แอปพลิเคชันทดสอบการยืนยันตัวบุคคลบนระบบปฏิบัติการแอนดรอยด์

SINGLE SIGN ON APPLICATION FOR ANDROID

วรพรต เดชรัตน์

สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SINGLE SIGN ON APPLICATION FOR ANDROID

Woraprot Dechrut

A COOPERATIVE EDUCATION SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR

THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)

DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2017

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	แอปพลิเคชันทดสอบการยืนยันตัวบุคคลบนระบบปฏิบัติการแอนดรอยด์ Single Sign On Application for Android
ชื่อนักศึกษา	นายวรพรต เดชรัตน์ รหัสนักศึกษา รหัสนักศึกษา 57050319
ปริญญา	วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2560
อาจารย์ที่ปรึกษา	อ.สันธนะ อู่อุดมยิ่ง

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) ประจำปีการศึกษา 2560

คณะกรรมการสอบ	ลายมือชื่อ
ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์ ประธานกรรมการ	
อ.สันธนะ อู่อุดมยิ่ง กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา แอปพลิเคชันทดสอบการยืนยันตัวบุคคลบนระบบปฏิบัติการแอนดรอยด์
ชื่อนักศึกษา นายวรพรต เดชรัตน์ รหัสนักศึกษา รหัสนักศึกษา 57050319
ปริญญา วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา วิทยาการคอมพิวเตอร์
คณะ วิทยาศาสตร์
มหาวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา 2560
อาจารย์ที่ปรึกษา อ.สันธนะ อู่อุดมยิ่ง

บทคัดย่อ

เนื่องจากทีมพัฒนาแอปพลิเคชัน Streaming ของบริษัทตลาดหลักทรพย์แห่งประเทศไทย (SET) มีการพัฒนาแอปพลิเคชันทดสอบการยืนยันตัวบุคคลให้เทสเตอร์ทดสอบ เพื่อใช้สร้างชุดคำสั่งสำหรับทดสอบ แทนแอปพลิเคชันแยกของแต่ละโบรกเกอร์ที่อยู่ระหว่างการพัฒนา ให้สามารถส่งชุดคำสั่งไปให้แอปพลิเคชัน Streaming ทำการประมวลผลการเข้าใช้งานแบบ Single Sign On (SSO) ได้ แต่แอปพลิเคชันทดสอบการยืนยันตัวบุคคลนั้นยังขาดฟังก์ชันที่จำเป็น และส่วนติดต่อผู้ใช้งานยากต่อการเข้าใจ จึงจำเป็นต้องพัฒนาแอปพลิเคชันทดสอบการยืนยันตัวบุคคลให้มีฟังก์ชันใหม่ๆ และส่วนติดต่อผู้ใช้งานรูปแบบใหม่ เพื่อตอบสนองความต้องการให้กับเทสเตอร์ของทีมพัฒนา ในการพัฒนาผู้จัดทำได้วิเคราะห์ระบบเดิมของแอปพลิเคชันทดสอบการยืนยันตัวบุคคล แล้วสร้างระบบใหม่ขึ้นมาด้วยการนำเทคโนโลยีของ Firebase มาใช้งาน พร้อมทั้งออกแบบส่วนติดต่อผู้ใช้งานตามหลัก Material Design เพื่อสร้างฟังก์ชันและการใช้งานที่มีประสิทธิภาพ ซึ่งแอปพลิเคชันทดสอบการยืนยันตัวบุคคลถูกพัฒนาด้วยภาษา Java และ XML ในโปรแกรม Android Studio

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Single Sign On Application for Android
Students	Mr. Woraprot Dechrut Student ID 57050319
Degree	Bachelor of Science (Computer Science)
Department	Computer Science
Faculty	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2017
Advisor	Mr. Santana Oudomying

Abstract

Since the application development team of the Market for Stock Exchange of Thailand (SET) has developed a test app for testers. To use to create a test case. Instead of the separate application of each broker under development. A set of instructions can be sent to the Streaming application to perform Single Sign On (SSO) access, but the Single Sign On Application for Android is missing the required functionality. And the user interface is difficult to understand. It's important to develop Single Sign On Application for Android Single Sign On Application for Android for new functionality. And the new user interface. To meet the needs of the Tester. In development, analysis of the legacy application testing system Then create a new system with the use of Firebase technology to use and design a user interface based on Material Design to create functional and effective use. Single Sign On Application for Android developed in Java and XML in Android Studio.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

สหกิจศึกษาเล่มนี้สำเร็จลุล่วงได้ด้วยดี เนื่องมาจากความกรุณาและความร่วมมือของทุกท่าน ขอขอบพระคุณ ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์ และอ.สันธนะ อู่อุดมยิ่ง อาจารย์ที่ปรึกษาสหกิจศึกษา ที่คอยให้คำปรึกษา ดูแลอย่างใกล้ชิดและช่วยเหลือแนะนำแนวทางแก้ปัญหา รวมทั้งตรวจสหกิจศึกษานี้ให้มีความสมบูรณ์ถูกต้อง ผู้จัดทำสหกิจจึงขอขอบพระคุณอย่างสูงไว้ ณ ที่นี้

ขอขอบพระคุณคณะอาจารย์ภาควิชาวิทยาการคอมพิวเตอร์ทุกท่านที่ให้ความรู้และตั้งใจสอนตลอดเวลาการศึกษา 4 ปี รวมถึงหน่วยงานของบริษัทตลาดหลักทรัพย์แห่งประเทศไทย ที่เข้าร่วมโครงการสหกิจศึกษาครั้งนี้ รวมถึงผู้ที่เกี่ยวข้องได้แก่ คุณธนวัฒน์ มาสิลีรังสี ที่คอยดูแลและให้คำปรึกษาในการทำงาน รวมถึงพี่ๆ ทุกคนในโครงการที่ให้คำแนะนำ และแนวทางการเรียนรู้การทำงานตลอดระยะเวลาที่เข้าร่วมโครงการสหกิจศึกษา

ท้ายที่สุดนี้ขอขอบคุณบิดา มารดา ที่ให้การสนับสนุนการศึกษา ตลอดจนคอยเลี้ยงดู อบรมสั่งสอนและเป็นกำลังใจ ซึ่งเป็นแรกผลักดันในการทำโครงงานสหกิจศึกษาให้สำเร็จลุล่วงไปได้ด้วยดี รวมถึงเพื่อนๆ และบุคคลอื่น ๆ ที่ไม่ได้กล่าวถึง ผู้จัดทำสหกิจศึกษาจึงขอขอบคุณทุกท่านเป็นอย่างสูงมา ณ โอกาสนี้

