

ระบบล็อกเกอร์อัจฉริยะควบคุมผ่าน NodeMCU ESP-32S และ iOS
แอปพลิเคชัน
SMART LOCKER CONTROLLED BY NODEMCU ESP-32S AND iOS
APPLICATION



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

ระบบล็อกเกอร์อัจฉริยะควบคุมผ่าน NodeMCU ESP-32S และ iOS
แอปพลิเคชัน
SMART LOCKER CONTROLLED BY NODEMCU ESP-32S AND iOS
APPLICATION



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SMART LOCKER CONTROLLED BY NODEMCU ESP-32S AND
iOS APPLICATION



THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN MECHATRONICS ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2018

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2561

ภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบล็อกเกอร์อัจฉริยะควบคุมผ่าน NodeMCU ESP-32S และ iOS แอปพลิเคชัน
SMART LOCKER CONTROLLED BY NODEMCU ESP-32S AND iOS APPLICATION

ผู้จัดทำ นายจิรัฏฐ์ สีลาเกียรติวงศ์ 58010195
นายกীরติ พึ่งธรรมสาร 58010108



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบล็อกเกอร์อัจฉริยะควบคุมผ่าน NodeMCU ESP-32S และ iOS แอปพลิเคชัน

โดย

นายจิรัฏฐ์ สีลาเกียรติวงศ์ 58010195

นายกীরติ พึ่งธรรมสาร 58010108

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร.คงศักดิ์ อนันตหิรัญรัตน์

ผู้ช่วยศาสตราจารย์เทพจิตร เขยโกศา

ปีการศึกษา 2561

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการศึกษาการออกแบบและพัฒนาระบบควบคุมตู้ล็อกเกอร์อัจฉริยะผ่าน NodeMCU ESP-32S และ iOS แอปพลิเคชัน เพื่อลดต้นทุนในการทำตู้ล็อกเกอร์สาธารณะอย่างเห็นกันทุกวันนี้ เดิมทีในการควบคุมตู้ล็อกเกอร์จะควบคุมด้วยคอมพิวเตอร์ 1 เครื่อง โดยส่วนใหญ่ซึ่งส่งผลให้การจัดทำตู้ล็อกเกอร์สาธารณะมีค่าใช้จ่ายค่อนข้างสูงจึงศึกษาให้ระบบทั้งหมดของตู้ล็อกเกอร์ถูกควบคุมผ่านเครือข่ายไร้สาย หรือ IoT (Internet of Things) โดยตู้ล็อกเกอร์นี้สามารถใช้งานผ่านทาง iOS แอปพลิเคชันที่ได้จัดทำขึ้น โดยใช้การสแกน QR Code เป็นส่วนประกอบของระบบเพื่อเพิ่มความสะดวกรบายให้แก่ผู้ใช้งาน และยังมีส่วนของการลงทะเบียนเพื่อเข้าใช้งานเพื่อความปลอดภัยของตู้ล็อกเกอร์อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SMART LOCKER CONTROLLED BY NODEMCU ESP-32S AND iOS APPLICATION

By

Mr.Jirat Leelakiatiwong 58010195

Mr.Keerati Fungtammasan 58010108

Advisors

Asst.Prof.Dr. Kongsak Anuntahirunrat

Asst.Prof. Thepjit Cheypoca

Academic Year 2018

ABSTRACT

This thesis proposal is to study and develop the “Smart locker controlled by NodeMCU ESP-32S and iOS Application system” which help reduce the cost of making public lockers that we see nowadays. Almost every public lockers are controlled by computer which make its high costs. Therefore, we make the entire system of the locker controlled via the wireless network called IoT (Internet of Things). These lockers can be used by iOS applications that we created using QR Code scanning as a component of the system to make it convenience to users also the register and log in system for the security of the lockers too.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงได้เป็นอย่างดี ด้วยการให้ความดูแลและให้คำแนะนำของ ผู้ช่วยศาสตราจารย์ ดร.คงศักดิ์ อนันตหิรัญรัตน์ และ ผู้ช่วยศาสตราจารย์ เทพจิตร เขยโกคา ซึ่งเป็นอาจารย์ที่ปรึกษาในการจัดทำโครงงานนี้ ที่คอยชี้แนะแนวทางปรับปรุงแก้ไข ข้อบกพร่องต่างๆ ด้วยความเอาใจใส่ และให้ความกรุณาเอื้อเฟื้อสถานที่รวมถึงอุปกรณ์ที่จำเป็นต่างๆ ทำให้ปริญญานิพนธ์นี้เสร็จสมบูรณ์ คณะผู้จัดทำมีความซาบซึ้งในความกรุณาของอาจารย์ที่อนุเคราะห์ช่วยเหลือ และขอขอบพระคุณท่านเป็นอย่างสูง ขอขอบพระคุณอาจารย์ทุกท่านที่ช่วยอบรม สั่งสอน ชี้แนะแนวทางที่ดีแก่คณะผู้จัดทำเสมอมา ขอขอบพระคุณเพื่อนๆ ที่คอยช่วยเหลือและแก้ปัญหาในการทำงาน

สุดท้ายนี้คณะผู้จัดทำขอกราบขอบพระคุณบิดา และมารดา และครอบครัวญาติพี่น้อง ที่คอยอยู่ให้กำลังใจ และให้การสนับสนุนในหลายๆ อย่าง ทำให้โครงงานนี้สำเร็จลุล่วงไปได้จนมาเป็นปริญญานิพนธ์ฉบับนี้



ผู้จัดทำ

นายจิรัฏฐ์

ลีลาเกียรติวงศ์

นายกิริติ

พุงธรรมสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VIII
สารบัญตาราง	VI
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ประโยชน์ของโครงการ	1
1.4 ขอบเขตของโครงการ	2
1.5 แนวทางการดำเนินงาน	2
1.6 รายละเอียดของปริญญานิพนธ์	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	4
2.1 Internet of Things (IoT)	4
2.2 ไมโครคอนโทรลเลอร์	6
2.3 ฐานข้อมูล	9
2.4 Application Programming Interface (API)	12
2.5 Thunkable	13
2.6 คิวอาร์โค้ด (QR Code)	14
2.7 รีเลย์ (Relay)	15
2.8 หลักการทำงานเบื้องต้น	17

สารบัญ(ต่อ)

	หน้า
บทที่ 3 การออกแบบตู้ล็อคเกอร์อัจฉริยะ	19
3.1 การออกแบบด้านโครงสร้าง	19
3.2 การออกแบบวงจรไฟฟ้าควบคุมการเปิด/ปิด	20
3.3 การออกแบบ iOS แอปพลิเคชัน	21
3.4 การส่งข้อมูล iOS แอปพลิเคชัน	27
บทที่ 4 ผลการทดลองและการวิเคราะห์	32
4.1 การทดลองใช้งานตู้ล็อคเกอร์อัจฉริยะ	32
4.2 การทดลองการสแกน QR Code	39
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ	40
5.1 การดำเนินงานจัดทำปฏิญานิพนธ์	40
5.2 ปัญหาที่พบและแนวทางการแก้ไข	40
5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ	41
เอกสารอ้างอิง	42
ภาคผนวก	43
ประวัติผู้วิจัย	63

สารบัญรูป

รูปที่	หน้า
2.1 Diagram อธิบายการเชื่อมต่อ Gateway เข้ากับ Local Network	5
2.2 PINOUT ของบอร์ด NodeMCU-32S	7
2.3 Chip ESP-WROOM32	8
2.4 โลโก้ Firebase	10
2.5 ภาพรวม Service Architecture ของ Cloudinary	11
2.6 ตัวอย่างการจัดการข้อมูลด้วย Airtable	12
2.7 API	13
2.8 หน้า Design ของ Thunkable	13
2.9 ตัวอย่าง Block ของ Thunkable	14
2.10 ตัวอย่างคิวอาร์โค้ด (QR Code)	15
2.11 ภาพตัวอย่าง Relay แบบ 4 Channel	16
2.12 วงจรการทำงานของ Relay	16
2.12 NodeMCU ESP-32S เชื่อมต่อกับอุปกรณ์ต่างๆ	17
2.13 การรับส่งข้อมูลระหว่าง NodeMCU ESP-32S กับแอปพลิเคชัน	18
3.1 การออกแบบโครงสร้างตู้ล็อคเกอร์	19
3.2 การออกแบบวงจรไฟฟ้า	20
3.3 วงจรไฟฟ้าติดตั้ง LED RGB Strip	20
3.4 Flowchart iOS แอปพลิเคชัน	21
3.5 แอปพลิเคชันหน้าล๊อคอิน (Log in)	22
3.6 แอปพลิเคชันหน้าลงทะเบียน (Sign up)	23
3.7 แอปพลิเคชันหน้าโฮม (Home)	24
3.8 แอปพลิเคชันหน้าสแกน (Scan)	25
3.9 หน้า PIN ของแอปพลิเคชัน	26
3.10 การเก็บข้อมูลผู้สมัครใน Firebase	27
3.11 การเก็บข้อมูลในส่วนของการทำงานของล็อคเกอร์	28

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.12 การเก็บข้อมูลใน Airtable	28
3.13 การเก็บข้อมูลใน Cloundinary	29
3.14 การทำงานของ Web API	29
3.15 Login & Sign up Flowchart	30
3.16 Scanning Flowchart	31
4.1 ข้อมูลที่ใส่ในการลงทะเบียน	32
4.2 ข้อมูล Firebase	33
4.3 ข้อมูลใน Airtable	33
4.4 อีเมลเพื่อยืนยันการลงทะเบียน	33
4.5 การ Log in เข้าสู่ระบบ	34
4.6 หน้าจอแสดงการใช้งานตู้ล็อกเกอร์	34
4.7 การตั้งค่าอ่านข้อมูล QR1 ใน Firebase	34
4.8 QR Code ของ QR1	35
4.9 รูปที่ใช้ในการสแกน QR Code	35
4.10 รูปที่สแกนที่ปรากฏในฐานข้อมูล Cloundinary	36
4.11 ใส่ PIN เพื่อใช้งานตู้ล็อกเกอร์	36
4.12 ข้อมูลการใช้งานตู้ล็อกเกอร์ใน Firebase	37
4.13 หน้าจอแสดงการใช้งานตู้ล็อกเกอร์	37
4.14 หน้าจอปรากฏข้อความการใช้งาน	38
4.15 ข้อมูลการใช้งานตู้ล็อกเกอร์ใน Firebase	38
4.16 หน้าจอแสดงการใช้งานตู้ล็อกเกอร์	38

สารบัญตาราง

ตารางที่	หน้า
2.1 การส่งข้อมูลของแต่ละ Access Technologies	5
2.2 ความจุข้อมูล QR Code	15
4.1 ผลการทดลองการสแกน QR Code	39



บทที่ 1

บทนำ

ในบทนี้จะกล่าวถึงความเป็นมาและความสำคัญของปฏิญานิพนธ์ วัตถุประสงค์ของโครงการ ขอบเขตของโครงการ ประโยชน์ที่คาดว่าจะได้รับ และรายละเอียดของปฏิญานิพนธ์ โดยมีรายละเอียดที่ได้กล่าวถึงดังต่อไปนี้

1.1 ความเป็นมาและความสำคัญของปฏิญานิพนธ์

Internet of Things (IoT) เป็นนวัตกรรมที่เชื่อมต่อทุกสิ่งอย่างเข้าสู่อินเทอร์เน็ต เพื่อให้สิ่งเหล่านั้นสามารถทำงานได้อย่างชาญฉลาดผ่านโพรโทคอลการสื่อสารทั้งแบบใช้สายและไร้สาย โดยสิ่งของต่างๆ มีวิธีการระบุตัวตนได้รับรู้สภาพแวดล้อมได้ และมีการโต้ตอบและทำงานร่วมกันได้ ความสามารถในการสื่อสารนี้จะนำไปสู่นวัตกรรมและบริการใหม่อีกมากมาย ซึ่ง IoT สามารถนำมาประยุกต์ใช้ได้กับหลายๆ สิ่งอย่างกว้างขวาง เช่น การใช้งานภายในครัวเรือน หรือการใช้งานตามสาธารณะ เพื่อให้ผู้ใช้งานมีความสะดวกสบายมากยิ่งขึ้น และบางครั้งก็ใช้เพื่อลดต้นทุนในการทำงาน ยกตัวอย่างเช่น ตู้ลิ้นชักเกอร์ตามที่สาธารณะ ตู้ลิ้นชักเกอร์เหล่านั้นมีการใช้คอมพิวเตอร์เพื่อควบคุมระบบทั้งหมดนั้นจึงทำให้มีราคาที่สูง

ในทางกลับกันหากนำ IoT มาประยุกต์ใช้งานกับตู้ลิ้นชักเกอร์ โดยให้ผู้ใช้งานสามารถใช้งานตู้ลิ้นชักเกอร์ โดยใช้เพียงโทรศัพท์เครื่องเดียว และควบคุมระบบของตู้ลิ้นชักเกอร์ด้วยอินเทอร์เน็ตเพื่อลดต้นทุนและยังช่วยเพิ่มคุณภาพ อีกทั้งยังทำให้ระบบมีความปลอดภัยมากยิ่งขึ้น และทำให้ผู้ใช้งานมีความสะดวกสบายมากยิ่งขึ้นเช่นกัน

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อพัฒนาระบบของตู้ลิ้นชักเกอร์สาธารณะที่ทำงานด้วยระบบ IoT
2. เพื่อพัฒนาความรู้ทางด้านแอปพลิเคชันของมือถือ และความรู้ทางด้าน IT
3. เพื่อประยุกต์ความรู้ทางด้านคอมพิวเตอร์และอิเล็กทรอนิกส์เข้าด้วยกัน

1.3 ประโยชน์ของโครงการ

1. สามารถนำระบบมาใช้ได้จริงอย่างมีประสิทธิภาพ
2. สามารถลดต้นทุนของระบบตู้ลิ้นชักเกอร์ที่เห็นตามที่สาธารณะ
3. เพิ่มความสะดวกสบายให้แก่ผู้ใช้งาน
4. ได้รับความรู้ด้านการเขียนแอปพลิเคชัน ซึ่งสามารถมาประยุกต์ใช้กับงานอื่นๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขอบเขตของโครงการ

1. ระบบสามารถสั่งการเปิด-ปิดได้ตามที่ผู้ใช้งานต้องการ โดยผู้ใช้งานสามารถใส่รหัสผ่านของตัวเองเพื่อความปลอดภัยของการทำงาน ทั้งการสร้างรหัสผ่านเพื่อปิดล็อกเกอร์ และยืนยันรหัสผ่านเพื่อเปิดล็อกเกอร์เช่นกัน
2. ผู้ใช้งานสามารถลงทะเบียนสร้าง Username และ Password ของตัวเองเพื่อใช้งานล็อกเกอร์ โดยผู้ใช้งานจำเป็นต้องยืนยันอีเมล ที่ผู้ใช้ได้ลงทะเบียนไว้ โดยระบบจะส่งลิงค์ยืนยันตัวตนผ่านทางอีเมล และผู้ใช้งานจำเป็นต้องเข้าไปที่ลิงค์นั้น เพื่อให้ลงทะเบียนสำเร็จ
3. ระบบสามารถทราบได้ว่าล็อกเกอร์ตู้ใดมีการใช้งานอยู่ เพื่อป้องกันไม่ให้ผู้ใช้งานท่านอื่นสามารถมาใช้ล็อกเกอร์ที่ผู้ใช้งานใช้อยู่ได้
4. ผู้ใช้งานสามารถใช้งานตู้ล็อกเกอร์ ด้วยการสแกน QR Code ที่ติดไว้ในตู้ล็อกเกอร์แต่ละตู้ โดยแต่ละตู้จะมี QR Code ที่ไม่เหมือนกัน
5. ระบบสามารถส่งข้อมูลทั้งหมดของล็อกเกอร์ ไปยังเซิร์ฟเวอร์ได้ทันทีเมื่อมีข้อมูลใหม่เกิดขึ้นด้วยการเชื่อมต่อผ่านอินเทอร์เน็ต

1.5 แนวทางการดำเนินงาน

1. ศึกษาค้นคว้าข้อมูลที่เกี่ยวข้องในขั้นตอนนี้ ผู้พัฒนาได้ทำการศึกษาการทำงานของ IoT ศึกษาการเขียนแอปพลิเคชันของมือถือ และศึกษาหาอุปกรณ์ที่จำเป็นต้องใช้
2. จัดเตรียมวัสดุที่เกี่ยวข้องกับการทำโครงการ ในขั้นตอนนี้จะทำการเตรียมอุปกรณ์ที่เกี่ยวข้อง ได้แก่ Microprocessor ตู้ล็อกเกอร์ที่จะใช้ในการติดตั้งระบบรีเลย์ รวมถึงค้นหาโปรแกรมที่จะใช้ในการเขียนแอปพลิเคชัน และหาเซิร์ฟเวอร์ที่จะใช้ในการเก็บข้อมูลบนฐานข้อมูลออนไลน์
3. ออกแบบแอปพลิเคชันและระบบในขั้นตอนนี้ จะทำการออกแบบระบบที่ต้องการ โดยการเขียน Flowchart ของแอปพลิเคชัน รวมถึงการออกแบบการจ่ายไฟให้ตู้ล็อกเกอร์หลังจากการสแกน QR Code
4. พัฒนาระบบการสแกน QR Code เพื่อใช้งานล็อกเกอร์ในขั้นตอนนี้ จะทำการเขียนแอปพลิเคชันที่ได้ออกแบบข้างต้น โดยจะเริ่มจากการเขียนในส่วนของการสแกน QR Code โดยจะออกแบบให้เมื่อมีการสแกน QR Code เกิดขึ้นจะส่งข้อมูลไปยังฐานข้อมูลออนไลน์ และระบบการตั้งรหัสผ่านหลังจากสแกนเพื่อสั่งให้ล็อกเกอร์เปิด หรือปิด และทดลองการใช้งานด้วยอุปกรณ์จริงเพื่อตรวจสอบว่าสามารถทำงานจริงได้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. พัฒนาระบบการสมัครสมาชิกเพื่อเข้าใช้ตู้ล็อกเกอร์ ในขั้นตอนนี้จะทำการเขียนแอปพลิเคชัน ให้ระบบสามารถจดจำข้อมูลของผู้ใช้งานที่ลงทะเบียนไว้ และส่งอีเมลไปให้ผู้ลงทะเบียน แล้วนำมารวมเข้ากับระบบการสแกน QR Code เพื่อให้หลังจากเข้าสู่ระบบสามารถทำการสแกนได้

6. ติดตั้งอุปกรณ์เข้ากับตู้ล็อกเกอร์ในขั้นตอนนี้จะทำการติดตั้งอุปกรณ์ต่างๆ กับตู้ล็อกเกอร์ โดยพิจารณาจากความสูงของตู้ล็อกเกอร์ กับตำแหน่งของตัวล็อกให้สามารถใช้งานได้อย่างราบรื่น

7. ออกแบบหน้าใช้งานในขั้นตอนนี้จะออกแบบหน้าใช้งานให้ดูน่าใช้ และสามารถใช้งานได้ง่าย โดยจะออกแบบให้ผู้ที่มาใช้งานครั้งแรกสามารถใช้งานได้เลย โดยไม่จำเป็นต้องศึกษาอะไรเพิ่มเติม

8. ตรวจสอบการใช้งานและปรับปรุงในขั้นตอนนี้ จะทำการตรวจสอบระบบเพื่อค้นหาข้อผิดพลาดต่างๆ ในการทำงาน และแก้ไขไม่ให้เกิดปัญหา รวมถึงการปรับปรุงให้แอปพลิเคชันมีความเสถียรมากขึ้น

1.6 รายละเอียดของปฏิญญานิพนธ์

ปฏิญญานิพนธ์นี้ประกอบด้วย 5 บท ดังนี้

บทที่ 1 บทกล่าวนำวัตถุประสงค์ของโครงการ ขอบเขตในการค้นคว้า ประโยชน์ที่คาดว่าจะได้รับ และรายละเอียดของโครงการ

