็นอย่างดี

2.3.2 แอปพลิเคชันแอนดรอยด์



รูปที่ 2.16 ส่วนประกอบของแอปพลิเคชัน

2.3.2.2 Activity

คือ ส่วนของหน้า User Interface หนึ่งหน้าของAndroid ซึ่งรวมไปจนถึงการจัดการต่างๆ ภายในหน้านั้นๆ ระหว่างUser กับตัว Application เช่นการควบคุม Button หรือ View ต่างๆ ของหน้า User Interface ที่ได้กำหนดไว้ใน Activity โดย User Interface อาทิ

เช่น หน้าโทรออก, หน้าแสดงเว็บไซต์, หน้าแสดง Contact List,และอื่นๆ

2.3.2.3 Service

คือส่วนการทำงานที่ไม่มีหน้าจอที่ติดต่อกับผู้ใช้ service นั้นจะทำงานอยู่ในส่วน
ของ background เช่น โปรแกรมเล่นเพลงต่างๆ ก็จะมีหน้าจอที่ติดต่อกับผู้ใช้ (นั่นคือส่วน
ของ activities) และเมื่อผู้ใช้เลือกเพลงเสร็จแล้วกดเล่น ไฟล์เพลงก็จะถูกเล่นโดยมีการ
ทำงานแบบ services หลีกจากผู้ใช้กด Back หรือ Home หน้าจอของเครื่องเล่นเพลง
(activities) ก็จะถูกเก็บไป แต่ในส่วนของservices ที่เล่นเพลงนั้นก็ยังคงเล่นเพลงต่อไป

2.3.2.4 Broadcast Receiver

คือส่วนที่จะรับเอา broadcast ต่างๆ มาทำงาน หรือส่ง broadcast นั้นต่อไป การ
broadcast ที่ว่านี้ส่วนใหญ่จะเป็นของระบบเองที่ทำการ broadcast มา เช่น เมื่อเครื่องถูก
ชาร์จไฟ หรือมีการโทรเข้า หรือมีการรับข้อความ หรือจำนวนแบตเตอรี่ลดลง ถ้ามี
เหตุการณ์ต่างๆ พวกนี้เกิดขึ้น ระบบก็จะทำการส่ง broadcast ไปให้รู้โดยทั่วกัน และถ้าใน
โปรแกรมเราต้องการนำค่าต่างๆ ที่ระบบbroadcast นั้นมาใช้งาน เราก็จะสร้าง Broadcast
receivers นี้แหละขึ้นมารับเอาข้อมูลไปทำงาน หรือนอกจากนี้ก็ยังสามารส่ง broadcast ที่
สร้างขึ้นมาจากที่ระบบมีไว้ก่อนหน้าและส่งไปได้ด้วย.

2.3.2.5 Content Provider

เป็นส่วนของการจัดการข้อมูลต่างๆ ที่ถูกแชร์กันในระบบ ไม่ว่าข้อมูลนั้นจะอยู่ที่
ไหนก็ตามจะเป็นไฟล์ของระบบ ใน database ที่อยู่ในระบบ หรือจะเป็นข้อมูลที่ถูกเก็บไว้
ในเว็บ และสามารถที่จะแก้ไขข้อมูลต่างๆนั้นได้ ถ้า content provider นั้นให้สิทธิ
ยกตัวอย่างเช่น ในระบบของ Android นั้นจะมี content providers ที่เห็นได้ชัดอยู่ตัวหนึ่ง
คือ content providers เพื่อจัดการข้อมูลของรายชื่อในโทรศัพท์ ทั้งนี้ก็เพื่อให้ app ที่เรา
เขียนขึ้นใช้งานข้อมูลดังกล่าวได้ผ่าน content provider นั้น นอกจากนี้ content provider ก็
ยังสามารถจัดการข้อมูลที่ไม่ได้ทำการแชร์ไว้ แต่ในโปรแกรมเราเองได้อีกด้วย

2.3.3 ภาษาไพทอน

Python เป็นภาษาเขียนโปรแกรมระดับสูงที่ใช้กันอย่างกว้างขวางในการเขียนโปรแกรม
สำหรับวัตถุประสงค์ทั่วไป ภาษา Python นั้นสร้างโดย Guido van Rossum และถูกเผยแพร่ครั้งแรก
ในปี 1991 Python นั้นเป็นภาษาแบบ interpret ที่ถูกออกแบบโดยมีปรัชญาที่จะทำให้โค้ด
อ่านได้ง่ายขึ้น และโครงสร้างของภาษานั้นจะทำให้โปรแกรมเมอร์สามารถเข้าใจแนวคิดการ
เขียนโค้ดโดยใช้บรรทัดที่น้อยลงกว่าภาษาอย่าง C++ และ Java ซึ่งภาษานั้นถูกกำหนดให้มี
โครงสร้างที่ตั้งใจให้การเขียนโค้ดเข้าใจง่ายทั้งในโปรแกรมเล็กไปจนถึงโปรแกรมขนาดใหญ่

Python นั้นมีคุณสมบัติเป็นภาษาเขียนโปรแกรมแบบไดนามิกส์และมีระบบการจัดการ

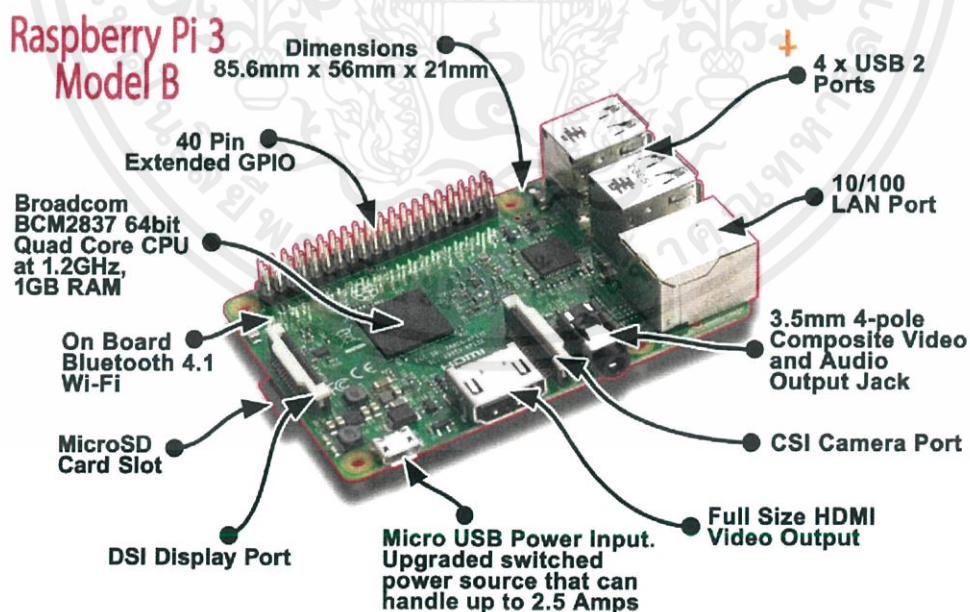
หน่วยความจำอัตโนมัติและสนับสนุนการเขียนโปรแกรมหลายรูปแบบ ที่ประกอบไปด้วย การเขียนโปรแกรมเชิงวัตถุ imperative การเขียนโปรแกรมแบบฟังก์ชัน และการเขียนโปรแกรมแบบขั้นตอน มันมีไลบรารีที่ครอบคลุมการทำงานอย่างหลากหลาย

ตัวแปรในภาษา Python นั้นมีให้ใช้ในหลายระบบปฏิบัติการ ทำให้โค้ดของภาษา Python สามารถรันในระบบต่างๆ ได้อย่างกว้างขวาง CPython นั้นเป็นการพัฒนาในตอนต้นของ Python ซึ่งเป็นโปรแกรมแบบ open source และมีชุมชนสำหรับเป็นต้นแบบในการพัฒนา เนื่องจากมันได้มีการนำไปพัฒนากระจายไปอย่างหลากหลาย variant CPython นั้นจึงถูกจัดการโดยองค์กร ไม่แสวงหาผลกำไรอย่าง Python Software Foundation

2.4 ทฤษฎีที่เกี่ยวข้องทางด้านฮาร์ดแวร์

2.4.1 บอร์ด Raspberry Pi 3 Module B

คือบอร์ดคอมพิวเตอร์ขนาดเล็กรองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) Pidora (Fedora) และ Arch Linux เป็นต้น โดยในโครงการนี้จะใช้ระบบปฏิบัติการ minbian ซึ่งถือเป็น Raspbian ที่ตัดความสามารถบางประการออกจึงทำให้ใช้ทรัพยากรหน่วยความจำน้อยกว่า บอร์ด Raspberry Pi ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ



รูปที่ 2.17 บอร์ด Raspberry Pi 3 Module B

รุ่น 3 Module B นี้มีคุณสมบัติดังนี้คือ

- 1) Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 2) 1GB RAM
- 3) BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 4) 40-pin GPIO
- 5) 4 USB 2 ports
- 6) 4 Pole stereo output and composite video port
- 7) Full size HDMI
- 8) CSI camera port for connecting a Raspberry Pi
- 9) DSI display port for connecting a Raspberry Pi touchscreen display
- 10) Micro SD port for loading your operating system and storing data
- 11) Upgraded switched Micro USB power source up to 2.5A

2.4.2 Raspberry Pi Camera Module

Raspberry Pi Camera Module เป็น โมดูลกล้องสำหรับต่อใช้งานร่วมกับบอร์ด Raspberry Pi ขนาดความละเอียด 5 ล้าน pixel สามารถถ่ายวิดีโอระดับ HD ที่ความละเอียด 1080p, 720p และ 640x480 ด้วยอัตราแสดงผล 30 (1080p), 60 (720p และ 640x480) และ 90 (640x480) เฟรมต่อวินาที



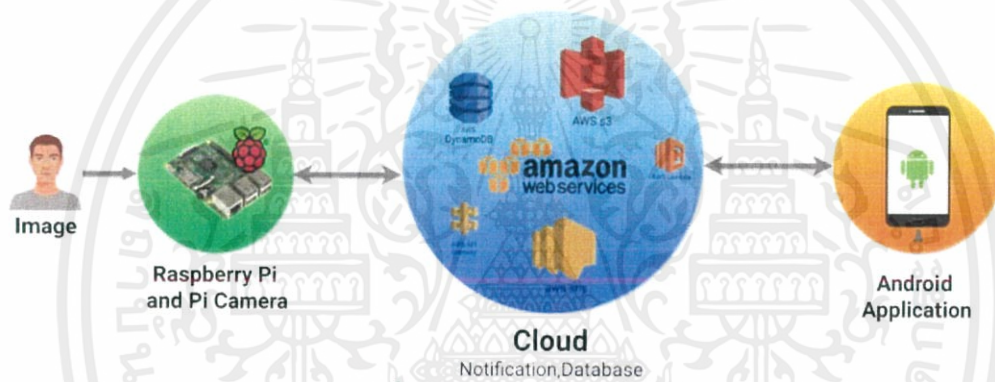
รูปที่ 2.18 Raspberry Pi Camera Module

บทที่ 3

การออกแบบและการพัฒนา

ในบทที่แล้วได้นำเสนอเรื่องทฤษฎีที่เกี่ยวข้องกับโครงงาน ในบทนี้จะนำเสนอเรื่องการออกแบบและการพัฒนาระบบ ซึ่งจะกล่าวถึงสถาปัตยกรรมของระบบ การออกแบบแผนภาพ UML การออกแบบ Use Case Diagram การออกแบบการทำงานของอุปกรณ์ฮาร์ดแวร์ การออกแบบการทำงานในส่วนของเทคโนโลยีที่ใช้พัฒนา IoT และการออกแบบส่วนติดต่อผู้ใช้งาน (User Interface)

