ล็อคเกอร์ฝากของอัจฉริยะ

SMART LOCKER

ยอดธง ไชยประสิทธิ์

ศุภกร ขำสม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ล็อคเกอร์ฝากของอัจฉริยะ

นายยอดธง	ไชยประสิทธิ์	60015038
นายศุภกร	ขำสม	60015047
อาจารย์สรยุทธ	กลมกล่อม	อาจารย์ที่ปรึกษา

ปีการศึกษาที่ 2562

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการออกแบบ และพัฒนาล็อคเกอร์ฝากของอัจฉริยะ ที่มีลักษณะที่เป็นตู้ล็อคเกอร์สามารถใช้งานได้ทั้งหมดจำนวน 3 ช่อง โดยใช้ Kotlin ในการสร้าง Mobile Application เพื่อเข้าใช้งานระบบ และปลดล็อคแต่ละตู้ได้อย่างอิสระ สามารถเก็บข้อมูลการใช้งานตู้ล็อคเกอร์ไว้ในระบบฐานข้อมูล ที่ใช้ firebase เป็นแม่ข่ายระบบฐานข้อมูล เพื่อตรวจสอบการใช้งานได้ในภายหลัง ในกระบวนการออกแบบทางด้านฮาร์ดแวร์ได้ใช้ไมโครคอนโทรลเลอร์ ESP32 เชื่อมต่อกับอุปกรณ์ฐานข้อมูลที่บันทึกไว้ จากนั้นจะทำการประมวลผลข้อมูลที่ได้กับฐานข้อมูลภายในที่กำหนดไว้ เพื่อส่งผลไปควบคุมกลอนไฟฟ้าที่ติดอยู่ภายในล็อคเกอร์แต่ละช่อง และส่งข้อมูลการใช้งานผ่านระบบเครือข่ายคอมพิวเตอร์แบบไร้สาย ไปยังระบบฐานข้อมูลที่ติดตั้งไว้บนไมโครคอนโทรลเลอร์ ESP32 โดยได้ออกแบบส่วนแสดงผลบนหน้าจอ Mobile Application แสดงสถานะการใช้งานของตู้ล็อคเกอร์ในแต่ละช่อง โดยใช้การเขียนโปรแกรมด้วยภาษา Kotlin ดึงข้อมูลจากระบบฐานข้อมูลออกมาแสดงผล ซึ่งตู้ล็อคเกอร์ที่ได้ออกแบบ และพัฒนานี้ สามารถเข้าใช้งานได้โดยอิสระต่อกันในแต่ละช่องโดยการใช้ User และ Password 1 ID ต่อล็อคเกอร์ 1 ช่อง และสามารถเรียกดูข้อมูลการใช้งานปัจจุบัน และย้อนหลังได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SMART LOCKER

Mr.Yodthong Chaiprasit 60015038
Mr.Suphakorn Khamsom 60015047
Mr.Sorayut Glomglome Advisor

Academic Year 2562

ABSTRACT

This thesis presents the design and development of Smart Locker. Smart Locker is locker that can be used in a total of 3 boxes. The box uses Kotlin to create a Mobile application to access the system and unlock each box freely. Able to keep usage information of locker in a database system that uses firebase as a database server for the process of checking usage later. The hardware design process include of using the ESP32 microcontrollers connect to the database for processing the obtained data with the internal database. In order to control the electric bolt lock attached to each box and send usage information via the wireless network to the database system installed on the ESP32. The design on the Mobile Application display shows the active status of each box in the locker by programming Kotlin, pulling data from the database to display. The Smart Locker can be used to access each box independently by using User and Password 1 ID per box and can view current usage and usage history.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีด้วยความช่วยเหลือจากหลายฝ่ายทั้งในทางตรงและทางอ้อม ปริญญานิพนธ์ฉบับนี้จะสำเร็จลงไม่ได้หากปราศจากความช่วยเหลือของบุคคลเหล่านี้ ขอขอบคุณ อาจารย์ที่ปรึกษา คือ อาจารย์สรยุทธ กลมกล่อม เป็นผู้ให้คำแนะนำ คำปรึกษา และให้ความช่วยเหลือตลอดการทำโครงงาน ซึ่งทำให้การทำงานต่าง ๆ เป็นไปได้อย่างราบรื่นและสำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณอาจารย์และบุคลากรต่าง ๆ ในสาขาวิชาวิศวกรรมคอมพิวเตอร์ที่ได้ให้คำแนะนำและคำสั่งสอนความรู้ต่าง ๆ มาโดยตลอด

ขอขอบคุณรุ่นพี่และเพื่อนหลาย ๆ คน ในภาควิชาวิศวกรรมคอมพิวเตอร์ได้ให้คำแนะนำและคำปรึกษาอย่างเต็มที่และแบ่งปันความรู้ในทุก ๆ ด้าน

ในท้ายที่สุดนี้ ขอขอบพระคุณ บิดา มารดา และความครัวที่ได้เลี้ยงดู สั่งสอน และให้การสนับสนุน พร้อมทั้งให้โอกาสในการศึกษาและให้กำลังใจเสมอมา

ยอดธง ไชยประสิทธิ์

ศุภกร ขำสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ V

สารบัญรูปภาพ VII

บทที่ 1 บทนำ 1

1.2 วัตถุประสงค์โครงงาน 2

1.3 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน 2

1.4 ขอบเขตของโครงงาน 2

1.5 ตารางการดำเนินงาน 4

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 5

2.1 ล็อคเกอร์ฝากของอัจฉริยะ 5

2.2 เทคโนโลยีที่เกี่ยวข้องกับการรับข้อมูล 5

2.3 Kotlin 7

บทที่ 3 การออกแบบและการพัฒนา 19

3.1 ภาพรวมการทำงานในระบบ 19

3.2 การเข้าระบบรับ-ฝากของใน Mobile application 20

3.3 รูปของ mobile application 21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.4 โครงสร้างของระบบ 22

3.5 Flowchart การทำงานของล็อคเกอร์ 23

3.6 Use case Diagram 24

3.7 บล็อกไดอะแกรมแสดงการทำงานของ Mobile Application.....25

3.8 ER Diagram 26

บทที่ 4 การใช้งานและการทดลอง.....27

4.1 การทดสอบการเข้า Mobile application บนสมาร์ทโฟน.....27

4.2 การทดสอบฝากของเข้าระบบ 28

4.3 การทดสอบรับของออกจากระบบ.....30

4.4 การทดสอบการใส่รหัสผิดในการเปิดล็อคเกอร์ 32

4.4 ทดสอบการใช้กลอนไฟฟ้ากับ Mobile Application 33

บทที่ 5 บทสรุปและข้อเสนอแนะ.....36

5.1 บทสรุป.....36

5.2 ปัญหาและอุปสรรค.....36

5.3 แนวทางแก้ปัญหา.....37

5.4 แนวทางการพัฒนาต่อ 37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป **หน้า**

1.1 ล็อคเกอร์ของสถาบันในห้องสมุด 1

1.2 ตารางการดำเนินงาน 4

2.1 Kotlin 7

2.2 Hello World ของ Kotlin 8

2.3 ตัวอย่างโค้ด Android ในภาษา Kotlin 13

2.4 หลักการเขียนโค้ดของ Java และ Kotlin 14

2.5 ตัวอย่างโค้ดที่ไม่จำเป็น 14

2.6 Similar with Swift 15

2.7 Functions Parameters 15

2.8 CLASS INHERITANCE.OVERRIDING 16

2.9 Low Learning Curve 16

2.10 Free Lambda 17

2.11การรองรับภาษาของ Kotlin 18

3.1 รูปของ mobile application 21

3.2 ไดอะแกรมของระบบ 22

3.3 Flowchart การทำงานของล็อคเกอร์ 23

3.4 Use case Diagram 24

3.5 บล็อกไดอะแกรมแสดงการทำงานของ Mobile Application 25

3.6 ER Diagram 26

4.1 หน้าการเข้า Mobile Application 27

4.2 (a) ช่องบอกสถานะของล็อคเกอร์ 28

4.2 (b) หน้าการฝากของในล็อคเกอร์ 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป **หน้า**

4.2 (c) การกรอกข้อมูล 28

4.3 (a) การเปิดตู้ในการฝากของสำเร็จ 29

4.3 (b) ช่องบอกสถานะของล็อคเกอร์ที่ใช้งานไปแล้ว 29

4.4 (a) ช่องบอกสถานะของล็อคเกอร์ที่ใช้งานไปแล้ว 30

4.4 (b) เลือกช่องล็อคเกอร์ที่จะทำการรับของคืน 30

4.4 (c) ใส่รหัสเปิดตู้ล็อคเกอร์ 30

4.5 (a) การยืนยันตัวตนรับของคืน 31

4.5 (b) ยืนยันคืนช่องล็อคเกอร์ 31

4.5 (c) เปิดช่องสำเร็จ 31

4.6 ทดสอบการใส่รหัสผ่านไม่ถูกต้อง 32

4.7 ทำการเปิดล็อคเกอร์เพื่อการฝากของ 33

4.8 ทำการฝากของเข้าไปและทำการปิดล็อคเกอร์ 34

4.9 ทำการเปิดล็อคเกอร์เพื่อทำการรับของคืน 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากปัจจุบันการพัฒนาสิ่งต่างๆ ในการดำรงชีวิตของผู้คนนั้นถูกพัฒนาไปได้อย่างรวดเร็ว ซึ่งปัจจุบันเริ่มมีการพัฒนาอุปกรณ์ที่สามารถป้องกันภัยจากโจรกรรมได้เพื่อเพิ่มประสิทธิภาพให้แก่ อุปกรณ์นั้นๆ ในด้านต่างๆ เช่น กล้องวงจรปิด สัญญาณกันขโมย เป็นต้น และต้องมีความปลอดภัยในระดับหนึ่งด้วยแต่เมื่อเราอยู่นอกบ้านนั้นเราจะป้องกันสิ่งของมีค่าที่อยู่ในตัวเราได้อย่างไร ผู้จัดทำโครงงานจึงเล็งเห็นว่าจะสร้างสิ่งของอย่างนึงที่ใช้การเชื่อมต่อกับการรักษาความปลอดภัยให้เป็นประโยชน์ และได้ข้อสรุปว่าจะทำล็อคเกอร์อัจฉริยะ (Smart Locker) เนื่องจาก ในปัจจุบันบริการรับฝากของยังไม่ค่อยมีให้เห็นตามที่สาธารณะสักเท่าไหร่อย่างเช่น หอสมุดหรือโรงยิม ทำให้ต้องระมัดระวังในการป้องกันของมีค่าล็อคเกอร์ บริการสาธารณะที่มีอยู่ทั่วไปแต่เป็นบริการที่เสียเงินในการรับฝากของและมีราคาค่อนข้างแพง บริการของผู้จัดทำโครงงานจึงคิดใช้ล็อคเกอร์รับฝากของในห้องสมุดของมหาวิทยาลัยโดยให้บริการฟรี เอามาแทนบริการรับฝากของจากเดิมที่ฝากผ่านเจ้าหน้าที่หรือล็อคเกอร์ของห้องสมุดที่มีบริการการใส่กุญแจแล้วล็อครหัส 4 ตัว

รูป 1.1 ล็อคเกอร์ของสถาบันในห้องสมุด

http://www.lib.kmitl.ac.th/2018/TH/images/pdf/KMITLCentralLibrary_HowtoUseLocker.pdf