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง และความรู้เบื้องต้นเกี่ยวกับ IoT, ไมโครคอนโทรลเลอร์, ฐานข้อมูลออนไลน์ และ QR Code

บทที่ 3 หลักการออกแบบโครงสร้าง วงจรควบคุม และแอปพลิเคชันทั้งหมดที่เกี่ยวข้องกับตู้ล็อกเกอร์อัจฉริยะ

บทที่ 4 ผลการทดลองของการทำงานของระบบของตู้ล็อกเกอร์อัจฉริยะ และการทดลองความเร็วในการสแกน QR Code ของแอปพลิเคชัน

บทที่ 5 สรุปผลการดำเนินงาน และแนวทางการพัฒนา

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงความรู้ทั่วไปเกี่ยวกับ IoT (Internet of Things), ไมโครคอนโทรลเลอร์, ซอฟต์แวร์ที่ใช้ในการเขียนแอปพลิเคชัน, ฐานข้อมูลออนไลน์, QR Code และอุปกรณ์ที่ใช้ โดยมีรายละเอียดที่จะได้กล่าวดังต่อไปนี้

2.1 Internet of Things (IoT)

เทคโนโลยี Internet of Things (IoT) หรือ “อินเทอร์เน็ตในทุกสิ่ง” หมายถึง การที่สิ่งต่างๆ ถูกเชื่อมโยงทุกอย่างเข้าสู่โลกอินเทอร์เน็ต ทำให้มนุษย์สามารถสั่งการควบคุมใช้งานอุปกรณ์ต่างๆ ผ่านทางเครือข่ายอินเทอร์เน็ต เช่น การสั่งเปิด-ปิด อุปกรณ์เครื่องใช้ไฟฟ้า รถยนต์ โทรศัพท์มือถือ เครื่องมือสื่อสาร เครื่องใช้สำนักงาน เครื่องมือทางการแพทย์ เครื่องจักรในโรงงาน อุตสาหกรรม อาคาร บ้านเรือน เครื่องใช้ในชีวิตประจำวันต่างๆ ผ่านเครือข่ายอินเทอร์เน็ต เป็นต้น โดยเทคโนโลยีนี้จะเป็นทั้งประโยชน์อย่างมหาศาล และความเสี่ยงไปพร้อมๆ กัน เพราะหากระบบรักษาความปลอดภัยของอุปกรณ์และเครือข่ายอินเทอร์เน็ตไม่ดีพอ จะทำให้ผู้ไม่ประสงค์ดีเข้ามากระทำการที่ไม่พึงประสงค์ต่ออุปกรณ์ข้อมูลสารสนเทศหรือความเป็นส่วนตัวของคุณได้ ดังนั้นการพัฒนาไปสู่ Internet of Things จึงมีความจำเป็นต้องพัฒนามาตรการและเทคนิคในการรักษาความปลอดภัยไอทีควบคู่กันไปด้วย

ตัวแปรสำคัญสำหรับ Internet of Things ที่ใช้ในการสื่อสารนั้นไม่เพียงแต่ Internet Network เพียงเท่านั้น แต่ยังมีตัวแปรอื่นเข้ามาเกี่ยวข้องอีกนั่นคือ Sensor Node ต่างๆ จำนวนมากที่ทำให้เกิด Wireless Sensor Network (WSN) ให้กับอุปกรณ์ต่างๆ สามารถเชื่อมต่อเข้ามาได้ ซึ่ง WSNs นี้เองสามารถตรวจจับปรากฏการณ์ต่างๆ (Physical Phenomena) ในเครือข่ายได้ด้วย ยกตัวอย่างเช่น แสง อุณหภูมิ ความดัน เป็นต้น เพื่อส่งค่าไปยังอุปกรณ์ในระบบให้ทำงานหรือสั่งงานอื่นๆ ต่อไป

การพัฒนา Internet of Things นั้นนอกจากจะพัฒนาเทคโนโลยีในฝั่ง Hardware ได้แก่ Processors, Radios และ Sensors ซึ่งจะถูกรวมเข้าด้วยกันเรียกว่า A Single Chip หรือ System on A Chip (SoC) แล้วก็ยังพัฒนา WSN ไปพร้อมๆ กันด้วย และเมื่อพูดถึงการเชื่อมต่อปัจจุบันได้มีการพัฒนาเทคโนโลยีสำหรับการเชื่อมต่อสำหรับ Internet of Things หรือ Access Technology มีอยู่ 3 ตัวได้แก่

1. Bluetooth 4.0
2. IEEE 802.15.4e

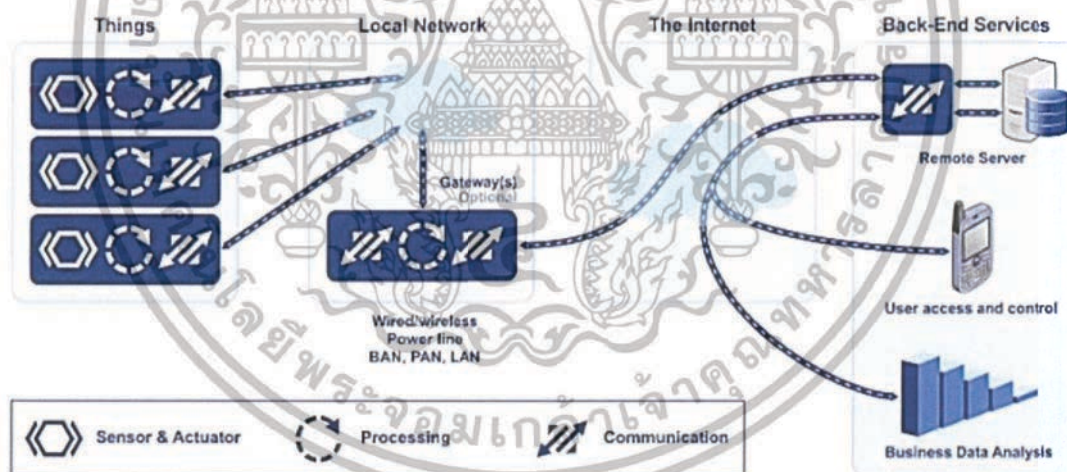
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. WLAN IEEE 802.11™ (WiFi) มาตรฐานการทำงานของระบบเครือข่ายไร้สาย โดยในแต่ละ Access Technologies นั้นมีการส่งข้อมูลที่แตกต่างกันดังตารางที่ 2.1

ตารางที่ 2.1 การส่งข้อมูลของแต่ละ Access Technologies

	IEEE 802.15.4e	Bluetooth	WLAN IEEE 802.11
Frequency	868/915 MHz 2.4 GHz	2.4 GHz	2.4, 5.8 GHz
Data Rate	250 Kbps	723 Kbps	11-105 Mbps
Power	Very Low	Low	High

เมื่อมีโครงข่าย Sensor Nodes แล้วก็จำเป็นจะต้องมี Gateway Sensor Nodes เพื่อจะเชื่อมต่อไปยังโลกอินเทอร์เน็ตด้วย โดยตัว Gateway นี้จะทำหน้าที่เชื่อมต่อไปยังเครือข่าย Internet ให้อุปกรณ์ทั้งหมดในโครงข่าย Sensor Nodes ทั้งหมดส่งข้อมูลเข้าสู่อินเทอร์เน็ตได้นั่นเอง และ Gateway ที่ว่านี้ก็จะต้องอยู่ภายใต้ Local Network ซึ่งจะมีการกำหนดกันต่อไปว่า Gateway ภายใต้อันไหนจะให้อุปกรณ์ที่เชื่อมต่อไปยังอินเทอร์เน็ตได้ด้วยหรือไม่ถ้าไม่ได้อุปกรณ์ที่เชื่อมต่อเข้ามาใน Gateway ก็อาจจะสื่อสารกันได้เฉพาะภายใน Local Network เองได้เท่านั้นดังรูปที่ 2.1



รูปที่ 2.1 Diagram อธิบายการเชื่อมต่อ Gateway เข้ากับ Local Network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัจจุบันมีการแบ่งกลุ่ม Internet of Things ออกตามตลาดการใช้งานเป็น 2 กลุ่ม ได้แก่

1. Industrial IoT คือ แบ่งจาก Local Network ที่มีหลายเทคโนโลยีที่แตกต่างกันในโครงข่าย Sensor Nodes โดยตัวอุปกรณ์ IoT Device ในกลุ่มนี้จะเชื่อมต่อแบบ IP Network เพื่อเข้าสู่อินเทอร์เน็ต

2. Commercial IoT คือ แบ่งจาก Local Communication ที่เป็น Bluetooth หรือ Ethernet (Wired or Wireless) โดยตัวอุปกรณ์ IoT Device ในกลุ่มนี้จะสื่อสารภายในกลุ่ม Sensor Nodes เดียวกันเท่านั้นหรือเป็นแบบ Local Devices เพียงอย่างเดียวอาจไม่ได้เชื่อมสู่อินเทอร์เน็ต

2.2 ไมโครคอนโทรลเลอร์

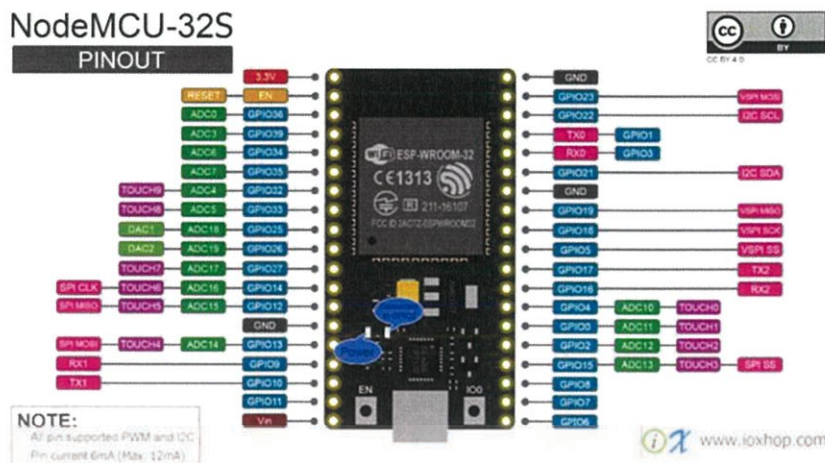
2.2.1 NodeMCU ESP-32S

บอร์ด Node32 ESP-32S เป็นบอร์ดที่นำ SoC อย่าง ESP-32 ของบริษัท Espressif มาออกแบบเป็นบอร์ดพัฒนา โดย ESP-32S เป็นไอซีที่พัฒนาเพิ่มเติมจาก ESP8266 ที่นิยมอย่างมาก เนื่องจากมันเป็นไมโครคอนโทรลเลอร์ที่มีราคาถูก และมี WiFi ติดมาด้วย แถมยังใช้ Arduino เขียนโปรแกรมได้ด้วย แต่สำหรับ ESP-32S ตัวใหม่นี้ยังเพิ่มความสามารถเข้าไปอีก มีการเชื่อมต่อ Bluetooth Low-Energy (BLE, BT4.0, Bluetooth Smart) เข้าไปอีก และยังได้เพิ่มเป็น GPIO ถึง 30 ขา ดังรูปที่ 2.2

ภาพรวมของ Node32S

- ใช้ ESP-WROOM-32 จาก Espressif ซึ่งเป็น WiFi/BLE SoC (System On Chip)
- Breadboard Friendly มีขนาดกว้าง 0.9" วางบน Breadboard จะเหลือข้างละ 1 ช่อง
- ใช้ USB2Serial ตระกูล FTDI ชิปเพื่อการโหลดโปรแกรมแบบอัตโนมัติ ความเร็วสูงสุดถึง 921000
- JST 2 mm Connector สำหรับเสียบแบตเตอรี่
- มีวงจรชาร์จ Lithium Ion และ Lithium Polymer (1 Cell) พร้อมทั้งไฟแสดงสถานะ
- มีวงจร PTC Fuse ตัดกระแสไฟเกินที่ 500 mA
- 3.3 V 600 mA On-board Voltage Regulator
- Push Button Switch ที่ขา IO0 และ EN (Reset)
- Crystal 32.768 KHz เพื่อใช้เลี้ยงวงจร RTC
- เหมาะสำหรับงาน พัฒนาด้านแบบ อุปกรณ์รูปแบบ Portable และ Wearable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 PINOUT ของบอร์ด NodeMCU ESP-32S

ภาพรวม ESP-32S

จากรูปที่ 2.3 โมดูล-ESP-WROOM32 ถือเป็นหัวใจหลักของบอร์ด NodeMCU ESP-32S ซึ่งจะมีความสามารถดังต่อไปนี้

- สองแกนหน่วยประมวลผล Tensilica LX6
- ความเร็วสูงสุด 240 MHz
- แรมภายใน 520 kB
- WiFi Transceiver B/G/N
- Bluetooth Dual-mode Classic และ BLE
- ช่วงไฟ 2.2 V ถึง 3.6 V
- กินกระแส 2.5 μ A (ในสถานะ Deep Sleep)
- 32 GPIO
- 10 ช่องของ Capacitive Touch
- 18 ช่องของ Analog-to-Digital Converter (ADC)
- ช่องของ SPI อินเทอร์เฟส
- ช่องของ UART อินเทอร์เฟส
- 2 ช่องของ I²C อินเทอร์เฟส
- 16 ช่องของ PWM
- 2 ช่องของ Digital-to-Analog Converters (DAC)
- 2 ช่องของ I²S อินเทอร์เฟส
- Hardware Accelerated Encryption (AES, SHA2, ECC, RSA-4096)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 Chip ESP-WROOM32

WiFi ถือเป็นหัวใจสำคัญ และเป็นจุดเด่นของ ESP-32S ด้วยการรวมส่วนของ WiFi มาในชิปพร้อมกับไมโครคอนโทรลเลอร์ทำให้สามารถประหยัดพื้นที่โดยรวมของระบบได้ รวมทั้งทำให้สามารถจัดการเรื่องของพลังงานได้ง่ายมากยิ่งขึ้น นอกจากนี้การใช้งาน WiFi ใน ESP-32S ยังง่ายกว่าชิป WiFi ตัวอื่นมาก โดยมีคุณสมบัติพิเศษคือ สามารถเลือกโหมดการใช้งาน WiFi ได้ 3 โหมดคือ โหมด AP (Access Point) โหมด STA (Station) และโหมด AP+STA ซึ่งทั้ง 3 โหมดจะมีการใช้งานที่แตกต่างกันเล็กน้อย การเลือกโหมดใช้งานจะเลือกตามลักษณะงานที่นำไปใช้เป็นหลัก

2.2.2 โหมด AP

AP ย่อมาจาก Access Point เป็นโหมดที่จะใช้ ESP-32S เป็นตัวปล่อยสัญญาณ WiFi ออกไป เพื่อให้อุปกรณ์ต่างๆ มาเชื่อมต่อด้วย ทั้งนี้ ESP-32S จะรองรับอุปกรณ์ที่มาเชื่อมต่อด้วยเพียง 1 ตัวเท่านั้น เหมาะสำหรับงานที่ไม่ต้องการใช้งานอินเทอร์เน็ตหรือติดต่อกับอุปกรณ์อื่นในวงแลน นอกจากนี้โหมดนี้ยังเหมาะกับการนำไปควบคุมอุปกรณ์ต่างๆ ผ่าน WiFi อีกด้วย ซึ่งงานควบคุมที่สามารถย้ายที่ได้ และไม่ต้องใช้งานผ่านอินเทอร์เน็ตจะเหมาะกับการใช้งาน ESP-32S ในโหมด AP มาก

2.2.3 โหมด STA

STA ย่อมาจาก Station เป็นโหมดที่จะใช้ ESP-32S ไปเชื่อมต่อกับอุปกรณ์ปล่อยสัญญาณอื่นๆ เช่น Router โทรศัพท์มือถือที่เปิด Hotspot การใช้งานในโหมดนี้นิยมใช้กับงานที่ต้องการเชื่อมต่ออินเทอร์เน็ต หรือมีการสื่อสารกับอุปกรณ์หลายๆ ในวงแลน ในงานด้าน IoT Smart Home และ Smart Farm มักใช้งานในโหมดนี้เพื่อส่งข้อมูลจากเซนเซอร์ขึ้นไปบนระบบ Cloud และใช้โหมดนี้เพื่อเชื่อมต่อกับระบบ Cloud รับคำสั่งมีสิ่งอุปกรณ์ต่างๆ ให้ทำงานผ่านอินเทอร์เน็ต

2.2.4 โหมด AP+STA

เป็นโหมดที่จะให้โมดูล WiFi ภายใน ESP-32 ทำงานทั้ง 2 โหมด นั่นคือ AP และ STA ร่วมกัน ข้อดีของการใช้โหมดนี้คือ อุปกรณ์ภายนอกสามารถเชื่อมต่อเข้ามาที่ ESP-32S ได้ และ ESP-32S สามารถไปเชื่อมต่อกับอุปกรณ์ภายนอกได้ ในโหมดนี้มักใช้กับงานที่ต้องการใช้งานอินเทอร์เน็ต

และมีการเคลื่อนย้ายบ่อยๆ หรือสามารถเปลี่ยนสถานที่ใช้งานได้ โดยในโหมด AP จะทำให้อุปกรณ์ต่างๆ สามารถเข้ามาตั้งค่าการเชื่อมต่อ WiFi กับ Router หรืออุปกรณ์อื่นๆ ได้ ในโหมด STA ก็จะทำหน้าที่ตั้งไว้มาใช้เชื่อมต่อกับ Router หรืออุปกรณ์อื่นๆ ที่ได้ตั้งค่าไว้

โดยที่นี้จะเลือกใช้โหมด STA หรือ Station ในการทำงานเนื่องจากมีการ Online ซึ่งทำให้สามารถอัปโหลดข้อมูลขึ้น Cloud ส่งงาน, แก้ไข, เรียกดูข้อมูล เรียกได้ว่าอิสระมากๆ ต่อการใช้งาน และยังสามารถสื่อสารกับอุปกรณ์หลายตัวในวงแลนได้อีกด้วย

2.3 ฐานข้อมูล

ฐานข้อมูล หรือ Database คือ กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ซึ่งกันและกัน โดยไม่ได้บังคับว่าข้อมูลทั้งหมดนี้จะต้องเก็บไว้ในแฟ้มข้อมูลเดียวกันหรือแยกเก็บหลายๆ แฟ้มข้อมูลระบบฐานข้อมูล (Database System) คือ ระบบที่รวบรวมข้อมูลต่างๆ ที่เกี่ยวข้องกันเข้าไว้ด้วยกันอย่างมีระบบมีความสัมพันธ์ระหว่างข้อมูลต่างๆ ที่ชัดเจน ในระบบฐานข้อมูลจะประกอบด้วยแฟ้มข้อมูลหลายแฟ้มที่มีข้อมูล เกี่ยวข้องสัมพันธ์กันเข้าไว้ด้วยกันอย่างเป็นระบบ และเปิดโอกาสให้ผู้ใช้สามารถใช้งาน และดูแลรักษาป้องกันข้อมูลเหล่านี้ ได้อย่างมีประสิทธิภาพ โดยมีซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่างผู้ใช้และโปรแกรมต่างๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล เรียกว่า ระบบจัดการฐานข้อมูล หรือ DBMS (Data Base Management System) มีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่ายสะดวกและมีประสิทธิภาพ การเข้าถึงข้อมูลของผู้ใช้อาจเป็นการสร้างฐานข้อมูล การแก้ไขฐานข้อมูล หรือการตั้งค่าถามเพื่อให้ได้ข้อมูลมา โดยผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างของฐานข้อมูล