3.1 สถาปัตยกรรมของระบบ

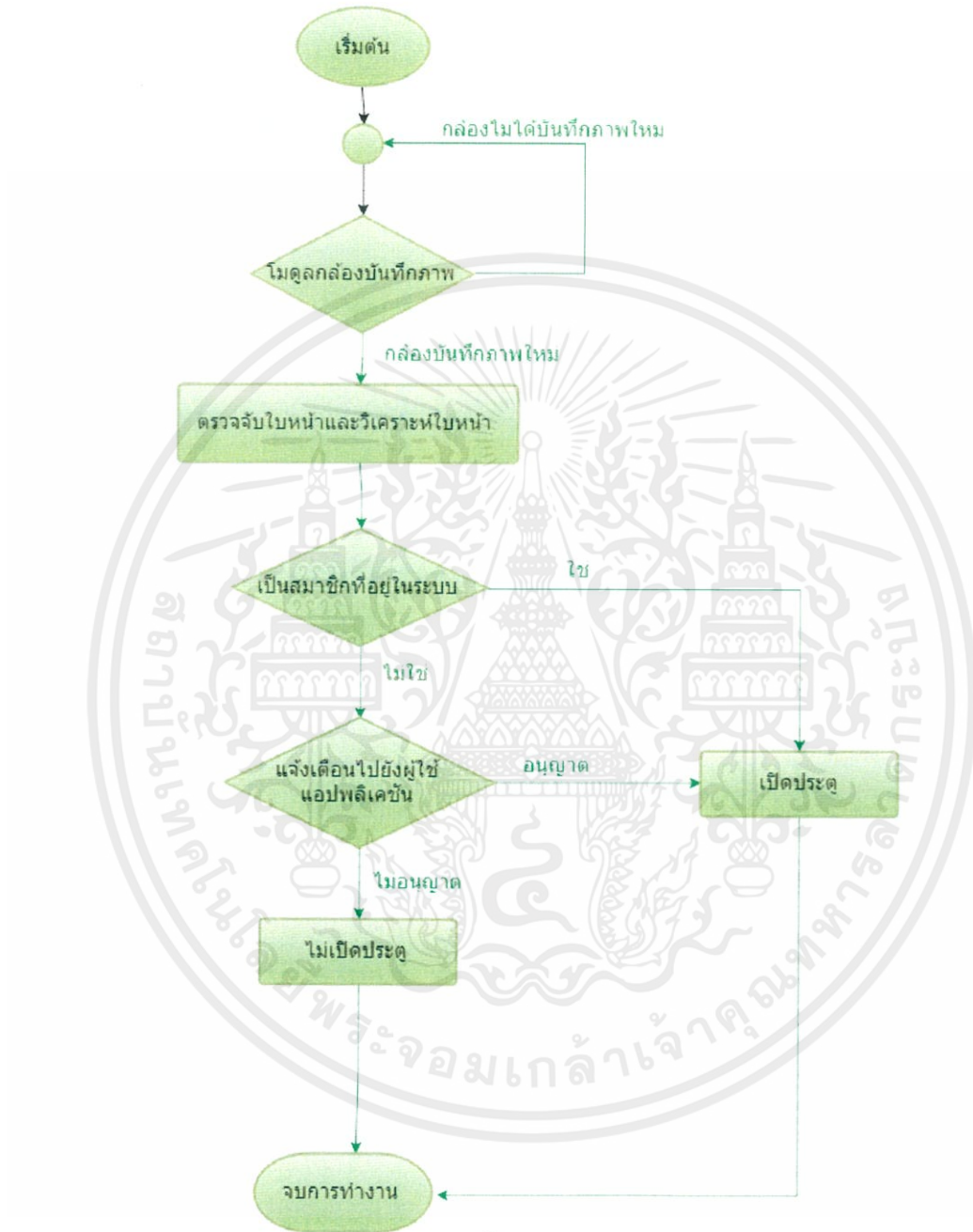


รูปที่ 3.1 ภาพรวมของระบบ

จากรูปที่ 3.1 แสดงการทำงานออกซึ่งแบ่งได้เป็น 3 ส่วนหลัก คือส่วนการทำ Face Recognition ส่วนการส่งข้อความและบันทึกผลผ่านเทคโนโลยีคลาวด์ และส่วนการพัฒนาแอปพลิเคชันบนอุปกรณ์แอนดรอยด์ โดยมีระบบการทำงานดังนี้ เริ่มจาก โมดูลกล้องจะทำการรับภาพใบหน้าของผู้ใช้แล้ว จะนำไปประมวลผลตรวจจับใบหน้าและวิเคราะห์ใบหน้า ว่าตรงกับใบหน้าที่ได้มีการเทรนเก็บไว้ก่อนหน้าในฐานข้อมูลหรือไม่ หลังจากที่ Raspberry pi ได้ประมวลผลภาพใบหน้าที่รับเข้ามาเสร็จสิ้น ถ้าผลที่ได้ตรงกับรหัสบุคคลที่ได้ทำการบันทึกไว้ กลอนประตูจะปลดล็อกระบบก็จะส่งสัญญาณ ไปแจ้งเตือนที่แอปพลิเคชันผ่านคลาวด์ แต่ถ้าพบว่า ไม่มีข้อมูลของผู้ใช้งานอยู่ในฐานข้อมูลหรือไม่ได้ทำการบันทึกไว้ ระบบจะแจ้งเตือนด้วยข้อความรวมไปถึงส่งรูปภาพเข้าสู่แอปพลิเคชันมือถือของผู้ที่ได้ทำการลงทะเบียนอุปกรณ์กับแอปพลิเคชันให้ทราบว่ามีบุคคลมาหา โดยในแอปพลิเคชัน จะสามารถตรวจสอบประวัติว่ามีใครบ้างที่มาหาพร้อมระยะเวลา สามารถเพิ่ม-ลบสมาชิก เพิ่ม-ลบอุปกรณ์ และสามารถควบคุมการปลดล็อกกลอนประตูผ่านอินเทอร์เน็ต ซึ่งในขั้นตอนการส่งข้อความและการบันทึกข้อมูลลงฐานข้อมูลนั้นจะใช้บริการต่างๆของคลาวด์ เช่น

AWS aws lambda, AWS DynamoDB, AMAZON S3, AMAZON SNS เป็นต้น

3.1.1 ขั้นตอนการทำงานของระบบ

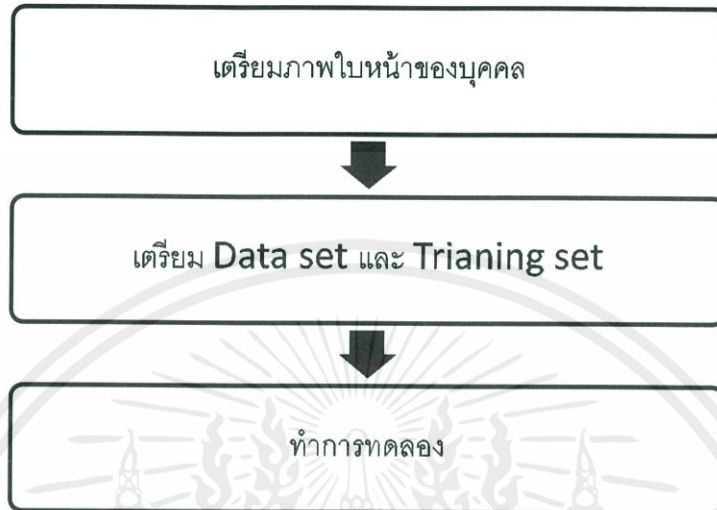


รูปที่ 3.2 แผนผังแสดงขั้นตอนการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 อัลกอริทึมที่ใช้ในการประมวลผลในขั้นตอนรู้จำใบหน้า

สำหรับการเปรียบเทียบอัลกอริทึมแบบต่างๆ เพื่อเปรียบเทียบหาวิธีที่แม่นยำที่สุดภายใต้เงื่อนไขต่างๆ มีขั้นตอนดังนี้



รูปที่ 3.3 ขั้นตอนการเปรียบเทียบอัลกอริทึมแบบต่างๆ

3.2.1 ฐานข้อมูลรูปภาพที่ใช้ในการทดลอง

ฐานข้อมูลภาพใบหน้าที่นำมาใช้ทดลองนั้น คือ 'AT&T Facedatabase' หรือเดิมคือ 'The ORL Facedatabase', 'Yale Facedatabase A', 'Extended Yale Facedatabase B' ซึ่งมีคุณสมบัติแตกต่างกันดังนี้

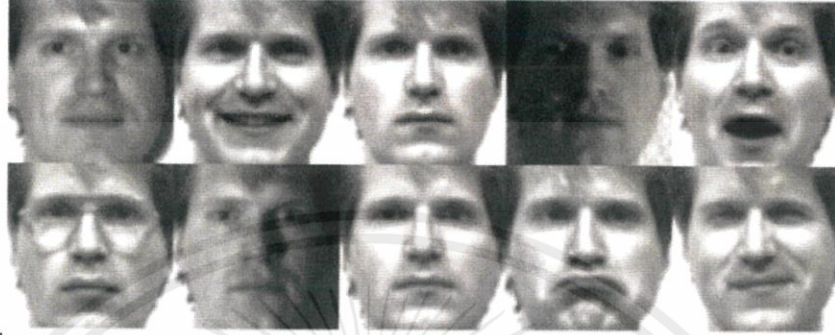
- 1) AT&T Facedatabase ประกอบด้วยรูปภาพของบุคคลที่แตกต่างกัน 40 คน แบ่งออกเป็นบุคคลละ 10 ภาพแตกต่างกันในระดับสี่เทา



รูปที่ 3.4 ตัวอย่างของฐานข้อมูลใบหน้า AT&T Facedatabase

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) Yale Facedatabase A ประกอบด้วยรูปภาพของบุคคลที่แตกต่างกัน 15 บุคคลคน แบ่งออกเป็นบุคคลละ 11 ภาพแตกต่างกันด้วยการแสดงอารมณ์ในรูปแบบต่างๆ เช่น ยิ้ม ดีใจ เสียใจ เป็นต้น ซึ่งเป็นภาพในระดับสีเทา



รูปที่ 3.5 ตัวอย่างของฐานข้อมูลใบหน้า Extended Yale Facedatabase B

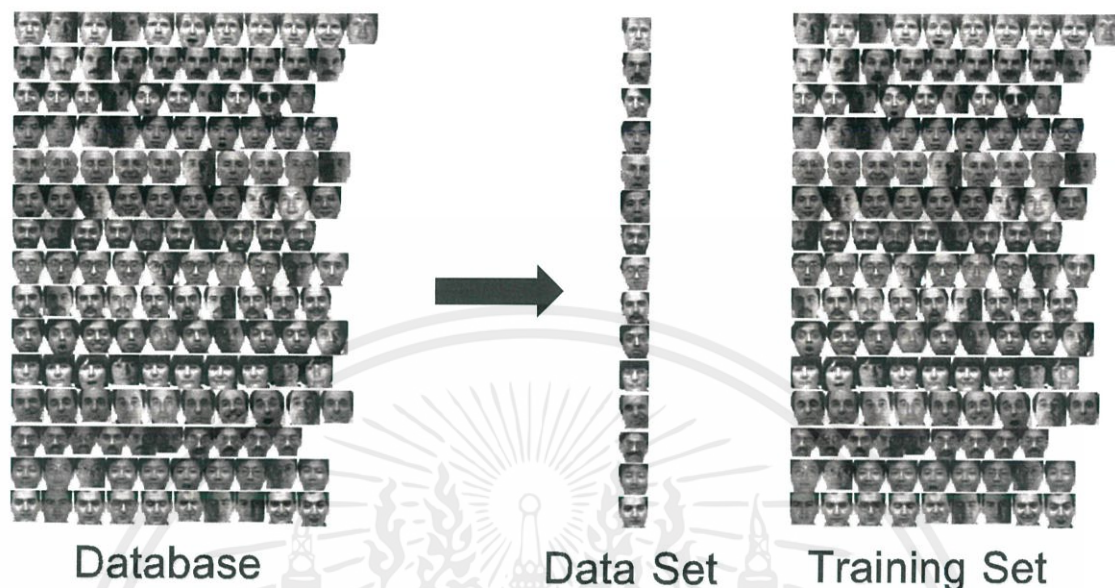
- 3) Extended Yale Facedatabase B ประกอบด้วยรูปภาพจำนวน 16128 ของบุคคลที่แตกต่างกัน 28 บุคคลคน ซึ่งแต่ละภาพเป็นระดับสีเทาที่แตกต่างกันด้วยความเข้มของแสงในรูปแบบต่างๆ



รูปที่ 3.6 ตัวอย่างของฐานข้อมูลใบหน้า Extended Yale Facedatabase B

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 การเตรียม Data set และ Training Set



รูปที่ 3.7 การเตรียม Data set และ Training Set

รูปภาพในแต่ละฐานข้อมูลนั้นจะถูกแบ่งออกเป็น Data set และ Training Set ซึ่ง Training Set คือชุดภาพเพื่อสำหรับให้โปรแกรมเรียนรู้ในแต่ละใบหน้า ส่วน Data set คือชุดภาพไว้สำหรับทดสอบว่าโปรแกรมที่เขียน โดยอัลกอริทึมในแต่ละอัลกอริทึมมีประสิทธิภาพมากแค่ไหน

สำหรับเตรียมการทดลอง ภาพของบุคคลบุคคลในแต่ละชุดรูปภาพจะถูกสุ่มดึง 1 ภาพ แล้วจัดเก็บลง Data set ส่วนภาพที่เหลือจะถูกจัดเก็บลงใน Training Set ซึ่งจะมีการสุ่มแบบนี้ในแต่ละชุดรูปภาพเป็นจำนวนครั้งที่กำหนด สำหรับการทดลองนี้ จะสุ่มเป็นจำนวน 5 ครั้ง เพื่อให้ได้ผลลัพธ์ที่แม่นยำมากขึ้น

3.2.3 ขั้นตอนการทดลอง

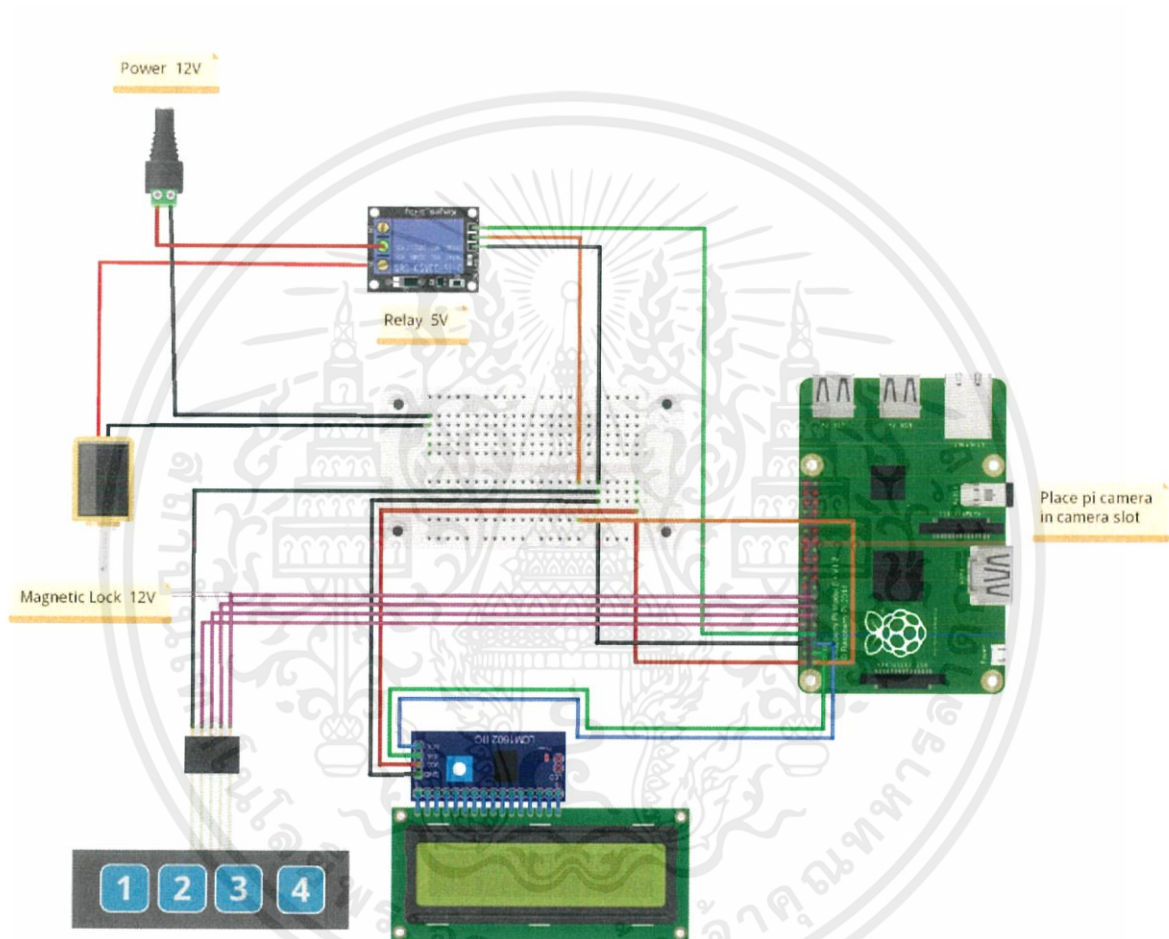
- 1) Face Detection
- 2) Feature Extraction ประกอบด้วย 3 อัลกอริทึมดังนี้ คือ
 - Eigenfaces
 - Fisherfaces
 - Local Binary Patterns Histograms
- 3) Classifier
 - k-Nearest Neighbor ประกอบด้วยสมการ ดังนี้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1) Euclidean Distance
- 2) Cosine Distance
- 3) Chi-Square Distance