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราอาจจะลืมรหัสผ่านได้หรือบางที่อาจมีคนอื่นมาโจรกรรมเดารหัสผ่านเราได้และอาจจะไม่ปลอดภัยเพราะเราไม่ได้เฝ้าของที่อยู่ในล็อคเกอร์นั้นตลอดเวลา ถ้าเราขนของมามากทำให้อาจเกิดความไม่ระมัดระวังได้ หรือบางอย่างที่เราไม่ใช้ก็ฝากเอาไว้ได้และค่อยมารับของคืน ในปัจจุบันล็อคเกอร์ของห้องสมุดนั้นไม่สามารถรับ-ฝากของได้ข้ามวัน และถ้าเราลืมของนั้นๆไว้ผู้ฝากจะต้องมีค่าปรับจะทำให้มีเสียค่าปรับ 50 บาท ทำให้การฝากของอาจจะทำให้เป็นการฝากแค่เวลาอันสั้นๆ และอาจจะทำให้หลงลืมและมาเสียค่าปรับไปฟรีๆได้ ผู้จัดทำโครงงานจึงเล็งเห็นความสำคัญของความปลอดภัย จึงได้ดัดแปลงล็อคเกอร์รับฝากของโดยใช้ Mobile Application มาประยุกต์ใช้ควบคุมการเปิดปิดของกลอนไฟฟ้าและบันทึกข้อมูล ลงฐานข้อมูลแล้วนำข้อมูลมาแสดงผลผ่านระบบอินเตอร์เน็ต ของที่อยู่ในล็อคเกอ์ถ้าไม่ได้ใช้ข้อมูลหรือ Password ที่ถูกต้องในการเปิด ล็อคเกอร์ก็จะไม่สามารถถูกเปิดได้โดยเด็ดขาด ทำให้ช่วยในเรื่องของความปลอดภัยสูง ส่วนการแสดงผลของข้อมูลผ่านจอ Mobile Application ก็จะทำให้สามารถรู้ได้ว่า ล็อกเกอร์ที่เท่าไหร่ชั้นไหนถูกใช้งานไปใน เวลาไหนบ้าง ทำให้ผู้ที่ต้องการจะฝากของสามารถตรวจสอบข้อมูลการใช้งานได้และไม่ต้องกังวลใจใน เรื่องของหาย

1.2 วัตถุประสงค์โครงงาน

1) ศึกษาและประยุกต์ใช้งาน Mobile Application
2) ศึกษาและประยุกต์ใช้งาน ESP32 ในการติดต่อ Server
3) ศึกษาและประยุกต์ใช้งาน Firebase ในการเก็บฐานข้อมูลของผู้ใช้งาน
4) ศึกษาและระบบการรับ-ฝากของโดยใช้เทคโนโลยีที่กล่าวมาข้างต้น

1.3 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน

1) อำนวยความสะดวกแก่คนที่จะมาฝากของชั่วคราว
2) สามารถตรวจสอบข้อมูลการใช้งานของล็อคเกอร์ได้
3) ไม่ต้องกังวลว่าของในล็อคเกอร์จะหายไป เพราะตู้จะไม่ถูกเปิดหากไม่ใช้รหัสผ่านที่ถูกต้อง
4) เข้าใจการทำงานของ Mobile Application

1.4 ขอบเขตของโครงงาน

การทำงานโครงการ เข้า Mobile Application สมัครการใช้งานทำการกรอกข้อมูล เลือกตู้ที่มีสถานะว่างดูจะสถานะว่างของตู้จากหน้าจอ Mobile Application ทำการฝากของ จะได้รับรหัสในการเปิดรับของในวันที่มาเอาของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4.1 Hardware

1) การทำงานด้าน Hardware

1.1 ตู้ล็อคเกอร์เป็นล็อคเกอร์ที่สามารถสั่งล็อคได้ด้วยกลอนไฟฟ้า ซึ่งไฟดับก็ยังสามารถล็อคอยู่ จะปลดล็อคไม่ได้

1.2 ทำงานโดยบอร์ด ESP32 ส่งข้องมูลติดต่อกับ Server โดยใช้ Wi-Fi

1.3 หน้าจอ Mobile Application แสดงสถานะของตู้ที่ถูกใช้งานไปและตู้ที่ว่าง

1.4.2 Software

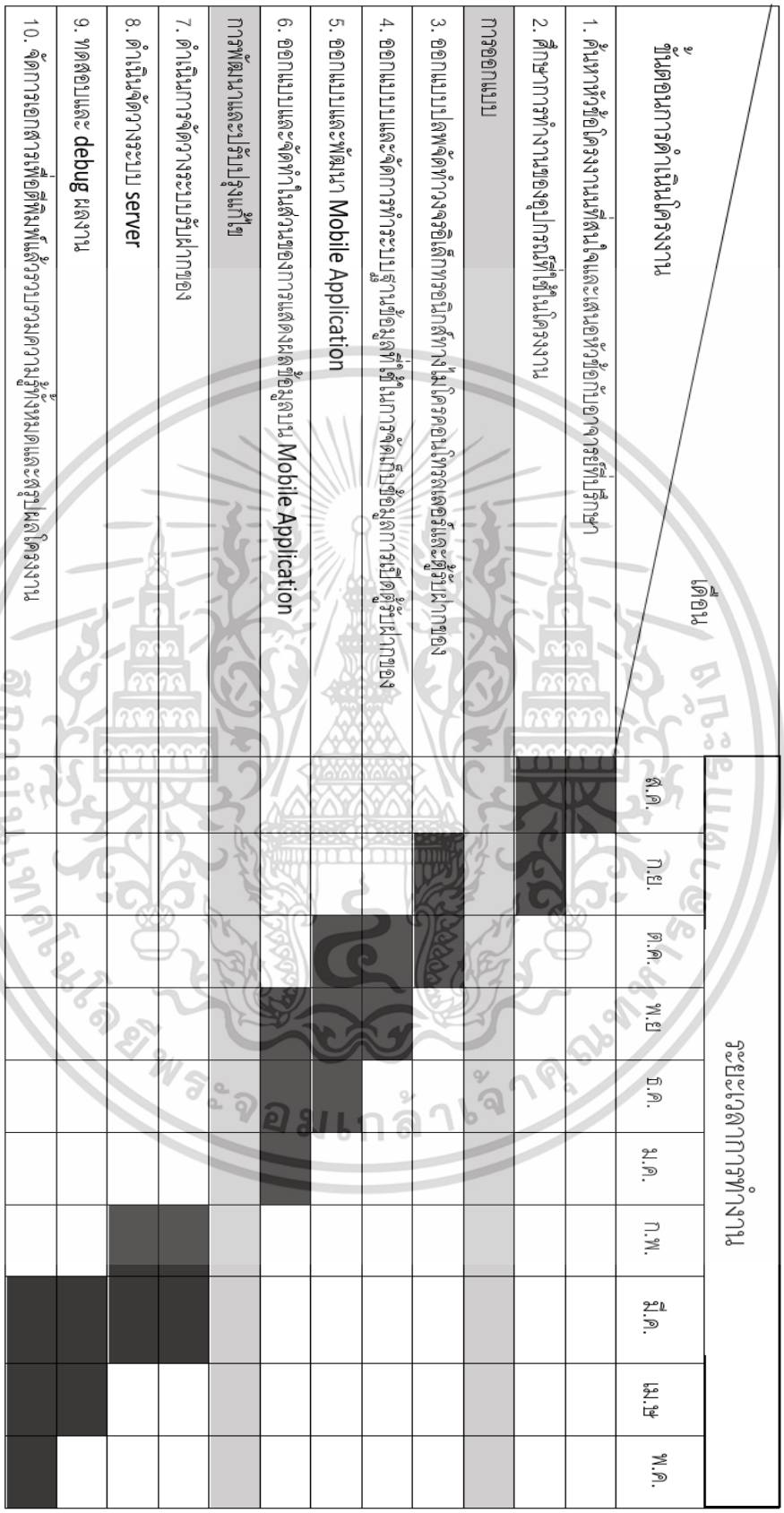
2) การทำงานด้าน Mobile Application

2.1 รับข้อมูล Input จากผู้ใช้ แล้วมีUser และ Password ในการปลดล็อค

2.2 วิเคราะห์ข้อมูล และจัดเก็บลงฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ตารางการดำเนินงาน



รูป 1.2 ตารางการดำเนินงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ล็อคเกอร์ฝากของอัจฉริยะ

ล็อคเกอร์รับฝากของโดยใช้ Mobile Application มาประยุกต์ใช้ควบคุมการเปิดปิดของกลอนไฟฟ้าและบันทึกข้อมูล ลงฐานข้อมูลแล้วนำข้อมูลมาแสดงผลผ่านระบบอินเตอร์เน็ต ของที่อยู่ในล็อคเกอร์ถ้าไม่ได้ใช้ข้อมูลหรือ Password ที่ถูกต้องในการเปิด ล็อคเกอร์ก็จะไม่สามารถถูกเปิดได้โดยเด็ดขาด ทำให้ช่วยในเรื่องของความปลอดภัยสูง ส่วนการแสดงผลของข้อมูลผ่านจอ Mobile Application ก็จะทำให้สามารถรู้ได้ว่า ล็อกเกอร์ที่เท่าไหร่ชั้นไหนถูกใช้งานไปใน เวลาไหนบ้าง ทำให้ผู้ที่ต้องการจะฝากของสามารถตรวจสอบข้อมูลการใช้งานได้และไม่ต้องกังวลใจในเรื่องของหาย

2.2 เทคโนโลยีที่เกี่ยวข้องกับการรับข้อมูล

จากที่ได้ทำการศึกษาขอ้มูลเกี่ยวกับการทำล็อคเกอร์รับฝากของโดยทั่วไปแล้ว จะใช้ Application ในการใช้งานโดยใช้ controller คอยรับส่งข้อมูล อุปกรณ์และระบบต่างๆ

2.2.1 Controller

บอร์ด ESP32 เป็นบอร์ดไมโครคอนโทรลเลอร์ขนาดเล็กราคาถูก สามารถเขียนโปรแกรมติดต่อกับ ภายนอกได้ง่าย

2.2.2 อุปกรณ์ต่างๆที่ใช้ในการทำล็อคเกอร์อัจฉริยะ

2.2.2.1 กลอนลิ้นชักไฟฟ้า

เปิดปิดอุปกรณ์ เมื่อเราจ่ายไฟ 12V โซลินอยด์ตัวนี้จะทำการดูดเหล็กที่ยื่นออกมาเข้าไป และเมื่อไม่ได้จ่ายไฟ สปริงจะดันกลอนกลับมาที่เดิม

2.2.2.2 รีเลย์

ใช้กระแสไฟฟ้าเป็นตัวสั่งการ แทนที่จะใช้มือกด เหมือนปุ่ม button หรือสวิทซ์อื่นๆ

2.2.3 ภาษาที่ใช้ทำ Mobile Application

ใช้ในการสมัครสมาชิกในการรับ-ฝากของจากตู้ล็อคเกอร์เพื่อความปลอดภัยในการฝากของ และสะดวกต่อการใช้งานได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.1 Kotlin

เป็นภาษาโอเพ่นซอร์สที่มีรากฐานมาจากจาวาเวอร์ช่วลแมชชีนหรือ JVM ที่ใช้ในการพัฒนาแอพบนแอนดรอยด์Kotlin เป็นภาษาลูกของภาษาโปรแกรมมิ่งสัญชาติรัสเซียที่มีชื่ออย่าง JetBrains ซึ่ง Kotlin นี้ก็ได้เข้ามาเป็นส่วนหนึ่งของ Android Studio 3.0 เวอร์ชั่นล่าสุดด้วย

2.2.3.2 Java

เป็นภาษาที่ได้รับความนิยมสูงสุดทั้งแอพบน iOS และแอนดรอยด์ ด้วยการพอร์ตโค้ดไปมาข้ามแพลตฟอร์ม และนำโค้ดเดิมมาใช้ใหม่ได้ ไม่เพียงแพลตฟอร์มอุปกรณ์พกพาเท่านั้น โค้ดเดียวกันนี้ยังพอร์ตไปรันบนเวอร์ช่วลแมชชีนและบราวเซอร์ต่างๆ ได้ด้วย ทำให้มีผู้ใช้งานจำนวนมากฝึกจนถึงขั้นเชี่ยวชาญ และรวมตัวเป็นชุมชนออนไลน์เพื่อช่วยเหลือเพื่อนร่วมอุดมการณ์ที่ใช้โอเพ่นซอร์สตัวนี้ รวมถึงมีไลบรารีและทูลให้นำมาใช้ฟรีเต็มไปหมดจนทำให้ผู้ใช้งานทำงานง่ายขึ้นมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Kotlin