2.3.1 Firebase

จากรูปที่ 2.4 Firebase Realtime Database เป็น NoSQL Cloud Database ที่เก็บข้อมูลในรูปแบบของ JSON และมีการ Sync ข้อมูลแบบ Realtime กับทุก Devices ที่เชื่อมต่อแบบอัตโนมัติ ในเสี้ยววินาที รองรับการทำงานเมื่อ Offline (ข้อมูลจะถูกเก็บไว้ใน Local จนกระทั่งกลับมา Online ก็จะทำการ Sync ข้อมูลให้อัตโนมัติ) รวมถึงมี Security Rules ให้สามารถออกแบบเงื่อนไขการเข้าถึงข้อมูลทั้งการ Read และ Write ได้ตั้งใจ ทั้ง Android, iOS และ Web



รูปที่ 2.4 โลโก้ Firebase

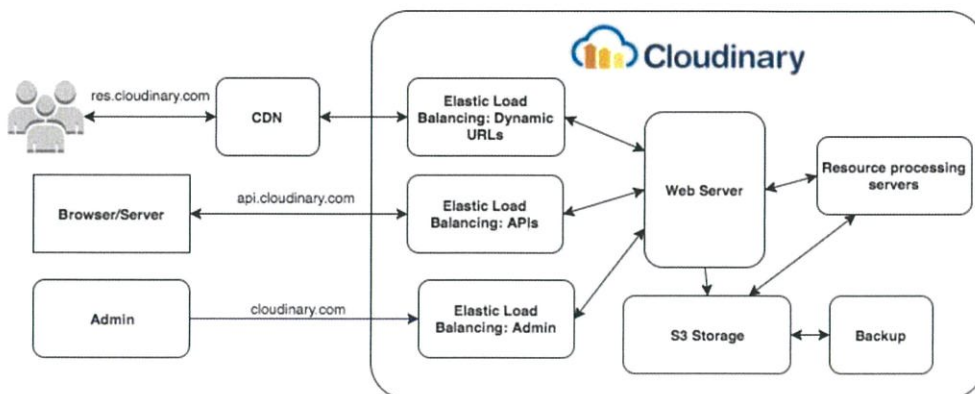
2.3.2 Cloudinary

Cloudinary คือ Software-as-a-Service (SaaS) หรือซอฟต์แวร์แอปพลิเคชันที่ให้บริการผ่านอินเทอร์เน็ต Cloudinary เป็นแพลตฟอร์มที่ช่วยในการจัดการรูปภาพ และวิดีโอสำหรับเว็บไซต์และแอปพลิเคชัน Cloudinary มี Solution แบบ End-to-End สำหรับรูปภาพและวิดีโอทั้งหมดของคุณตั้งแต่อัปโหลด (Upload), การจัดเก็บ (Storage), การบริหารการจัดการ (Administration), เครื่องมือการจัดการ (Manipulation) และการส่งมอบ (Delivery) การอัปโหลดการประมวลผลและการส่งมอบจะทำบนเซิร์ฟเวอร์ของ Cloudinary และปรับขนาดโดยอัตโนมัติเพื่อจัดการกับการโหลดของผู้ใช้งานให้เหมาะสมกับอุปกรณ์ (Device) นั้นๆ ระบบของ Cloudinary จะทำทุกอย่างบน Cloud รูปภาพและวิดีโอของคุณจะถูกจัดส่งผ่าน Server ที่มีประสิทธิภาพสูงผ่านทาง CDN (Content Delivery Networks) รูปภาพและวิดีโอที่อัปโหลดไปยัง Cloudinary ทั้งหมดจะถูกเก็บไว้อย่างปลอดภัย ดังรูปที่ 2.5

การจัดการ Cloudinary ประกอบด้วย

- Server ประสิทธิภาพสูงที่สนับสนุนการอัปโหลด และดาวน์โหลดได้อย่างรวดเร็ว
- พื้นที่จัดเก็บที่พร้อมใช้งานสูงและมั่นใจได้ว่าทรัพยากรจะพร้อมใช้งานและปลอดภัยอยู่เสมอ
- Server การประมวลผลที่มีประสิทธิภาพสูงสำหรับการสร้างภาพ และวิดีโอที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ภาพรวม Service Architecture ของ Cloudinary

ข้อดีของ Cloudinary

- ช่วยลดเวลาจากการศึกษาและพัฒนา และสามารถลดค่าใช้จ่ายลง 90 %
- สามารถใช้ร่วมกับ Framework มากมายได้อย่างง่ายดาย
- มี Features ในการจัดการรูปภาพที่ดีและมากมาย
- มีการจัดการรูปภาพ, APIs สำหรับอัปโหลด, เรียกใช้งาน และระบบการจัดการไว้ในระบบ Cloud
- เว็บไซต์ทำงานได้ดีขึ้น และสามารถโหลดได้อย่างรวดเร็วบนอุปกรณ์ใดก็ได้

2.3.3 Airtable

เปรียบเสมือนโปรแกรม Spreadsheet เช่น Excel ที่ใช้งานกันอยู่กับงานทั่วไป แต่ในแต่ละแถวของ Spreadsheet นั้นสามารถบริหารจัดการคอลัมน์ได้อย่างอิสระ ไม่ว่าจะเป็นการกำหนดประเภทของคอลัมน์ให้เป็นแบบ Text, Dropdown, รูปภาพ, ไฟล์แนบ อีกทั้งยังสามารถบริหารจัดการข้อมูลทั้งหมดได้ในรูปแบบของฐานข้อมูลอีกด้วย ดังรูปที่ 2.6

Opportunity Name	Status	Priority	Estimated Value	Account
Qualification Count 4 Sum \$41,054				
1 BPS Pilot	Qualification	Medium	\$10,000	Bear Paw Solutions
2 Acetube inquiry	Qualification	Medium	\$15,133	Acetube
3 Timbershadow expansion	Qualification	Very High	\$6,154	Timbershadow
4 LKS req	Qualification	Medium	\$9,767	Leonard Krower & Sons
Proposal Count 7 Sum \$145,830				
5 BPS second use case	Proposal	Very Low / Deprioritize	\$24,791	Bear Paw Solutions
6 Galerprises renewal	Proposal	Low	\$23,205	Galerprises
7 EYS renewal	Proposal	Low	\$23,503	Edge Yard Service
8 Payless inbound	Proposal	Low	\$20,999	Payless Cashways
9 Robinetworks renewal	Proposal	Medium	\$24,692	Robinetworks
10 SI expansion	Proposal	Low	\$12,687	Sunlight Intelligence
11 Payless Cashways EU	Proposal	High	\$15,953	Payless Cashways

รูปที่ 2.6 ตัวอย่างการจัดการข้อมูลด้วย Airtable

ส่วนประกอบของ Airtable

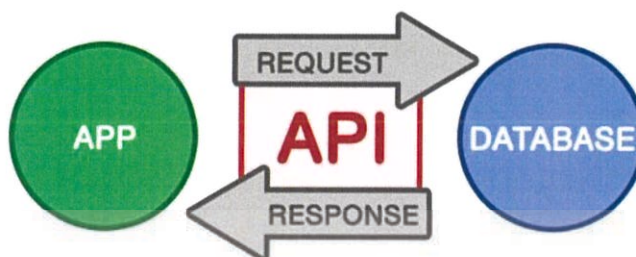
- Bases เปรียบเสมือนกับไฟล์หลัก มองภาพคล้ายกับเป็นฐานข้อมูล
- Tables โดยในหนึ่ง Base มีหลาย Tables ได้ มองคล้ายกับ Sheet ใน Spreadsheet หรือ Table ใน Database นั่นเอง
- Views เป็นมุมมองข้อมูลต่างๆ ใน Tables ซึ่งสามารถปรับแต่งให้เข้ากับลักษณะงานที่ต้องการได้
- Fields เป็นข้อมูลในแต่ละคอลัมน์ของ Tables ซึ่งสามารถจัดเก็บข้อมูลได้หลายชนิด เช่น Text, Dropdown, File, Image ฯลฯ
- Records คือ ข้อมูลในแต่ละแถวของ Tables นั่นเอง

2.4 Application Programming Interface (API)

API ย่อมาจาก Application Programming Interface คือ ช่องทางการเชื่อมต่อ, ช่องทางหนึ่งที่จะเชื่อมต่อกับเว็บไซต์ผู้ให้บริการ API จากที่อื่น เป็นตัวกลางที่ทำให้โปรแกรมประยุกต์เชื่อมต่อกับโปรแกรมประยุกต์อื่น หรือเชื่อมการทำงานเข้ากับระบบปฏิบัติการ ตัวอย่างเช่น Twitter มีหลายเว็บที่มีการเชื่อมโยงข้อมูลกับ Twitter ทั้งเป็นการอ่านข้อมูลจาก Twitter หรือส่งข้อมูลเข้า Twitter เองก็ตาม ซึ่งล้วนอาศัยการเชื่อมต่อแลกเปลี่ยนข้อมูลกัน ด้วย API นั่นเอง และอีกหนึ่งตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น Google Maps API คือ บริการของ Google อีกรูปแบบหนึ่งที่สามารถนำข้อมูลของ Google Maps ที่ทาง Google ให้บริการ โดยส่วนมากจะนำมาใช้กับเว็บไซต์ ของบริษัทฯหรือเว็บไซต์ห้างร้านต่างๆ เพื่อเป็นอีกช่องทางที่ให้ลูกค้ารู้ว่าบริษัทฯ หรือห้างร้านนั้น ดังรูปที่ 2.7



รูปที่ 2.7 Application Programming Interface (API)

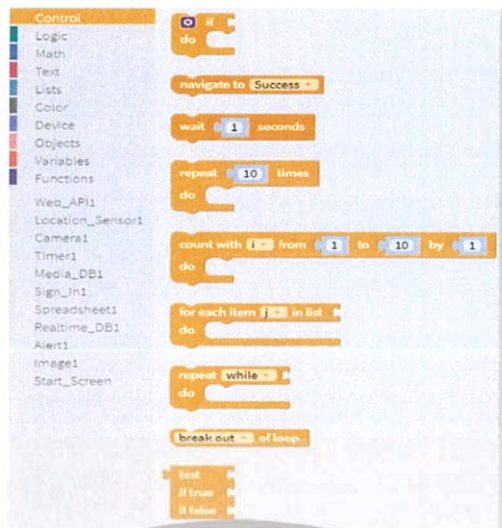
2.5 Thinkable

Thinkable X เป็นโปรแกรมสร้างแอปพลิเคชันที่สามารถทำงานได้หลายแพลตฟอร์ม นั่นหมายความว่าทุกแอปพลิเคชันที่พัฒนาด้วย Thinkable X จะสามารถทำงานได้ทั้งบนอุปกรณ์มือถือ ระบบแอนดรอยด์และ iOS แต่เดิมการสร้างแอปพลิเคชันหนึ่งต้องใช้เงินนับแสนดอลลาร์ และทีมวิศวกรสองทีม ทีมหนึ่งสำหรับระบบแอนดรอยด์ ส่วนอีกทีมสำหรับระบบ iOS แต่ขณะนี้ Non-coders ก็สามารถพัฒนาแอปพลิเคชันของตนเองบนแพลตฟอร์มหนึ่งได้อย่างง่ายดาย และแอปพลิเคชันเหล่านี้จะทำงานได้ทั้งบนอุปกรณ์แอนดรอยด์ ไอโฟน และไอแพด ดังรูปที่ 2.8 และรูปที่ 2.9



รูปที่ 2.8 หน้า Design ของ Thinkable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 ตัวอย่าง Block ของ Thunkable

สิ่งที่ทำให้ Thunkable โดดเด่นในแวดวง No-code/Low-code คือ โปรแกรมลากและวางแบบ Visual รวมทั้งภาษาที่ง่าย ไม่ว่าใครก็ใช้ได้ แต่ก็ยังทรงอำนาจ และสามารถปรับแต่งได้ตามความต้องการ จึงเหมาะสำหรับการสร้างแอปพลิเคชันที่แตกต่างไปจากแอปพลิเคชันอื่นๆ ที่พัฒนาโดยวิศวกรซอฟต์แวร์ โดยวิธีการนี้ได้รับการคิดค้นโดยคณะนักวิจัยของ Google และ MIT ซึ่งเป็นสถาบันที่บ่มเพาะและให้กำเนิด Thunkable

2.6 คิวอาร์โค้ด (QR Code)

คิวอาร์โค้ด (QR Code) คือ สัญลักษณ์สี่เหลี่ยม ที่เริ่มเห็นแพร่หลายมากขึ้น ไม่ว่าจะบนจากหนังสือพิมพ์หรือนิตยสาร ย่อมาจาก Quick Response เป็นบาร์โค้ด 2 มิติ ที่มีต้นกำเนิดมาจากประเทศญี่ปุ่น โดยบริษัท Denso-Wave (เดนโซ-เวฟ) ตั้งแต่ปีค.ศ. 1994 คุณสมบัติของ QR Code คือ เป็นสัญลักษณ์แทนข้อมูลต่างๆ ที่มีการตอบสนองที่รวดเร็ว ซึ่งส่วนใหญ่จะนำมาใช้กับสินค้า, สื่อโฆษณาต่างๆ เพื่อให้ข้อมูลเพิ่มเติม หรือจะเป็น URL (ยูอาร์แอล) เว็บไซต์ หรือเก็บข้อมูลต่างๆ ที่ต้องการประยุกต์ใช้ได้หลากหลาย

2.6.1 ประโยชน์ของ QR Code

สามารถนำ QR Code มาประยุกต์ใช้ได้หลากหลายรูปแบบ เช่น แสดง URL ของเว็บไซต์, ข้อความ, เบอร์โทรศัพท์ และข้อมูลที่เป็นตัวอักษรได้อีกมากมาย ปัจจุบัน QR Code ถูกนำไปใช้ในหลายๆ ด้าน เนื่องจากความรวดเร็ว เพราะทุกวันนี้คนส่วนใหญ่จะมีมือถือกันทุกคน และมีถือเดียวนี้ก็มีกล้องเกือบทุกรุ่นแล้ว ประโยชน์ที่เห็นได้ชัดที่สุดของ QR Code คือ การแสดง URL ของเว็บไซต์ เพราะ URL โดยปกติแล้วจะจดจำยากเพราะยาวและบางทีก็จะซับซ้อนมาก แต่ด้วย QR Code เพียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แค่ยกมือถือมาสแกน QR Code ที่พบเห็นตามผลิตภัณฑ์ต่างๆ เช่น นามบัตร, นิตยสาร ฯลฯ แล้วมือถือจะลิงค์เข้าเว็บเพจที่ QR Code นั้นๆ บันทึกข้อมูลอยู่โดยอัตโนมัติ

2.6.2 วิธีใช้งาน QR Code

วิธีใช้งาน QR Code ต้องใช้งานผ่านโทรศัพท์มือถือ ที่มีสัญลักษณ์ QR Code อยู่ภายในตัวเครื่อง เพียงนำกล้องที่อยู่บนมือถือสแกนบน QR Code แล้วรอสักครู่เครื่องจะอ่าน QR Code สีดำออกมาเป็นตัวหนังสือที่มีข้อมูลมากมาย เช่น รายละเอียดสินค้า โปรโมชั่น สถานที่ตั้งของบริษัท ร้านค้า เว็บไซต์ เบอร์โทรศัพท์ หากอยู่บนนามบัตร เจ้าของนามบัตรก็จะใส่ทั้งชื่อ อีเมล ฯลฯ รวมทั้งสามารถใช้ QR Code สื่อบอกความเข้าใจได้ด้วย เพียงพิมพ์ QR Code ลงบนการ์ดผู้ที่ได้รับการ์ดนำโทรศัพท์มือถือที่มีกล้องมาสแกน เพียงเท่านี้ก็รู้ใจความด้วยไอทีแล้ว ดังรูปที่ 2.10 และตารางที่ 2.2



รูปที่ 2.10 ตัวอย่างคิวอาร์โค้ด (QR Code)

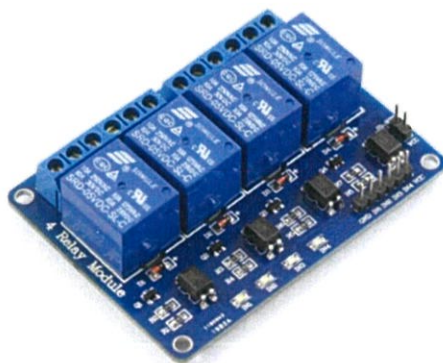
ตารางที่ 2.2 ความจุข้อมูล QR Code

ความจุข้อมูล QR Code	
ตัวเลขอย่างเดียว	มากที่สุด 7089 ตัวอักษร
ตัวอักษรผสมตัวเลข	มากที่สุด 4296 ตัวอักษร
ไบนารี (8 บิต)	มากที่สุด 2953 ไบต์
คันจิ/คะนะ	Max 1817 ตัวอักษร

2.7 รีเลย์ (Relay)

รีเลย์ (Relay) เป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็ก เพื่อใช้ในการดึงดูดหน้าสัมผัสของคอนแทคให้เปลี่ยนสถานะ โดยการป้อนกระแสไฟฟ้าให้กับขดลวด เพื่อทำการปิดหรือเปิดหน้าสัมผัสคล้ายกับสวิตช์อิเล็กทรอนิกส์ ซึ่งสามารถนำรีเลย์ไปประยุกต์ใช้ในการควบคุมวงจรต่างๆ ในงานอิเล็กทรอนิกส์มากมาย ดังรูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



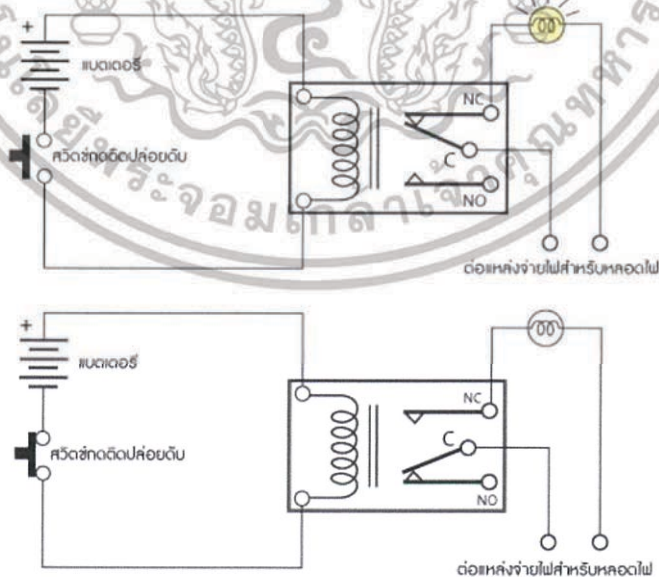
รูปที่ 2.11 ภาพตัวอย่าง Relay แบบ 4 Channel

จุดต่อใช้งานมาตรฐาน ประกอบด้วย

จุดต่อ NC ย่อมาจาก Normal Close หมายความว่าปกติปิด หรือหากยังไม่จ่ายไฟให้ขดลวดเหนี่ยวนำหน้าสัมผัสจะติดกัน โดยทั่วไปมักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการให้ทำงานตลอดเวลา

จุดต่อ NO ย่อมาจาก Normal Open หมายความว่าปกติเปิด หรือหากยังไม่จ่ายไฟให้ขดลวดเหนี่ยวนำหน้าสัมผัสจะไม่ติดกัน โดยทั่วไปมักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการควบคุมการเปิดปิด เช่น โคมไฟสนามหรือหน้าบ้าน

จุดต่อ C ย่อมาจาก Common คือ จุดรวมที่ต่อมาจากแหล่งจ่ายไฟ ดังรูปที่ 2.12



รูปที่ 2.12 วงจรการทำงานของ Relay

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อคำนึงในการใช้งานรีเลย์ทั่วไป