3.3 โครงสร้างของฮาร์ดแวร์

3.3.1 การทำงานของวงจรไฟฟ้า

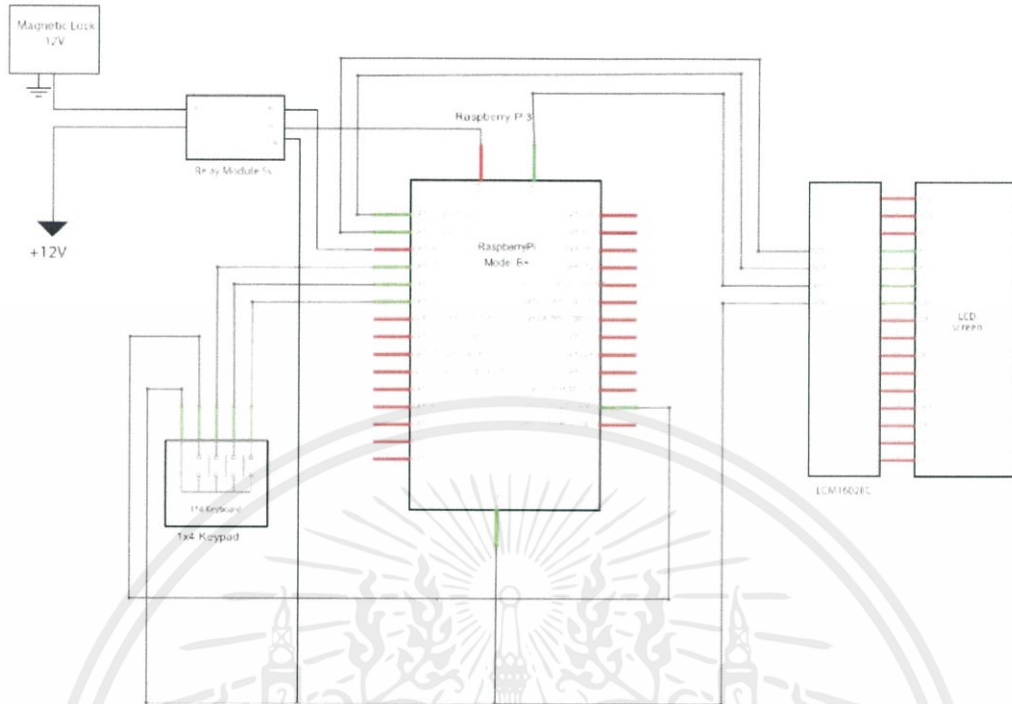


รูปที่ 3.8 การทำงานของวงจรไฟฟ้า

วงจรประกอบไปด้วย

- 1) Raspberry Pi 2 Model B
- 2) 1602 LCD (Blue Screen) 16x2 LCD
- 3) Relay Module 5V
- 4) Membrane Keypad Matrix Switch 1x4
- 5) กลอนแม่เหล็กไฟฟ้า กลอนไฟฟ้า Magnetic Lock 60 kg 12V

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 รายละเอียด Pin 1 ของการทำงานของวงจรไฟฟ้า

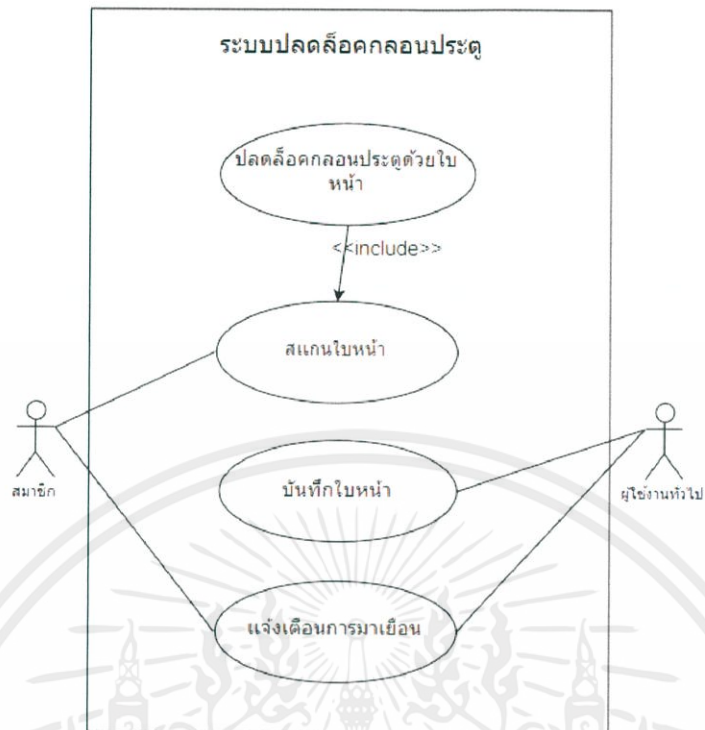
ออกแบบวงจรโดย

- 1) ต่อพินขนาด 5 โวลต์จากบอร์ดเข้ากับ โมดูลจอ 16x2 LCD เพื่อเป็นพาวเวอร์ซัพพลาย
- 2) ล็อกแบบแม่เหล็ก (Magnetic Lock) ขนาด 12 โวลต์จะเชื่อมต่อกับบอร์ดและพาวเวอร์ซัพพลาย (ไฟฟ้ากระแสสลับ 12V) ผ่านโมดูลรีเลย์ขนาด 5 โวลต์ โดยพาวเวอร์ซัพพลาย จะเชื่อมต่อเข้ากับช่อง Common(c) ส่วนล็อกแบบแม่เหล็ก (Magnetic Lock) จะเชื่อมต่อเข้ากับช่อง NC
- 3) ต่อพินขนาด 3.3 โวลต์จากบอร์ดเข้ากับ โมดูลรีเลย์เพื่อเป็นพาวเวอร์ซัพพลาย
- 4) ปุ่มแบบ 1x4 คีย์แพด

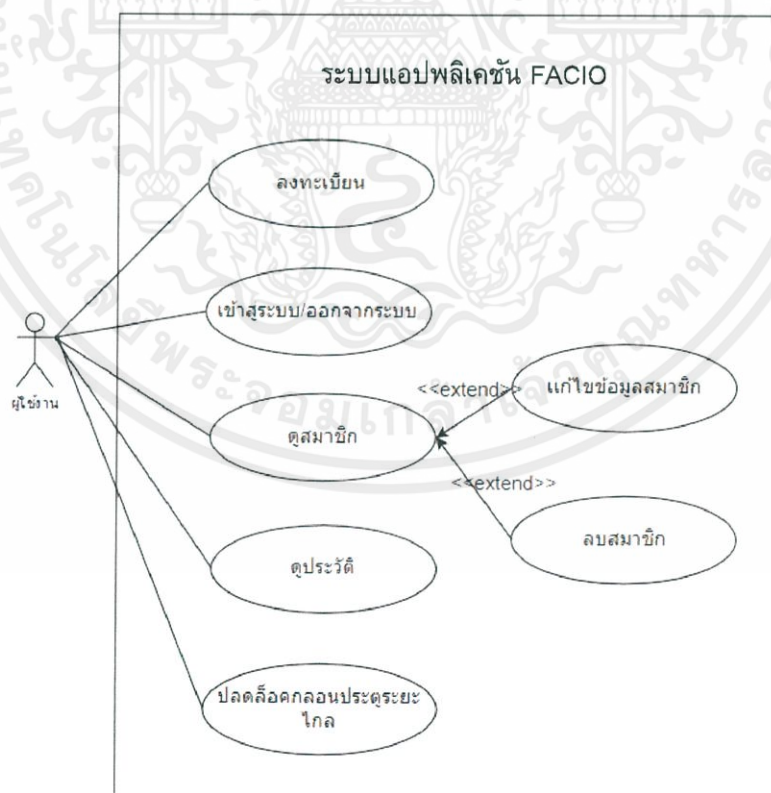
3.4 แผนภาพ UML (Unified Modeling Language Diagram)

3.4.1 Use Case Diagram

ระบบปลดล็อกกลอนแบบรู้จำใบหน้าจะประกอบด้วยเจ้าของอุปกรณ์, สมาชิก และผู้ใช้ทั่วไป ซึ่งระบบปลดล็อกกลอนแบบรู้จำใบหน้าสามารถแสดงเป็น Use case Diagram ของระบบได้ดังนี้



รูปที่ 3.10 แผนภาพ Use case ของระบบปลดลือกคกลอนแบบรู้จำใบหน้า



รูปที่ 3.11 แผนภาพ Use case ของระบบแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดขั้นตอนต่างๆ ใน Use Case Diagram ของระบบกลอนประสูร์จำไบหน้า

ตารางที่ 3.1 Use case ลงทะเบียน

| | |
|----------------------|---|
| Use case : ลงทะเบียน | |
| Actor | เจ้าของอุปกรณ์ |
| Purpose | เพื่อสมัครเป็นสมาชิกสำหรับใช้งานแอปพลิเคชัน |
| Description | ใส่ข้อมูลส่วนตัว รวมไปถึงรหัสประจำเครื่องอุปกรณ์ที่ต้องการเชื่อมต่อกับแอปพลิเคชัน |
| Precondition | เปิดเข้าใช้งานแอปพลิเคชัน |
| Main Flows | <ol style="list-style-type: none"> 1. เปิดเข้าแอปพลิเคชัน 2. กดปุ่ม Sign Up |

ตารางที่ 3.2 Use case เข้าสู่ระบบ/ออกจากระบบ

| | |
|-----------------------------------|---|
| Use case : เข้าสู่ระบบ/ออกจากระบบ | |
| Actor | เจ้าของอุปกรณ์ |
| Purpose | เพื่อเข้าสู่ระบบเพื่อใช้งานหรือออกจากระบบ |
| Description | เจ้าของอุปกรณ์ใส่ชื่อผู้ใช้และรหัสผ่านก่อนจะเลือกปุ่ม Login |
| Precondition | เปิดเข้าใช้งานแอปพลิเคชันหรือเมื่อต้องการออกจากระบบ |
| Main Flows | <ol style="list-style-type: none"> 1. เปิดเข้าแอปพลิเคชัน 2. ใส่ชื่อผู้ใช้และรหัสผ่าน 3. กดปุ่ม Log in |

ตารางที่ 3.3 Use case ดูประวัติผู้มาเยือน

| | |
|------------------------------------|---|
| Use case : จัดการประวัติผู้มาเยือน | |
| Actor | เจ้าของอุปกรณ์ |
| Purpose | เพื่อดูหรือแก้ไขประวัติของคนที่มาเยือนซึ่งได้บันทึกไว้ |
| Description | เจ้าของอุปกรณ์เลื่อนหน้าแอปพลิเคชันไปยัง History |
| Precondition | เข้าสู่หน้า History |
| Main Flows | <ol style="list-style-type: none"> 1. เปิดเข้าแอปพลิเคชัน 2. เข้าสู่ระบบ(กรณีที่ไม่ได้เข้าสู่ระบบค้างไว้) 3. เลื่อนหน้าจอไปที่ History |

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 Use case ดูสมาชิก

| | |
|----------------------------|--|
| Use case : ดูสมาชิก | |
| Actor | เจ้าของอุปกรณ์ |
| Purpose | เพื่อดูอุปกรณ์ที่เชื่อมต่ออยู่ |
| Description | เจ้าของอุปกรณ์เลื่อนหน้าแอปพลิเคชันไปยัง Device |
| Precondition | เข้าสู่หน้า Device |
| Main Flows | <ol style="list-style-type: none"> 1. เปิดเข้าแอปพลิเคชัน 2. เข้าสู่ระบบ(กรณีที่ไม่ได้เข้าสู่ระบบค้างไว้) 3. เลื่อนหน้าจอไปที่ Member |

ตารางที่ 3.5 Use case แก้ไขข้อมูลสมาชิก

| | |
|--------------------------------------|--|
| Use case : แก้ไขข้อมูลส่วนตัว | |
| Actor | เจ้าของอุปกรณ์ |
| Purpose | เพื่อแก้ไขข้อมูลส่วนตัวของเจ้าของอุปกรณ์ เช่น ชื่อ ที่อยู่ เบอร์โทรศัพท์ เป็นต้น |
| Description | เจ้าของอุปกรณ์ใส่ข้อมูลที่ต้องการเปลี่ยน |
| Precondition | เข้าสู่ระบบ |
| Main Flows | <ol style="list-style-type: none"> 1. เปิดเข้าแอปพลิเคชัน 2. เข้าสู่ระบบ(กรณีที่ไม่ได้เข้าสู่ระบบค้างไว้) 3. เลื่อนหน้าจอไปที่ Member 4. กดปุ่มเข้าสู่สมาชิกที่ต้องการ 5. กดปุ่ม Edit เพื่อแก้ไขข้อมูล 6. กดปุ่ม done เพื่อยืนยันและบันทึกข้อมูลที่แก้ไข |

ตารางที่ 3.6 Use case ลบสมาชิก

| Use case : ลบสมาชิก | |
|---------------------|--|
| Actor | เจ้าของอุปกรณ์ |
| Purpose | เพื่อปลดล็อคอุปกรณ์ทางไกล |
| Description | เจ้าของอุปกรณ์เลื่อนหน้าแอปพลิเคชัน ไปยัง Device |
| Precondition | เข้าสู่หน้า Device |
| Main Flows | <ol style="list-style-type: none"> 1. เปิดเข้าแอปพลิเคชัน 2. เข้าสู่ระบบ(กรณีที่ไม่ได้เข้าสู่ระบบค้างไว้) 3. กดปุ่มเข้าสู่สมาชิกที่ต้องการ 4. กดปุ่ม delete เพื่อแก้ไขข้อมูล 5. กดปุ่ม done เพื่อยืนยัน |

ตารางที่ 3.7 Use case ปลดล็อคกลอนผ่านแอปพลิเคชัน

| Use case : ปลดล็อคกลอนผ่านแอปพลิเคชัน | |
|---------------------------------------|---|
| Actor | เจ้าของอุปกรณ์ |
| Purpose | เพื่อปลดล็อคอุปกรณ์ทางไกล |
| Description | เจ้าของอุปกรณ์เลื่อนหน้าไปยัง Home |
| Precondition | เข้าสู่หน้า Device |
| Main Flows | <ol style="list-style-type: none"> 1. เปิดเข้าแอปพลิเคชัน 2. เข้าสู่ระบบ(กรณีที่ไม่ได้เข้าสู่ระบบค้างไว้) |