วรพรต เดชรัตน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก
บทคัดย่อภาษาอังกฤษ ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ฉ
สารบัญรูป ..... ช
บทที่ 1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของสหกิจศึกษา 1
1.2 วัตถุประสงค์ของสหกิจศึกษา 1
1.3 ขอบเขตของสหกิจศึกษา 2
1.4 ประโยชน์ที่คาดว่าจะได้รับ 2
1.5 อุปกรณ์ที่ใช้ในการพัฒนา 3
บทที่ 2 ทฤษฎีและเทคโนโลยีที่ใช้ในการพัฒนาระบบ 4
2.1 การเข้าใช้งานระบบแบบ Single Sign On (SSO) 4
2.2 Progressive Web App (PWA) 5
2.2.1 การทำงานของ Progressive Web App (PWA) 5
2.3 ระบบปฏิบัติการแอนดรอยด์ 7
2.3.1 โครงสร้างของระบบปฏิบัติการแอนดรอยด์ 7
2.4 แอนดรอยด์สตูดิโอ (Android Studio) 8
2.5 อะโดบี โฟโตชอป (Adobe Photoshop) 9
2.6 อะโดบี อิลลัสเตรเตอร์ (Adobe Illustrator) 10
2.7 ไฟร์เบส (Firebase) 10
2.7.1 ตัวอย่างบริการของ Firebase ที่ใช้ในการพัฒนา 10
2.8 จาวา (Java) 12
2.9 กิท (Git) 13
2.10 มาเทอเรียลดีไซน์ (Material Design) 13
2.10.1 แนวทางการออกแบบหรือหลักการออกแบบ 13
2.11 สัดส่วนทองคำ (Golden Ratio) 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 วิธีการดำเนินงานสหกิจศึกษา 15

3.1 กระบวนการทำงานของระบบ 15

3.2 วิเคราะห์ระบบ 16

3.2.1 วิเคราะห์ระบบเดิม 16

3.2.2 วิเคราะห์ระบบใหม่ 17

3.3 การออกแบบระบบ 18

3.3.1 แผนภาพยูสเคส (Use Case Diagram) 18

3.3.2 ขั้นตอนการทำงานของระบบ (Sequence Diagram) 21

3.3.2.1 ฟังก์ชันการสร้างและบันทึกชุดคำสั่ง (Profile) 21

3.3.2.2 ฟังก์ชันบันทึกประวัติชุดคำสั่ง (History) 22

3.3.2.3 ฟังก์ชันจัดการตัวเลือกในฟอร์มบนดาต้าเบส (Config Database) 23

3.4 ส่วนติดต่อผู้ใช้งาน (User Interface) 24

3.5 การออกแบบโลโก้ 27

3.6 ตัวอย่างโค้ดโปรแกรม 28

3.6.1 ตัวอย่างโค้ดเพื่อตรวจสอบการเข้าใช้งาน 28

3.6.2 ตัวอย่างโค้ดเพื่ออ่านข้อมูล 29

3.6.3 ตัวอย่างโค้ดเพื่อเขียนข้อมูล 32

บทที่ 4 ผลการพัฒนาแอปพลิเคชัน 34

4.1 ฟังก์ชัน Profile 34

4.2 ฟังก์ชัน Save History 35

4.3 ฟังก์ชัน Config Database 35

บทที่ 5 สรุปผลและข้อเสนอแนะ 37

5.1 สรุปผลการดำเนินงานสหกิจศึกษา 37

5.2 ข้อเสนอแนะในการพัฒนา 37

เอกสารอ้างอิง 38

ภาคผนวก 40

ภาคผนวก ก 41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
1.1 ตาราง Use case อธิบายการทำงาน Login	19
1.2 ตาราง Use case อธิบายการทำงาน Create test case	19
1.3 ตาราง Use case อธิบายการทำงาน Show history	19
1.4 ตาราง Use case อธิบายการทำงาน Config database	20
1.5 ตาราง Use case อธิบายการทำงาน Send test case	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 การทำงานของ Single Sign On	5
2.2 การทำงานของ Service Worker	6
2.3 ตัวอย่างการสร้างไอคอนของเว็บไซต์บนหน้า Home ของสมาร์ทโฟน	6
2.4 โครงสร้างของระบบปฏิบัติการแอนดรอยด์	7
2.5 ตัวอย่างหน้า GUI ของโปรแกรม Android Studio	9
2.6 ตัวอย่างหน้า GUI ของโปรแกรม Adobe Photoshop	9
2.7 ตัวอย่างหน้า GUI ของโปรแกรม Adobe Illustrator	10
2.8 ตัวอย่างหน้าจัดการ Security Rules ใน Firebase Console	11
2.9 ตัวอย่างหน้าแสดงข้อมูลที่ถูกเก็บใน Database ในรูปแบบ NoSQL	11
2.10 ตัวอย่างหน้าจัดการ Account ต่างๆ ใน Firebase Console	12
2.11 ตัวอย่างโลโก้ที่ใช้สัดส่วนทองคำ	14
3.1 สถาปัตยกรรมของแอปพลิเคชันเดิม	16
3.2 สถาปัตยกรรมของแอปพลิเคชันใหม่	17
3.3 แผนภูมิยูสเคสของแอปพลิเคชันทดสอบการยืนยันตัวบุคคล	18
3.4 Sequence Diagram ฟังก์ชันการสร้างและบันทึกชุดคำสั่ง (Profile)	21
3.5 Sequence Diagram ฟังก์ชันการบันทึกประวัติชุดคำสั่ง (History)	22
3.6 Sequence Diagram ฟังก์ชันจัดการตัวเลือกในฟอร์มบนดาต้าเบส (Config Database)	23
3.7 หน้าจอเริ่มต้นเมื่อเข้าแอปพลิเคชัน	24
3.8 หน้าจอแสดงรายการประวัติชุดคำสั่ง	24
3.9 หน้าจอแสดงข้อมูลของรายการประวัติชุดคำสั่ง	25
3.10 หน้าจอแสดงฟังก์ชัน Profile	25
3.11 หน้าจอแสดงรายการชุดคำสั่งที่เคยสร้างไว้จากฟังก์ชัน Profile	26
3.12 หน้าจอแสดงข้อมูลของรายการของชุดคำสั่งที่เคยสร้างไว้	26
3.13 หน้าจอแสดงรายการที่สามารถจัดการได้	27
3.14 โลโก้แอปพลิเคชันทดสอบการทดสอบการยืนยันตัวบุคคล	27
3.15 ตัวอย่างโค้ดการบันทึกข้อมูลลง Firebase Realtime Database	28
3.16 ตัวอย่างโค้ดการอ่านข้อมูลจาก Firebase Realtime Database	30
3.17 ตัวอย่างโค้ดการเขียนข้อมูลลง Firebase Realtime Database	32
3.18 หน้าแสดงข้อมูลของ Firebase Realtime Database บน Firebase Console	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.1 หน้าต่างการทำงานของฟังก์ชัน Profile	34
4.2 หน้าต่างการทำงานของฟังก์ชัน Save	35
4.3 หน้าต่างการทำงานของฟังก์ชัน Config	36
ก.1 หน้าเว็บโหลดโปรแกรม Android Studio	41
ก.2 หน้าแสดงรายละเอียดก่อนติดตั้ง	41
ก.3 หน้าแสดงตัวเลือกส่วนประกอบที่จะติดตั้ง	42
ก.4 หน้าแสดงตำแหน่งการติดตั้งไฟล์โปรแกรม	42
ก.5 หน้าแสดงการติดตั้งโปรแกรม	43
ก.6 หน้าแสดงขั้นตอนสุดท้ายของการติดตั้ง	43
ก.7 หน้าเว็บของ Firebase	44
ก.8 หน้าเว็บแสดงฟอร์มการเข้าสู่ระบบ	44
ก.9 หน้าเว็บแสดงโปรเจคทั้งหมดของผู้ใช้งานc	45
ก.10 หน้าเว็บแสดงฟอร์มกรอกข้อมูลของโปรเจค	45
ก.11 หน้าเว็บแสดงหน้าหลัก Firebase Console	46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของสหกิจศึกษา