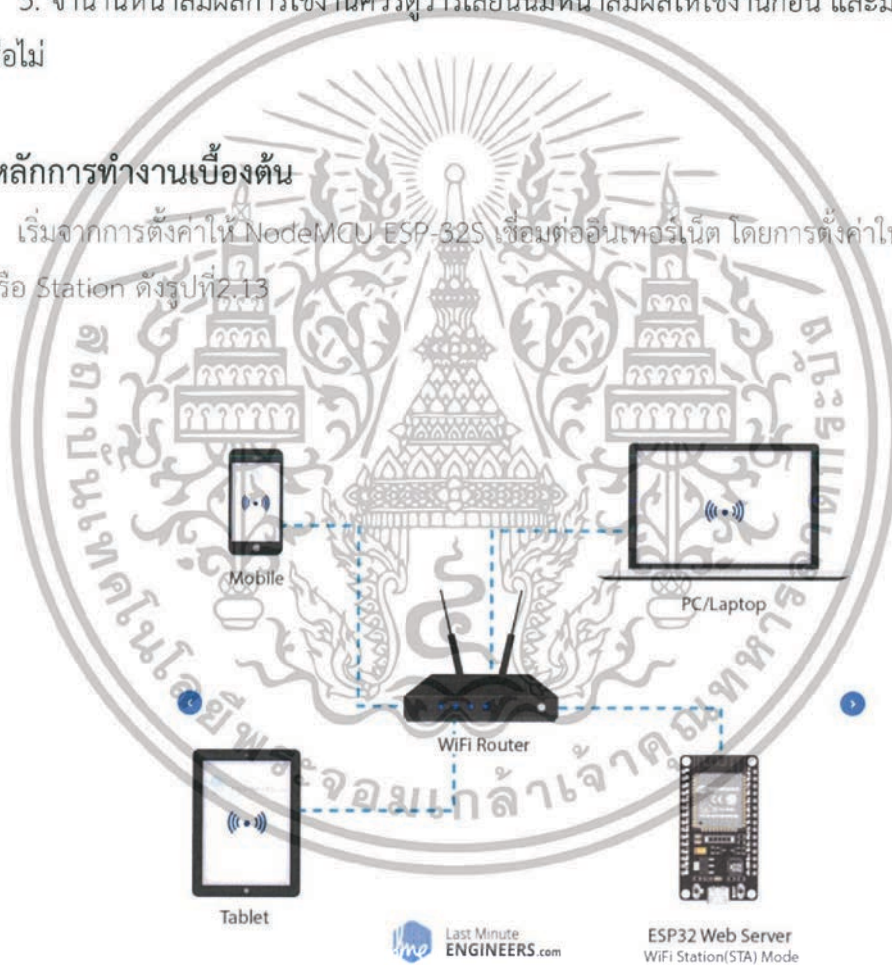
1. แรงดันใช้งาน หรือแรงดันที่ทำให้รีเลย์ทำงานได้ หากดูที่ตัวรีเลย์จะระบุค่าแรงดันใช้งานไว้ (หากใช้ในงานอิเล็กทรอนิกส์ ส่วนมากจะใช้แรงดันกระแสตรงในการใช้งาน) เช่น 12 VDC คือต้องใช้แรงดันที่ 12 VDC เท่านั้น หากใช้มากกว่านี้ขดลวดภายในตัวรีเลย์อาจจะขาดได้ หรือหากใช้แรงดันต่ำกว่ามากรีเลย์จะไม่ทำงาน ส่วนในการต่อวงจรนั้นสามารถต่อขั้วใดก็ได้ เพราะตัวรีเลย์จะไม่ระบุขั้วต่อไว้ (นอกจากชนิดพิเศษ)

2. การใช้งานกระแสผ่านหน้าสัมผัส ซึ่งที่ตัวรีเลย์จะระบุไว้ เช่น 10 A 220 AC คือหน้าสัมผัสของรีเลย์นั้นสามารถทนกระแสได้ 10 A ที่ 220 VAC แต่การใช้ก็ควรจะใช้งานที่ระดับกระแสต่ำกว่านี้จะเป็นการดีกว่า เพราะถ้ากระแสผ่านหน้าสัมผัส ของรีเลย์จะละลายเสียหายได้

3. จำนวนหน้าสัมผัสการใช้งานควรดูว่ารีเลย์นั้นมีหน้าสัมผัสให้ใช้งานกี่อัน และมีขั้วคอมมอนด้วยหรือไม่

2.8 หลักการทำงานเบื้องต้น

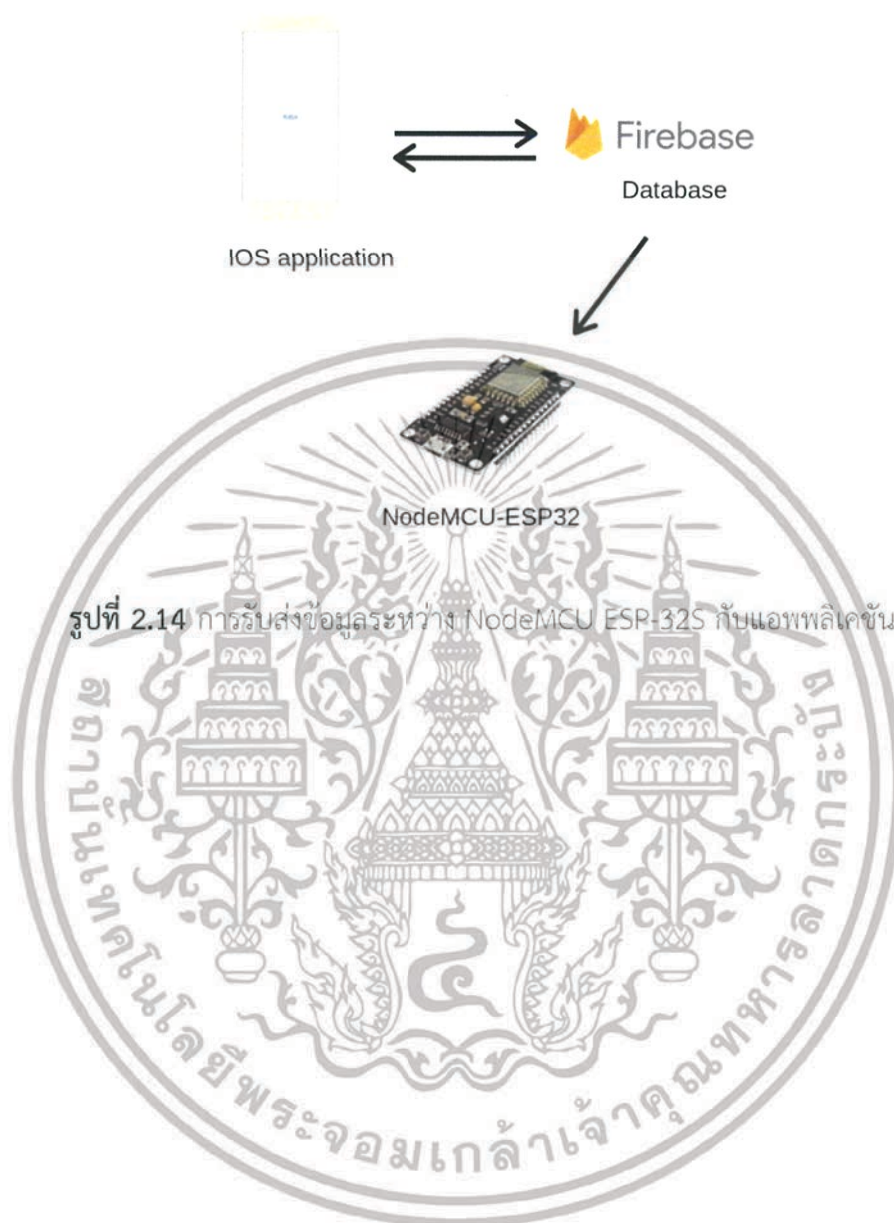
เริ่มจากการตั้งค่าให้ NodeMCU ESP-32S เชื่อมต่ออินเทอร์เน็ต โดยการตั้งค่าให้เป็นโหมด STA หรือ Station ดังรูปที่ 2.13



รูปที่ 2.13 NodeMCU ESP-32S เชื่อมต่อกับอุปกรณ์ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และตั้งค่าให้กับ NodeMCU ESP-32S รับค่าจาก Firebase ว่าค่าที่ได้รับมาจะให้ทำการเปิดหรือปิด ล็อคเกอร์ โดยค่าที่ได้รับมาถูกส่งมาจาก iOS แอปพลิเคชันที่ได้เขียนไว้อีกทีหนึ่ง ดังรูปที่ 2.14



รูปที่ 2.14 การรับส่งข้อมูลระหว่าง NodeMCU ESP-32S กับแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

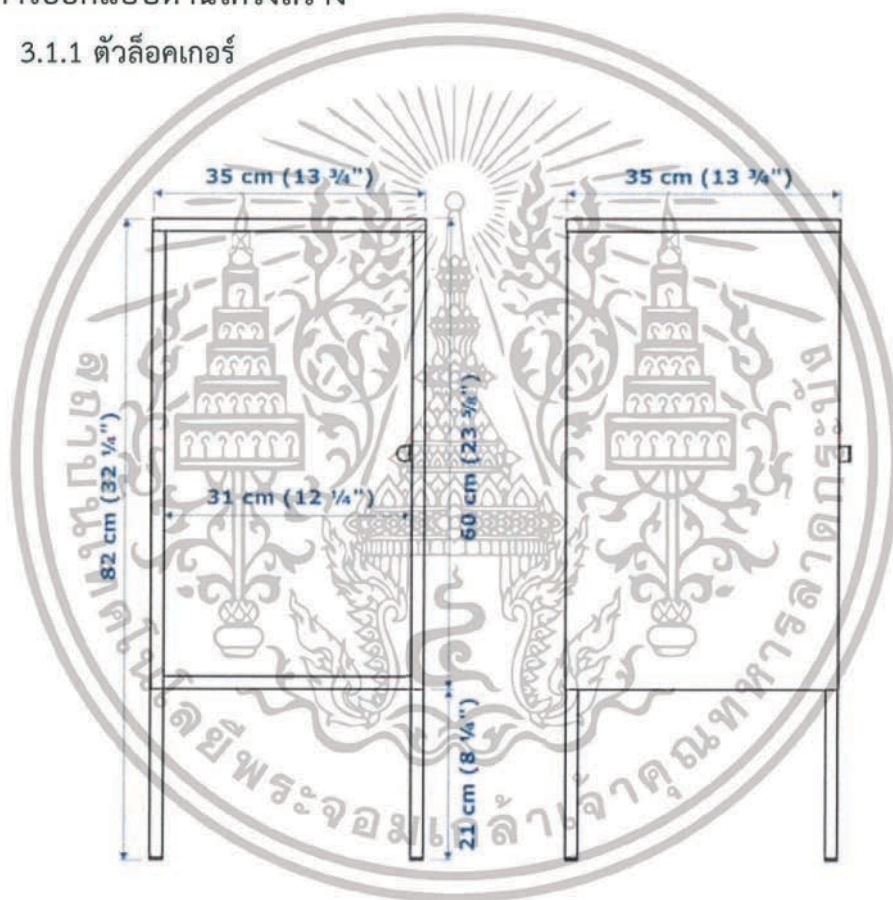
บทที่ 3

การออกแบบตู้ล็อคเกอร์อัจฉริยะ

ในบทนี้จะกล่าวถึงการออกแบบด้านโครงสร้าง, วงจรไฟฟ้าควบคุมระบบ และแอปพลิเคชันที่ส่งผลต่อการทำงานของตู้ล็อคเกอร์อัจฉริยะ และการพัฒนาระบบควบคุม เพื่อให้ตู้ล็อคเกอร์อัจฉริยะทำงานได้อย่างมีประสิทธิภาพ

3.1 การออกแบบด้านโครงสร้าง

3.1.1 ตู้ล็อคเกอร์

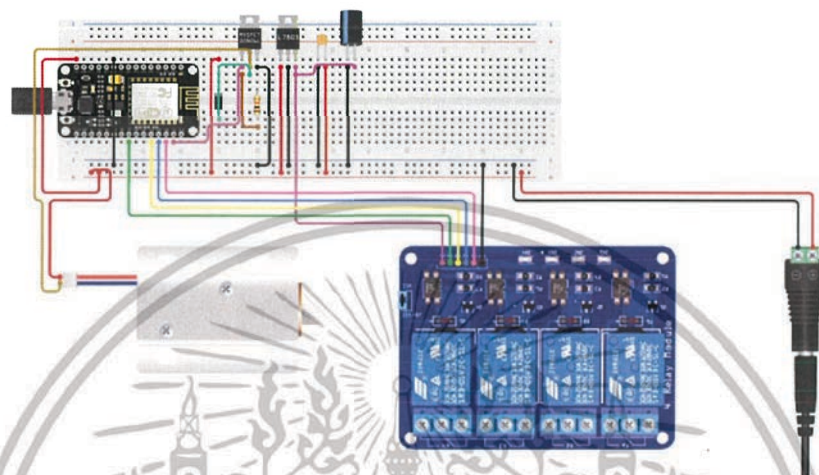


รูปที่ 3.1 การออกแบบโครงสร้างตู้ล็อคเกอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

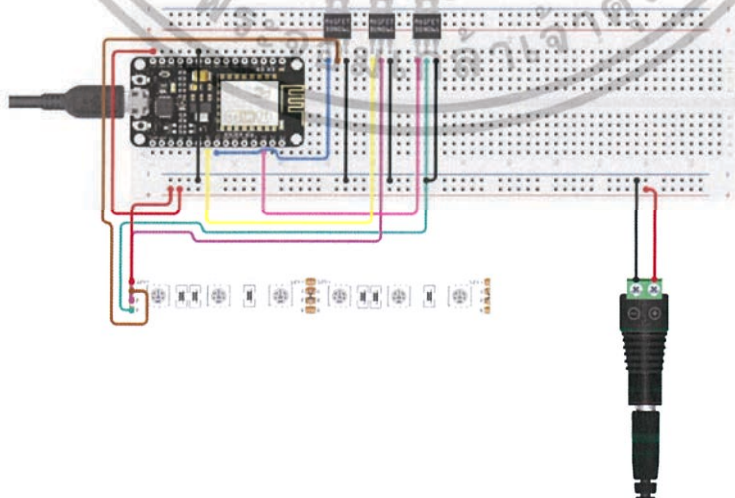
3.2 การออกแบบวงจรไฟฟ้าควบคุมการเปิด/ปิด

ในขั้นตอนแรกของการทำงานเนื่องจากตัวลีดและรีเลย์ทำงานที่ 12 V จึงต้องใช้ตัวแปลงไฟ 12 V ต่อเข้ากับรีเลย์, ตัวลีด และ NodeMCU แล้วต่อขา Output จาก NodeMCU ไปยังรีเลย์ เพื่อให้รีเลย์ควบคุมการทำงานของตัวลีดตามสัญญาณของ NodeMCU ดังรูปที่ 3.2



รูปที่ 3.2 การออกแบบวงจรไฟฟ้า

ติดตั้ง LED RGB Strip เพื่อแสดงสถานะของลีดเกอร์ว่าว่าง หรือถูกใช้งาน โดยต่อ LED RGB Strip กับ วงจร Amplifier เนื่องจาก NodeMCU ESP-32S จ่ายกระแสไฟฟ้าได้ที่ 40 mA ซึ่งเมื่อทำการต่อกับวงจร Amplifier จะขยายกระแสไฟฟ้าเป็น 1 AMP ทำให้ LED RGB สว่างขึ้น ตามลักษณะการใช้งานที่ระบุไว้ที่ 1 AMP ดังรูปที่ 3.3

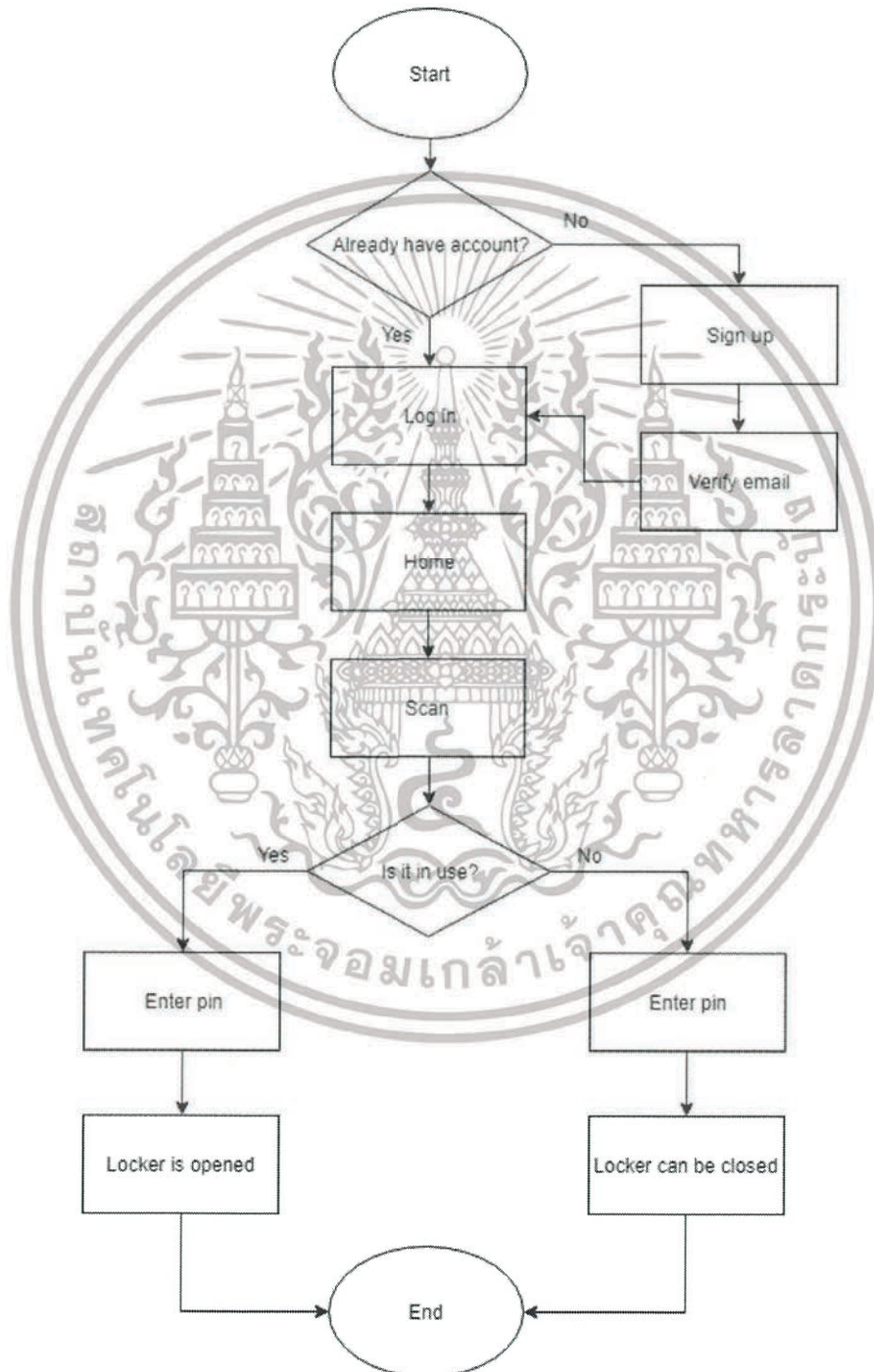


รูปที่ 3.3 วงจรไฟฟ้าติดตั้ง LED RGB Strip

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบ iOS แอปพลิเคชัน

การออกแบบ iOS แอปพลิเคชัน GUI (Graphical User Interface) คือ ส่วนที่ติดต่อกับผู้ใช้ โดยใช้ภาพสัญลักษณ์ ต้องการออกแบบให้เรียบง่ายสามารถใช้งานได้สะดวก โดยการใช้งานนั้นจะต้องสมัครมี Account เป็นของตัวเองเพื่อที่จะสามารถรู้ว่าล็อคเกอร์ที่ใช้อยู่ใครเป็นผู้ใช้ซึ่งมีลำดับของแต่ละหน้าดังรูปที่ 3.4