ตารางที่ 3.8 Use case บันทึกใบหน้า

| Use case : บันทึกใบหน้า | |
|-------------------------|--|
| Actor | เจ้าของอุปกรณ์, สมาชิก, ผู้ใช้งานทั่วไป |
| Purpose | สำหรับผู้ใช้งานทั่วไปที่ต้องการเป็นสมาชิกหรือเจ้าของอุปกรณ์และสมาชิกต้องการบันทึกรูปใหม่ |
| Description | กดปุ่มบันทึกใบหน้า |
| Precondition | - |
| Main Flows | 1. บันทึกใบหน้า (Adding Member) |

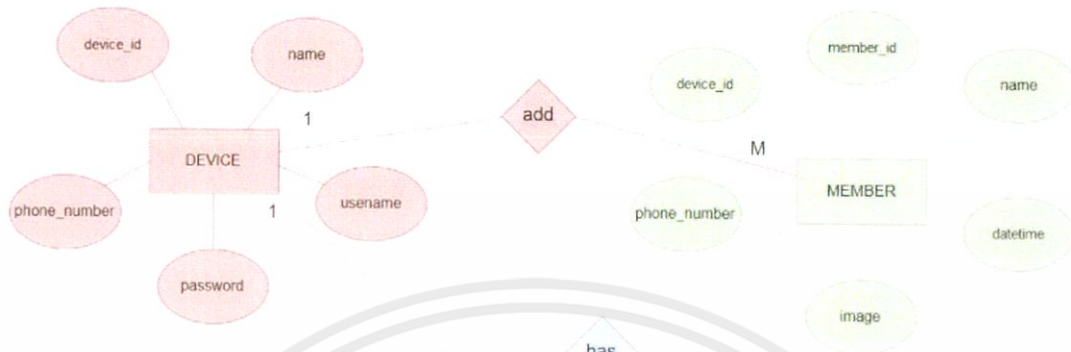
ตารางที่ 3.9 Use case สแกนใบหน้า

| Use case : สแกนใบหน้า | |
|-----------------------|--|
| Actor | เจ้าของอุปกรณ์, สมาชิก, ผู้ใช้งานทั่วไป |
| Purpose | สำหรับผู้ใช้งานทุกประเภทเมื่อต้องการจะปลดล็อคกลอนประตู |
| Description | กดปุ่มบันทึกสแกนใบหน้า |
| Precondition | - |
| Main Flows | 1. กดปุ่มบันทึกสแกนใบหน้า (Scan Face) |

ตารางที่ 3.10 Use case สแกนใบหน้า

| Use case : แจ้งเตือนการมเยือน | |
|-------------------------------|--|
| Actor | สมาชิก, ผู้ใช้งานทั่วไป |
| Purpose | เมื่อสมาชิกหรือผู้ใช้งานทั่วไปต้องการแจ้งให้เจ้าของอุปกรณ์ทราบมาตนมเยือน |
| Description | กดปุ่มแจ้งเตือน |
| Precondition | - |
| Main Flows | 1. กดปุ่มแจ้งเตือน |

3.4.2 ER diagram



รูปที่ 3.12 ER Diagram ของระบบปลดล็อคกล้องประตูรั้วหน้า

ตารางที่ 3.11 รายละเอียด DEVICE

| | | | | | |
|----------|------------------|-------------|-----------------|-----------------|----------------------|
| ← P.K. → | device_id | name | username | password | phone_ number |
|----------|------------------|-------------|-----------------|-----------------|----------------------|

ตารางที่ 3.12 MEMBER

| | | | | | |
|------------------|------------------|-------------|----------------------|-----------------|--------------|
| member_id | device_id | name | phone_ number | datetime | image |
|------------------|------------------|-------------|----------------------|-----------------|--------------|

ตารางที่ 3.13 HISTORY

| | | | | | |
|----------|---------------|-------------|--------------|-------------------|------------------|
| ← P.K. → | his_id | name | image | visitor_id | date_time |
|----------|---------------|-------------|--------------|-------------------|------------------|

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.14 รูปแบบของตาราง DEVICE ในฐานข้อมูล

| Key | Attribute | Type | Description |
|-------------|--------------|---------|--|
| Primary key | device_id | VARCHAR | id ของอุปกรณ์บอร์ด |
| - | name | VARCHAR | ชื่อของสมาชิกที่ลงทะเบียน |
| - | username | VARCHAR | username ที่ใช้ในการ login เข้าสู่ แอปพลิเคชัน |
| - | password | VARCHAR | password ที่ใช้ในการ login เข้าสู่ แอปพลิเคชัน |
| - | phone_number | VARCHAR | เบอร์โทรศัพท์ |

ตารางที่ 3.15 รูปแบบของตาราง MEMBER ในฐานข้อมูล

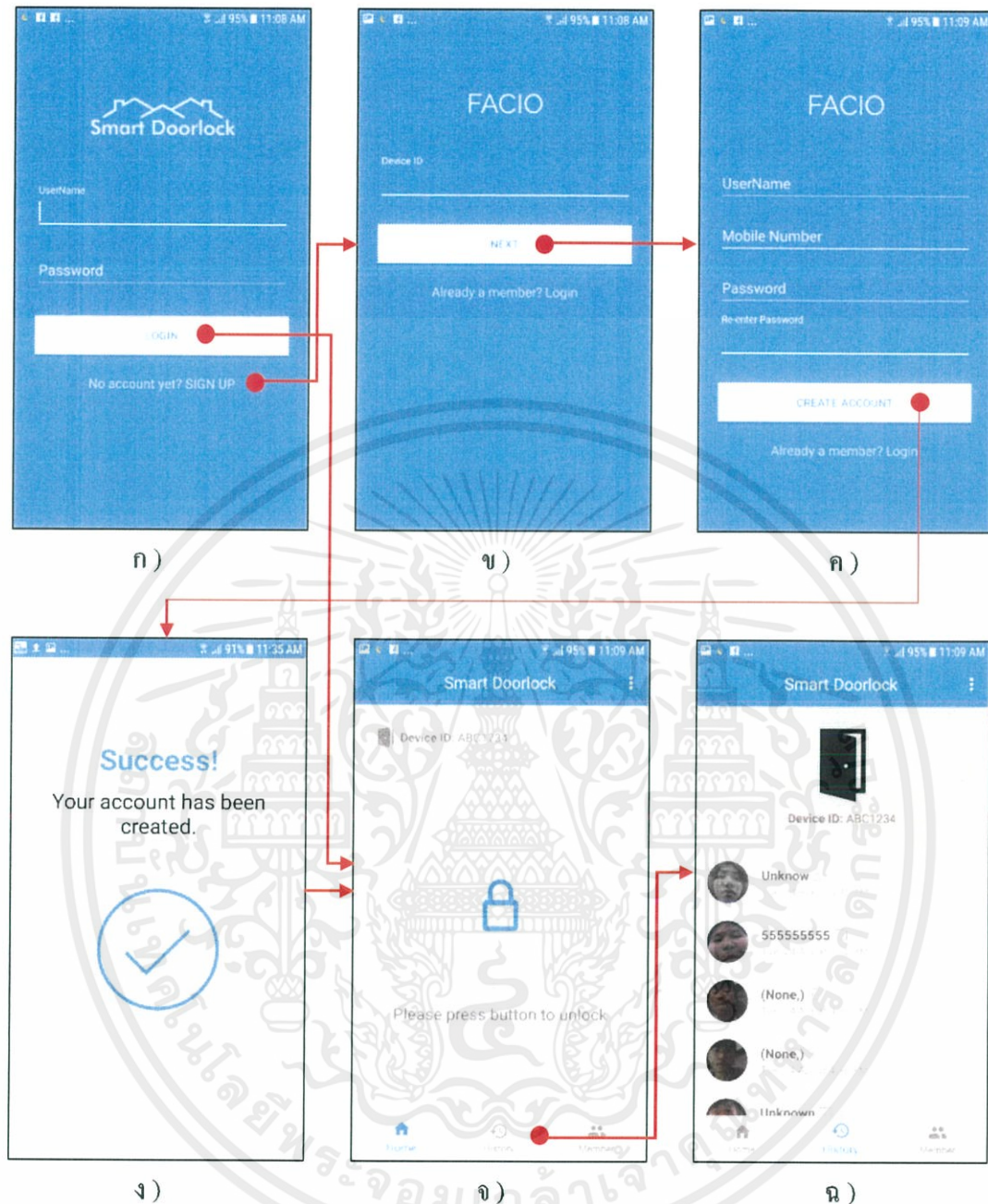
| Key | Attribute | Type | Description |
|-------------|--------------|----------|------------------------|
| Primary key | member_id | INT | id ของ member |
| | device_id | VARCHAR | id ของอุปกรณ์บอร์ด |
| - | name | VARCHAR | ชื่อของ member |
| | phone_number | VARCHAR | เบอร์โทรศัพท์ |
| | datetime | DATETIME | เวลาที่เพิ่มเป็นสมาชิก |
| | image | BLOB | รูปภาพของสมาชิก |

ตารางที่ 3.16 รูปแบบของตาราง HISTORY ในฐานข้อมูล

| Key | Attribute | Type | Description |
|-------------|-----------|----------|-----------------------------|
| Primary key | his_id | INT | id ของ history |
| - | member_id | INT | id ของ member |
| - | device_id | VARCHAR | id ของอุปกรณ์บอร์ด |
| - | datetime | DATETIME | วันและเวลาที่บันทึก |
| | image | BLOB | รูปภาพที่บันทึกของผู้ใช้งาน |

3.5 การออกแบบส่วนติดต่อผู้ใช้งาน (User Interface)

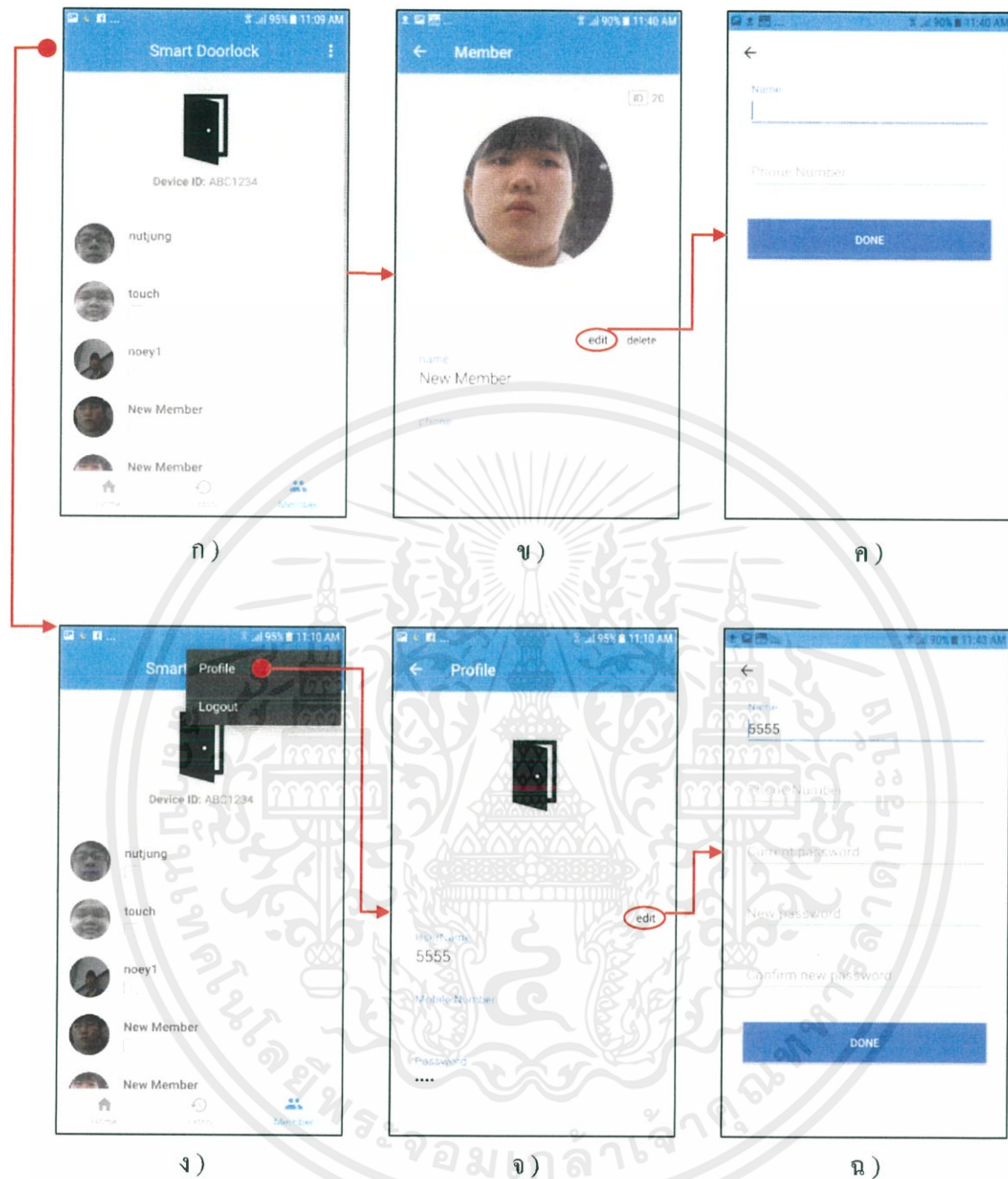
การออกแบบส่วนติดต่อผู้ใช้งานจะเป็นส่วนที่ผู้ใช้งานสามารถควบคุมการทำงานของอุปกรณ์ได้ทั้งจากระยะใกล้ และระยะไกล โดยการเข้าใช้งานแอปพลิเคชัน ผู้ใช้งานต้องทำการเข้าสู่ระบบด้วย Username และ Password ที่ได้ทำการลงทะเบียนไว้ แต่ถ้าผู้ใช้งานยังไม่เคยเข้าใช้งานแอปพลิเคชันมาก่อน ผู้ใช้งานต้องทำการลงทะเบียนก่อนการเข้าใช้ โดยภายในแอปพลิเคชันจะมีการแบ่งส่วนติดต่อผู้ใช้งานออกเป็นส่วนๆตามการทำงาน สำหรับส่วนติดต่อผู้ใช้งาน มีดังนี้



รูปที่ 3.13 Flow การทำงานส่วนติดต่อผู้ใช้งาน (1)

- ก) หน้า Login หรือหน้าเข้าสู่ระบบ
- ข) หน้า Sign Up หรือหน้าลงทะเบียน(กรอก Device ID)
- ค) หน้า Sign Up หรือหน้าลงทะเบียน (กรอกข้อมูล)
- ง) หน้า Success Register
- จ) หน้า ควบคุมการปลดล็อกทางไกล (Home)
- ฉ) หน้า แสดงประวัติ (History)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