Kotlin เป็นผลงานของบริษัท JetBrains บริษัทซอฟต์แวร์จากยุโรปตะวันออก (สำนักงานใหญ่ตั้งอยู่ที่สาธารณรัฐเช็ก มีสำนักงานอยู่ที่ประเทศรัสเซียด้วย) ซึ่งรู้จักกับบริษัทนี้ในฐานะผู้สร้าง IntelliJ IDEA ซอฟต์แวร์ IDE

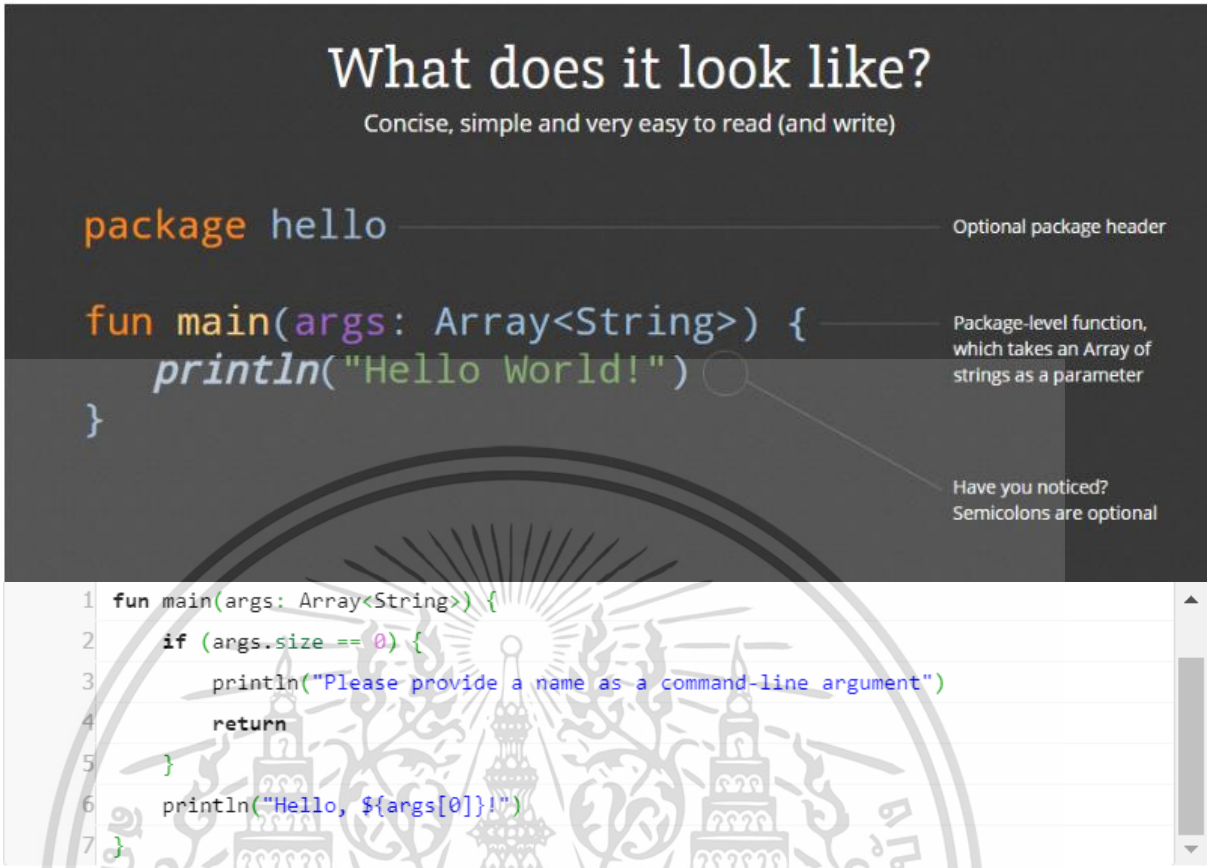
JetBrains มองเห็นข้อจำกัดของ Java (ถือเป็นคนละส่วนของ Java Platform และ JVM) และทำการพัฒนาภาษาโปรแกรมตัวใหม่ขึ้นมาเพื่อใช้แทน ภาษา Java แต่ยังคง Compile เป็น Bytecode เพื่อ Run บน JVM เหมือนเดิม

เนื่องจากทีมงานของ JetBrains ที่ทำการพัฒนาภาษานี้อยู่ที่เมือง Saint Peterburg ประเทศ Russia ได้นำชื่อของเกาะ KotlinIsland ในอ่าวด้านหน้าเมืองของ Saint Peterburg แล้วแต่งตั้งเป็นชื่อภาษาของ Kotlin นั่นเอง

ภาษา Kotlin ถือกำเนิดขึ้นในปี ค.ศ.2011 และได้ความนิยมมากขึ้นของโลกผู้ใช้งาน Java ควบคู่กับ JVM

รูป 2.1 Kotlin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.2 Hello World ของ Kotlin

Syntax ของ Kotlin เป็นการผสมของภาษาในตระกูล C ที่ใช้ปีกกา {} และไม่จำเป็นต้องมี Semicolon (;) เป็นตัวสุดท้ายของบรรทัด (จะทำการใส่หรือไม่ทำการใส่ก็ได้) จุดที่ต่างสำคัญอีกประการหนึ่งของ Kotlin คือการที่ประกาศประเภทของตัวแปร (Type) จะอยู่ด้านหลังชื่อตัวแปร (Variable name) โดยมีเครื่องหมาย Colon (:) คั่นตรงกลาง เช่น a: Int และ s: String ตรงนี้มีความคล้ายกับภาษาอย่างเช่น Go และ TypeScript

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวคิดของภาษา Kotlin

Andery Breslav หัวหน้าของทีมออกแบบภาษา Kotlin ให้สัมภาษณ์ไว้ว่าตั้งใจออกแบบภาษา Kotlin ขึ้นมาเป็นภาษาใหม่เพื่อใช้แทนของภาษา Java ที่มีจุดบกพร่องหลายอย่าง และไม่สามารถแก้ไขอะไรให้ดีขึ้นมากนักเพราะต้องทำการ Backward Compatibility ตอนแรก ทีมงานของ JetBtains สำรวจแล้วว่ามีภาษาโปรแกรมที่มีความต้องการหรือไม่ แต่พบว่าในท้องตลาดมีแค่เพียง Scala เท่านั้นที่มีความใกล้เคียงมากที่สุด แต่จริงๆ ปัญหาของ Scala คือการ Compile ที่ช้าและเรียนรู้ได้ยาก ส่งผมให้ทีมงานนั้นตัดสนใจกันสร้างภาษาที่ใหม่และมีความที่เข้าใจง่ายกว่าขึ้นมาแทน

แนวคิดของ Kotlin คือความเข้ากันได้ 100% กับ Java Platform เพื่อที่ใช้ประโยชน์จาก Library Framework API และเครื่องมือจำนวนมากที่มีอยู่บนโลกของ Java และสามารถนำมาผสมผสาน Code ภาษา Java/Kotlin ได้ และเป้าหมายของ Kotlin ก็คือการนำมาใช้แทน Java ในโครงการใหม่ๆ โดยการรักษา Code Java ในโครงการเก่าที่เขียนไปแล้วและยังสามารถทำงานร่วมกันได้

Kotlin เป็นภาษาที่ Statically-typed หรือการประกาศชนิดของตัวแปรอย่างชัดเจนเสมอด้วยเหตุผลนี้ เรื่องความปลอดภัยของปัญหาเรื่องหน่วยความจำ (Type Safety) และประสิทธิภาพ ต่างออกไปจากภาษา Groovy , Jython , Ruby ที่เป็นของ Dynamically-Typed

จุดเด่นอีกอย่างของภาษา Kotlin ที่เหนือกว่าภาษา Java คือ มีความสั้นกระชับกว่า ตัวเลขของ JetBrains ระบุว่า สามารถลดจำนวนของ Code ลงไปได้ถึง 40% ถ้าเปลี่ยน Program

ตัวอย่าง Code Java

```
1.  public class Customer {
2.    private String name;
3.    private String email;
4.    private String company;
5.
6.    public Customer(String name) {
7.      this(name, "", "");
8.    }
9.
10.   public Customer(String name, String email) {
11.     this(name, email, "");
12.
13.   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
14.
15.    public Customer(String name, String email, String company) {
16.        this.name = name;
17.        this.email = email;
18.        this.company = company;
19.    }
20.
21.    public String getName() {
22.        return name;
23.    }
24.
25.    public void setName(String name) {
26.        this.name = name;
27.    }
28.
29.    public String getEmail() {
30.        return email;
31.    }
32.
33.    public void setEmail(String email) {
34.        this.email = email;
35.    }
36.
37.    public String getCompany() {
38.        return company;
39.    }
40.
41.    public void setCompany(String company) {
42.        this.company = company;
43.    }
44.
45.    @Override
46.    public boolean equals(Object o) {
47.        if (this == o) return true;
48.        if (o == null || getClass() != o.getClass()) return false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
49.
50.         Customer customer = (Customer) o;
51.
52.         if (name != null ? !name.equals(customer.name) : customer.name != null) return false;
53.         if (email != null ? !email.equals(customer.email) : customer.email != null) return false;
54.         return company != null ? company.equals(customer.company) : customer.company == null;
55.     }
56.
57.     @Override
58.     public int hashCode() {
59.         int result = name != null ? name.hashCode() : 0;
60.         result = 31 * result + (email != null ? email.hashCode() : 0);
61.         result = 31 * result + (company != null ? company.hashCode() : 0);
62.         return result;
63.     }
64.
65.     @Override
66.     public String toString() {
67.         return "Customer{" +
68.                 "name='" + name + '\'' +
69.                 ", email='" + email + '\'' +
70.                 ", company='" + company + '\'' +
71.                 '}';
72.     }
73. }
```