รูปที่ 3.4 Flowchart iOS แอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 หน้าล็อกอิน (Log in)

หน้าล็อกอิน (Log in) เป็นหน้าเริ่มต้น โดยหน้านี้จะมีส่วนหัวเขียนเป็นชื่อแอปพลิเคชัน MECHALOCK ส่วนต่อมาก็คือ ช่อง Input Text สำหรับใส่ค่าสองช่อง ช่องแรกสำหรับใส่ Email ช่องที่สองสำหรับ Password ส่วนต่อมาเป็นปุ่ม ปุ่มแรกคือ Login ปุ่มนี้ใช้สำหรับเมื่อสมัครมี Account แล้ว ให้ใส่ค่าลงไป Input Text Email และ Password แล้วกดปุ่ม Login เพื่อไปขั้นตอนต่อไป ปุ่มที่สองคือ Sing in คือ ปุ่มกดเมื่อกดแล้วจะไปทีหน้าใน Sing up สำหรับสร้าง Account ไว้ใช้งานนั่นเอง ดังรูปที่ 3.5



รูปที่ 3.5 แอปพลิเคชันหน้าล็อกอิน (Log in)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 หน้าลงทะเบียน (Sign up)

หน้าลงทะเบียน (Sign up) เป็นหน้าที่เอาไว้ใช้ในการสมัคร Account โดยมีช่อง Input Text สีเทา 4 ช่อง ให้ใส่ค่า Full-name, Email, Password และ Student ID และปุ่ม 2 ปุ่ม ปุ่มแรก SUBMIT เป็นปุ่มกดเมื่อกรอกทุกอย่างถูกต้องและต้องการดำเนินการต่อ ปุ่มที่ 2 Back to Login เป็นปุ่มกดไว้ใช้กลับไปหน้า Log in ดังรูปที่ 3.6



รูปที่ 3.6 แอปพลิเคชันหน้าลงทะเบียน (Sign up)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 หน้าโฮม (Home)

หน้าโฮม (Home) คือ หน้า que แสดงสถานะของลืคเกอร์แต่ละตัว โดยลืคเกอร์แต่ละตัวแทนด้วยสัญลักษณ์สีเหลี่ยม โดยมีหมายเลขกำกับ 1, 2, 3 และ 4 ตามลำดับ สีฟ้าหมายถึงลืคเกอร์ว่างสามารถใช้งานได้ สีแดงหมายถึงลืคเกอร์ไม่ว่างถูกใช้งานอยู่ ส่วน Label ข้างล่างไว้แสดง Email ของ Account ที่ทำการ Log in และปุ่ม Sign out ไว้กดเมื่อต้องการออกจากระบบ โดยหน้า Home นี้สามารถ Swipe ไปทางซ้ายเพื่อไปสู่น้าสแกน ดังรูปที่ 3.7



รูปที่ 3.7 แอปพลิเคชันหน้าโฮม (Home)

3.3.4 หน้าสแกน (Scan)

หน้าสแกน (Scan) เป็นหน้าที่เอาไว้ใช้ในการอ่านค่า QR Code โดยการปุ่มสแกน เพื่อทำการถ่าย QR Code ติดไว้บนตู้ล็อกเกอร์เพื่อสั่งการเปิด โดยเมื่อทำการสแกนแล้ว จะไปที่หน้า PIN เพื่อทำการตั้งรหัสต่อ ดังรูปที่ 3.8



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5 หน้า PIN

หน้า PIN มีไว้เพื่อใช้ตั้งรหัสตู้ที่ว่างก่อนทำการเปิด หรือถ้าหากตู้ถูกใช้งานอยู่ก็ต้องใส่รหัสเดิมที่เคยถูกตั้งไว้ยืนยันก่อนที่ตู้จะทำการเปิด โดยรหัสจะเป็นตัวเลข จำนวน 4 หลัก ดังรูปที่ 3.9



รูปที่ 3.9 หน้า PIN ของแอปพลิเคชัน

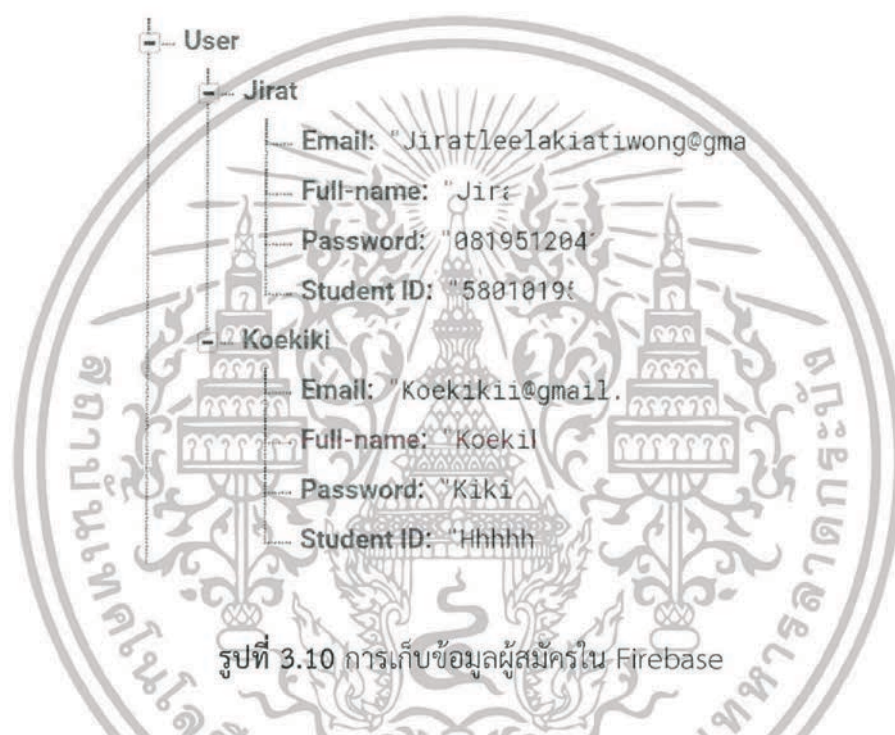
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การส่งข้อมูล iOS แอปพลิเคชัน

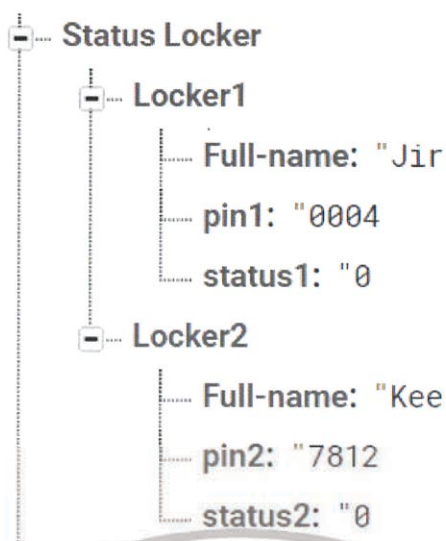
ในการส่งหรือรับข้อมูลของ iOS แอปพลิเคชัน ไปยัง Database ต่างๆ จะถูกส่งและรับข้อมูลในรูปแบบ Json โดยจะแบ่งเป็นหัวข้อ Database แต่ละตัวที่ iOS แอปพลิเคชัน ได้ส่งหรือรับข้อมูลต่างๆ ดังนี้

3.4.1 Firebase

ส่วนที่ 1. การเก็บข้อมูลผู้สมัคร เมื่อผู้สมัคร สมัครการใช้งานข้อมูลผู้ใช้งาน ได้แก่ Email, Full-name, Password และ StudentID จะถูกเก็บไว้ใน User ตามด้วย Email ของผู้ใช้ เขียนเป็น path ได้ดังนี้ User/UserEmail@xxxxmailcom ดังรูปที่ 3.10



ส่วนที่ 2. ส่วนการทำงานของล็อกเกอร์เมื่อผู้สมัครทำการสแกนเปิดล็อกเกอร์ข้อมูลถูกเก็บไว้ใน Path ดังนี้ Status Locker/Locker “x” โดย x แทนหมายเลขของล็อกเกอร์ โดยข้างใน Locker “x” ก็จะประกอบด้วย Full-name คือ ชื่อคนที่เปิด Locker “x” ตัวนี้ และ Pin “x” คือรหัสผ่านของ Locker “x” และ Status “x” สถานะของล็อกเกอร์ว่าเปิดหรือปิดอยู่นั่นเอง (1 คือ เปิด 0 คือ ปิด) ดังรูปที่ 3.11



รูปที่ 3.11 การเก็บข้อมูลในส่วนของการทำงานของล็อกเกอร์

3.4.2 Airtable

เมื่อผู้สมัคร สมัครใช้งานข้อมูลผู้ใช้งาน Full-name Email Password StudentID จะถูกเก็บไว้ในแต่ละ Colum เรียงตามที่ได้ออกแบบ และยังสามารใส่ไฟล์รูปภาพ Media เปลี่ยนสี เพิ่มสัญลักษณ์ และอื่นๆ อีกมากมาย ซึ่งสะดวกต่อการจัดเก็บตัดแปลงเพิ่มเติมมากกว่า ใน Firebase ดังรูปที่ 3.12

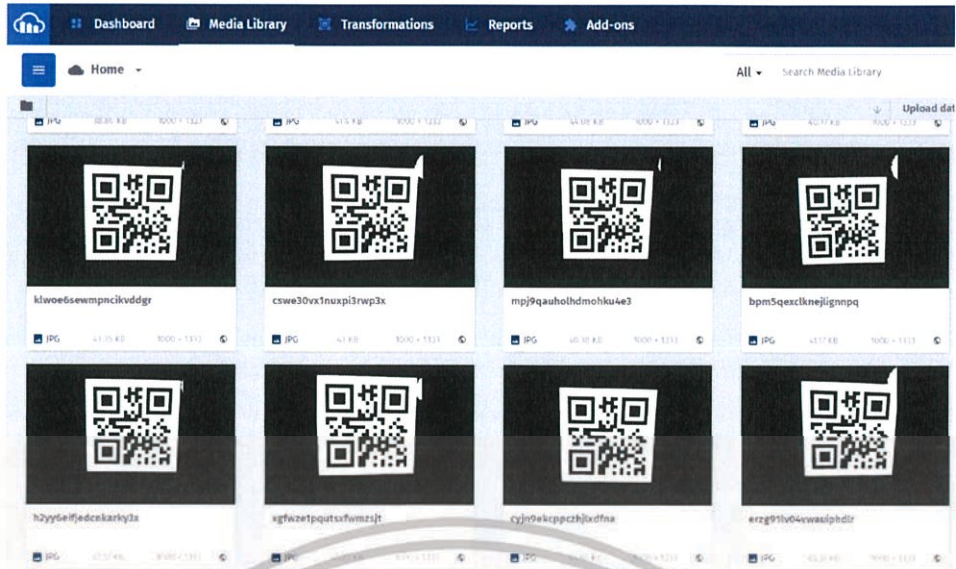
	A Full-name	A Email	A Password	A Student ID	Notes
1	Jirat	Jiratleelanatwong@gmail.com	Jib2j1	58010195	Good student
2	Koeki	Koekikii@gmail.com	Kiki	58010194	Bad student ever

รูปที่ 3.12 การเก็บข้อมูลใน Airtable

3.4.3 Cloudinary

เมื่อมีการสแกนภาพใน iOS แอปพลิเคชันจะถูกอัปโหลดขึ้น Cloudinary อัตโนมัติเพื่อมาทำการจัดการ, ปรับแต่ง จัดเก็บ อย่างเป็นระบบ ซึ่งสามารถใช้งานรูปภาพนี้ต่อ โดยแต่ละภาพจะมีชื่อและ URL ในการเข้าถึง จึงทำให้ง่ายต่อการจัดการและการเข้าถึงจากเว็บภายนอก ดังรูปที่ 3.13

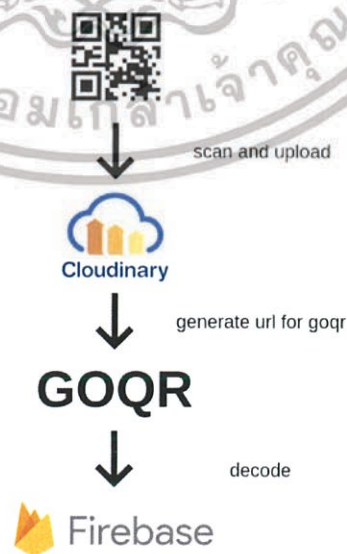
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 การเก็บข้อมูลใน Cloudinary

3.4.4 Web API

เนื่องจาก iOS แอปพลิเคชันที่เขียนไว้ไม่สามารถอ่านค่า QR Code ได้โดยตรง จึงต้องใช้ Web API ใช้ในการอ่านค่า QR Code โดย iOS แอปพลิเคชันจะทำการส่งรูป QR Code ที่ทำการสแกน และอัปโหลดขึ้น Cloudinary ส่งต่อไปยัง Web API โดย Web API ต้องอาศัยรูปจาก Cloudinary โดยจะเข้าถึงรูปจาก url ที่ Cloudinary สร้างเฉพาะขึ้น เพื่อทำการอ่านค่า Decode ของ QR Code นั้น โดยเมื่อทำการ Decode แล้วจะสามารถรู้ได้ว่าจะดำเนินงานกับลิสต์เกอร์ตัวไหน และส่งกลับมายัง Firebase โดยใช้บริการ Web API ของ GOQR ดังรูปที่ 3.14



รูปที่ 3.14 การทำงานของ Web API

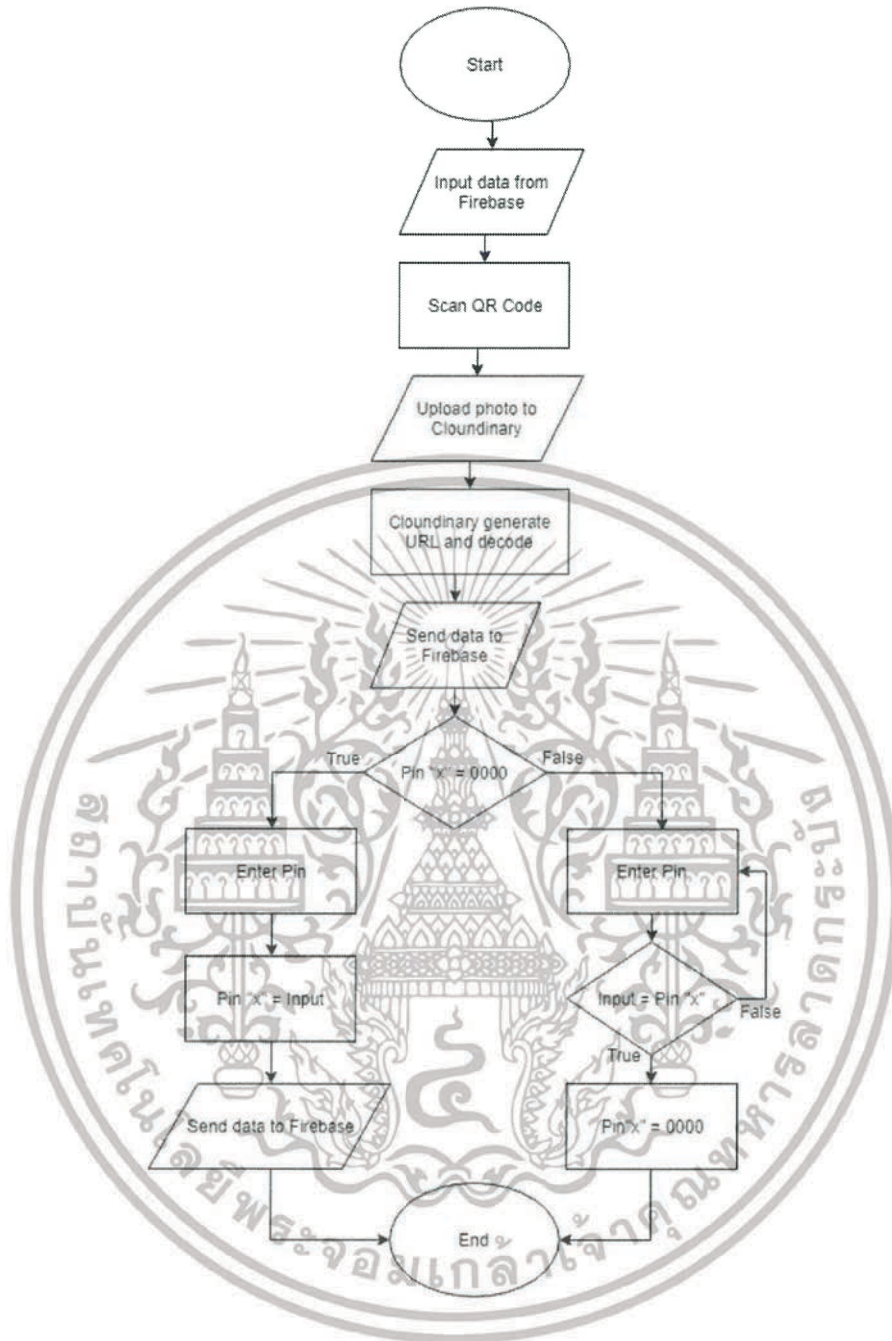
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 Login & Sign up Flowchart

จากรูปที่ 3.15 เป็นแผนภาพแสดงโปรแกรมการรับและส่งค่าจากข้อมูลที่ผู้ใช้งานใส่ ไปยังฐานข้อมูล Firebase ในส่วนของการเข้าใช้งานและลงทะเบียน โดยเมื่อเริ่มการทำงาน NodeMCU จะทำการเชื่อมต่อสัญญาณ WiFi จากนั้นจะเชื่อมต่อกับฐานข้อมูล Firebase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 Scanning Flowchart

จากรูปที่ 3.16 เป็นแผนภาพแสดงการทำงานของการทำงานของการสแกน QR Code เพื่อใช้งานตู้ล็อกเกอร์ โดยเมื่อมีการสแกน รูปถ่ายจะถูกเก็บไว้ยัง Cloudinary จากนั้นจะ Decode และส่งข้อมูลทั้งหมดไปยังฐานข้อมูล Firebase รวมถึง PIN ที่ผู้ใช้งานกำหนดด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลองและการวิเคราะห์

ในบทนี้จะกล่าวถึงผลการทดลอง ที่ได้จากการทดลองเพื่อนำมาใช้พัฒนาและปรับปรุงระบบ
ตู้ล็อกเกอร์อัจฉริยะ ซึ่งจะมีรายละเอียดดังต่อไปนี้

4.1 การทดลองใช้งานตู้ล็อกเกอร์อัจฉริยะ

การทดลองนี้จะอธิบายขั้นตอนการทดลองทุกขั้นตอนอย่างละเอียด และผลลัพธ์ที่ได้หลังจาก
การทดลองรวมถึงการเปลี่ยนแปลงของข้อมูลใน Database ต่างๆ

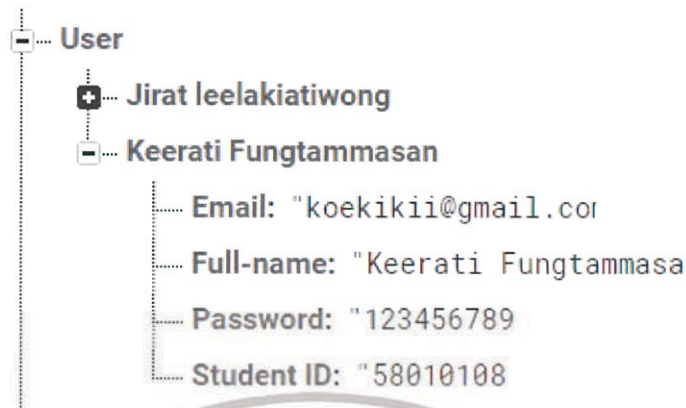
1. ทำการลงทะเบียนผู้ใช้ใหม่ในแอปพลิเคชัน โดยใช้ข้อมูลดังต่อไปนี้ ดังรูปที่ 4.1



รูปที่ 4.1 ข้อมูลที่ใส่ในการลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์จะได้ข้อมูลใน Firebase และ Airtable ดังรูปที่ 4.2 และรูปที่ 4.3

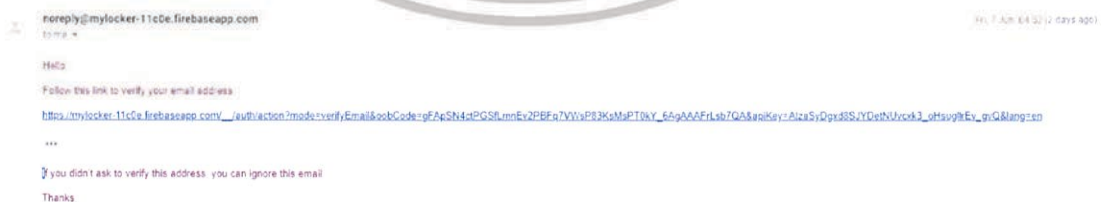


รูปที่ 4.2 ข้อมูลใน Firebase

	A Full-name	A Email	A Password	A Student ID
1	Jirat	jiratleelakiatiwong@gmail.com	jjb2j1	58010195
2	Photsawat	jmiefam@gmail.com	BKK	58010741
3	Keerati Fungtammasan	koekikii@gmail.com	123456789	58010108

รูปที่ 4.3 ข้อมูลใน Airtable

2. หลังจากยืนยันอีเมลดังรูปที่ 4.4 เรียบร้อยแล้ว ทำการ Log in เข้าสู่ระบบดังรูปที่ 4.5



รูปที่ 4.4 อีเมลเพื่อยืนยันการลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mechalock

Label

koekikii@gmail.com

.....