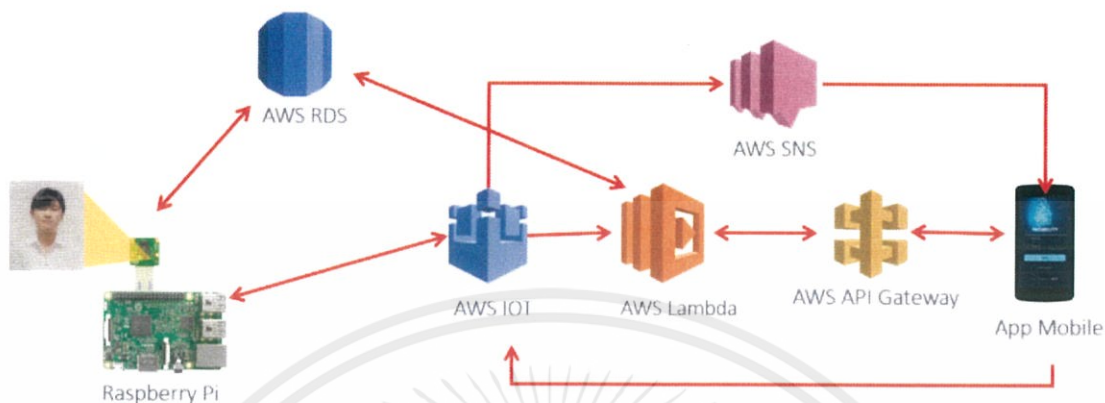


รูปที่ 3.14 Flow การทำงานส่วนติดต่อผู้ใช้งาน (2)

- ก) หน้าสมาชิก (Member)
- ข) หน้าแสดงข้อมูลสมาชิก แก้ไขสมาชิก
- ค) หน้าแก้ไขสมาชิก
- ง) หน้าแสดงรายการ Settings
- จ) หน้าข้อมูล Profile
- ฉ) หน้าแก้ไข Profile

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การออกแบบส่วนของระบบคลาวด์ (Cloud Architecture)

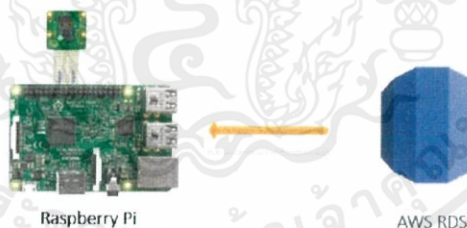


รูปที่ 3.15 การออกแบบระบบคลาวด์

ในส่วนของการติดต่อสื่อสารระหว่างอุปกรณ์เคลื่อนที่และบอร์ด Raspberry pi รวมไปถึงฐานข้อมูลระบบ นั้นจะใช้เทคโนโลยีคลาวด์ ซึ่งแบ่งการทำงานออกตามฟังก์ชันการใช้งานต่างๆ ดังนี้

1) การเก็บข้อมูลลงฐานข้อมูล

การเก็บข้อมูลลงฐานข้อมูลจะใช้เซอร์วิส AWS RDS ซึ่งเชื่อมต่อกับบอร์ด โดยตรง



รูปที่ 3.16 การเรียกดึงข้อมูลบนฐานข้อมูลผ่านบอร์ด Raspberry pi

2) การการเรียกดึงหรือแก้ไขข้อมูลบนฐานข้อมูล

การการดึงข้อมูลต่างๆ รวมไปถึงแก้ไขเปลี่ยนแปลง ทั้งข้อมูลสมาชิกและประวัติของผู้มาเยือน ด้วยอุปกรณ์โทรศัพท์มือถือจะต้องดึงข้อมูลจากฐานข้อมูลบนคลาวด์ซึ่งจะประมวลผลคำสั่งด้วยฟังก์ชันที่สร้างไว้อยู่บนเซอร์วิส AWS Lambda โดยอุปกรณ์เคลื่อนที่ที่จะต้องเรียกใช้งานผ่านเซอร์วิส AWS Gateway



รูปที่ 3.17 การทำการการเรียกดึงหรือแก้ไขข้อมูลบนฐานข้อมูล

3) การส่งข้อความผ่านเซอร์วิส AWS SNS

การแจ้งเตือนจากบอร์ดไปยังอุปกรณ์เคลื่อนที่ทำงานโดย บอร์ดจะส่งข้อความไปยัง AWS IOT เพื่อทำการเรียกใช้เซอร์วิส AWS SNS อีกต่อหนึ่ง



รูปที่ 3.18 การส่งข้อความผ่านเซอร์วิส AWS SNS

4) การปลดล๊อคกลอนประตูทางไกล

การปลดล๊อคกลอนประตูทางไกลจากอุปกรณ์โทรศัพท์มือถือจะทำงานผ่านการส่งข้อความไปยังเซอร์วิส AWS IOT ซึ่งเซอร์วิสนี้จะเป็นตัวส่งข้อความไปหาบอร์ด ซึ่งโปรโตคอลที่ใช้ในการส่งข้อความในขั้นตอนนี้คือ MQTT



รูปที่ 3.19 การปลดล๊อคกลอนประตูทางไกล

บทที่ 4

การทดลองและผลการทดลอง

4.1 ผลการทดลองอัลกอริทึมรู้จำใบหน้า

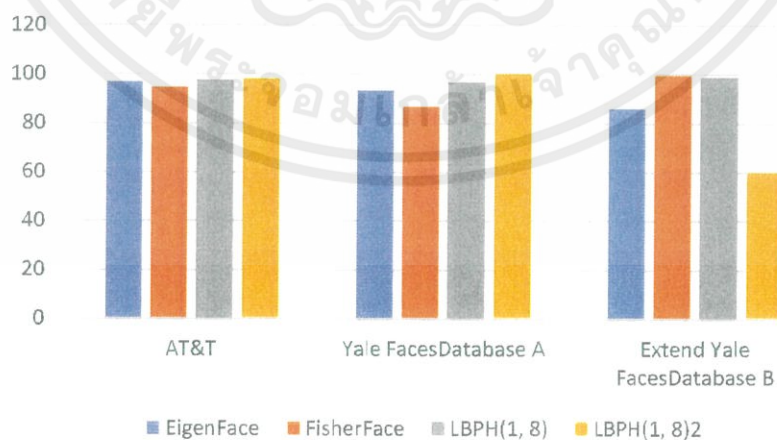
4.1.1 ผลทดลองจากฐานข้อมูลรูปภาพ

ผลทดลองจากฐานข้อมูลรูปภาพ AT&T FacesData, Yale FacesDatabase A, Extend Yale FacesDatabase B ด้วยอัลกอริทึม PCA หรือ Eigenfaces, LDA หรือ Fisherfaces, LBPH(P,R) เมื่อ R แทนจำนวนตำแหน่งรอบจุดศูนย์กลางและตัวแปร R แทนรัศมีระหว่างตำแหน่ง

ตารางที่ 4.1 ผลการทดลองความแม่นยำของอัลกอริทึมโดยใช้ Library OpenCV

| DataFaces | Feature Extraction | | | |
|-----------------------------|--------------------|-------------|-----------|-----------|
| | Eigenfaces | Fisherfaces | LBPH(1,8) | LBPH(2,8) |
| AT&T | 96.75 | 94.50 | 97.50 | 98.00 |
| Yale FacesDatabase A | 93.33 | 86.66 | 96.67 | 97.77 |
| Extend Yale FacesDatabase B | 86 | 99.5 | 98.75 | 99.5 |

กราฟแสดงการเปรียบเทียบของผลการทดลอง



รูปที่ 4.1 กราฟแสดงการเปรียบเทียบของผลการทดลอง

4.1.2 ทดลองรันโดยใช้ Code จาก Philipp Wagner <http://bytefish.de> ด้วยฐานข้อมูลรูปภาพ AT&T

ตารางที่ 4.2 ผลการทดลองความแม่นยำของอัลกอริทึมโดยใช้ Library OpenCV

| Feature Extraction | Classifier(k-Nearest Neighbor) | | | | | | | | |
|--------------------|--------------------------------|-------|-------|--------|-------|-------|------------|-------|-------|
| | Euclidean | | | Cosine | | | Chi-Square | | |
| | K=1 | K=3 | K=5 | K=1 | K=3 | K=5 | K=1 | K=3 | K=5 |
| Fisherfaces | 94.50 | 96.25 | 94.50 | 94.25 | 94.00 | 94.25 | 1 | 1.25 | 0.75 |
| LBPH(1,8) | 97.50 | 95.25 | 89.75 | 97.75 | 94.75 | 91.75 | 98.25 | 97.25 | 94.00 |
| LBPH(2,8) | 97.00 | 95.75 | 90.25 | 98.00 | 96.00 | 93.50 | 99.25 | 98.00 | 96.75 |

4.1.3 ทดลองรันผลโปรแกรมรับอินพุทเป็นวีดีโอสตรีม

ทดลองโดยใช้อัลกอริทึม LPBH(2, 8) และใช้กล้อง web-cam ของคอมพิวเตอร์โน้ตบุ๊ก ซึ่งบันทึกภาพของแต่ละบุคคลจำนวน 30 ภาพในขั้นตอนการเรียนรู้(training)



รูปที่ 4.2 ตัวอย่างภาพที่ใช้กล้อง web-cam ของคอมพิวเตอร์โน้ตบุ๊ก

ซึ่งเมื่อทดลองแล้วได้ผลการทดลองแบบต่างๆดังนี้
4.1.3.2 ผลการทดลองที่ถูกต้อง



รูปที่ 4.3 ผลการทดลองรับอินพุทเป็นวิดีโอสตรีมบุคคลเดียว



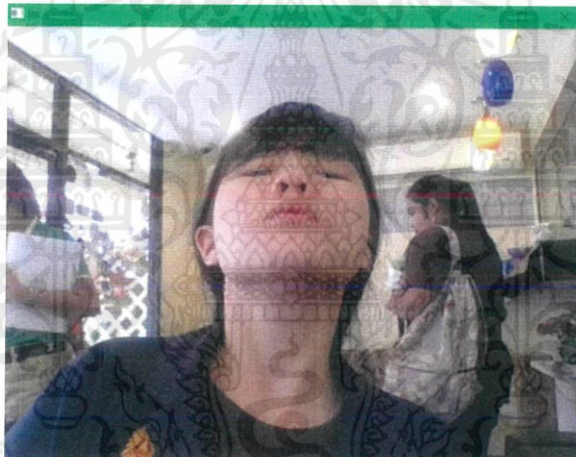
รูปที่ 4.4 ผลการทดลองรับอินพุทเป็นวิดีโอสตรีม 2 บุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

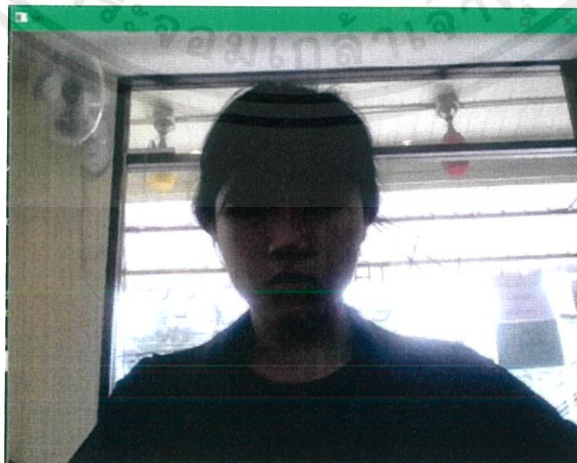
4.1.3.3 ผลการทดลองที่ผิดพลาด



รูปที่ 4.5 ผลการทดลองรับอินพุทเป็นวิดีโอสตรีมที่ผิดพลาดจากการก้มเกินไป



รูปที่ 4.6 ผลการทดลองรับอินพุทเป็นวิดีโอสตรีมที่ผิดพลาดจากการเงยเกินไป



รูปที่ 4.7 ผลการทดลองรับอินพุทเป็นวิดีโอสตรีมที่ผิดพลาดจากแสงที่มืดเกินไป

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดลองการประมวลผลใบหน้าบนบอร์ด Raspberry pi

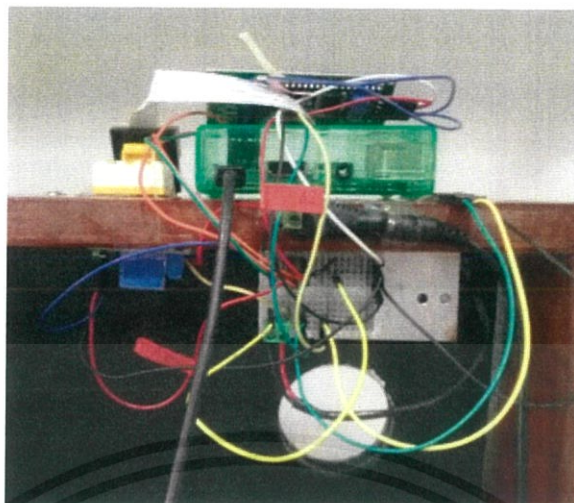


รูปที่ 4.8 ประตูล้างของระบบปลดล็อกกลอนประตูรั้วบ้าน (ด้านหน้า)



รูปที่ 4.9 ประตูล้างของระบบปลดล็อกกลอนประตูรั้วบ้าน (ด้านหลัง)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 ส่วนวงจรระบบปลดล็อกกลอนประตูรั้วหน้า

หน้าจอ LCD แสดงโหมดการทำงานต่างๆ โดยการใช้งานจะแบ่งออกเป็น 3 โหมด ได้แก่

- 1) ตรวจสอบใบหน้าเพื่อปลดล็อกกลอน
- 2) เพิ่มสมาชิก
- 3) แจ้งเตือนไปยังอุปกรณ์โทรศัพท์มือถือ

4.2.2 โหมดปลดล็อกกลอน

4.2.2.1 โหมดปลดล็อกกลอนด้วยใบหน้า

โหมดตรวจสอบใบหน้าเพื่อปลดล็อกกลอน หน้าจอ LCD จะถูกตั้งให้เป็นค่าเริ่มต้น

ปุ่ม “ตกลง” เพื่อ
เข้าทำงานในโหมดที่

ปุ่ม “ถัดไป” เพื่อไป

โหมดถัดไป

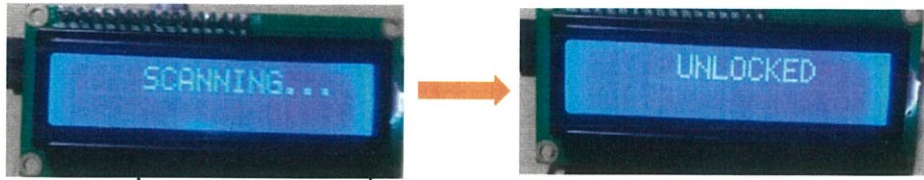
ปุ่ม “ยกเลิก”