ตัวอย่างโค้ดเดียวกันในภาษา Kotlin

```
1.  data class Customer(var name: String, var email: String = "",
2.                      var company: String = "")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Kotlin ที่ใช้ได้

Platform

แรกเริ่ม Kotlin ออกแบบมาเพื่อสำหรับ Platform Java และ Compile เพื่อ Run บน JVM เป็นหลัก (Kotlin/JVM) แต่ภายหลัง Kotlin ได้มีการขยายรูปแบบสู่การใช้งาน Android และทำการแปลงเป็น JavaScript ได้ (Kotlin/JS) ทีมงาน Kotlin กำลังพัฒนา Program ภาษา Kotlin สามารถทำการ Compile แบบ Native โดยไม่มีการ Run ผ่าน Virtual

IDE

เครื่องมือ IDE มีการสนับสนุน Kotlin คือ IntelliJ IDEA ของบริษัท JetBrains มีการปลั๊กอินสำหรับการ Eclipse และปลั๊กอินสำหรับ Android Studio (ใน Android Studio 3.0 จะผนวกมาให้ไม่มีการปลั๊กอิน) บริษัทที่นำ Kotlin ไปใช้งาน และบริษัทไอทีหลายๆบริษัท ต่างนำ Kotlin ไปใช้งานในบาง Project เช่น Evernote , Uber , Square เป็นต้น

ทำไม Google ถึงสนับสนุน Kotlin

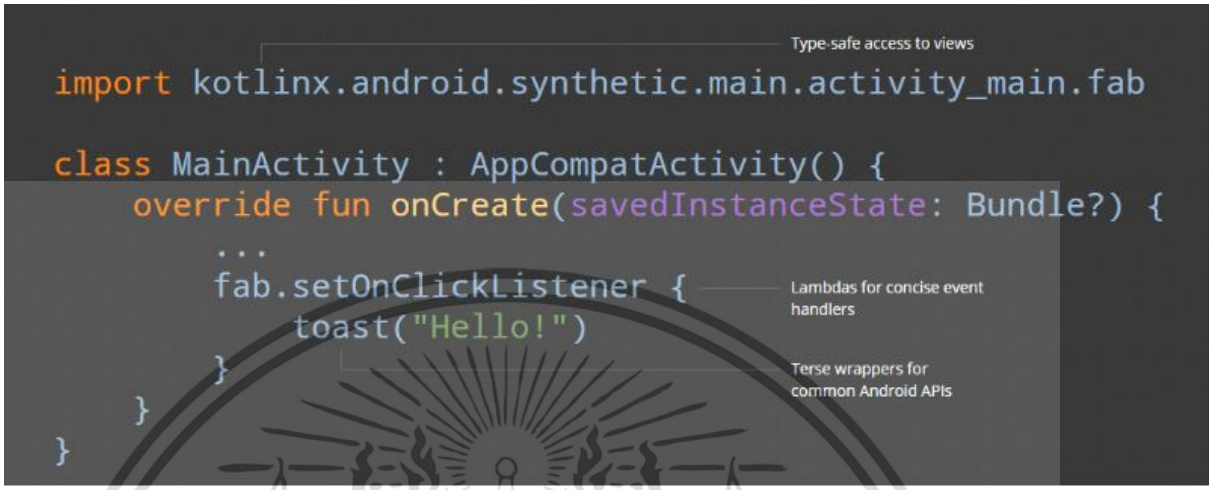
Google ได้อธิบายไว้ว่าเหตุผลสำคัญที่เลือก Kotlin เพราะมีความเข้ากันได้ 100 % กับภาษาอย่าง Java เดิม ช่วยให้ Project หนึ่งสามารถผสมผสาน Code เก่าที่เป็นภาษา Java และ Code ใหม่ที่เป็น Kotlin ได้ ทำให้ช่วยลดงานของผู้พัฒนาลงได้ เหตุผลอื่นๆ คือฟีเจอร์ของภาษา Kotlin มีทั้งเรื่องความปลอดภัยและประสิทธิภาพ รวมไปถึงการยอมรับจากผู้พัฒนาอีกด้วย ประเด็นเรื่องเครื่องมือพัฒนาก็เป็นสิ่งที่ Google ให้ความสำคัญ เนื่องด้วย Android Studio มีรากฐานมาจาก IntelliJ IDEA ที่เป็น Open sort ทำให้มีความมั่นใจได้ว่าการสนับสนุน Kotlin ใน IntelliJ IDEA จาก JetBrains ซึ่งเป็นบริษัทเดียวกัน ย่อมเดินหน้าต่อไปได้เป็นอย่างดี

สถานะของ Kotlin ในโลกของ Android

JetBrains จะเป็นฝ่ายที่เข้าไปพัฒนาฟีเจอร์ด้านต่างๆของ Kotlin ให้กับ Android Studio ร่วมกับที่งานของ Google และทั้งสองได้ทำการร่วมบริษัทตั้งมูลนิธิกลางมาดูแลความเรียบร้อยต่างๆ ของ Kotlin ต่อไปในฐานะองค์กรอิสระ (JetBrains ยืนยันไม่มีการขายบริษัทให้ Google) สถานะของ Kotlin ใน Android จะเป็นภาษาที่ได้รับการรองรับอย่างเป็นทางการเทียบเท่า Java (First-Class Supported Language) ส่วนในด้านของฟีเจอร์ของตัว IDE คือ Android Studio (เช่น Autocomplete , Refactorings) จะรองรับสมบูรณ์ใน Version 3.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Google ได้ออกมาบอกว่าเอกสารและคู่มือการพัฒนายังไม่สมบูรณ์ตามที่ต้องการ และจะมีการทยอยและปรับปรุงเอกสารเหล่านี้ให้ดีขึ้นเรื่อยๆ



รูป 2.3 ตัวอย่างโค้ด Android ในภาษา Kotlin

ข้อดีและข้อเสียที่ต่างกัน ระหว่าง Java และ Kotlin

- Null Safety การ Check ของ null

Java

```
@Override
public int getItemCount() {
    return mItems != null ? mItems.size() : 0;
}
```

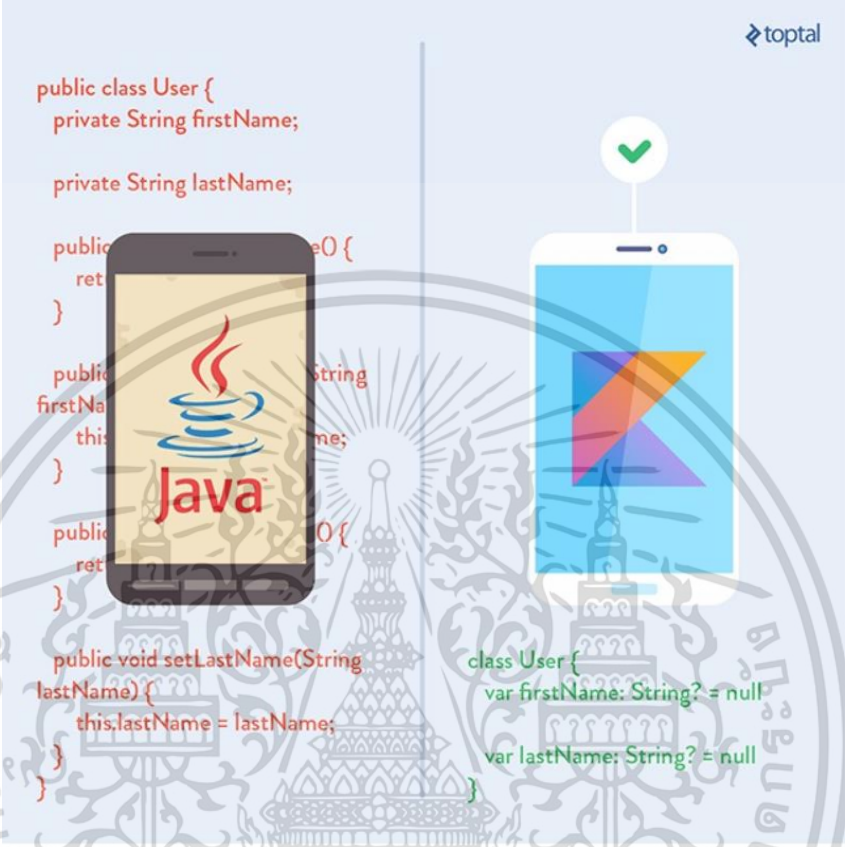
Kotlin

```
override fun getItemCount() = mItems.size
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

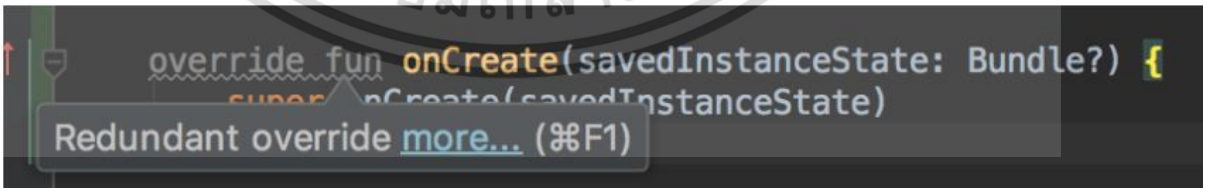
- Less code, Less bugs, Read easier

การเขียน Kotlin ทำให้เราเขียนโค้ดต่างๆสั้นขึ้น เช่น class model จะทำให้โค้ดสั้นขึ้นมาก



รูป 2.4 หลักการเขียนโค้ดของ Java และ Kotlin

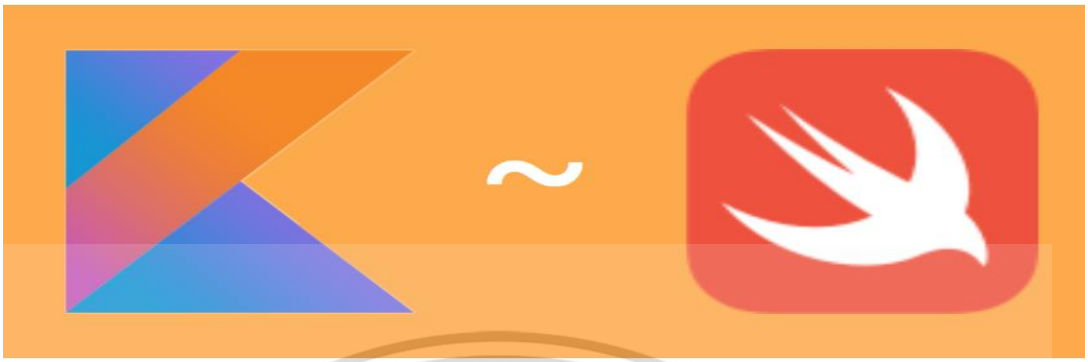
และ Kotlin มันจะเปลี่ยนนิสัยของเรา โค้ดอันไหนที่เราไม่จำเป็นต้องใช้มันจะขึ้นสีเทาๆ แบบนี้ ให้เราลบออก



รูป 2.5 ตัวอย่างโค้ดที่ไม่จำเป็น

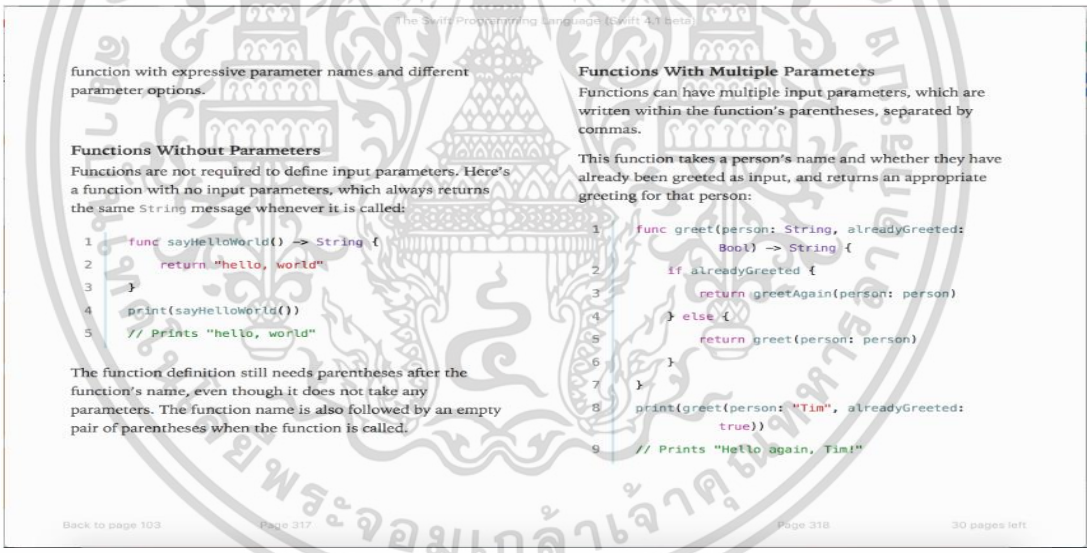
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Similar with Swift



รูป 2.6 Similar with Swift

ด้วยความที่มี syntax อะไรหลายๆอย่างคล้ายกัน ระหว่าง Swift กับ Kotlin ทำให้ iOS Developer และ Android Developer สามารถอ่านโค้ดซึ่งกันและกันได้เข้าใจมากขึ้นนั่นเอง



function with expressive parameter names and different parameter options.

Functions Without Parameters

Functions are not required to define input parameters. Here's a function with no input parameters, which always returns the same `String` message whenever it is called:

```
func sayHelloWorld() -> String {
    return "hello, world"
}
print(sayHelloWorld())
// Prints "hello, world"
```

The function definition still needs parentheses after the function's name, even though it does not take any parameters. The function name is also followed by an empty pair of parentheses when the function is called.