Login

Sign up ?

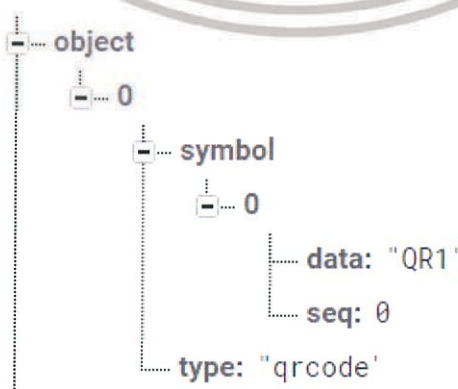
รูปที่ 4.5 การ Log in เข้าสู่ระบบ

หลัง Log in ระบบจะพาไปที่หน้า Home ซึ่งแสดงให้เห็นว่าตู้ล็อกเกอร์นั้นไม่มีการใช้งานทั้ง 4 ตู้ ดังรูปที่ 4.6



รูปที่ 4.6 หน้าจอแสดงการใช้งานตู้ล็อกเกอร์

ทำการสแกน QR Code โดยสามารถสั่งให้ระบบทำงานเมื่อได้รับข้อมูลใดๆ ก็ได้ผ่านทาง Firebase โดยในที่นี้จะสั่งให้ตู้ล็อกเกอร์ที่ 1 ทำงานเมื่อได้รับข้อมูลว่า "QR1" โดยจะสร้าง QR Code ผ่านทาง GOQR และป้อนข้อมูลว่า "QR1" ดังรูปที่ 4.7 ถึงรูปที่ 4.9



รูปที่ 4.7 การตั้งค่าอ่านข้อมูล QR1 ใน Firebase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 รูปที่ QR Code ของ QR1



รูปที่ 4.9 รูปที่ใช้ในการสแกน QR Code

3. หลังจากสแกนเสร็จรูปที่ใช้ในการสแกนจะถูกจัดเก็บไว้ใน Cloudinary และระบบจะให้การใส่ PIN เพื่อตั้ง Password ในการเปิด-ปิดตู้ล็อกเกอร์ โดยในที่นี้จะตั้ง PIN ว่า “1234” โดยข้อมูลทั้งหมดจะถูกส่งไปยัง Firebase ดังรูปที่ 4.10 ถึงรูปที่ 4.12

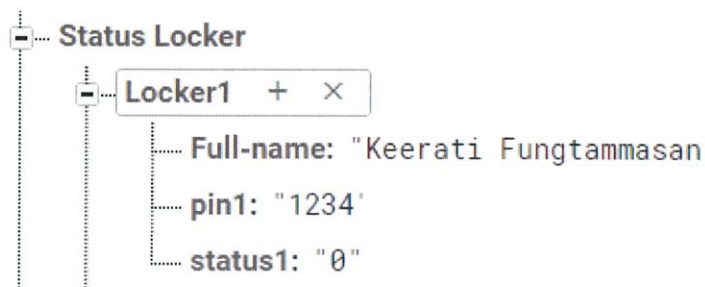


รูปที่ 4.10 รูปที่สแกนที่ปรากฏในฐานข้อมูล Cloudinary



รูปที่ 4.11 ใส่ PIN เพื่อใช้งานล็อกเกอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 ข้อมูลการใช้งานตู้ล็อกเกอร์ใน Firebase

และเมื่อดูในหน้า Home จะเห็นว่าตู้ล็อกเกอร์ 1 ได้มีผู้ใช้งานอยู่ดังรูปที่ 4.13



4. เสร็จสิ้นการใช้งานตู้ล็อกเกอร์ ต่อไปจะทำการเปิดตู้ล็อกเกอร์ โดยทำการสแกน QR Code เดิมซ้ำอีกรอบหนึ่ง จะปรากฏข้อความว่าต้องการเปิดตู้ล็อกเกอร์ใช่ไหมดังรูปที่ 4.14 และใส่ PIN เดิมที่กล่าวไว้ข้างต้น



รูปที่ 4.14 หน้าจอปรากฏข้อความการใช้งาน

ข้อมูลใน Firebase จะถูกรีเซ็ตใหม่ทั้งหมดดังรูปที่ 4.15 และเมื่อเข้าไปดูที่หน้า Home จะเห็นว่าตู้ล็อกเกอร์ที่ 1 กลับมาว่างอีกครั้งดังรูปที่ 4.16



รูปที่ 4.15 ข้อมูลการใช้งานตู้ล็อกเกอร์ใน Firebase



รูปที่ 4.16 หน้าจอแสดงการใช้งานตู้ล็อกเกอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองการสแกน QR Code

การทดลองการสแกน QR Code เพื่อทดสอบประสิทธิภาพในการสแกนทั้งทางด้านการทำงานว่าทำงานหรือไม่ และ ความรวดเร็วในการสแกน โดยทำการทดสอบ 10 รอบซึ่งใช้ QR Code อันเดียวกัน ซึ่งแต่ละรอบจะใช้รูปที่ใช้ในการสแกนไม่เหมือนกัน ดังตารางที่ 4.1

ตารางที่ 4.1 ผลการทดลองการสแกน QR Code

การทดลองครั้งที่	ผลการทดลอง	
	การสแกน	ความเร็ว (วินาที)
1	ทำงาน	5.53
2	ทำงาน	9.18
3	ทำงาน	9.02
4	ทำงาน	7.97
5	ทำงาน	9.70
6	ทำงาน	9.07
7	ทำงาน	8.77
8	ทำงาน	8.83
9	ทำงาน	6.16
10	ทำงาน	9.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการดำเนินงานและข้อเสนอแนะ

จากการทำโครงการนี้ทำให้ต้องคำนึงถึงระบบปฏิบัติการในโทรศัพท์ที่จะใช้ในการทำโครงการ อุปกรณ์ที่ใช้ในการทำชิ้นงานจริง โปรแกรมที่ใช้ในการเขียนแอปพลิเคชัน การทดลองการสแกน QR Code ทั้งด้านความเร็วและด้านเสถียรภาพ ความสะดวกสบายในการใช้งานจริง โดยทั้งหมดสามารถสรุปผลการดำเนินการได้ดังนี้

5.1 การดำเนินงานจัดทำปริญญานิพนธ์

โครงการนี้เป็นโครงการที่ทำการออกแบบ และพัฒนาแอปพลิเคชันควบคุมตู้ลิ้นชักเกอร์อัจฉริยะ ซึ่งการควบคุมการเปิด-ปิดของตู้ลิ้นชักเกอร์อัจฉริยะนั้น ทำได้ด้วยการควบคุมผ่านมือถือที่ใช้เป็นประจำในชีวิตประจำวัน ในการทำงานนั้นได้ร่วมมือกันทำงาน และศึกษาข้อมูลจากแหล่งต่างๆ เพื่อให้ระบบควบคุมตู้ลิ้นชักเกอร์อัจฉริยะทำงานได้มีประสิทธิภาพ ทำงานได้รวดเร็ว มีความสะดวกสบาย ง่ายต่อการใช้งาน และสามารถลดค่าใช้จ่ายในการผลิตได้จริง รวมถึงความปลอดภัยของตู้ลิ้นชักเกอร์ โดยตู้ลิ้นชักเกอร์จะทำให้ผู้ใช้งานสามารถใช้งานตู้ลิ้นชักเกอร์ได้ทุกที่ในชีวิตประจำวัน ส่งผลให้ผู้ที่มีความสะดวกสบายในการเดินทางไปสถานที่ต่างๆ ลดความเหนื่อยล้าในการเดินทาง และยังทำให้ทรัพย์สินของผู้ใช้งานมีความปลอดภัยเพิ่มขึ้นด้วย

การสร้างระบบควบคุมตู้ลิ้นชักเกอร์อัจฉริยะนั้น จำเป็นต้องมีความรู้ความเข้าใจหลากหลายด้าน อาทิเช่น ด้านไมโครคอนโทรลเลอร์ ด้านระบบควบคุม ด้านการเขียนภาษาคอมพิวเตอร์ ด้านไอที เป็นต้น ซึ่งต้องศึกษาอย่างเจาะลึกในด้านนั้นๆ ให้เข้าใจอย่างลึกซึ้งเพื่อให้สามารถทำงานได้ โดยมีข้อสงสัยน้อยที่สุด ความรู้ที่กล่าวมาข้างต้นนั้นในท้ายที่สุดจะนำมาประกอบกันให้เกิดการทำงานที่มีผลต่อกันเป็นระบบ เมื่อทำแล้วเจอปัญหาจึงมาช่วยกันแก้ให้การทำงานของระบบควบคุมตู้ลิ้นชักเกอร์อัจฉริยะนั้นมีประสิทธิภาพสูงสุด

5.2 ปัญหาที่พบและแนวทางการแก้ไข

1. ปัญหาเนื่องจากขาดอุปกรณ์ในการทำงาน เพราะชิ้นงานจำเป็นต้องมีอุปกรณ์เฉพาะเพื่อให้ได้คุณภาพชิ้นงานที่ออกมาดี ทำให้งานบางอย่างจำเป็นต้องทำหาวิธีทำชิ้นงานจากแหล่งอื่นๆ
2. ปัญหาเนื่องจากอุปกรณ์บางส่วนจำเป็นต้องสั่งซื้อ สั่งทำเป็นพิเศษ และส่วนประกอบของหุ่นยนต์บางชิ้นนั้นเป็นส่วนเฉพาะที่ต้องใช้เวลาในการหา หรือสร้างขึ้นมา ทำให้ต้องใช้เวลาอย่างมากในการศึกษาและหาแหล่งขายอุปกรณ์ชนิดนั้น

3. ปัญหาเนื่องจากระบบจำเป็นต้องใช้ทักษะที่มีความชำนาญในการทำเฉพาะด้าน ทำให้ต้องใช้เวลาในการเขียน
4. ปัญหาเนื่องจากชิ้นงานมีความคลาดเคลื่อนจากความเป็นจริง เพราะชิ้นงานต้องการทักษะความชำนาญ ทำให้จำเป็นต้องปรับแก้หลายครั้ง เกิดความล่าช้า
5. ปัญหาเนื่องจากไม่มีโทรศัพท์ที่รองรับระบบปฏิบัติการ Android ไว้ใช้งาน จึงจำเป็นต้องทำในระบบปฏิบัติการ iOS ซึ่งมีความยากมากกว่า Android ทำให้ต้องศึกษาเพิ่มเติมมากขึ้น

5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ

1. การพัฒนาระบบสแกนให้สามารถ Decode เองได้ภายในแอปพลิเคชันเพื่อส่งข้อมูลไปยังฐานข้อมูล โดยไม่ต้องผ่านเว็บ API ซึ่งจะส่งผลให้การสแกนมีความรวดเร็วมากยิ่งขึ้น
2. เพิ่มระบบการตัดเงินผ่านทางธนาคารเมื่อใช้งานตู้ล็อกเกอร์
3. เพิ่มจำนวนตู้ล็อกเกอร์ที่ใช้งาน
4. พัฒนาระบบความปลอดภัยในการใช้ตู้ล็อกเกอร์ เพื่อป้องกันการโจรกรรม
5. พัฒนาหน้าใช้งานของแอปพลิเคชันให้มีความสวยงามมากขึ้น เพื่อดึงดูดให้ผู้อื่นมาใช้งานแอปพลิเคชัน



เอกสารอ้างอิง

- [1] “ESP32 เบื้องต้น” [Online]. Available:
<https://www.ioxhop.com/article/62/esp32>
- [2] “Database คืออะไร” [Online]. Available:
<https://www.mindphp.com/คู่มือ/73-คืออะไร/2055-database-คืออะไร23.html>
- [3] “Firebase Realtime Database” [Online]. Available:
<https://medium.com/firebasethailand/รู้จัก-firebase-realtime-database-ตั้งแต่-zero-จนเป็น-hero-5d09210e6fd6>
- [4] “จัดการรูปภาพบนเว็บง่ายๆด้วย Cloudinary” [Online]. Available:
<https://engineering.thinknet.co.th/จัดการรูปภาพบนเว็บง่ายๆด้วย-cloudinary-3008afaf888c>
- [5] “Spreadsheet รวมตัวกับ Database” [Online]. Available:
<https://askboon.com/airtable-เมื่อ-spreadsheet-รวมตัวกับ-database/>
- [6] “QR Code (คิวอาร์ โค้ด) คืออะไรและมีประโยชน์อย่างไร” [Online]. Available:
<https://www.mindphp.com/บทความ/239-it-technology/1836-what-is-qr-code.html>
- [7] “รีเลย์ (Relay) คืออะไร?” [Online]. Available:
<http://www.pspstech.co.th/รีเลย์relayคืออะไร-15696.page>
- [8] “การใช้งาน Firebase บน ESP32” [Online]. Available:
<https://www.ioxhop.com/article/88/การใช้งาน-firebase-บน-esp32>
- [9] “ESP32 เบื้องต้นการใช้งาน WiFi” [Online]. Available:
<https://www.ioxhop.com/article/71/esp32-เบื้องต้น-บทที่-10-การใช้งาน-wifi>

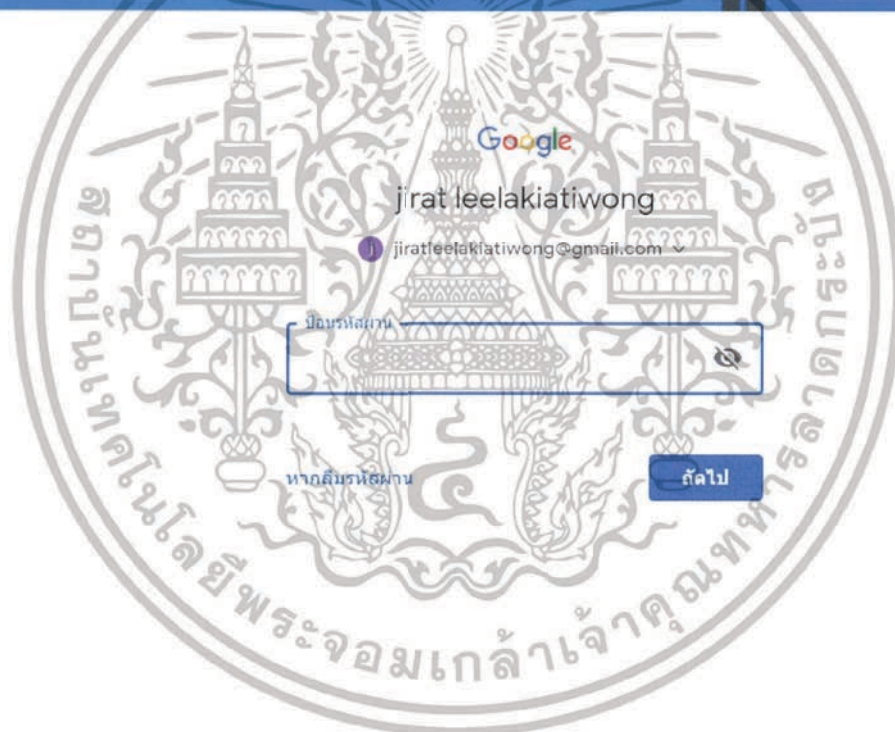
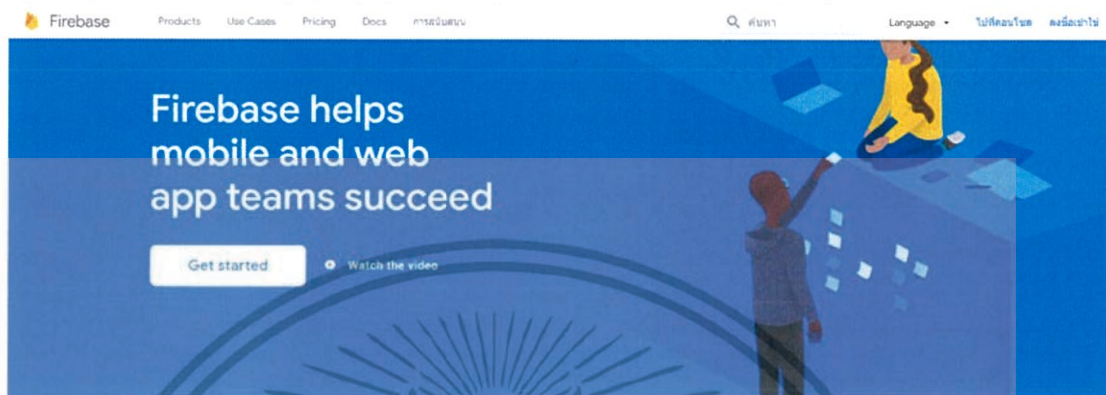