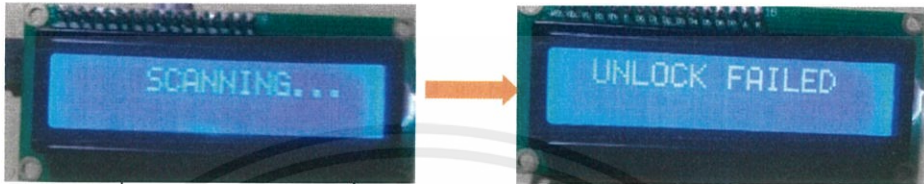
ปุ่ม “แจ้งเตือน” เพื่อแจ้ง
เตือนข้อความไปยังอุปกรณ์
โทรศัพท์มือถือ

รูปที่ 4.11 หน้าจอ LCD โหมดตรวจสอบใบหน้า

การทำงานในโหมดตรวจสอบใบหน้าเพื่อปลดล็อกนั้น เมื่อถูกเลือกจะทำงานตรวจสอบ
ใบหน้าเพื่อเปรียบเทียบกับภาพในฐานข้อมูลแล้วแสดงผลการสแกนออกทางหน้าจอ LCD



รูปที่ 4.12 หน้าจอแสดงเมื่อระบบทำการปลดล็อกสำเร็จ



รูปที่ 4.13 หน้าจอแสดงเมื่อระบบทำการปลดล็อกไม่สำเร็จ

4.2.2.2 โหมดปลดล็อกกลอนด้วยรหัสผ่าน



รูปที่ 4.14 หน้าจอโหมดปลดล็อกด้วยรหัสผ่าน

โดยในโหมดปลดล็อกด้วยรหัสผ่านจะทำงานผ่านการใส่รหัสที่ถูกตั้งค่าไว้ภายในอุปกรณ์ ถ้าใส่รหัสผ่านถูกต้องระบบจะทำการปลดล็อกในลำดับต่อไป

หลังจากนั้นในการปลดล็อกทั้งสองแบบ ข้อมูล เวลา รูป และเลขประจำตัวของผู้ใช้ในกรณีที่ผู้ใช้มีข้อมูลอยู่ในระบบและจะเป็น 0 ในกรณีที่ไม่มีจะถูกส่งเข้าไปเก็บในฐานข้อมูล AWS RDS ในตาราง HISTORY ซึ่งอยู่บนระบบคลาวด์

| his_id | device_id | name | image | date_time | visitor_id |
|--------|-----------|------|--------|-------------------------|------------|
| 22 | ABC1234 | | (BLOB) | 3:8 2018-04-18 15:19:06 | 51 |

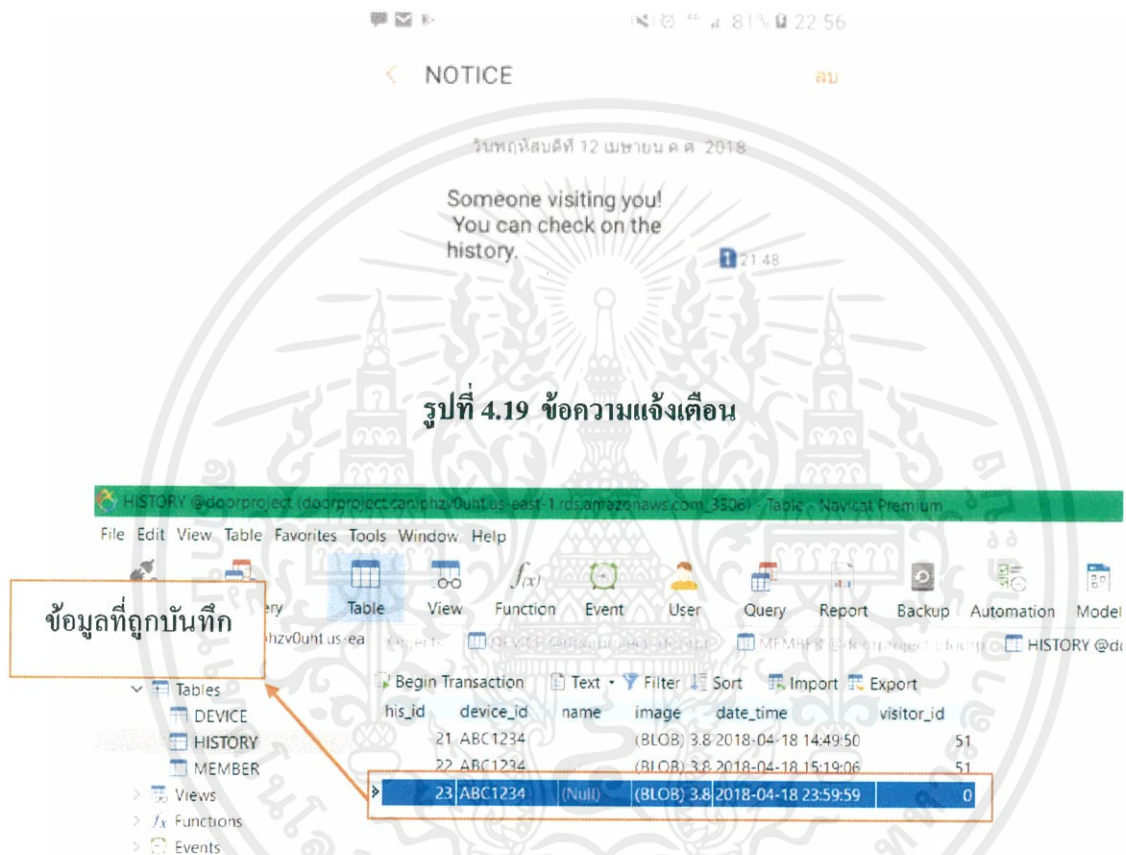
รูปที่ 4.15 ข้อมูลในตาราง HISTORY

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 โหมดแจ้งเตือน

ในการทำงานโหมดแจ้งเตือนซึ่งทำงานโดยการกดปุ่มหมายเลข 4 ระบบจะทำการส่ง SMS ไปยังหมายเลขเบอร์ที่ผูกไว้กับ Application บนมือถือด้วย AWS SNS Service

ซึ่งถ้าไม่ได้บันทึกเบอร์โทรศัพท์มือถือไว้จะไม่สามารถส่งข้อความแจ้งเตือนได้และระบบจะทำการแจ้งบนหน้าจอ LCD



รูปที่ 4.20 ข้อมูลในตาราง HISTORY

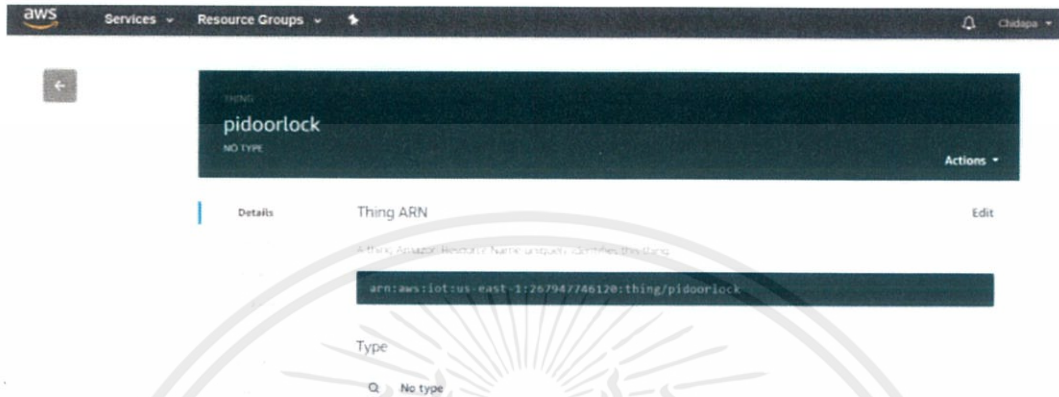
4.3 ผลการทดลองในส่วนของการประมวลผลบนคลาวด์

AWS Cloud Service ที่ใช้ ได้แก่

- 1) AWS IOT
- 2) AWS LAMBDA
- 3) AWS API GATEWAYS
- 4) AWS RDS

4.3.1 AWS IOT

สร้าง Thing ใน AWS IOT เพื่อนำไฟล์ Certificate และไฟล์ Primary Key มาผูกเข้ากับ อุปกรณ์บอร์ด Raspberry pi

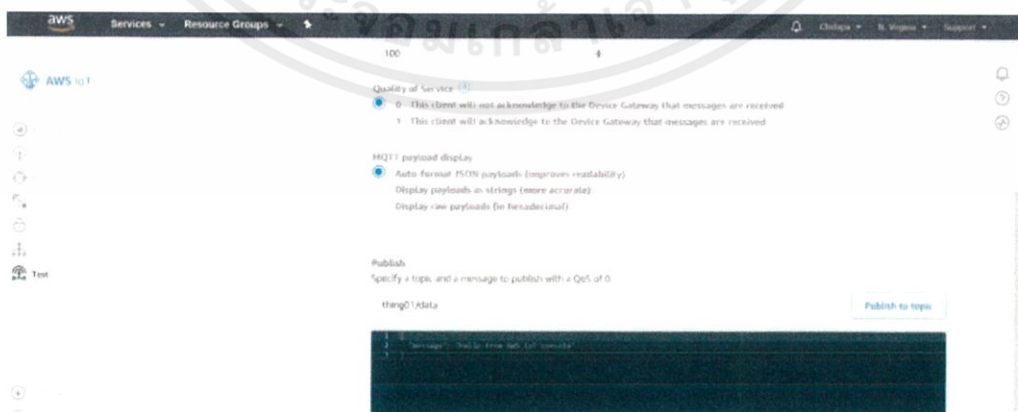


รูปที่ 4.21 THING ที่สร้างขึ้น

| Name | Size | Changed | Rights | Owner |
|--------------------------------|------|--------------------|-----------|-------|
| 68ad7b0ef8-certificate.pem.crt | 2 KB | 21/3/2018 17:13:39 | rw-r--r-- | pi |
| 68ad7b0ef8-private.pem.key | 2 KB | 21/3/2018 17:14:35 | rw-r--r-- | pi |
| 68ad7b0ef8-public.pem.key | 1 KB | 21/3/2018 17:14:27 | rw-r--r-- | pi |
| CA.pem | 2 KB | 21/3/2018 17:14:49 | rw-r--r-- | pi |

รูปที่ 4.22 ไฟล์ Certificate และ key ต่างๆ

การทดลองการเชื่อมต่อระหว่างอุปกรณ์บอร์ด Raspberry pi ทำได้โดยทดลอง Publish ข้อความจาก AWS IOT ไปยังอุปกรณ์ซึ่งได้ทำการ Subscribe Topic เอาไว้โดยผ่านโปรโตคอล MQTT