เนื่องจากปัจจุบันมีนักลงทุนเพิ่มขึ้นตลอดเวลา นักลงทุนเหล่านี้ส่วนมากเลือกที่จะลงทุนกับการซื้อขายหุ้นเพื่อให้ได้รับผลตอบแทนที่มากขึ้นและสะดวก โดยอดีตนั้น วิธีการลงทุนนั้นจำเป็นจะต้องไปติดต่อกับทางบริษัทเพื่อขอซื้อหุ้น หรือติดต่อกับนักลงทุนคนอื่นเพื่อที่จะขายหุ้น แต่ในปัจจุบันเทคโนโลยีเกี่ยวกับสมาร์ทโฟนนั้นมีการเจริญเติบโตขึ้น จึงได้มีการนำเทคโนโลยีมาพัฒนาเป็นแอปพลิเคชันให้นักลงทุนที่ซื้อขายหุ้น เพื่อสร้างความสะดวกสบายให้แก่นักลงทุนเหล่านี้ และแอปพลิเคชันที่ใช้ซื้อขายหุ้น เช่น Streaming เป็นต้น โดยวิธีการซื้อขายหุ้นของนักลงทุนจำเป็นจะต้องสมัครใช้งานกับโบรกเกอร์หรือตัวแทนผู้ทำหน้าที่ซื้อขายหุ้นให้แก่นักลงทุน เมื่อโบรกเกอร์มีจำนวนมากขึ้น โบรกเกอร์เหล่านี้จึงเกิดความประสงค์ที่จะให้แอปพลิเคชันของตนไม่ซ้ำกับโบรกเกอร์อื่น จึงได้มีการพัฒนาแอปพลิเคชันแยกของแต่ละโบรกเกอร์ออกมาจากแอปพลิเคชัน Streaming เพื่อให้เกิดความแตกต่าง เช่น ส่วนติดต่อผู้ใช้งานจะไม่เหมือนกัน เป็นต้น โดยความสามารถของแอปพลิเคชันแยกของแต่ละโบรกเกอร์นั้นทำได้แค่ดูข้อมูลของหุ้นเท่านั้น เมื่อนักลงทุนต้องการจะซื้อขายหุ้นตัวใดตัวหนึ่งนั้น จำเป็นจะต้องเข้ามาซื้อขายในแอปพลิเคชัน Streaming ดังนั้นนักลงทุนของบางโบรกเกอร์จำเป็นจะต้องใช้ 2 แอปพลิเคชัน และการใช้งานของทั้ง 2 แอปพลิเคชันจำเป็นต้องลงชื่อเข้าใช้งานแต่ละแอปพลิเคชันเสียก่อนด้วย Username และ Password เดียวกัน จึงเกิดปัญหาในการเข้าใช้งานเพราะจะต้องพิมพ์ Username และ Password ถึง 2 ครั้ง และในภายหลังได้มีการเพิ่มการเข้าใช้งานแบบ Single Sign-On (SSO) หรือการยืนยันตัวบุคคลที่รองรับการลงชื่อเข้าใช้งานแอปพลิเคชันเพียงครั้งเดียว แล้วสามารถเข้าใช้งานแอปพลิเคชันหลายแอปพลิเคชันได้ โดยไม่จำเป็นจะต้องพิมพ์ Username และ Password ซ้ำ ดังนั้นในการพัฒนาแอปพลิเคชันแยกของโบรกเกอร์นั้นเทสเตอร์ของกลุ่มนักพัฒนาจำเป็นต้องทดสอบการเข้าใช้งานแบบ Single Sign-On (SSO) กับแอปพลิเคชัน Streaming ตลอดเวลา และระหว่างการพัฒนาแอปพลิเคชันแยกของโบรกเกอร์อยู่นั้นในบางกรณีเทสเตอร์ของกลุ่มนักพัฒนาไม่สามารถที่จะทดสอบ Single Sign-On (SSO) กับแอปพลิเคชัน Streaming ได้ จึงทำให้การทดสอบแอปพลิเคชันนั้นเป็นไปได้ยาก แอปพลิเคชันทดสอบการยืนยันตัวบุคคลจะแก้ปัญหาที่กล่าวมา โดยกลไกของแอปพลเคชัน คือ จำลองชุดคำสั่งแทนแอปพลิเคชันของโบรกเกอร์ที่อยู่ระหว่างการพัฒนา ให้สามารถส่งไปให้แอปพลิเคชัน Streaming ทำการตรวจสอบการเข้าใช้งานแบบ Single Sign-On (SSO) ต่อไปได้

จากปัญหาดังกล่าวผู้จัดทำได้เล็งเห็นความสำคัญจึงพัฒนาแอปพลิเคชันทดสอบการยืนยันตัวบุคคลต่อจากระบบเดิม เพื่อช่วยลดระยะเวลาการในการทดสอบแต่ละครั้งได้อย่างรวดเร็วและมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกเหนือจากผู้จัดทำได้พัฒนาแอปพลิเคชันทดสอบการยืนยันตัวบุคคลแล้วผู้จัดทำได้รับมอบหมายให้พัฒนาระบบดาวน์โหลดไฟล์ติดตั้งของแอปพลิเคชันที่พัฒนาขึ้นภายในทีมจากระบบเดิมซึ่งเป็นเว็บไซต์บนเบราว์เซอร์ให้เป็น Progressive Web App (PWA) เพื่อช่วยลดระยะเวลาในการเข้าถึงข้อมูลไฟล์ติดตั้งอย่างรวดเร็ว พัฒนาฟังก์ชัน TFEX Summary ปรับปรุงหน้า Quote ให้มีประสิทธิภาพ แก้การแสดงผลหน้าต่างๆ ให้ตรงตาม Requirement ให้กับแอปพลิเคชัน Streaming และพัฒนาฟังก์ชัน Switching ให้กับแอปพลิเคชัน Streaming Fund