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

1. ข้อมูลผลการทดลองการเชื่อมต่อการใช้งาน Firebase

1.1 สมัครและ Login บัญชีผู้ใช้ <https://firebase.google.com/>

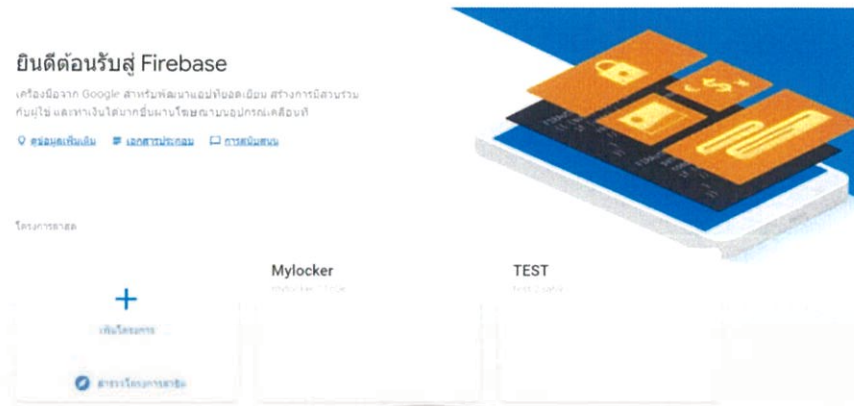


1.2 กดไปที่ Console มุมขวา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

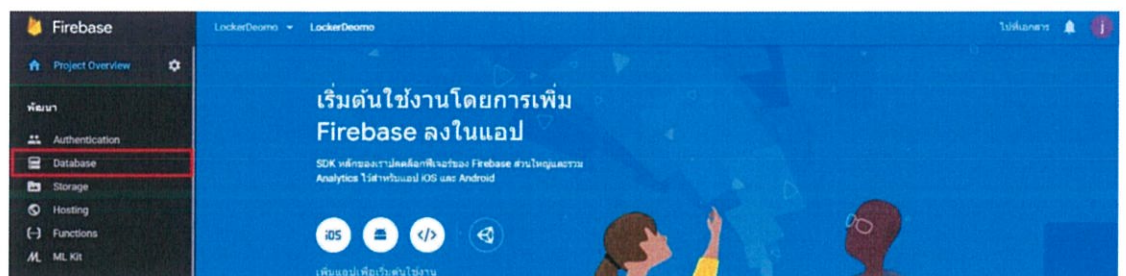
1.3 กดที่ปุ่ม + เพิ่มโครงการ



1.4 ตั้งชื่อโครงการและเลือกจันยอมรับจากนั้นกด สร้างโครงการ



1.5 จากนั้นกดเลือก Database

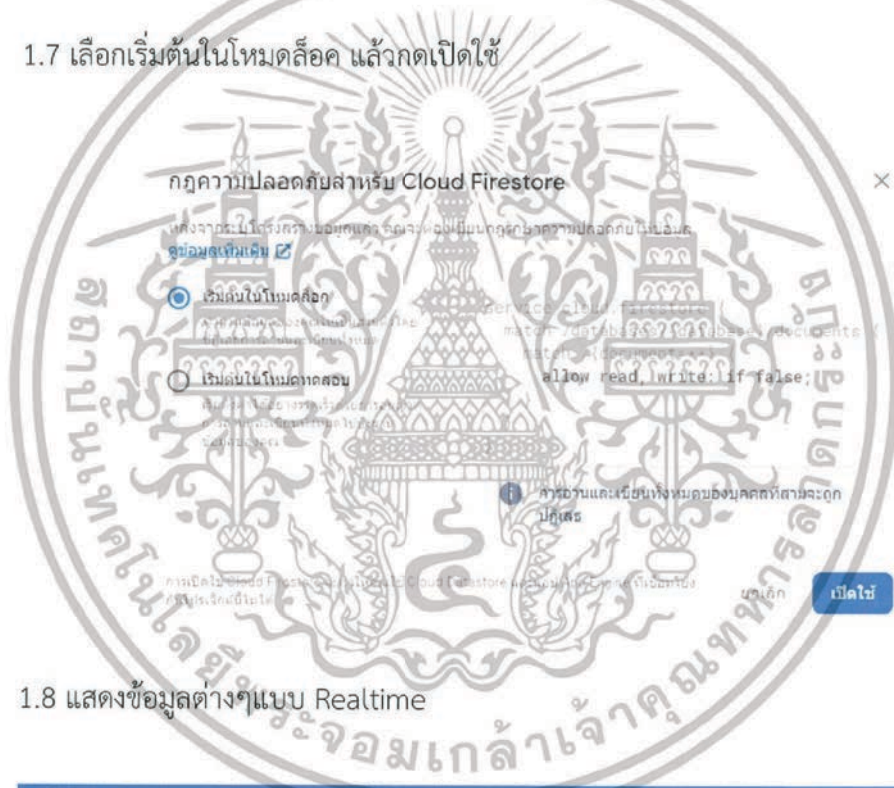


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

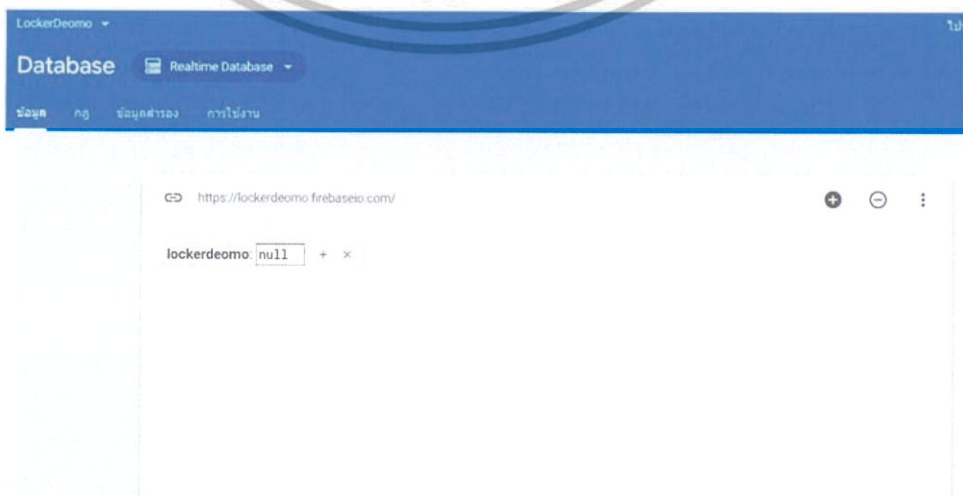
1.6 กตสร้างฐานข้อมูล



1.7 เลือกเริ่มต้นในโหมดล็อก แล้วกดเปิดใช้



1.8 แสดงข้อมูลต่างๆแบบ Realtime



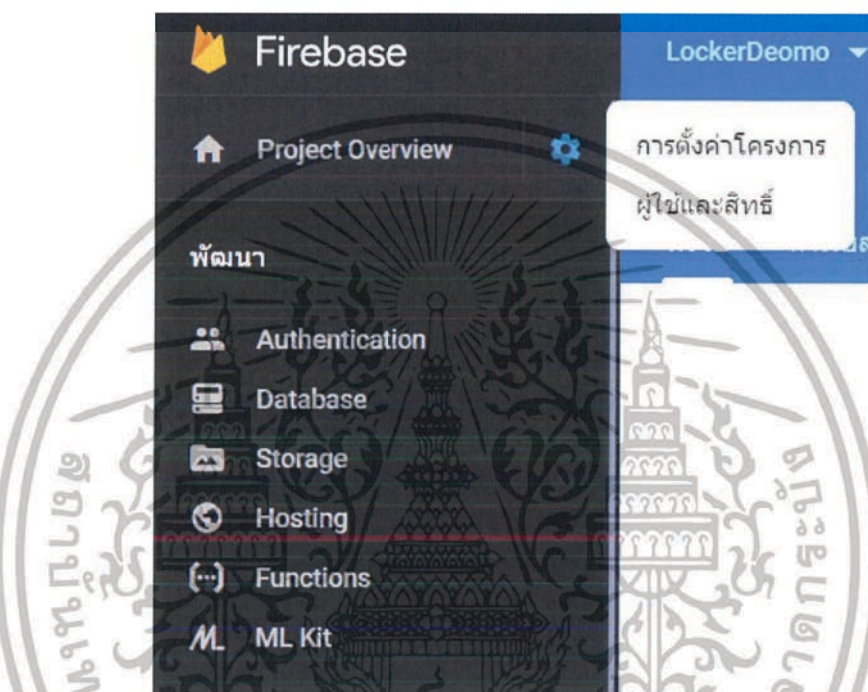
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การเชื่อมต่อ Firebase กับ Thunkable

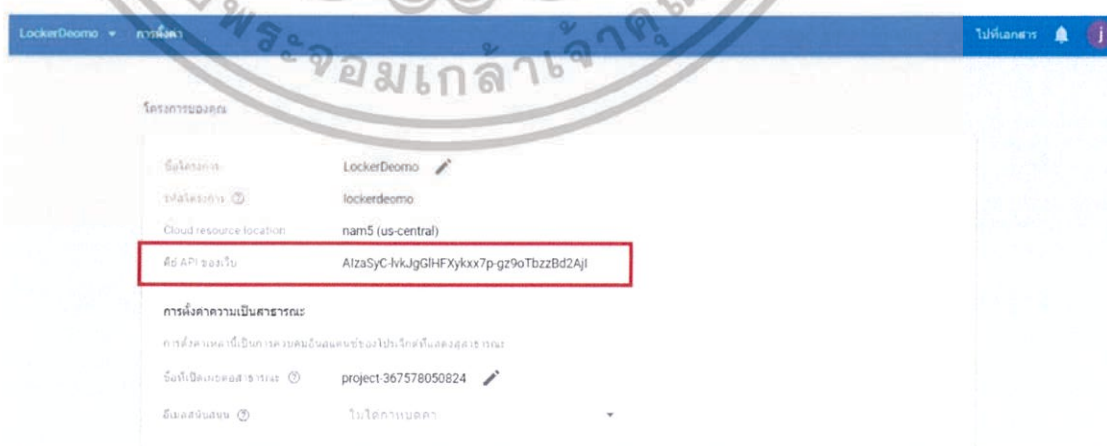
2.1 Copy URL ในการเข้าถึง Firebase ที่เป็นส่วนเชื่อมต่อ Project



2.2 กดที่รูปเฟืองแล้วกดการตั้งค่าโครงการ



2.3 จากนั้นจะแสดงหน้าขึ้นมาให้ Copy คีย์ API ซึ่งเป็นรหัสการเข้าถึง Project ไว้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 เข้า Thunkable จากนั้นกดที่ Realtime_DB1 ด้านล่างจากนั้นใส่ค่า API Key และ Database URL ที่ได้ Copy ไว้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การเข้าใช้งาน Airtable

3.1 เริ่มจากสมัครและ login ที่เว็บ <https://airtable.com/>

Airtable Product Blocks Templates Universe Pricing Sign in Sign up

Create, your way

Part spreadsheet, part database, and entirely flexible, teams use Airtable to organize their work, their way.

Enter your email address **Get started**

Already using Airtable? [Sign in](#)

Airtable

[Sign in](#) [Forgot password](#)

Email

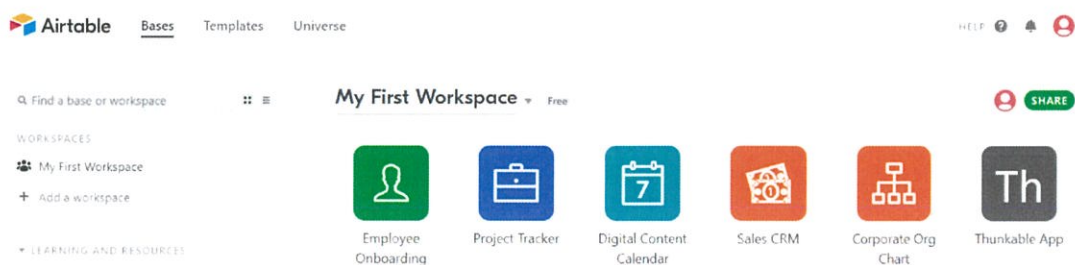
Password

[Sign in](#)

[Sign in with Google](#)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 กติที่ Thinkable App



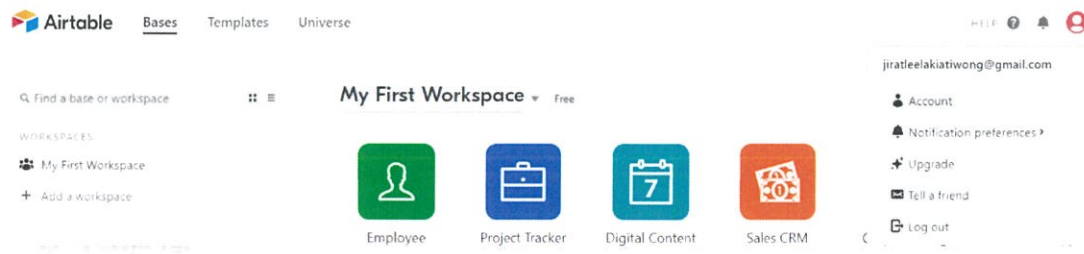
3.3 ข้อมูลจาก iOS แอปพลิเคชันที่ถูกส่งมา

	A Full-name	A Email	A Password	A Student ID	Notes
1	Jirat	Jiratleelakiwong@gmail.com	jjb2j1	58010195	Good studentw
2	Koeki	Koekki@gmail.com	Kki	56010154	EANG bang
3	Photsawat	Imeifam@gmail.com	BKK	58010741	Kasertsart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การเชื่อมต่อ Airtable กับ Thinkable

4.1 ทำการ Login แล้วกดที่ Account



4.2 Copy Airtable API เพื่อนำไปเชื่อมต่อกับ Thinkable



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 และมีค่าหนึ่งที่ต้องใช้นั้นคือ Base ID เข้าไปที่ <https://airtable.com/api> แล้วกดที่ Thunkable App



Standard API

After you've created and configured the schema of an Airtable base from the graphical interface, your Airtable base will provide its own API to create, read, update, and destroy records.

Select a base to view API documentation

My First Workspace

- Employee Onboarding
- Project Tracker
- Digital Content Calendar
- Sales CRM
- Company Directory
- Thunkable App

Activate Wind

4.4 Copy ค่า Base ID

```

curl node.js

EXAMPLE USING ENVIRONMENT VARIABLE
# Shell:
$ export AIRTABLE_API_KEY=YOUR_API_KEY

# Node:
const base = require('airtable').base('appWUo2adNy0x2ARy');

EXAMPLE USING CUSTOM CONFIGURATION
var Airtable = require('airtable');
Airtable.configure({
  endpointUrl: 'https://api.airtable.com',
  apiKey: 'YOUR_API_KEY'
});
var base = Airtable.base('appWUo2adNy0x2ARy');

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 เข้า Thinkable เลือก Spreadsheet1 และใส่ค่า API Key, Base ID ที่ได้ทำการ Copy มา

Spreadsheet1

Simple Advanced

2

APIKey	keyr1yvL0YT3QITKc	×
BaseID	appWUo2adNy0x2ARy	×
TableName	Table 1	×
ViewName	Grid view	×

Web_API1 Location_Sidebar1 Camera1 Table1

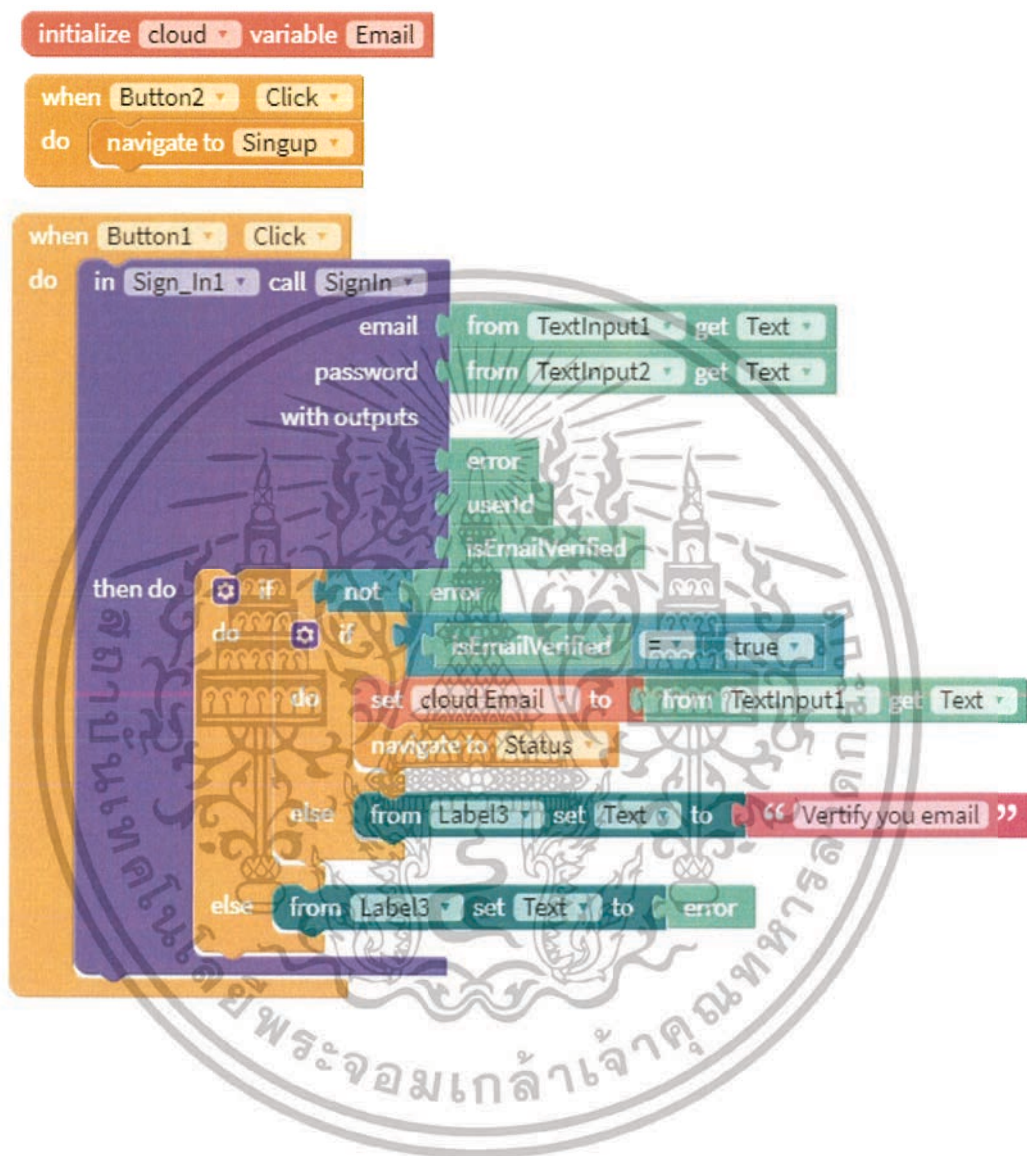
Media_DB1 Spreadsheet1

Activate Windows
Go to Settings to activate Windows.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Thunkable Block ควบคุมการทำงานของ iOS แอปพลิเคชัน pl

5.1 Block การทำงานหน้า Login



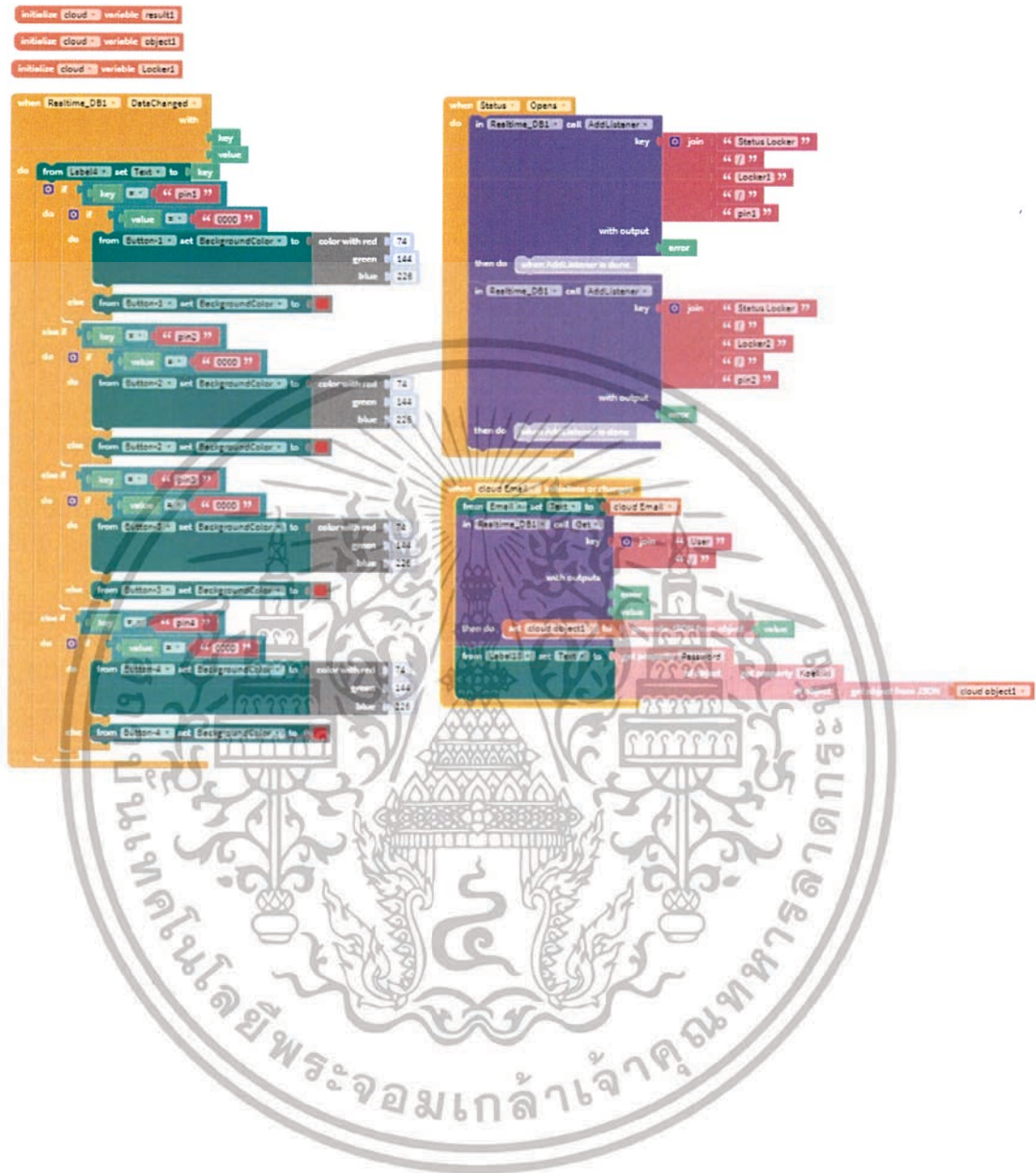
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 Block การทำงานหน้า Sing in

The image displays a complex block-based programming script for a login system. The script begins with an initialization block for a cloud variable named 'Full-name'. It then enters a 'when Button1 Click' event, followed by a series of conditional checks. First, it verifies if any of the four input fields (input1, input2, input3, input4) are empty. If they are, it triggers a message box: 'Please complete filling the data'. If all fields are filled, it proceeds to call the 'Save' function in Realtime DB, creating a new object with the following fields: 'Full-name' (from input1), 'Email' (from input2), 'Password' (from input3), and 'Student ID' (from input4). The script then updates the cloud variable 'Full-name' with the value from input1. Subsequently, it checks for a 'Button2 Click' event, which triggers a navigation to 'Screen1'. The script also includes error handling for 'Signin1' and 'Signin2' calls, showing messages like 'You've successfully entered the data' or 'Please complete filling the data'.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 Block การทำงานหน้า Home



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

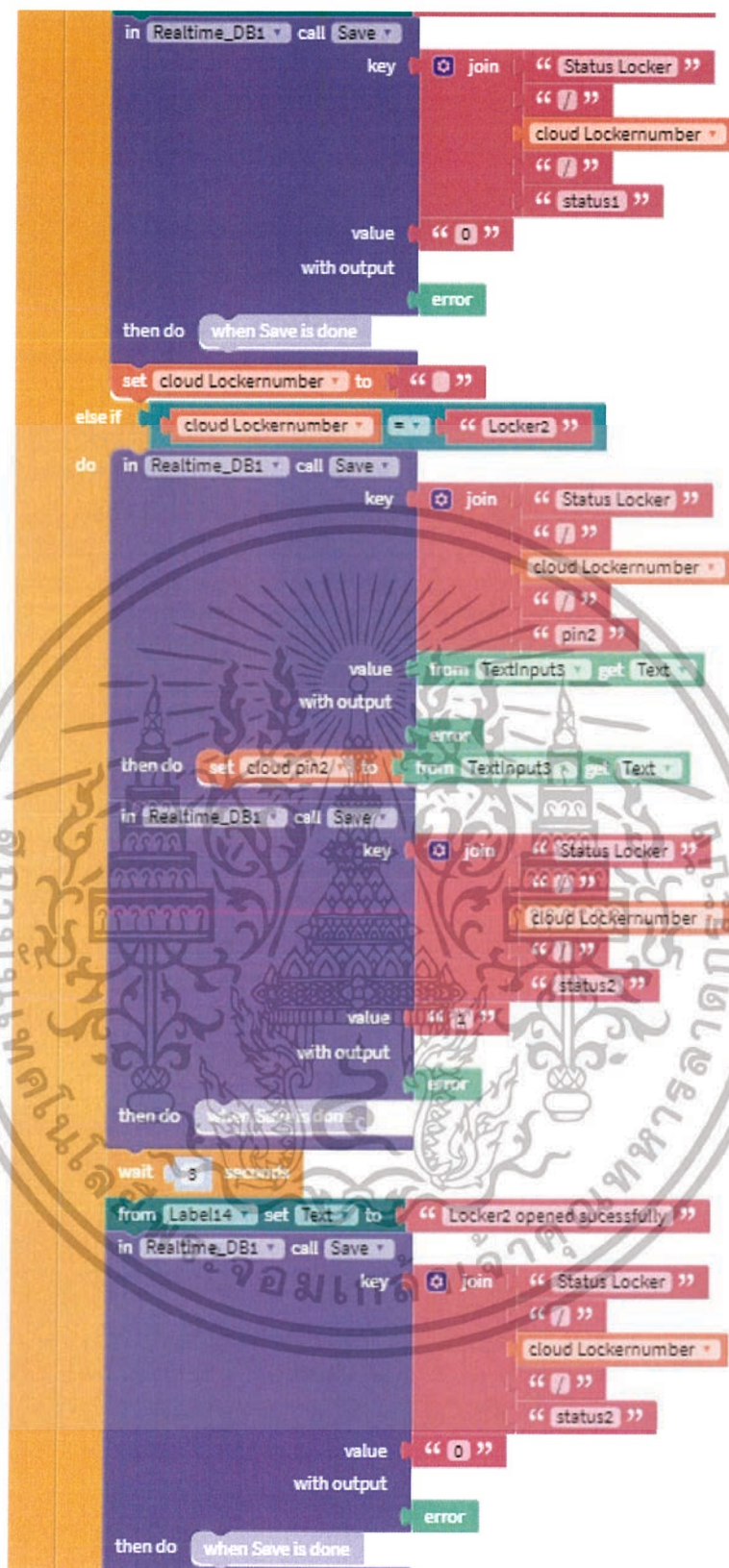
5.4 Block หน้า PIN