Functions With Multiple Parameters

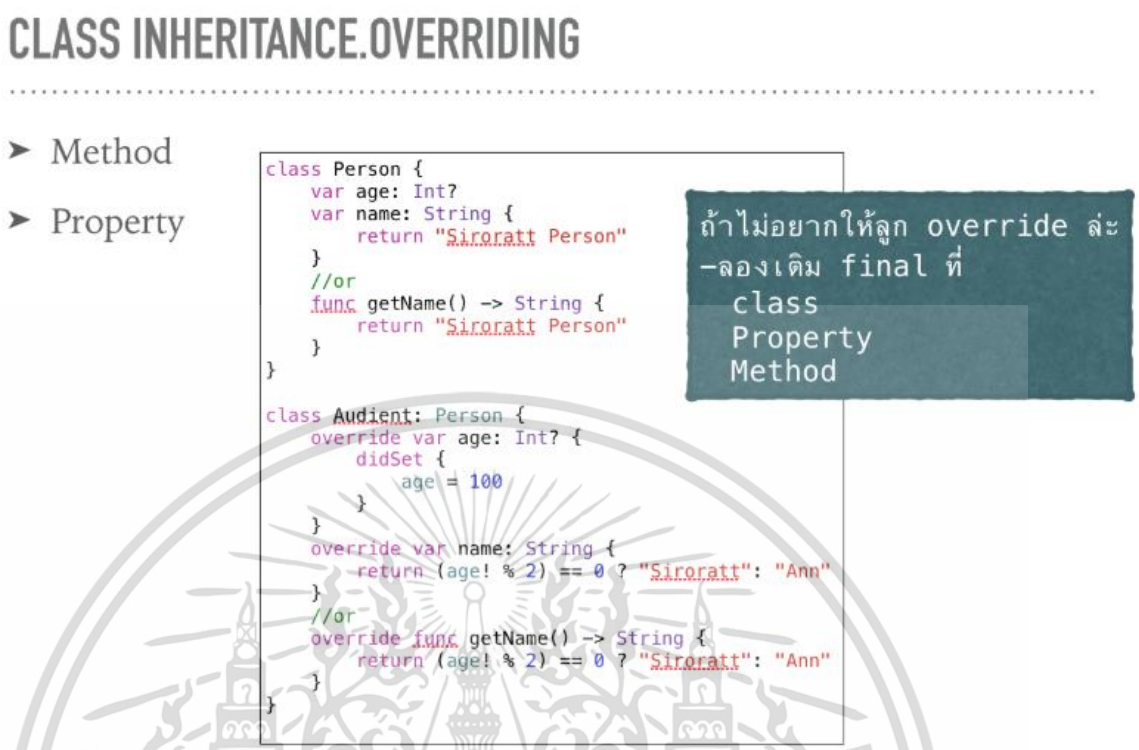
Functions can have multiple input parameters, which are written within the function's parentheses, separated by commas.

This function takes a person's name and whether they have already been greeted as input, and returns an appropriate greeting for that person:

```
func greet(person: String, alreadyGreeted:
        Bool) -> String {
    if alreadyGreeted {
        return greetAgain(person: person)
    } else {
        return greet(person: person)
    }
}
print(greet(person: "Tim", alreadyGreeted:
        true))
// Prints "Hello again, Tim!"
```

รูป 2.7 Functions Parameters

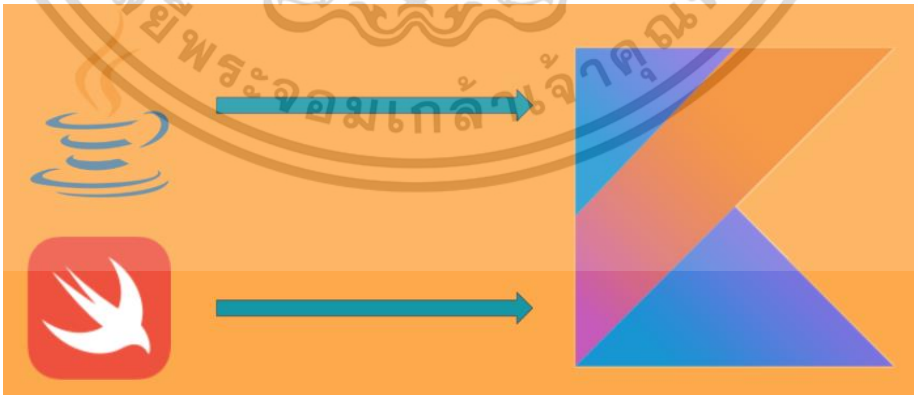
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.8 CLASS INHERITANCE.OVERRIDING

- Low Learning Curve

ถ้าเคยเขียน Android หรือเขียน iOS ด้วย Swift สามารถเรียนรู้ได้ไม่ยาก จากสาเหตุข้อข้างบน และก็ไม่ได้ต่างจากการเขียนแอปด้วย Java

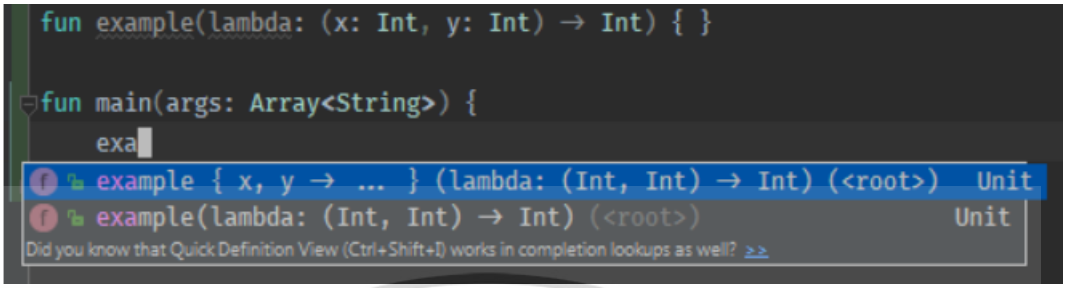


รูป 2.9 Low Learning Curve

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Free Lambda

Kotlinสามารถใช้ lambda ได้ตามใจชอบซึ่งสามารถใช้ Kotlin ได้มากขึ้นกว่าใน Java



รูป 2.10 Free Lambda

- Work together with JAVA

สามารถเพิ่ม Kotlin ใน project เก่าๆได้เลย และสามารถทำงานร่วมกันได้ ไม่ต้องเสียเวลามาเขียนโค้ดขึ้นมาใหม่ใน project เดิม และนอกจากจะทำงานร่วมกันกับ Java ได้แล้ว ยังทำงานร่วมกันกับ js ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Tool Support

สาเหตุที่ทำงานด้วยกันได้ เพราะ tool ที่มีสามารถรองรับ Kotlin ได้ เพราะทาง Android ก็ support ทั้งสองภาษา

รูป 2.11 การรองรับภาษาของ Kotlin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการพัฒนา

3.1 ภาพรวมการทำงานในระบบ

3.1.1 Smartlocker

3.1.1.1 Kotlin

1) สามารถเก็บข้อมูล firebase ของผู้ใช้งานแต่ละผู้ใช้ได้

2) สามารถทำ mobile application ได้

3) สามารถออกแบบระบบของแต่ละผู้ใช้ได้

4) สามารถสำรองข้อมูลของผู้ใช้ไปยัง firebase

5) สามารถลบข้อมูลของผู้ใช้ได้

3.1.1.2 mobile application

1) สามารถจัดการกับล็อคเกอร์ของแต่ละผู้ใช้ได้

2) สามารถสมัครการรับฝากของของแต่ละผู้ใช้ได้

3) สามารถรับ-ฝากได้ตลอดเวลาแต่ผู้ใช้จะต้องอยู่หน้าล็อคเกอร์

4) สามารถ mobile application ได้ เพื่อทำงานเข้าระบบในการสมัครสมาชิกเข้าระบบของตู้ล็อคเกอร์ หรือทำการรับฝากของ

3.1.1.3 Firebase

1) สามารถเก็บข้อมูลของผู้ใช้งานในระบบจากการเปิดและปิดของช่องล็อคเกอร์ของผู้ใช้งานได้

2) สามารถแยกไม่ให้ผู้ใช้อื่นเข้าระบบล็อคเกอร์อื่นได้ถึงจะมีข้อมูลระบบ

3) ผู้ใช้ที่อยู่ในระบบจะเข้าถึงได้เฉพาะข้อมูลของล็อคเกอร์ของตัวผู้ใช้เองแต่ไม่สามารถเข้าล็อคเกอร์อื่นที่ไม่ใช่ของตัวเองได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การเข้าระบบรับ-ฝากของใน Mobile application

ต้องมีการสมัครเข้าระบบของ mobile application เพื่อที่จะเข้าระบบในการทำรายการในการรับ-ฝากของในตู้ล็อคเกอร์ของแต่ละผู้ใช้งาน คือ

3.2.1 ทำการเข้า mobile application ตู้ล็อคเกอร์ของผู้วิจัย เพื่อทำการรับ-ฝากของในล็อคเกอร์

3.2.2 เมื่อเข้า mobile application มาแล้ว เริ่มต้นกดปุ่ม locker 1 เมื่อเข้าสู้การเลือกล็อคเกอร์ที่จะต้องการรับ-ฝากของ

3.2.3 เมื่อกดปุ่ม locker 1 เข้ามาแล้วจะมีการทำการให้เลือก LOCKER ของแต่ละล็อคเกอร์ ว่าผู้ใช้งานต้องการฝากตู้ล็อคเกอร์ช่องไหน และดูสถานะ ว่าล็อคเกอร์ช่องนั่นว่างหรือไม่ แล้วทำการกดปุ่มไปยัง LOCKER นั้นๆ

3.2.4 ถ้าผู้ใช้มี User และ Password ในการเข้าล็อกเกอร์อยู่แล้ว แค่เข้า Login ได้ตามปกติ

3.2.5 แต่ถ้าผู้ใช้ไม่มี User และ Password ในการเข้าล็อกเกอร์ ผู้ใช้ต้องทำการกดสมัครก่อนเพื่อทำการสมัครบัญชีการเข้าใช้งาน ถึงจะสามารถเข้าระบบในการฝากรับ-ของของได้

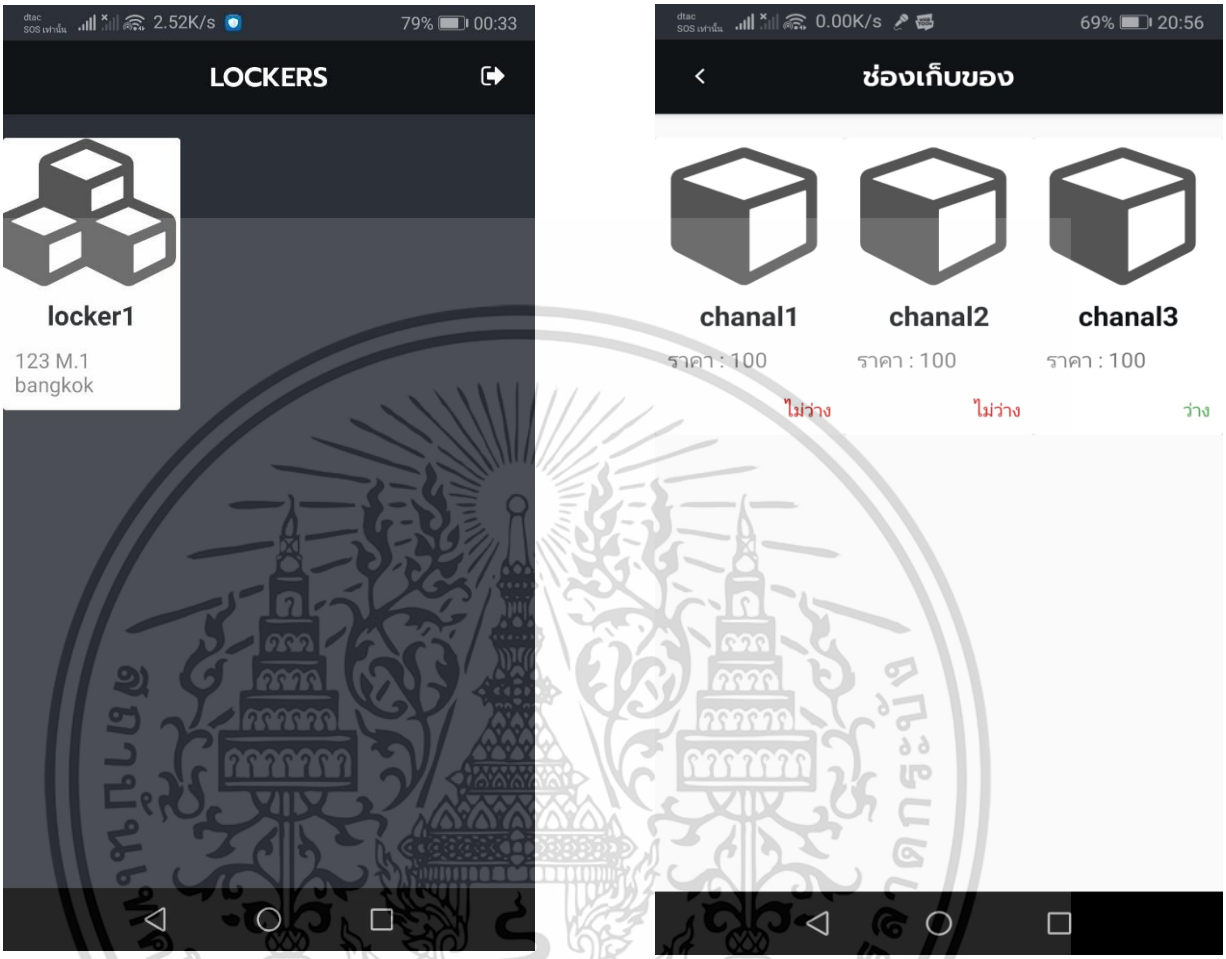
3.2.6 เมื่อเข้ามาแล้วก็ทำการกดเปิดช่อง เพื่อทำการฝากของที่เราต้องการ