รูปที่ 4.23 การ Publish ข้อความบนหน้า Test ของ AWS IOT

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pi@raspberrypi: /var/working/lcd
(venv) pi@raspberrypi:/var/working/lcd $ python mqtt_test.py
{
  "message": "Hello from AWS IoT console"
}

```

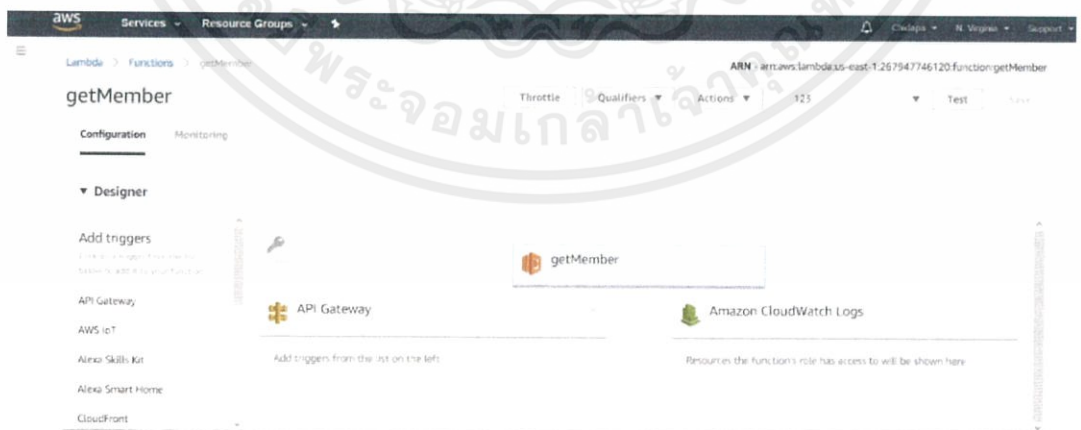
รูปที่ 4.24 ข้อความที่ได้รับบนหน้า Commands ของบอร์ด Raspberry pi

4.3.2 AWS LAMBDA

สร้าง Function ต่างๆเพื่อเชื่อมต่อ Application และบอร์ด Raspberry pi กับฐานข้อมูล AWS RDS

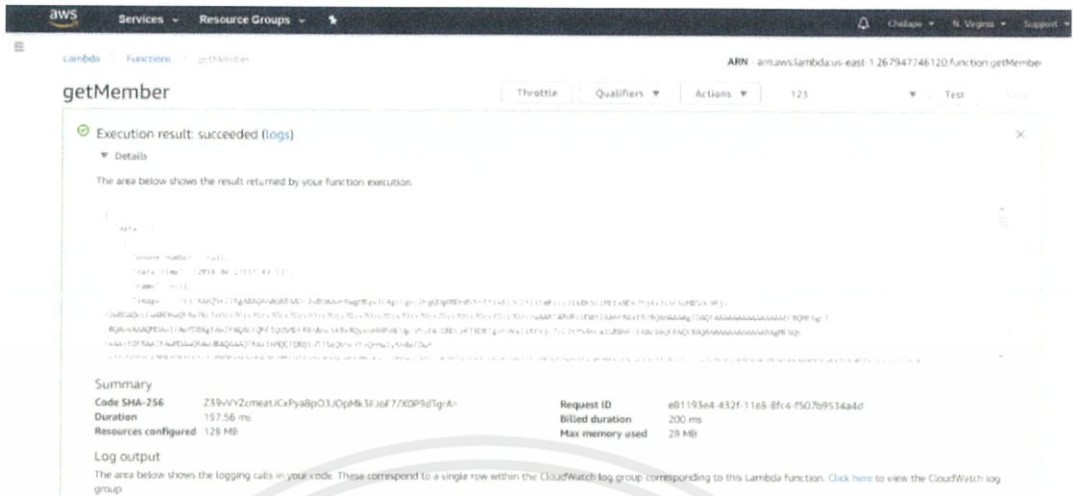


รูปที่ 4.25 functions ที่สร้างขึ้น



รูปที่ 4.26 รายละเอียดการออกแบบ Function

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



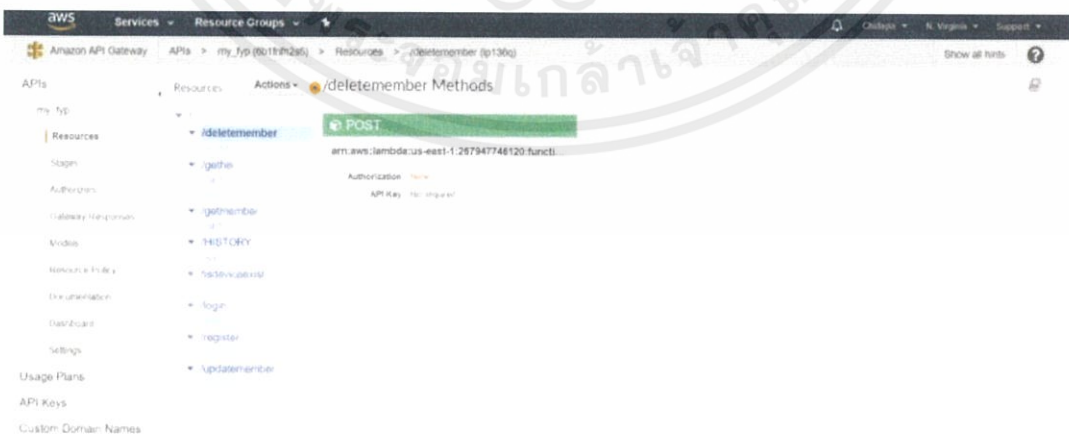
รูปที่ 4.27 ทดสอบการเรียกใช้ Function เมื่อการเรียกใช้สำเร็จ

4.3.3 AWS API GATEWAYS

เป็นตัวกลางในการเรียกใช้ functions ใน Lambda

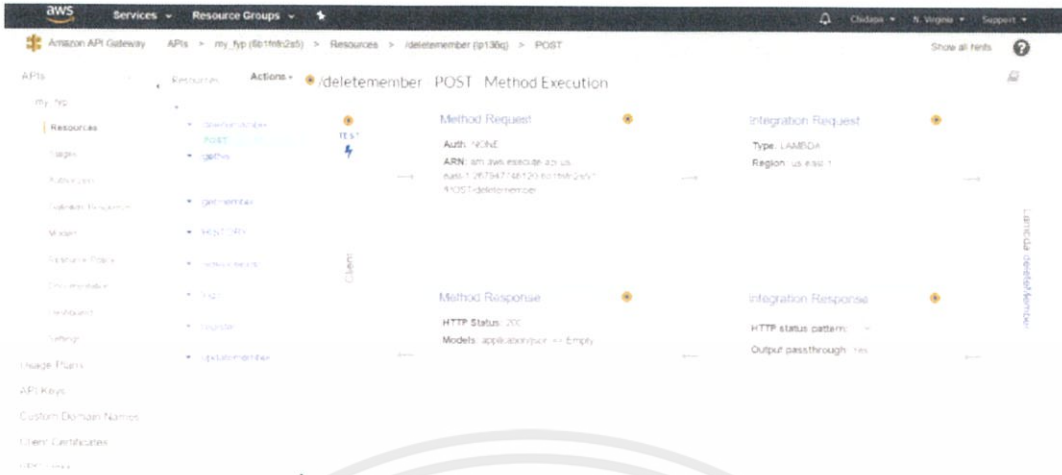


รูปที่ 4.28 API ที่สร้างขึ้น

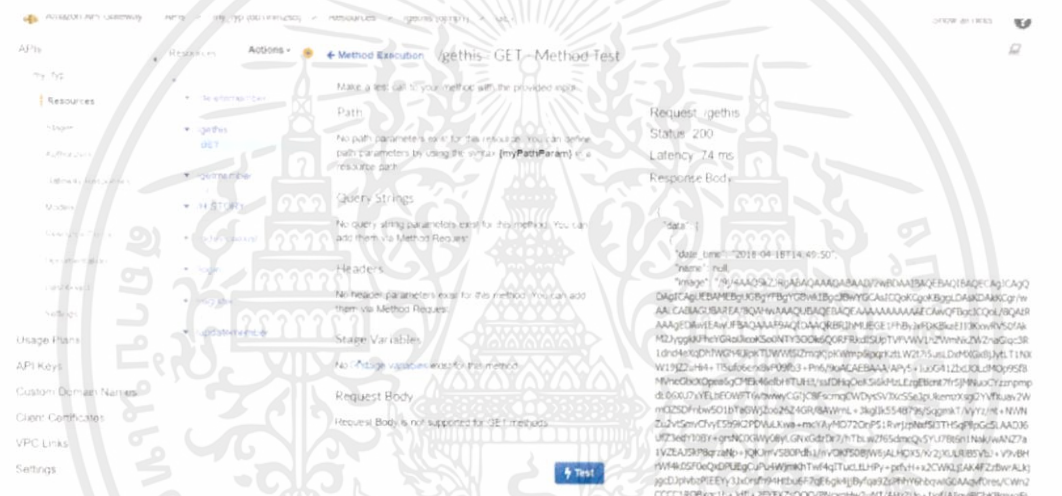


รูปที่ 4.29 Resources และ Methods ของแต่ละ Resource ที่สร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

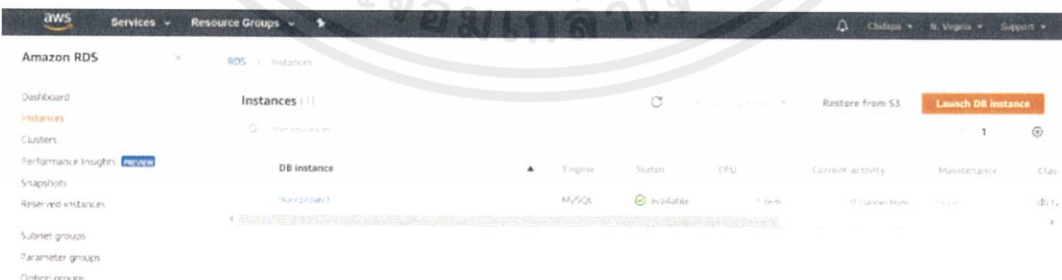


รูปที่ 4.30 แผนผังการประมวลผลของ Method



รูปที่ 4.31 การทดสอบ API

4.3.4 AWS RDS



รูปที่ 4.32 ฐานข้อมูลที่สร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การทดลองใช้แอปพลิเคชัน

จากการที่พัฒนาระบบตามที่ได้มีการออกแบบในบทที่ 3 แล้วขั้นต่อไปคือการทดสอบระบบ วัตถุประสงค์คือ เพื่อทดสอบการทำงานของแอปพลิเคชัน การควบคุมกลอนประตู และการติดต่อกับ Web Service API ให้ถูกต้องตามที่ได้มีการออกแบบไว้ โดยมีขอบเขตการทดลองการทำงานของแอปพลิเคชันดังนี้

- 1) การเข้าสู่ระบบ (Login)
- 2) การสมัครสมาชิก (Sign Up)
- 3) การปลดล็อกทางไกล
- 4) การตรวจสอบประวัติ (History)
- 5) การเรียกดูสมาชิก (Member)
- 6) การแก้ไขข้อมูลของสมาชิก (Edit) และการลบสมาชิก (Delete)
- 7) การแสดงแถบเมนู
- 8) การแก้ไขข้อมูล Profile และ Logout

4.4.1 การเข้าสู่ระบบ (Log in)

4.4.1.1 วัตถุประสงค์

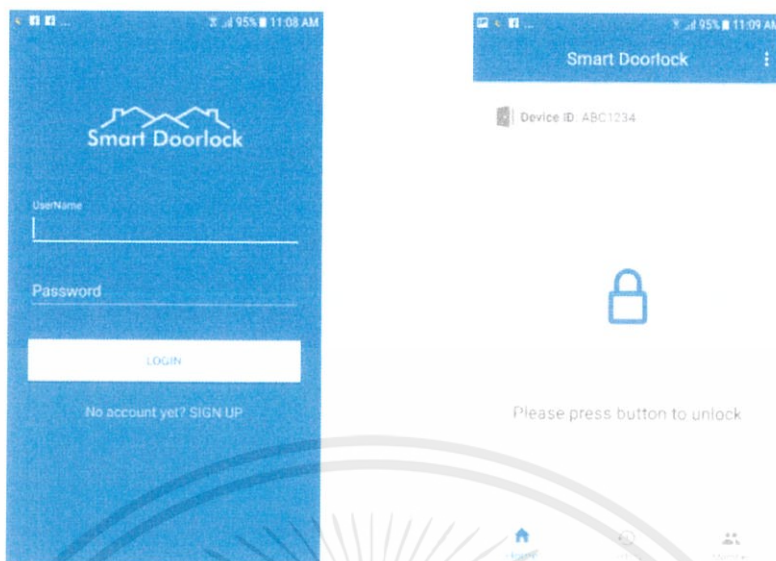
เพื่อทดสอบการล็อกอินเข้าสู่แอปพลิเคชัน

4.4.1.2 ขั้นตอนการทดลอง

ก่อนที่ผู้ใช้จะเข้าใช้งานแอปพลิเคชันได้นั้น ต้องทำการล็อกอินเข้าสู่ระบบก่อน โดย กรอก User Name และ Password และกดปุ่ม “Login” ดังรูปที่ 4.7 ก)

4.4.1.3 ผลการทดลอง

สามารถเข้าสู่ระบบได้ แอปพลิเคชันสามารถติดต่อกับ API Service ได้อย่างถูกต้องและเข้าสู่หน้า Home



ก)

ข)

รูปที่ 4.33 ส่วนแสดงผลการเข้าสู่ระบบ

ก) ส่วนการ Login

ข) ส่วนหน้าแรกของการเข้าสู่ระบบ

4.4.2 การสมัครสมาชิก (Sign Up)

4.4.2.1 วัตถุประสงค์

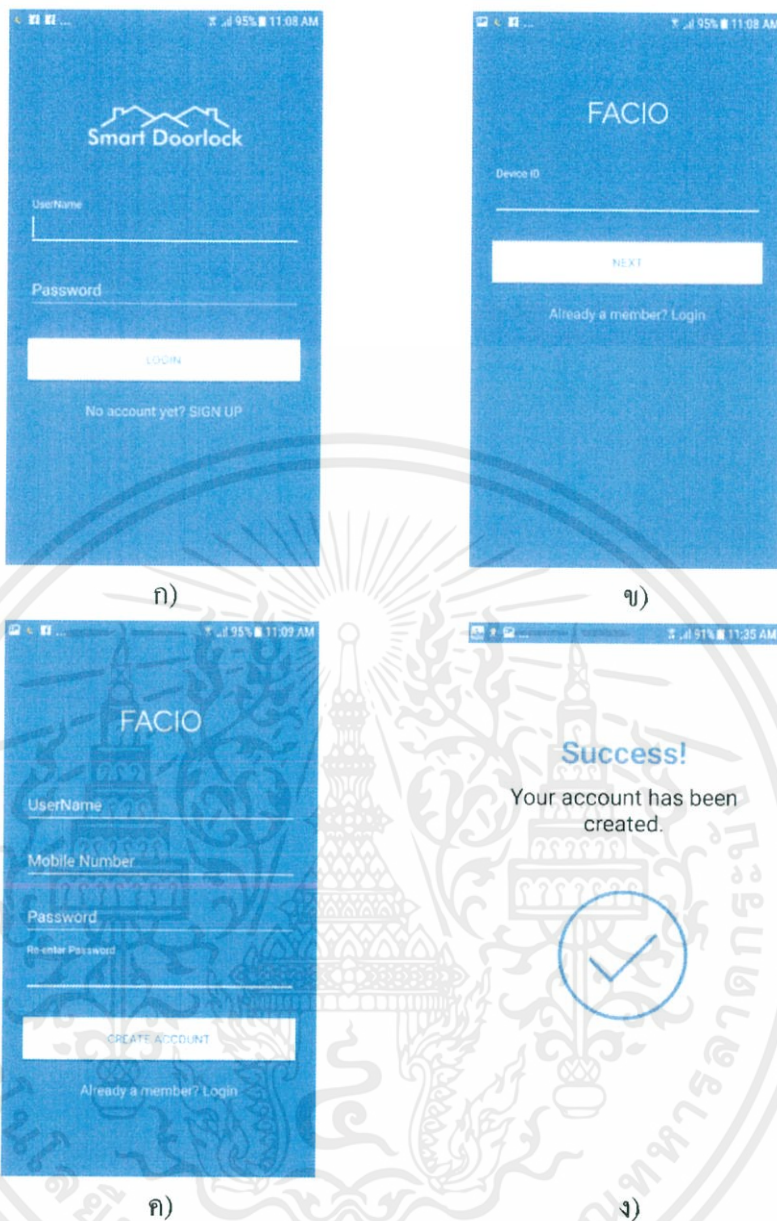
เพื่อทดสอบการสมัครสมาชิกเพื่อเข้าสู่แอปพลิเคชันของผู้ใช้งาน สำหรับการใช้งานแอปพลิเคชันครั้งแรก

4.4.2.2 ขั้นตอนการทดลอง

สำหรับการใช้งานแอปพลิเคชันครั้งแรก ผู้ใช้งานต้องทำการสมัครสมาชิกก่อน โดยกดปุ่ม “SIGN UP” ที่หน้าแรกของแอปพลิเคชัน จากนั้นทำการกรอก Device ID และ กดปุ่ม “NEXT” และกรอก User Name, Mobile Number, Password และ Re-enter Password แล้วกดปุ่ม “CREATE ACCOUNT” หลังจากสมัครสมาชิกเสร็จระบบจะแสดงผลสำเร็จและเข้าสู่แอปพลิเคชัน

4.4.2.3 ผลการทดลอง

แอปพลิเคชันสามารถเข้าสู่ระบบได้โดยเข้าสู่หน้า Home และสามารถแสดงข้อมูลได้อย่างครบถ้วน



รูปที่ 4.34 ส่วนแสดงผลการสมัครสมาชิก

- ก) หน้าแรกแสดงปุ่มสมัครสมาชิก
- ข) ส่วนการใส่ Device ID
- ค) ส่วนการใส่ข้อมูลการสมัครสมาชิก
- ง) ส่วนแสดงผลเมื่อทำการสมัครสมาชิกสำเร็จ

4.4.3 การปลดล็อกทางไกล

4.4.3.1 วัตถุประสงค์

เพื่อทดสอบการปลดล็อกทางไกลและตรวจการติดต่อระหว่างอุปกรณ์ต่างๆของระบบ

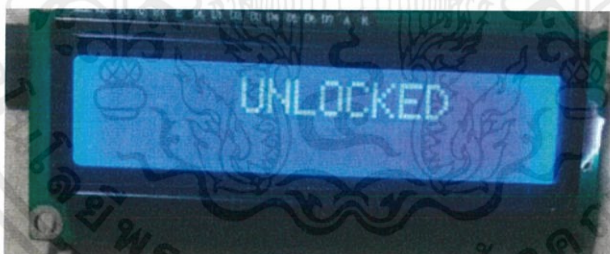
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.3.2 ขั้นตอนการทดลอง

กดปุ่ม “Unlock” ที่แสดงเป็น ImageButton ดังรูป 4.35

4.4.3.3 ผลการทดลอง

สามารถปลดล็อกทางไกลได้สำเร็จ โดยแสดงเป็นข้อความ “UNLOCKED” ที่หน้าจอ LCD



4.4.4 การตรวจสอบประวัติ (History)

4.4.4.1 วัตถุประสงค์

เพื่ออำนวยความสะดวกให้แก่ผู้ใช้งานในการตรวจสอบประวัติ ว่ามีบุคคลใดมาเยี่ยมเวลาใดบ้าง

4.4.4.2 ขั้นตอนการทดลอง

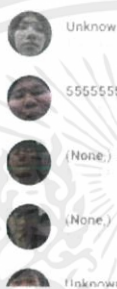
กดเลือกแถบ “History” ที่ Bottom Navigation Menu

4.4.4.3 ผลการทดลอง

แอปพลิเคชันสามารถแสดง list ของ History ได้อย่างถูกต้องครบถ้วนโดยแสดง ชื่อสมาชิก, ID Member, เวลา และรูปของบุคคลที่มาเยือน



Device ID: ABC1234



รูปที่ 4.37 ส่วนแสดงประวัติ (History)

4.4.5 การเรียกดูสมาชิก (Member)

4.4.5.1 วัตถุประสงค์

เพื่ออำนวยความสะดวกให้แก่ผู้ใช้งานในการเรียกดูว่ามีบุคคลใดเป็นสมาชิกบ้าง และดูรายละเอียดของสมาชิกแต่ละคนได้อย่างชัดเจน

4.4.5.2 ขั้นตอนการทดลอง

กดเลือกแถบ “Member” ที่ Bottom Navigation Menu

4.4.5.3 ผลการทดลอง

แอปพลิเคชันสามารถแสดง list ของสมาชิกได้อย่างถูกต้องครบถ้วน โดยแสดง ชื่อสมาชิก, ID Member และรูปของMemberแต่ละคน



รูปที่ 4.38 ส่วนแสดงสมาชิก

4.4.6 การแก้ไขข้อมูลของสมาชิก (Edit) และการลบสมาชิก (Delete)

4.4.6.1 วัตถุประสงค์

เพื่ออำนวยความสะดวกให้แก่ผู้ใช้งานในการจัดการข้อมูลของสมาชิกแต่ละคน

4.4.6.2 ขั้นตอนการทดลอง

กดเลือกสมาชิกที่ต้องการ หลังจากนั้นระบบจะแสดงข้อมูลของสมาชิกคนนั้นดังรูป 4.39 ก) หากต้องการลบสมาชิกคนนั้นให้กดปุ่ม “DELETE” และหากต้องการแก้ไขให้กดปุ่ม “EDIT” แล้วแก้ไขข้อมูล Name/Phone Number ที่ต้องการและกดปุ่ม “DONE ” ดังรูป 4.39 ข)

4.4.6.3 ผลการทดลอง

แอปพลิเคชันสามารถแสดงข้อมูลของสมาชิกคนที่เลือกได้อย่างถูกต้องครบถ้วน และสามารถทำการ Edit/Delete สมาชิกได้อย่างถูกต้อง



edit delete
name
New Member
phone

ก) ข)
รูปที่ 4.39 ส่วนแสดงข้อมูลสมาชิกและการจัดการสมาชิก

- ก) ส่วนแสดงข้อมูลสมาชิก ปุ่ม “EDIT” และปุ่ม “DELETE”
ข) ส่วนการแก้ไขข้อมูลสมาชิก

4.4.7 การแก้ไขข้อมูล Profile และ Logout