1.2 วัตถุประสงค์ของสหกิจศึกษา

1) เพื่อทดสอบการเข้าใช้งานแบบ Single Sign-On(SSO) ของแอปพลิเคชัน Streaming
2) เพื่อช่วยลดระยะเวลาในการทดสอบแอปพลิเคชันวิเคราะห์หุ้นหรืออนุพันธ์
3) เพื่อให้ผู้ใช้งานสามารถดูประวัติการทดสอบแต่ละครั้งได้
4) เพื่อให้ผู้ใช้งานสามารถดาวน์โหลดไฟล์ติดตั้งของแอปพลิชันที่ต้องการได้อย่างรวดเร็ว

1.3 ขอบเขตของสหกิจศึกษา

1) เป็นแอปพลิเคชันสร้างชุดคำสั่งเพื่อส่งไปให้แอปพลิเคชัน Streaming ตรวจสอบการเข้าใช้งานแบบ Single Sign-On(SSO)
2) ผู้ทดสอบสามารถกรอกแบบฟอร์มทดสอบแต่ละครั้งได้
3) ผู้ทดสอบสามารถบันทึกแบบฟอร์มทดสอบไว้ใช้ในครั้งต่อไปได้
4) ผู้ทดสอบสามารถจัดการแบบฟอร์มที่บันทึกไว้ได้
5) ผู้ทดสอบสามารถดูประวัติการทดสอบแต่ละครั้งได้
6) ผู้ใช้งานสามารถดาวน์โหลดไฟล์ติดตั้งแอปพลิเคชันที่ต้องการได้อย่างรวดเร็ว

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) ด้านผู้ใช้งานแอปพลิเคชัน (User)
    - ได้แอปพลิเคชันที่สามารถส่งชุดคำสั่งเพื่อทดสอบการเข้าใช้งานแบบ Single Sign-On (SSO) บนแอปพลิเคชัน Streaming
    - ลดระยะเวลาในการทดสอบฟังก์ชัน Single Sign-On (SSO)
    - สามารถดาวน์โหลดไฟล์ติดตั้งแอปพลิเคชันที่ต้องการได้อย่างรวดเร็ว
2) ด้านผู้พัฒนาแอปพลิเคชัน (Developer)
    - ได้เรียนรู้การพัฒนาแอปพลิเคชันลงบนระบบปฏิบัติการแอนดรอยด์
    - ได้เรียนรู้วิธีการออกแบบส่วนติดต่อผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ได้เรียนรู้การพัฒนาเว็บแอปพลิเคชันโดยใช้เทคโนโลยี Progressive Web App (PWA)

1.5 อุปกรณ์ที่ใช้ในการพัฒนา

1) ฮาร์ดแวร์

- คอมพิวเตอร์ Notebook 1 เครื่อง
- สมาร์ทโฟนแอนดรอดย์ 1 เครื่อง

2) ซอฟต์แวร์

- Android Studio
- Google Chrome
- Adobe Photoshop CS6
- Adobe Illustrator CS6
- Git

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและเทคโนโลยีที่ใช้ในการพัฒนาระบบ

2.1 การเข้าใช้งานระบบแบบ Single Sign On (SSO)

Single Sign ON (SSO) นั้นเป็นการระบุตัวตนเพื่อใช้งานทรัพยากรที่อยู่ภายในระบบโดยทำการพิสูจน์ตัวตนเพียงครั้งเดียว ยกตัวอย่างเช่น บริษัทแห่งหนึ่งมีการใช้งาน Active Directory ในการระบุตัวตนในการใช้งานเครื่องคอมพิวเตอร์ในองค์กรอยู่แล้ว ซึ่งหากไม่มีการนำ SSO มาใช้สิ่งที่เกิดขึ้นคือ User ต้องทำการล็อกอินหลายครั้ง ครั้งแรกอาจเป็นการล็อกอินสำหรับใช้งาน Computer ครั้งต่อไปเป็นการล็อกอินเพื่อใช้งาน Internet ซึ่งทำให้เกิดความซ้ำซ้อน ในการล็อกอิน เพื่อใช้สำหรับตรวจสอบการใช้งาน Internet ของ User นั้นๆ แต่หากนำเทคโนโลยี SSO มาใช้สิ่งที่เกิดขึ้นก็คือ User ทำการล็อกอินเข้าใช้งานเครื่องคอมพิวเตอร์เพียงครั้งเดียวเท่านั้น

ประโยชน์ของ Single Sign-On

- ผู้ใช้บริการเกิดความสะดวกในการเชื่อมต่อบริการต่างๆ
- ผู้ใช้ไม่ต้องจำข้อมูลเข้าระบบและรหัสผ่านหลายๆ ตัว เพราะใช้การเข้าสู่ระบบเพียงครั้งเดียว
- ผู้ให้บริการไม่ต้องเก็บข้อมูลซ้ำซ้อนกันในระบบบริการแต่ละตัว
- ช่วยเพิ่มจำนวนผู้ใช้ให้กับบริการต่างๆ เพราะหลังจากเข้าสู่ระบบแล้ว สามารถเลือกใช้บริการต่างๆ ได้ทันที
- สามารถกำหนดหรือควบคุมสิทธิ์ของผู้ใช้จากทุกบริการได้ง่ายขึ้น
- ลดจำนวนการแจ้งลืมรหัสผ่านของระบบ เพราะใช้จากจุดเดียว

ตัวอย่างบริการที่ใช้ Single Sign-On

- บริการอิเล็กทรอนิกส์ภาครัฐ (e-Government Portal)
- บริการของกูเกิล (Google)
- บริการของไมโครซอฟต์ (Microsoft)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.1 การทำงานของ Single Sign On

2.2 Progressive Web App (PWA)

เป็นเทคโนโลยีสำหรับพัฒนาเว็บไซต์ให้คล้ายการทำงานของ Application มากที่สุดทั้งด้านประสบการณ์ผู้ใช้ หรือการแสดงผลต่างๆ ปัญหาที่ทำให้เกิดเทคโนโลยี Progressive Web App (PWA) คือ จากการวิจัยของ comScore Mobile Metrix พบว่าในปัจจุบัน ค่าเฉลี่ยของการดาวน์โหลดแอปพลิเคชันใหม่ลงบนสมาร์ทโฟนต่อเดือนอยู่ที่ 0 และในงานวิจัยกล่าวว่า 80% ของเวลาที่ผู้ใช้งานใช้สมาร์ทโฟน ใช้งานเพียง 3 แอปพลิเคชันหลักๆ เท่านั้น ในขณะที่ข้อมูลการเข้าใช้งานเว็บไซต์ต่อเดือนมีค่ามากกว่าหลายเท่า Google จึงพัฒนาเทคโนโลยี Progressive Web App (PWA) ขึ้นมา โดยมีแนวคิด ดังนี้

- Reliable ต้องทำงานได้แม้อยู่ในโหมดออฟไลน์
- Fast ต้องมีความเร็วในการใช้งาน
- Engaging แบ่งออกเป็น 3 ข้อ ดังนี้