3.2.7 เมื่อเสร็จภารกิจแล้วให้ผู้ใช้การทำการปิดล็อคเกอร์

3.2.8 เมื่อต้องการรับของคืนต้องกรอกรหัสผ่านที่ได้มา 4 หลัก เพื่อทำการเปิดล็อคเกอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

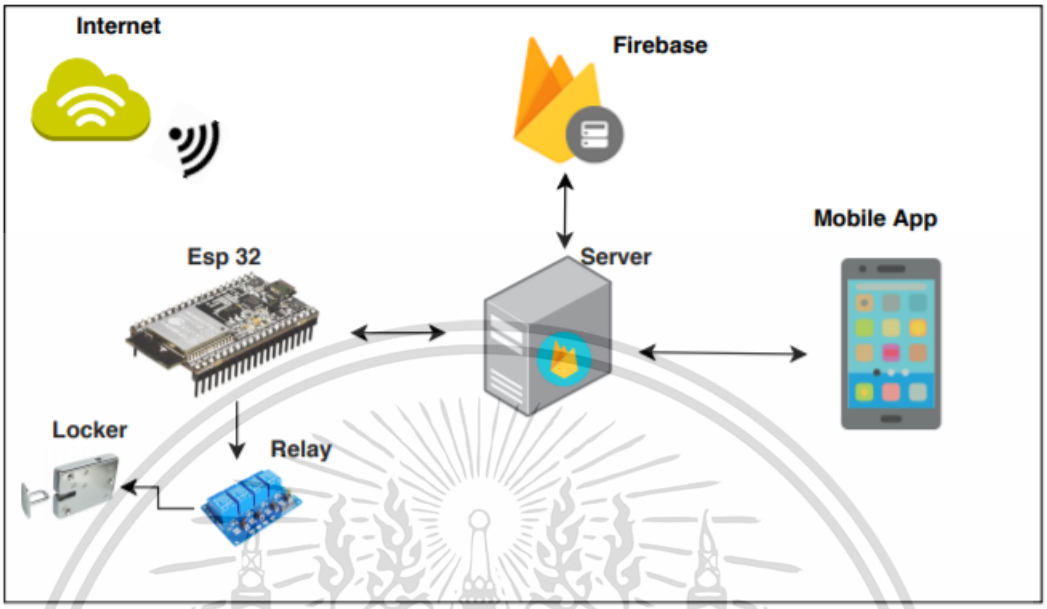
3.3 รูปของ mobile application



รูป 3.1 รูปของ mobile application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 โครงสร้างของระบบ



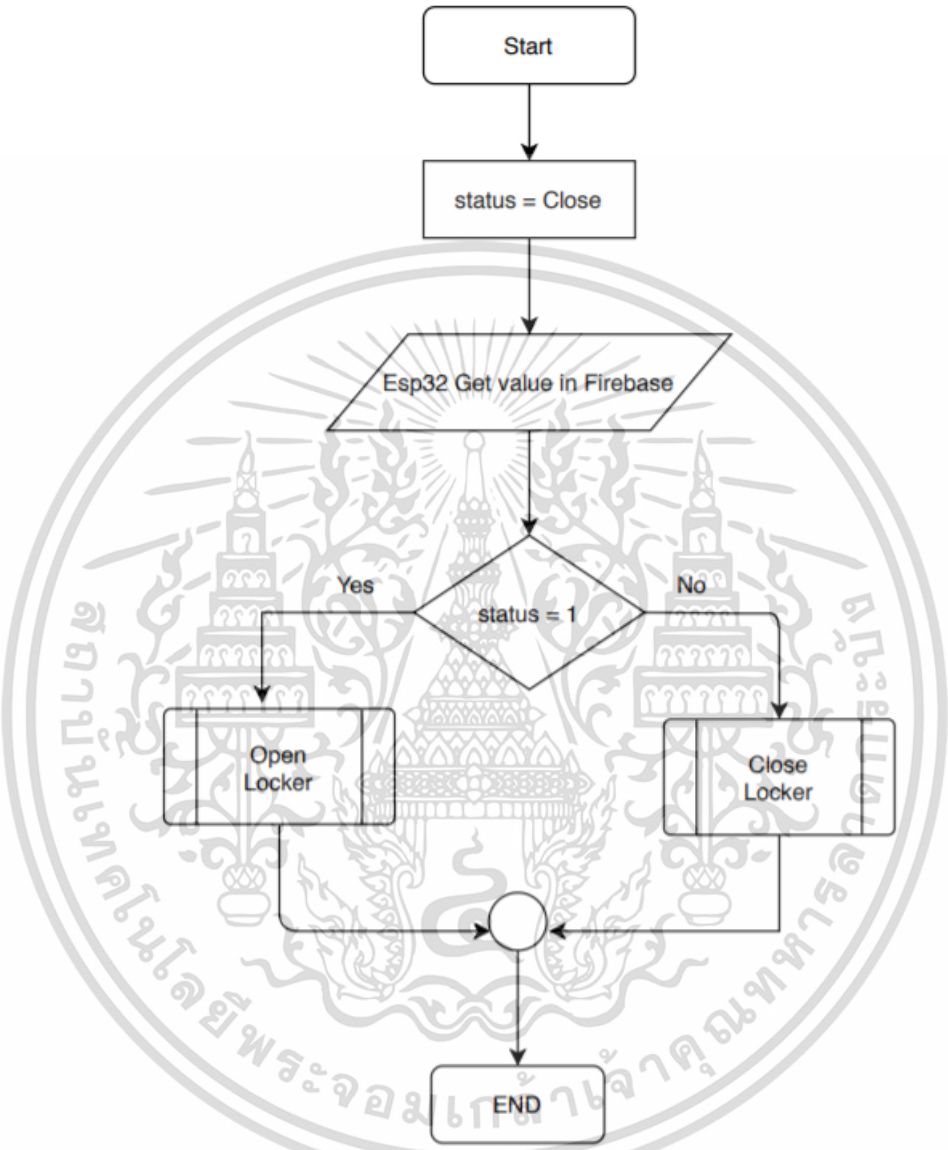
รูป 3.2 ไดอะแกรมของระบบ

จากรูปที่ 3.2 จะแบบออกเป็น 2 ส่วน คือ บอร์ด ESP32 เป็น server และควบคุมการเปิด-ปิดของล็อคเกอร์ และ Mobile application

1) ใช้ Firebase ในการเก็บข้อมูลของผู้ใช้งานในระบบเก็บข้อมูลและไฟล์ Kotlinสำหรับการทำ mobile application เพื่อเชื่อมต่อกับ Internet ในการเข้าระบบของล็อคเกอร์ของแต่ละ User ในแต่ละช่องล็อคเกอร์
2) ESP32 เป็นสื่อกลางในการควบคุมล็อคเกอร์ในการเปิด-ปิดล็อคเกอร์โดยใช้ในการรับคำสั่งเข้ามากับตัว Relay และบอกค่าสถานะของล็อคเกอร์ในแต่ละช่อง และใช้ Relay ในการควบคุมตัวของ Solenoid เพื่อใช้ในการสั่งการเปิด-ปิดของล็อคเกอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

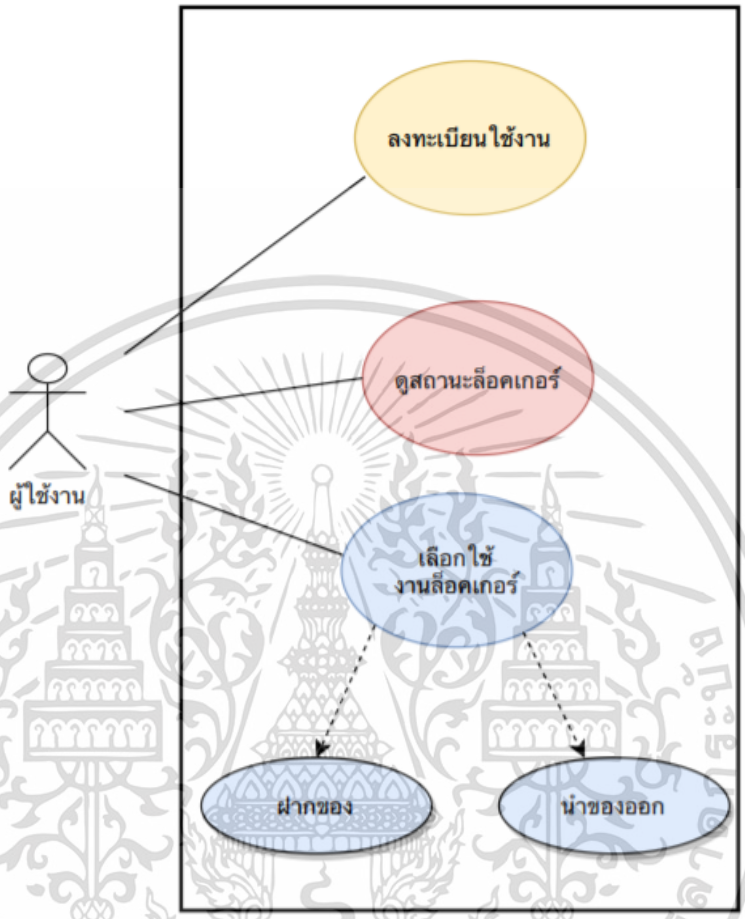
3.5 Flowchart การทำงานของล็อคเกอร์



รูป 3.3 Flowchart การทำงานของล็อคเกอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 Use case Diagram