```

initialize cloud variable openpin1
initialize cloud variable openpin2

when Button1 Click
do
  if cloud Lockernumber = "Locker1"
  do
    in Realtime_DB1 call Save
      key join "Status Locker"
      value "Locker1"
      with output cloud Lockernumber
    then do
      in Realtime_DB1 call Save
        key join "Full-name"
        value cloud Full-name
        with output
      error
    then do
      in Realtime_DB1 call Save
        key join "pin1"
        value from Textinput3 get Text
        with output
      then do
        set cloud pin1 to from Textinput3 get Text
      in Realtime_DB1 call Save
        key join "status"
        value "1"
        with output
      error
    then do
      when Save is done
      wait 3 seconds
      from Label14 set Text to "Locker1 opened successfully"
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



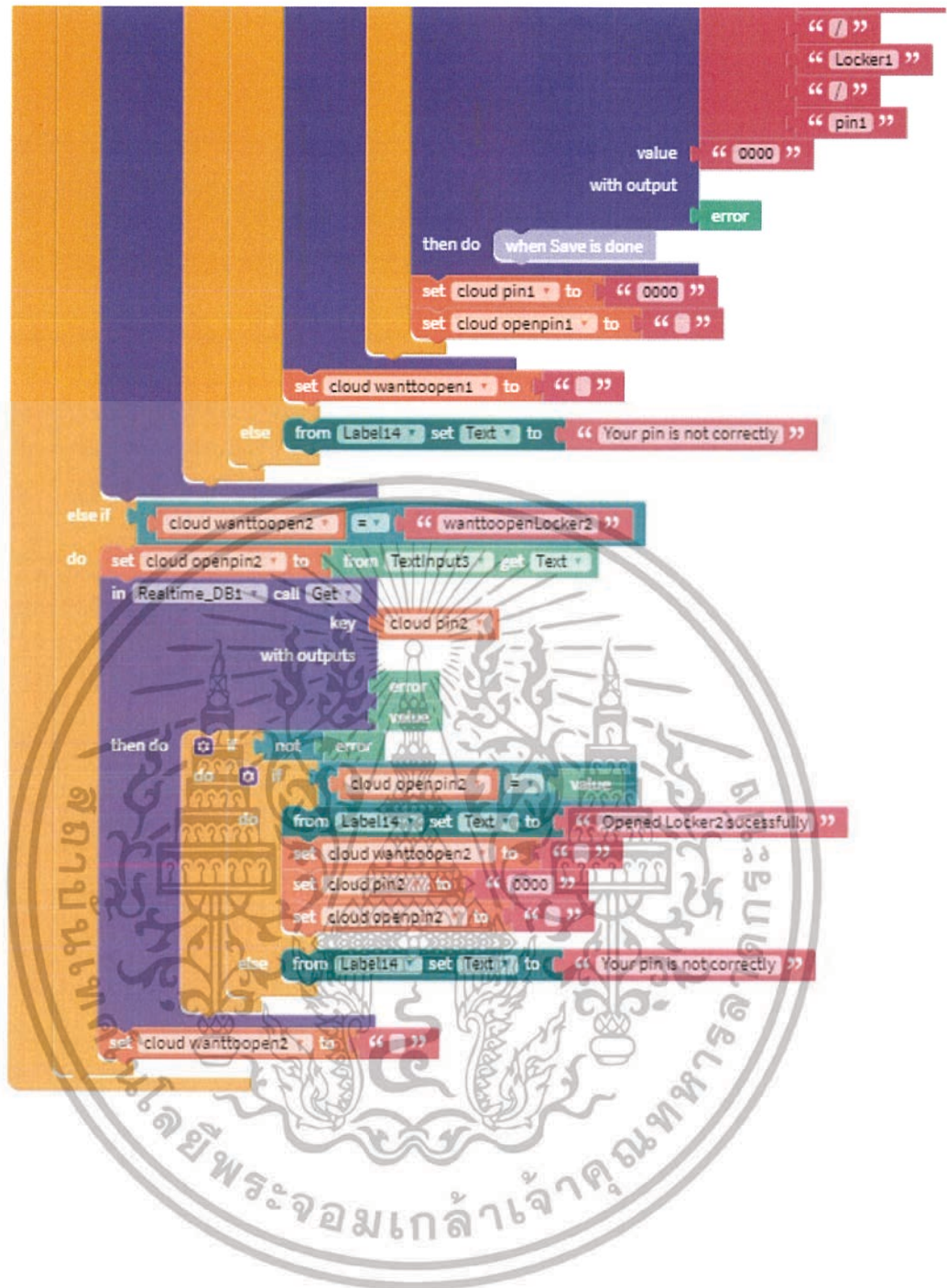
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

set cloud Lockernumber to "0"
else if cloud wanttoopen1 = "wanttoopenLocker1"
do
  set cloud openpin1 to from TextInputs get Text
  in Realtime_DB1 call Get
    key "pin1"
    with outputs
      error
      value
  then do
    if not error
    do
      if cloud openpin1 = value
      do
        from Label14 set Text to "Opened Locker1 sucessfully"
        from Alert1 set Title to "เปิด Locker1 แล้ว"
        from Alert1 set Message to "ต้องการจะปิดไหม?"
        from Alert1 set ConfirmButton to get object from JSON [{"Text":"YES","Dangerous":"false"}]
        from Alert1 set CancelButton to get object from JSON [{"Text":"NO","Dangerous":"false"}]
        in Realtime_DB1 call Save
          key join ["Status Locker", "0"]
          value [{"Locker1", "status1"}]
          with output
            error
        then do
          when Show is clicked
          wait 3 seconds
          in Realtime_DB1 call Save
            key join ["Status Locker", "0"]
            value [{"Locker1", "status1"}]
            with output
              tests: 11111
          then do
            when Show is clicked
            in Alert1 call Show
            with output
              wasConfirmed
          then do
            if not wasConfirmed
            do
              in Realtime_DB1 call Save
                key join ["Status Locker", "0"]

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 Block การทำงานหน้าสแกน

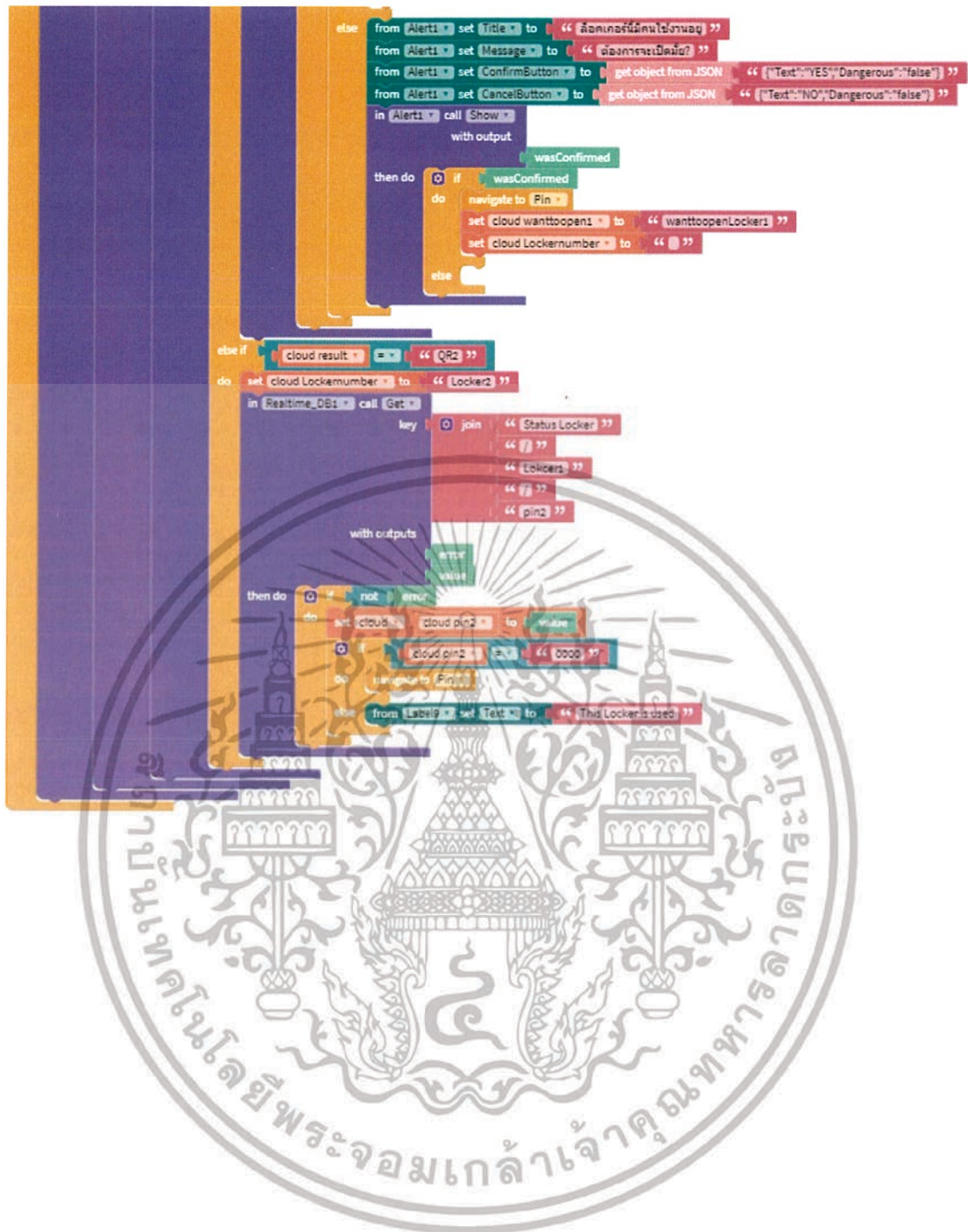
The image displays a Scratch script for a QR code scanning application. The script is organized into several nested blocks:

- Event:** when Button10 Click
- Do Loop:**
 - in Camera1:** call TakePhoto (with outputs: Photo, DidUserCancel, Error)
 - then do:**
 - from Image2:** set Picture to Photo
 - in Media_DB1:** call Upload (with outputs: media, error, mediaURL)
 - then do:**
 - from Web_API1:** set URL to (join: http://api.qrserver.com/v1/read-qr-code/?fileurl=, mediaURL)
 - in Web_API1:** call Get (with outputs: response, status, error)
 - then do:**
 - set cloud object to:** get object from JSON (response)
 - from Label1:** set text to (response)
 - set cloud result to:** get property data of object (response)
 - from Label2:** set text to (text)
 - if cloud result is QR?:**
 - do in Realtime_DB1:** call Get (key: join: Status Locker?, cloud result)
 - then do:**
 - if not error:**
 - set cloud pin1 to:** cloud pin1
 - cloud pin1:** navigate to #1
 - set cloud locknumber to:** cloud locknumber
 - set cloud locknumber to:** cloud locknumber

On the right side, there is a list of initialization blocks for cloud variables:

- initialize cloud variable wanttoopen1
- initialize cloud variable Locknumber
- initialize cloud variable wanttoopen2
- initialize cloud variable object
- initialize cloud variable result
- initialize cloud variable pin1
- initialize cloud variable pin2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้วิจัย

ชื่อ-นามสกุล นายกิริติ ฟูงธรรมสาร
 วัน เดือน ปีเกิด 8 มิถุนายน พุทธศักราช 2539
 ที่อยู่ปัจจุบัน 58/16 ถนนพระราม2 แขวงบางมด เขตจอมทอง กทม. 10150
 เบอร์โทรศัพท์ 086-3744201
 Email koekikii@gmail.com

ประวัติการศึกษา

พุทธศักราช 2546-2551 สำเร็จการศึกษาระดับชั้นประถมศึกษา
 จาก โรงเรียนสารสาสน์วิเทศบางบอน
 พุทธศักราช 2552-2554 สำเร็จการศึกษาระดับชั้นมัธยมศึกษาตอนต้น
 จาก โรงเรียนสวนกุหลาบวิทยาลัย
 พุทธศักราช 2555-2557 สำเร็จการศึกษาระดับชั้นมัธยมศึกษาตอนปลาย
 จาก โรงเรียนเตรียมวิศวกรรมศาสตร์ ไทย-เยอรมัน
 พุทธศักราช 2558-ปัจจุบัน ศึกษาต่อด้านปริญญาโท สาขาวิศวกรรมศาสตรบัณฑิต
 สาขาวิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้วิจัย

ชื่อ-นามสกุล	นายจิรัฏฐ์ สีลาเกียรติวงศ์
วัน เดือน ปีเกิด	23 มิถุนายน 2540
ที่อยู่ปัจจุบัน	75 รังสิต-นครนายก22 ตำบลประชาธิปไตย อำเภอดัญบุรี จังหวัดปทุมธานี รหัสไปรษณีย์ 12130
เบอร์โทรศัพท์	0633431040
Email	Jiratleelakiatiwong@gmail.com
ประวัติการศึกษา	
พุทธศักราช 2552-2557	สำเร็จการศึกษาระดับชั้นมัธยมศึกษาตอนต้น - ปลาย โรงเรียนศรีอยุธยา ในพระอุปถัมภ์ฯ
พุทธศักราช 2558-ปัจจุบัน	ศึกษาระดับอุดมศึกษา หลักสูตรวิศวกรรมแมคคาทรอนิกส์ ภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้