4.4.7.1 วัตถุประสงค์

เพื่ออำนวยความสะดวกให้กับผู้ใช้งานในการแสดง/แก้ไข Profile และ Logout

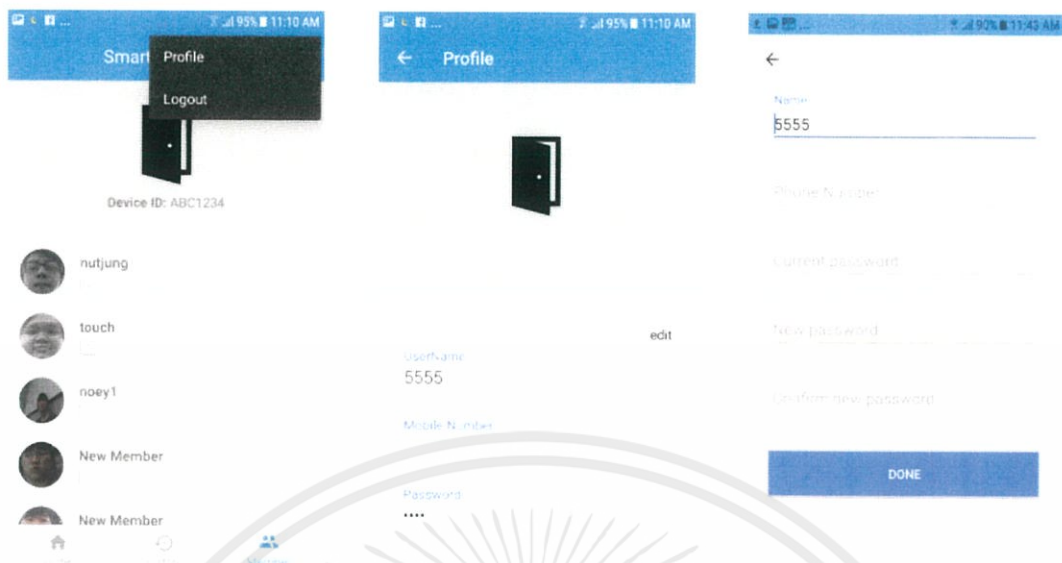
4.4.7.2 ขั้นตอนการทดลอง

เลือกแถบ Settings หากต้องการออกจากระบบให้กดเลือก Logout หากต้องการแสดงข้อมูล Profile กดเลือก Profile และหากต้องการแก้ไขให้กดปุ่ม “EDIT” แล้วแก้ไขข้อมูล Username, Mobile Number, Password ที่ต้องการและกดปุ่ม “DONE” ดังรูป 4.40

ก)

4.4.7.3 ผลการทดลอง

แอปพลิเคชันสามารถแสดงข้อมูลของผู้ใช้งานได้อย่างถูกต้อง



ก)

ข)

ค)

รูปที่ 4.40 การแก้ไขข้อมูล Profile และ Logout

- ก) ส่วนแสดงข้อมูลสมาชิก ปุ่ม “EDIT” และปุ่ม “DELETE”
- ข) ส่วนการแก้ไขข้อมูลสมาชิก
- ค) ส่วนแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผล

กลอนประจักษ์จำใบหน้า จะใช้โมดูลกล้องของบอร์ด Raspberry Pi ในการจับภาพใบหน้าของบุคคล ซึ่งเป็นระบบที่มีความสามารถในการจดจำใบหน้าได้อย่างแม่นยำ สามารถระบุได้ว่าบุคคลนั้นเป็นสมาชิกหรือไม่ ซึ่งถ้าหากเป็นสมาชิกกลอนประจักษ์จะปลดล็อกอัตโนมัติ แต่ถ้าหากไม่ใช่สมาชิก เจ้าของบ้านสามารถควบคุมการปลดล็อกกลอนประจักษ์ได้ทั้งจากระยะไกล ผ่านแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ซึ่งจะเชื่อมโยงกันผ่านบริการของคลาวด์ โดยจะมีระบบแจ้งเตือนโดย SMS เข้าสู่โทรศัพท์มือถือ เจ้าของบ้านสามารถตรวจสอบประวัติย้อนหลังได้ สามารถเพิ่ม-ลบจำนวนสมาชิกเองได้ ปัญหาและอุปสรรค

- 1) การศึกษาและเลือกใช้เครื่องมือในการพัฒนาระบบการจดจำใบหน้านั้นใช้เวลาในการศึกษาค่อนข้างมาก เนื่องจาก Library OpenCV ที่ใช้ในการพัฒนาระบบการจดจำใบหน้านั้น เมื่อทดลองแล้วให้ผลลัพธ์ที่ไม่ดีมากนักมีความแม่นยำในการระบุตัวตนน้อย จึงพยายามหา Library ตัวอื่น ๆ รวมไปถึงวิธีอื่นๆที่ให้ผลลัพธ์ที่ไม่คลาดเคลื่อน
- 2) ผลลัพธ์จากการจดจำใบหน้ามีความแม่นยำในการระบุตัวตนน้อย เนื่องจากแสงมีผลกระทบต่อการถอดรหัสภาพค่อนข้างมาก
- 3) การประมวลผลการจดจำใบหน้าของสองบุคคลขึ้นไป มีความแม่นยำในการระบุตัวตนน้อยเป็นผลมาจากใบหน้าที่มีลักษณะที่ใกล้เคียงกัน
- 4) เนื่องจากมีจำนวนภาพในระบบฐานข้อมูลเป็นจำนวนมาก เป็นผลให้การประมวลผลการจดจำใบหน้าใช้เวลาาน ซึ่งสามารถแก้ไขปัญหาได้โดยการลดขนาดของภาพ แต่การลดขนาดของภาพลง ก็ส่งผลกระทบต่อความแม่นยำในการระบุตัวตนด้วยเช่นกัน
- 5) เนื่องจากการพัฒนาระบบการจดจำใบหน้านั้นใช้เครื่องมือต่างๆที่หลากหลาย และเป็นเรื่องใหม่สำหรับคณะผู้จัดทำ จึงใช้เวลาานในการศึกษาเรียนรู้เครื่องมือ วิธีการใช้ และองค์ประกอบต่างๆที่ใช้ในการพัฒนาในส่วนของการออกแบบส่วนติดต่อกับผู้ใช้งานแอปพลิเคชันแอนดรอยด์ เพื่อให้ผู้ใช้งานสามารถใช้งานได้ง่ายนั้น จำเป็นต้องคำนึงถึงหลายปัจจัย คณะผู้จัดทำยังไม่มีหลักการที่ดีในการออกแบบ จึงใช้เวลาานในการศึกษาตัวอย่างการออกแบบ เพื่อให้ได้ส่วนติดต่อกับผู้ใช้ที่ดีและใช้งานง่าย

5.2 แนวทางการพัฒนาต่อ

- 1) การประมวลผลการจดจำใบหน้ามีความแม่นยำในการระบุตัวตนมากขึ้น และไม่มี ความผิดพลาดในการจดจำใบหน้า แม้จะทำการตรวจจับใบหน้า มากกว่า 2 บุคคลก็ ตาม
- 2) การประมวลผลการจดจำใบหน้า สามารถทำการวิเคราะห์และประมวลผลได้อย่าง รวดเร็ว
- 3) เพิ่มความสามารถต่างๆ ให้กับแอปพลิเคชันเพื่อให้สามารถทำงานได้แบบเรียลไทม์โดย แท้จริง กล่าวคือ สามารถตรวจสอบใบหน้าของคนแปลกหน้าได้ผ่านแอปพลิเคชัน ทันทีที่ระบบตรวจจับได้
- 4) เจ้าของบ้านสามารถเพิ่มอุปกรณ์กลอนประตูได้มากกว่า 1 อุปกรณ์



บรรณานุกรม

- [1] Qasim Hasan Mezher Al-shebani. **Embedded Door Access Control Systems Based on Face Recognition.** Faculty of Engineering and Information Sciences University of Wollongong , 2014
- [2] Tony DiCola. **Raspberry Pi Face Recognition Treasure Box.**
- [3] Pranav Aggarwal. Face Recognition Based Door Lock System Using Opencv and C# with Remote Access and Security Features. K.J Somaiya College Of Engineering, Vidyanagar.
- [4] 2014Philipp Wagner. **Face Recognition with Python.** 2012
- [5] อมรศักดิ์ อมรชนานันท์, มาลีรัตน์ โสคานิธิ. การรู้จำใบหน้าแบบหลายมุมมองโดยใช้เทคนิคผสมผสานการแบ่งภาพและการจับคู่ภาพมุมมองจริง. ภาควิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ. 2012
- [6] Timo Ojala, Matti Pietikäinen and Topi Mäenpää. **Multiresolution Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns.** Machine Vision and Media Processing Unit Infotech Oulu, University of Oulu P.O.Box 4500, FIN - 90014 University of Oulu
- [7] Di Huang, Caifeng Shan, Mohsen Ardebilian, Yunhong Wang, and Liming Chen. **Local Binary Patterns and Its Application to Facial Image Analysis: A Survey**

- [8] Md. Abdur Rahim, Md. Najmul Hossain, Tanzillah Wahid and Md. Shafiul Azam. **Face Recognition using Local Binary Patterns (LBP)**. Pabna University of Science and Technology, Bangladesh. 2013
- [9] จุฑามาศ มาบรรดิษ, หัตถยา อินตะวิกุล และ ปาริฉัตร เสริมวุฒิสาร. **การรู้จำอารมณ์บนใบหน้า โดยใช้วิธีการวิเคราะห์องค์ประกอบหลัก และการวิเคราะห์จำแนกประเภทเชิงเส้น**. ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ กาแพงแสน มหาวิทยาลัยเกษตรศาสตร์
- [10] Philipp Wagner. **facerec**. [ออนไลน์]. เข้าถึงได้จาก : <https://bytefish.de/dev/facerec> (วันที่ค้นข้อมูล: 9 ตุลาคม 2560).
- [11]AMZN. **Amazon Web Service**. [ออนไลน์]. เข้าถึงได้จาก : <https://aws.amazon.com/> (วันที่ค้นข้อมูล: 9 ตุลาคม 2560).
- [12]Google. **Android Studio**. [ออนไลน์]. เข้าถึงได้จาก : <https://developer.android.com/> (วันที่ค้นข้อมูล: 15 พฤศจิกายน 2560).
- [13]Adobe Systems. **Adobe Experience Design**. [ออนไลน์]. เข้าถึงได้จาก : <https://www.adobe.com/sea/> (วันที่ค้นข้อมูล: 15 พฤศจิกายน 2560).
- [14]Postdot Technologies. **Postman**. [ออนไลน์]. เข้าถึงได้จาก : <https://www.getpostman.com/company> (วันที่ค้นข้อมูล: 22 มกราคม 2561).