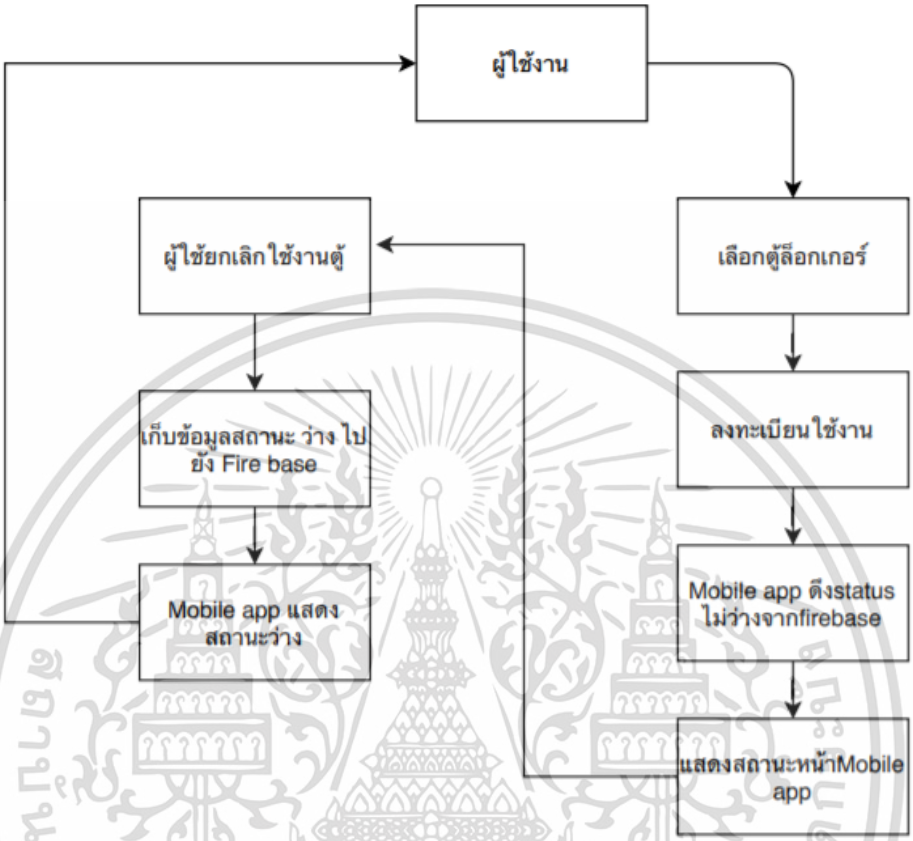


รูป 3.4 Use case Diagram

1) การลงทะเบียน – ผู้ใช้งานสามารถลงทะเบียนได้จาก mobile application ของล็อคเกอร์ฝากของได้ และผู้ดูแลสามารถดูได้ว่าผู้ใช้งานลงทะเบียนไปแล้วจากฐานข้อมูลที่มีในระบบ
2) ดูสถานะตู้ – ผู้ใช้งานสามารถดูสถานะตู้จาก mobile application ในหน้าจอ Mobile Application ที่จะแสดงสถานะของตู้ไว้ ว่าช่องไหนพร้อมใช้งาน และช่องไหนใช้บริการไปแล้ว และผู้ดูแลสามารถตรวจสอบได้ว่าสถานะของตู้ที่ผู้ใช้งานใช้งาน สามารถตรวจสอบได้
3) เลือกตู้ใช้งาน – ผู้ใช้สามารถเลือกใช้งานของตู้ที่มีสถานะว่างได้ตามต้องการ ของผู้ใช้งานไม่ว่าจะการฝากของหรือนำของออกจากล็อคเกอร์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

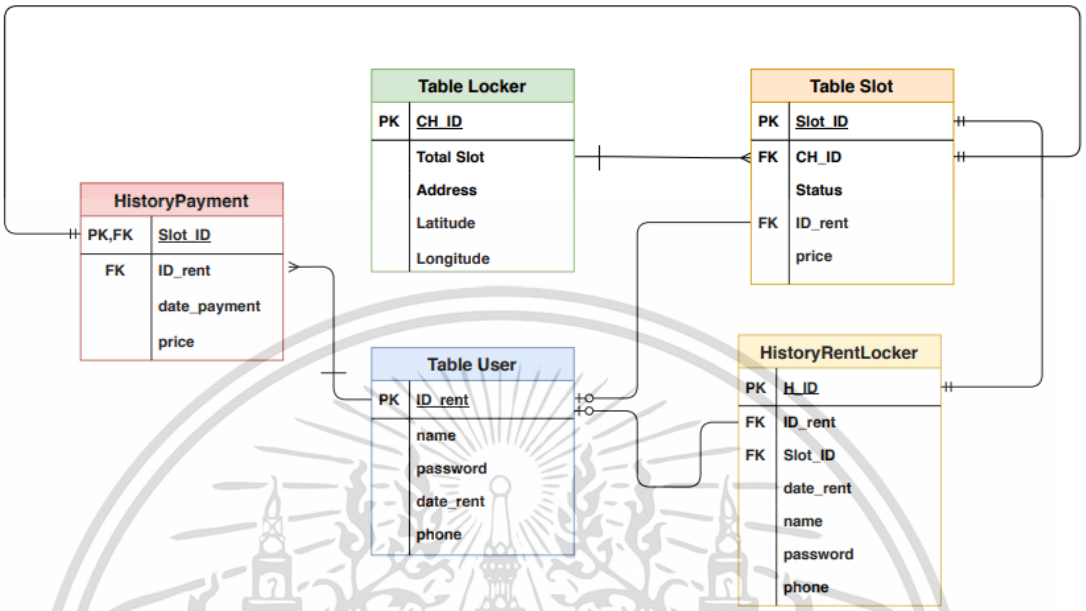
3.7 บล็อกไดอะแกรมแสดงการทำงานของ Mobile Application



รูป 3.5 บล็อกไดอะแกรมแสดงการทำงานของ Mobile Application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 ER Diagram



รูป 3.6 ER Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง

การทดลองในบทนี้จะเป็นการทดลองใช้ Mobile Application ในการเข้าระบบ และการทดลองเปิด-ปิดกลอนไฟฟ้าของ ESP32

1) การทดสอบเข้าหน้า Mobile Application บนสมาร์ทโฟน
2) การทดสอบฝากของเข้าระบบ
3) การทดสอบรับของออกจากระบบ
4) การทดสอบการใส่รหัสผ่านผิดในการเปิดล็อคเกอร์
5) ทดสอบการใช้งานกลอนไฟฟ้ากับ Mobile Application

4.1 การทดสอบการเข้า Mobile application บนสมาร์ทโฟน

4.1.1 วัตถุประสงค์

เพื่อทดสอบคุณภาพของ Mobile application บนสมาร์ทโฟนใช้งานได้

4.1.2 วิธีการทดลอง

การเข้า Mobile application โดยใช้สมาร์ทโฟน ดังรูปที่ 4.1

รูป 4.1 หน้าการเข้า Mobile Application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

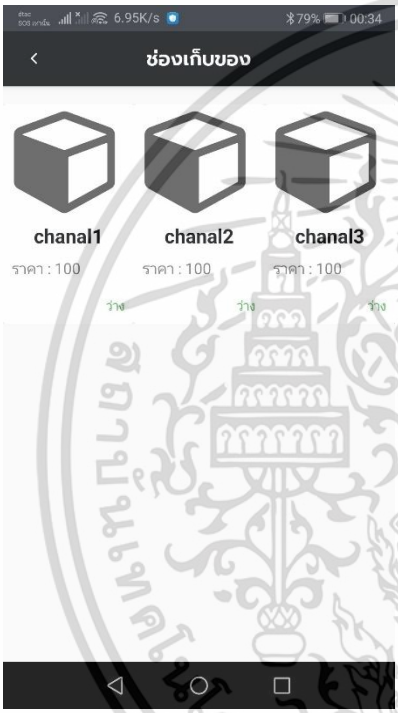
4.2การทดสอบฝากของเข้าระบบ

4.2.1 วัตถุประสงค์

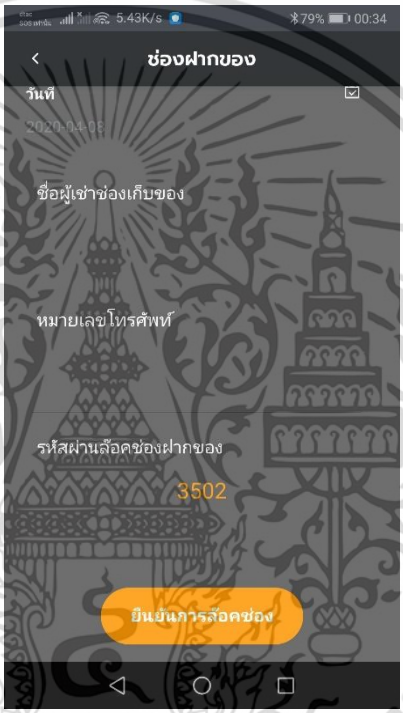
เพื่อความปลอดภัยในการฝากของโดยไม่ต้องกังวลว่าของจะหาย

4.2.2วิธีการทดลอง

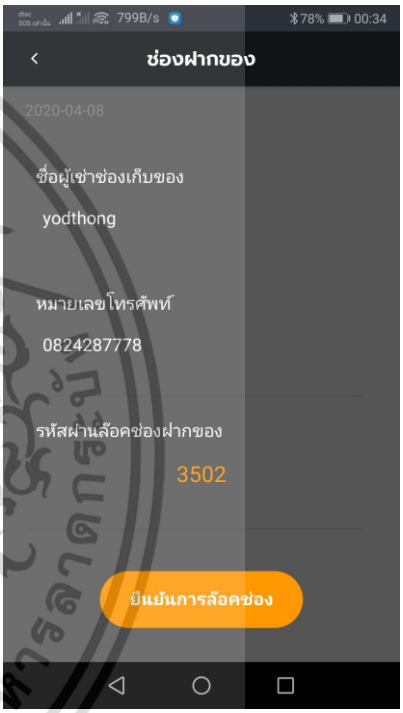
เข้าไปหน้าแอพพลิเคชันแล้วกดเลือกช่องล็อคเกอร์ที่ผู้ใช้ต้องการฝากของและทำการใส่ User เบอร์โทรศัพท์ และจำรหัส 4 ตัวไว้ สามารถทำการฝากของได้



รูป 4.2 (a)



รูป 4.2 (b)



รูป 4.2 (c)

รูป 4.2 (a) ช่องบอกสถานะของล็อคเกอร์

- ทำการเลือกช่องล็อคเกอร์ที่จะทำการฝากของ และดูสถานะของตู้แต่ละช่อง

(b) หน้าการฝากของในล็อคเกอร์

- จะเป็นช่องให้ผู้ใช้งานทำการกรอกชื่อและเบอร์โทรศัพท์เพื่อทำการฝากตู้ และมีรหัสตู้สุ่มมาให้ 4 หลักในการรับของคืน

(c) การกรอกข้อมูล

- กรอกข้อมูลเพื่อทำการฝากของในล็อคเกอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 4.3 (a) **รูป 4.3 (b)**

รูป 4.3 (a) การเปิดตู้ในการฝากของสำเร็จ

- จะแสดงข้อความบอกว่าผู้ใช้งานทำการสมัครเข้าฝากของสำเร็จ และทำการฝากของได้

(b) ช่องบอกสถานะของล็อคเกอร์ที่ใช้งานไปแล้ว

- จะบอกสถานะว่าล็อคเกอร์ช่องไหนฝากของไปแล้วจะแสดงสถานะว่า ไม่ว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

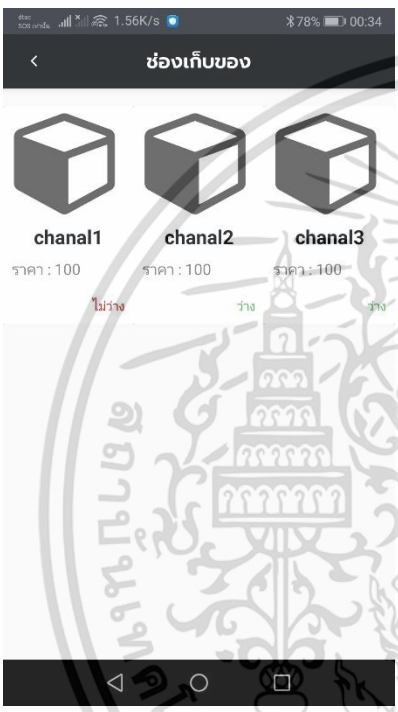
4.3การทดสอบรับของออกจากระบบ

4.3.1 วัตถุประสงค์

เพื่อให้ผู้ใช้งานทดสอบการรับของคืนจากระบบ Mobile Application

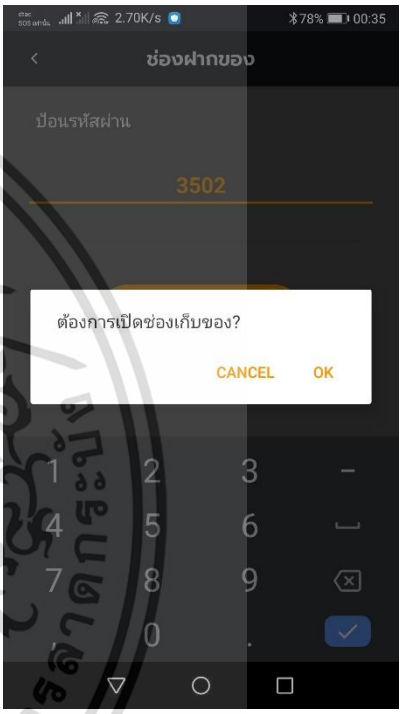
4.2.2วิธีการทดลอง

ทำการเลือกช่องที่เราทำการฝากของไป แล้วใส่รหัสผ่าน 4 หลัก เพื่อทำการเข้าระบบของล็อคเกอร์ในการรับของคืน