  1) Homescreen เข้าถึงสะดวก สามารถเปิดหน้าเว็บได้จาก Icon App

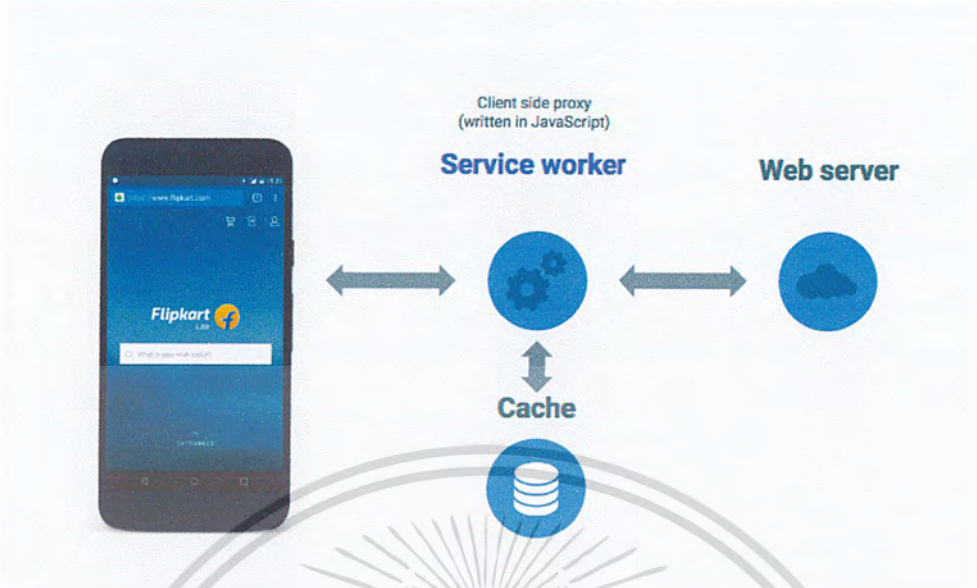
  2) Immersive นักพัฒนาสามารถควบคุมประสบการณ์การใช้งานของผู้ใช้งานได้

  3) Notification สามารถสร้างการแจ้งเตือนแบบ Notification ให้กับผู้ใช้งานได้

2.2.1 การทำงานของ Progressive Web App (PWA)

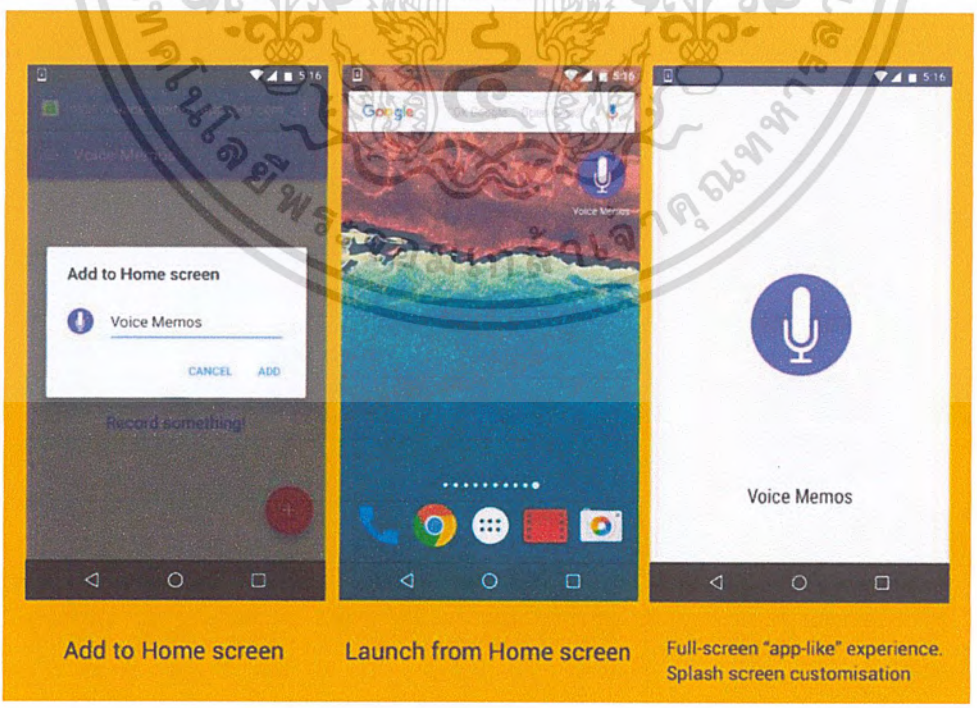
1) Service Worker เป็นตัวกำหนด Cache สิ่งต่างๆที่จำเป็นในเว็บไซต์ของนักพัฒนา เช่น ปุ่มต่างๆ เป็นต้น เพื่อให้การเข้าใช้งานครั้งต่อไปไม่จำเป็นต้องโหลด Element เหล่านี้ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 การทำงานของ Service Worker

2) Manifest.json เป็นไฟล์ JSON ที่สามารถกำหนดความสามารถให้เว็บไซต์ได้ เช่น สร้างไอคอนหน้า Home บนสมาร์ทโฟน กำหนดขนาดการแสดงผลของเว็บไซต์แบบเต็มจอ กำหนดการแสดงผลแบบแนวตั้งหรือแนวนอน



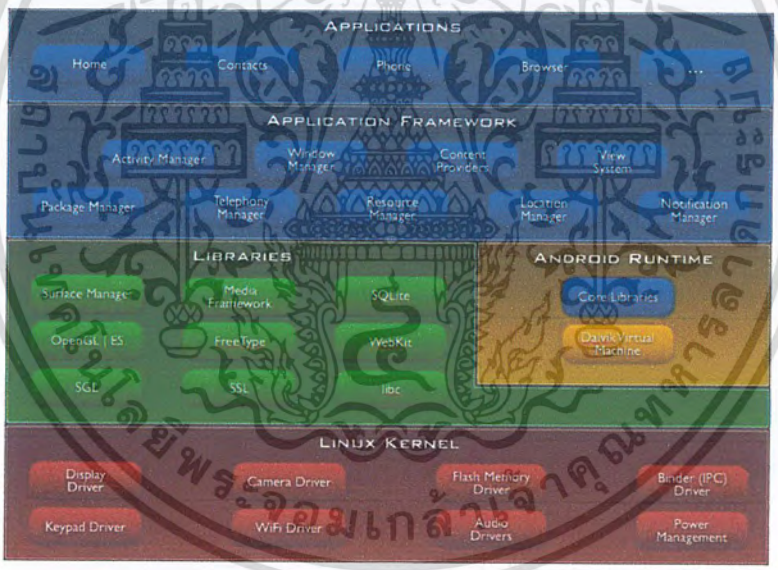
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.3 ตัวอย่างการสร้างไอคอนของเว็บไซต์บนหน้า Home ของสมาร์ทโฟน

2.3 ระบบปฏิบัติการแอนดรอยด์

ระบบปฏิบัติการแอนดรอยด์ คือ ระบบปฏิบัติการสำหรับอุปกรณ์พกพา เช่น โทรศัพท์มือถือ แท็บเล็ตคอมพิวเตอร์ เน็ตบุ๊ก ทำงานบนลินุกซ์ เคอร์เนล เริ่มพัฒนาโดยบริษัทแอนดรอยด์ (อังกฤษ: Android Inc.) จากนั้นบริษัทแอนดรอยด์ถูกซื้อโดยกูเกิล และนำแอนดรอยด์ไปพัฒนาต่อ ภายหลังถูกพัฒนาในนามของ Open Handset Alliance ทางกูเกิลได้เปิดให้นักพัฒนาสามารถแก้ไขโค้ดต่างๆ ด้วยภาษาจาวา และควบคุมอุปกรณ์ผ่านทางชุด Java libraries ที่กูเกิลพัฒนาขึ้น มี Android SDK (Software Development Kit) เป็นเครื่องมือสำหรับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ อีกทีหนึ่ง โดยระบบปฏิบัติการ แอนดรอยด์ เริ่มพัฒนาเมื่อปี พ.ศ. 2550 โดยบริษัทแอนดรอยด์ร่วมกับ Google จากนั้นนเมื่อปี พ.ศ.2550 ได้มีการร่วมมือกันกว่า 30 บริษัทชั้นนำเพื่อพัฒนาระบบ

2.3.1 โครงสร้างของระบบปฏิบัติการแอนดรอยด์



รูปที่ 2.4 โครงสร้างของระบบปฏิบัติการแอนดรอยด์

จากโครงสร้างของระบบปฏิบัติการแอนดรอยด์ สามารถแบ่งส่วนๆได้ ดังนี้

1. **Applications** ส่วน Application หรือส่วนของโปรแกรมที่มีมากับระบบปฏิบัติการ หรือเป็นกลุ่มของโปรแกรมที่ผู้ใช้งานได้ทำการติดตั้งไว้ โดยผู้ใช้งานสามารถเรียกใช้โปรแกรมต่างๆได้โดยตรง ซึ่งการทำงานของแต่ละโปรแกรมจะเป็นไปตามที่ผู้พัฒนาโปรแกรมได้ออกแบบและเขียนโปรแกรมเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

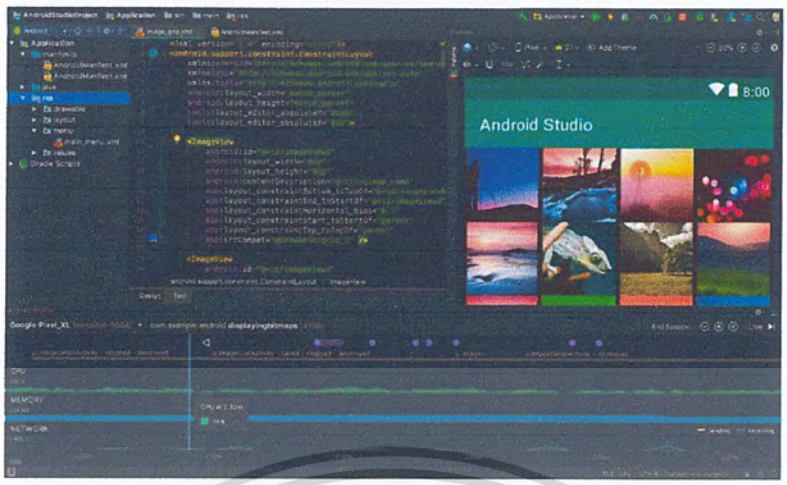
2. **Application Framework** เป็นส่วนที่มีการพัฒนาขึ้นเพื่อให้นักพัฒนาสามารถพัฒนาโปรแกรมได้สะดวก และมีประสิทธิภาพมากยิ่งขึ้น โดยนักพัฒนาไม่จำเป็นต้องพัฒนาในส่วนที่มีความยุ่งยากมากๆ เพียงแค่ทำการศึกษาถึงวิธีการเรียกใช้งาน Application Framework ในส่วนที่ต้องการใช้งาน แล้วนำมาใช้งาน ซึ่งมีหลายกลุ่มด้วยกัน
3. **Libraries** เป็นส่วนของชุดคำสั่งที่พัฒนาด้วย C/C++ โดยแบ่งชุดคำสั่งออกเป็นกลุ่มตามวัตถุประสงค์ของการใช้งาน เช่น Surface Manage จัดการเกี่ยวกับการแสดงผล, Media Framework จัดการเกี่ยวกับการการแสดงภาพและเสียง, Open GL | ES และ SGL จัดการเกี่ยวกับภาพ 3 มิติ และ 2 มิติ, SQLlite จัดการเกี่ยวกับระบบฐานข้อมูล เป็นต้น
4. **Android Runtime** จะมี Darvik Virtual Machine ที่ถูกออกแบบมา เพื่อให้ทำงานบนอุปกรณ์ที่มี หน่วยความจำ(Memmory), หน่วยประมวลผลกลาง (CPU) และพลังงาน(Battery)ที่จำกัด ซึ่งการทำงานของ Darvik Virtual Machine จะทำการแปลงไฟล์ที่ต้องการทำงาน ไปเป็นไฟล์ .DEX ก่อนการทำงาน เหตุผลก็เพื่อให้มีประสิทธิภาพเพิ่มขึ้นเมื่อใช้งานกับ หน่วยประมวลผลกลางที่มีความเร็วไม่มาก ส่วนต่อมาคือ Core Libraries ที่เป็นส่วนรวบรวมคำสั่งและชุดคำสั่งสำคัญ โดยถูกเขียนด้วยภาษาจาวา (Java Language)
5. **Linux Kernel** เป็นส่วนที่ทำหน้าที่หัวใจสำคัญ ในจัดการกับบริการหลักของระบบปฏิบัติการ เช่น เรื่องหน่วยความจำ พลังงาน ติดต่อกับอุปกรณ์ต่างๆ ความปลอดภัย เครือข่าย เป็นต้น

2.4 แอนดรอยด์สตูดิโอ (Android Studio)

Android Studio เป็น IDE Tool จาก Google ไว้พัฒนา Android โดยพัฒนาจากแนวคิดพื้นฐานมาจาก IntelIJ IDEA คล้าย ๆ กับการทำงานของ Eclipse และ Android ADT Plugin โดยวัตถุประสงค์ของ Android Studio คือต้องการพัฒนาเครื่องมือ IDE ที่สามารถพัฒนา App บน Android ให้มีประสิทธิภาพมากขึ้น ทั้งด้านการออกแบบ GUI ที่ช่วยให้สามารถ Preview ตัว App มุมมองที่แตกต่างกันบน Smart Phone แต่ล่ะรุ่น สามารถแสดงผลบางอย่างได้ทันทีโดนไม่ต้องทำการรัน App บน Emulator รวมทั้งยังแก้ไขปรับปรุงในเรื่องของความเร็วของ Emulator ที่ยังเจอปัญหากันอยู่ในปัจจุบัน