รูป 4.4 (a)

รูป 4.4 (b)



รูป4.4 (c)

รูป 4.4 (a) ช่องบอกสถานะของล็อคเกอร์ที่ใช้งานไปแล้ว

- จะบอกสถานะว่าล็อคเกอร์ช่องไหนฝากของไปแล้วจะแสดงสถานะว่า ไม่ว่าง

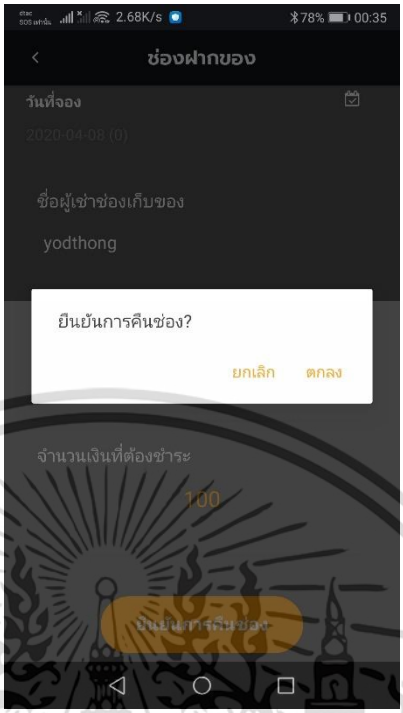
(b) เลือกช่องล็อคเกอร์ที่จะทำการรับของคืน

- กดช่องล็อคเกอร์ที่ผู้ใช้งานทำการฝากของไว้ก่อนหน้านี้

(c) ใส่รหัสเปิดตู้ล็อคเกอร์

- ใส่รหัสผ่าน 4 หลัก ที่สุ่มได้ตอนผู้ใช้งานทำการฝากของและทำการใส่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.5 (a) **รูป 4.5 (b)** **รูป 4.5 (c)**

รูป 4.5 (a) การยืนยันตัวตนรับของคืน

- เป็นการยืนยันตัวตนอีกครั้งในการรับของคืนจากตู้ล็อคเกอร์

(b) ยืนยันคืนช่องล็อคเกอร์

- กดเพื่อยืนยันว่าผู้ใช้งานต้องการเปิดล็อคเกอร์เพื่อทำการรับของคืน

(c) เปิดช่องสำเร็จ

- จะแสดงข้อความบอกว่าผู้ใช้งานทำการเปิดตู้ล็อคเกอร์ได้สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4การทดสอบการใส่รหัสผิดในการเปิดล็อคเกอร์

4.4.1 วัตถุประสงค์

เพื่อให้ผู้ใช้งานรับฝากของมีความปลอดภัยในการรับของ

4.4.2วิธีการทดลอง

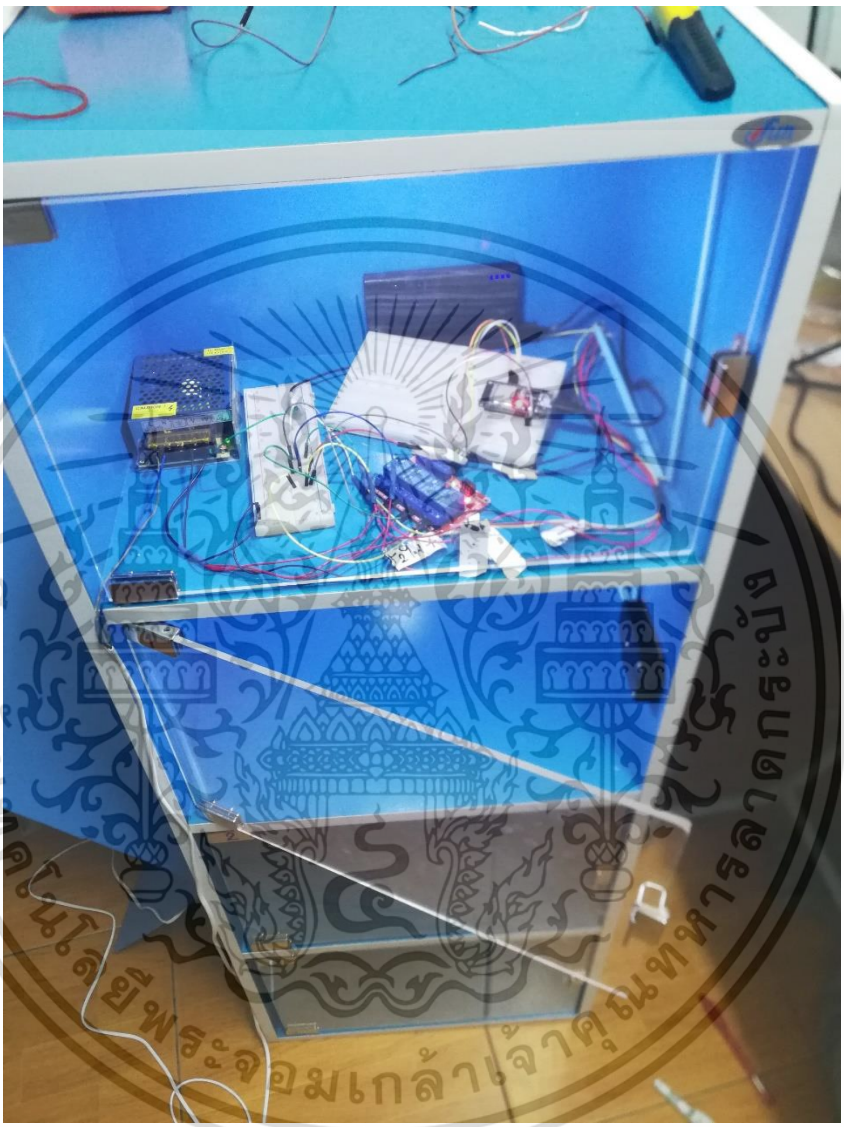
ทำการใส่รหัสผ่าน 4 ตัวที่ไม่ถูกต้อง ที่ Mobile Application ทำการสุ่มให้เพื่อเปิดล็อคเกอร์

รูป 4.6 ทดสอบการใส่รหัสผ่านไม่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5ทดสอบการใช้กลอนไฟฟ้ากับ Mobile Application

4.5.1 ทดสอบการเปิดล็อคเกอร์ เมื่อต้องการฝากของหรือทำการรับของ ผ่าน Mobile Application



รูป 4.7 ทำการเปิดล็อคเกอร์เพื่อการฝากของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 4.8 ทำการฝากของเข้าไปและทำการปิดล็อคเกอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 4.9 ทำการเปิดล็อคเกอร์เพื่อทำการรับของคืน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

ภายในโครงงานมีการทำงานร่วมกัน 2 ส่วนหลัก คือ ส่วน Mobile Application และส่วนของ ESP32

5.1.1 ส่วนของ Mobile Application

1) สามารถเชื่อมต่อกับ Firebase ของแต่ละผู้ใช้งานได้

2) สามารถใช้งาน Mobile Application ควบคุมการเปิด-ปิดของกลอนไฟฟ้าได้

3) สามารถจัดการกับตู้ล็อคเกอร์ของแต่ละผู้ใช้งานได้

4) สามารถดูฐานข้อมูลของแต่ละผู้ใช้งานได้

5.1.2 ส่วนของ ESP32

1) สามารถเชื่อมต่อกับ Mobile Application ในการเข้าระบบของผู้ใช้งานได้

2) สามารถเชื่อมต่อเข้ากับ server ได้เพื่อที่สั่งงานการเปิด-ปิดของ Relay ได้

3) สามารถเชื่อต่อ Relay เข้ากับกลอนไฟฟ้าเพื่อสั่งงานการเปิด-ปิดได้

4) สามารถควบคุมการเปิด-ปิดของแต่ละช่องของตู้ล็อคเกอร์ได้

5.2 ปัญหาและอุปสรรค

5.2.1 ส่วนของ Mobile Application

1) มีการเข้าถึง Mobile Application ได้ช้าเกินไป หน้าเว็บมีการดาวน์โหลดส่วนประกอบค่อนข้างเยอะ ทำให้แสดงผลช้า หากอินเทอร์เน็ตมีความเร็วต่ำ

2) เกิดการ delay ระหว่างการใช้งานในการเปิด-ปิดของตู้ล็อคเกอร์ ทำให้มีการคลาดเคลื่อนในการใช้งาน

5.2.2 ส่วนของ ESP32

1) กลอนไฟฟ้ามีความร้อนจากการใช้งานเป็นเวลานาน

2) มีความคลาดเคลื่อนของระบบอินเตอร์เน็ตที่ส่งไปยัง Mobile Application ทำให้การสั่งงานบางครั้งกดเปิด-ปิดได้ บางครั้งกดไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางแก้ปัญหา

5.3.1 ส่วนของ Mobile Application

1) ปรับปรุงหน้าแอพพลิเคชันให้มีความเสถียรมากขึ้นกับการใช้งาน

5.3.2 ส่วนของ ESP32

1) ใช้แหล่งจ่ายไฟที่พอเหมาะกับการใช้งานของกลอนไฟฟ้า เพื่อไม่ให้เกิดอาการร้อน

2) ปรับปรุง server ให้มีประสิทธิภาพในการใช้งานการเปิด-ปิดของแต่ละตู้

5.4 แนวทางการพัฒนาต่อ

5.3.1 ส่วนของ Mobile Application

1) ปรับปรุงรูปแบบแอพพลิเคชัน เพื่อแสดงผลบนสมาร์ทโฟนให้สวยงามขึ้น

2) ปรับปรุง Firebase ให้มีประสิทธิภาพมากขึ้น

3) เพิ่มความสามารถในการเข้าระบบให้เร็วขึ้น เพื่อเพิ่มประสิทธิภาพในการใช้งานของผู้ใช้งาน

5.3.2 ส่วนของ ESP32

1) ปรับปรุงความปลอดภัยของล็อคเกอร์แต่ละช่องให้มีประสิทธิภาพ และปลอดภัยมากขึ้น

2) เพิ่มฟีเจอร์ในการฝากของ เพื่อเพิ่มควาสามารถในการใช้งานให้มากขึ้น

3) ปรับปรุงการใช้งานในระบบให้มีความพร้อมต่อการใช้งานในเรื่องของพลังงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- เรียนรู้ ESP32, [ออนไลน์], เข้าถึงครั้งสุดท้าย 5/02/2563 https://medium.com/@redzero/%E0%B8%95%E0%B8%B4%E0%B8%94%E0%B8%95%E0%B8%B1%E0%B9%89%E0%B8%87-%E0%B8%A1%E0%B8%B2%E0%B8%A5%E0%B8%AD%E0%B8%87%E0%B9%80%E0%B8%A5%E0%B9%88%E0%B8%99%E0%B9%80%E0%B8%88%E0%B9%89%E0%B8%B2-esp32-%E0%B8%81%E0%B8%B1%E0%B8%99%E0%B9%80%E0%B8%96%E0%B8%AD%E0%B8%B0-75cbe79815c8
- รู้จัก Firebase Realtime Database ตั้งแต่ Zero จนเป็น Hero, [ออนไลน์], เข้าถึงครั้ง สุดท้าย 31/10/2562 https://medium.com/@jirawatee/รู้จัก-firebase realtime-database-ตั้งแต่-zero-จนเป็น-hero-5d09210e6fd6
- หนังสือ Arvind Ravulavaru. (2017). Learning Ionic Second Edition. Birmingham,UK: Packt Publishing Ltd
- เรียนรู้ Kotlin เข้าถึงครั้งสุดท้าย 15/20/2563 https://developer.android.com/kotlin/index.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้