การเขียน Android บน Android Studio จะมีขั้นตอนอยู่ 2 ขั้นตอนก็คือ ติดตั้ง Java SDK และดาวน์โหลด Android Studio มาติดตั้งก็จะสามารถใช้งานได้ทันที โดยที่เราไม่ต้องทำการติดตั้ง Android ADT Plugin แต่อย่างใด ซึ่งช่วยลดขั้นตอนการติดตั้งเครื่องมือต่าง ๆ ได้

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

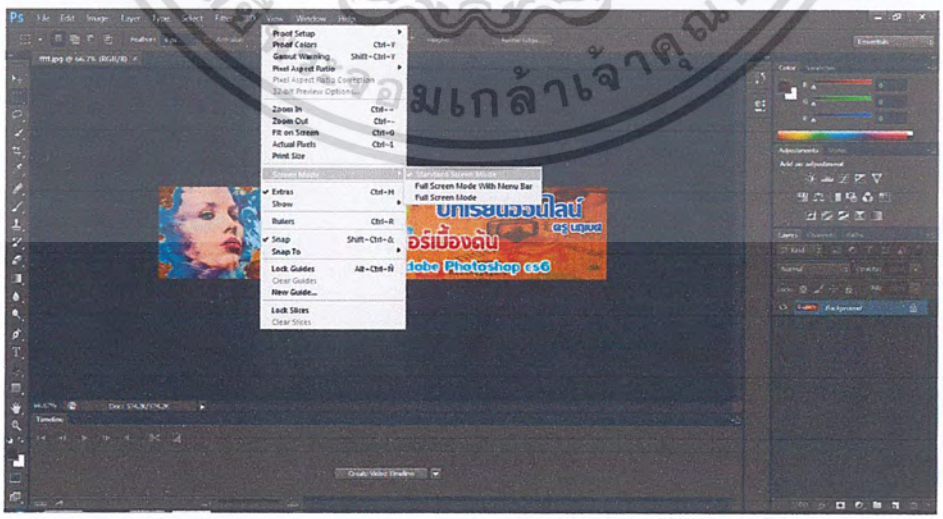


รูปที่ 2.5 ตัวอย่างหน้า GUI ของโปรแกรม Android Studio

2.5 อะโดบี โฟโตชอป (Adobe Photoshop)

เป็นโปรแกรมประยุกต์ที่มีความสามารถในการจัดการแก้ไขและตกแต่งรูปภาพ (photo editing and retouching) แบบแรสเตอร์ ผลิตโดยบริษัทอะโดบีซิสเต็มส์ ซึ่งผลิตโปรแกรมด้านการพิมพ์อีกหลายตัวที่ได้รับความนิยม เช่น Illustrator และ InDesign ปัจจุบันโปรแกรมโฟโตชอปได้พัฒนามาถึงรุ่น CC (Creative Cloud)

โดยโปรแกรมอะโดบี โฟโตชอป ผู้พัฒนานำมาใช้ออกแบบส่วนติดต่อผู้ใช้งานของแอปพลิเคชันทดสอบการยืนยันตัวบุคคล เพื่อเป็นต้นแบบในการเขียนโปรแกรม



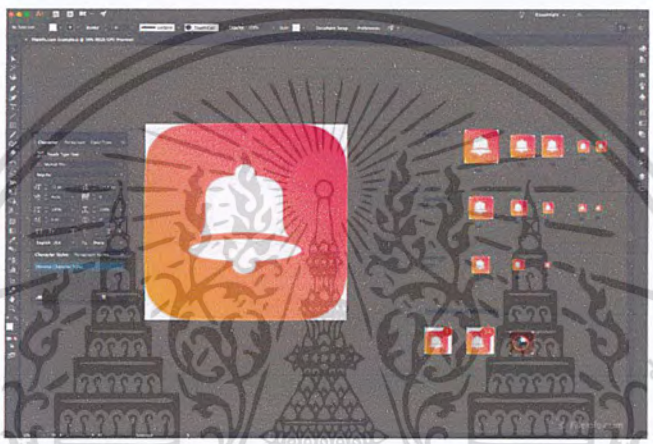
รูปที่ 2.6 ตัวอย่างหน้า GUI ของโปรแกรม Adobe Photoshop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 อะโดบี อิลลัสเตรเตอร์ (Adobe Illustrator)

เป็นโปรแกรมวาดภาพกราฟิกแบบเวกเตอร์ ซึ่งพัฒนาโดยบริษัทอะโดบีซิสเต็มส์ รุ่นแรกถูกพัฒนาขึ้นในปี ค.ศ. 1986 เพื่อใช้งานกับเครื่องแมคอินทอช และได้พัฒนารุ่นที่ 2 ออกมาให้ใช้งานได้กับวินโดวส์ ซึ่งได้รับความพึงพอใจ และ การตอบรับที่ดีจากผู้ใช้เป็นจำนวนมาก ปัจจุบันโปรแกรมอิลลัสเตรเตอร์ได้พัฒนามาถึงรุ่น CC (Creative Cloud)

โดยโปรแกรมอะโดบี อิลลักเตรเตอร์ ผู้พัฒนานำมาใช้ออกแบบไอคอนของแอปพลิเคชันทดสอบการยืนยันตัวบุคคล



รูปที่ 2.7 ตัวอย่างหน้า GUI ของโปรแกรม Adobe Illustrator

2.7 ไฟร์เบส (Firebase)

Firebase คือ Project ที่ถูกออกแบบมาให้เป็น API และ Cloud Storage สำหรับพัฒนา Realtime Application รองรับหลาย Platform ทั้ง iOS, Android หรือ Web ถูกสร้างขึ้นจากคุณสมบัติเสริมว่านักพัฒนาสามารถผสมและจับคู่เพื่อให้พอดีกับความต้องการของตน บริษัท ก่อตั้งขึ้นในปี 2011 โดยแอนดรูลีและเจมส์ เทมปลิน สินค้าเริ่มต้น Firebase ปีนฐานข้อมูลเรียลไทม์ซึ่งมี API ที่ช่วยให้นักพัฒนาในการจัดเก็บและซิงค์ข้อมูล โดย Google Firebase 2.0 กูลเกิลได้ซื้อกิจการ Firebase และมีการพัฒนาให้สามารถ จากบริการ backend เก็บข้อมูลอย่างเดียว มาเป็นแพลตฟอร์ม ครบวงจรสำหรับนักพัฒนา

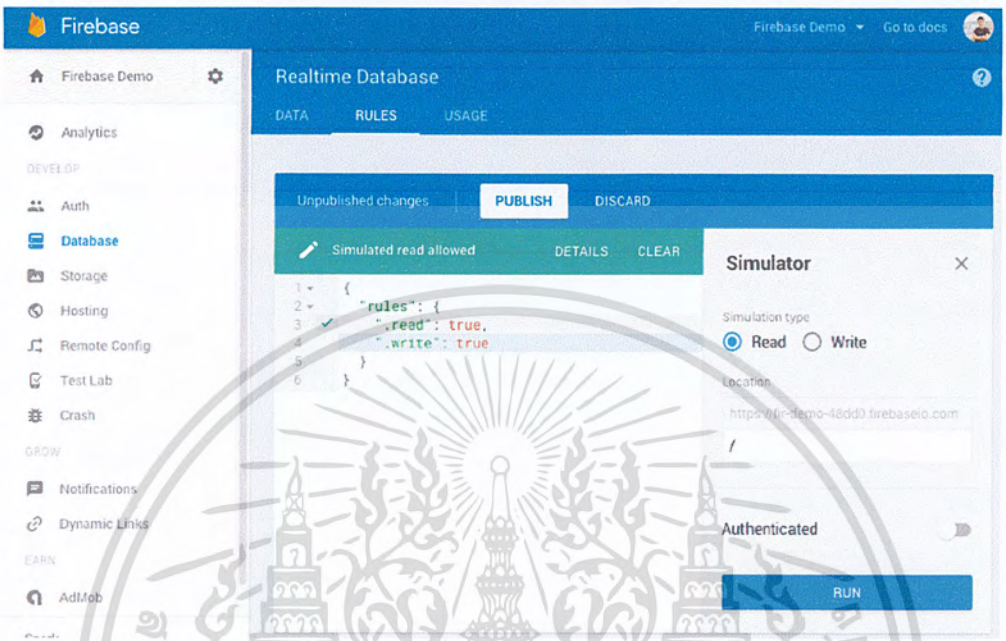
2.7.1 ตัวอย่างบริการของ Firebase ที่ใช้ในการพัฒนา

1) Firebase Realtime database

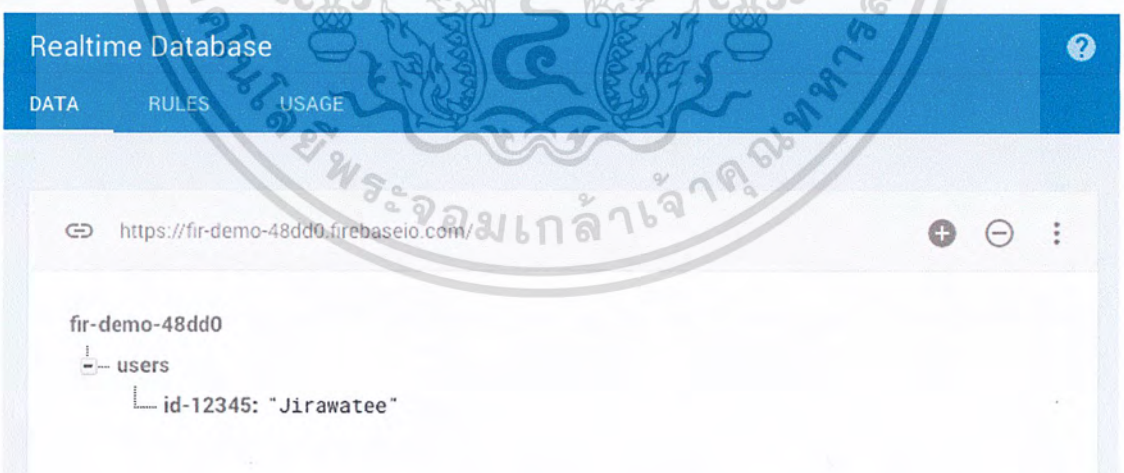
Firebase Realtime Database เป็น NoSQL cloud database ที่เก็บข้อมูลในรูปแบบของ JSON และมีการรับส่งข้อมูลแบบ Realtime กับทุกอุปกรณ์ที่เชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รองรับการทำงานแบบ offline และสามารถตั้งค่า Security Rules เพื่อออกแบบเงื่อนไขการเข้าถึงข้อมูลทั้งการอ่านและเขียน



รูปที่ 2.8 ตัวอย่างหน้าจัดการ Security Rules ใน Firebase Console

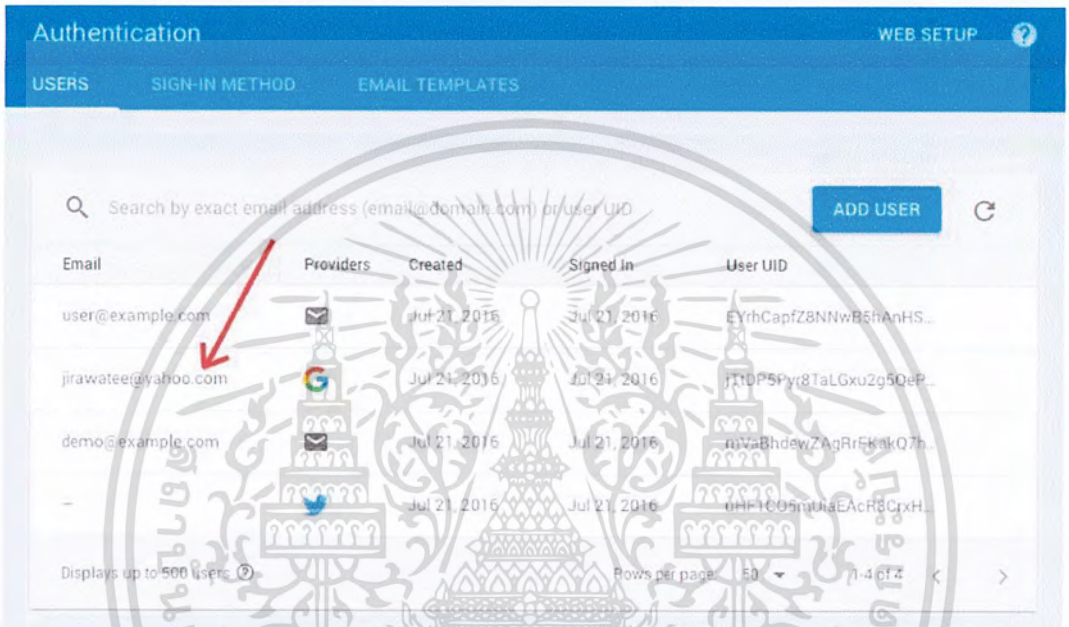


รูปที่ 2.9 ตัวอย่างหน้าแสดงข้อมูลที่ถูกเก็บใน Database ในรูปแบบ NoSQL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) **Firebase Authentication**

Firebase Authentication เป็นบริการที่เข้ามาจัดการ Backend ทั้ง การ Register, การ Sign-in, การ Reset password, โดยจะมี SDK ให้ทั้ง Android, iOS และ Web นำไปติดตั้งและใช้งาน ซึ่งรองรับการ Sign-in หลากหลายรูปแบบทั้งจาก Facebook, Twitter, Github, Google Account หรือแบบไม่ระบุตัวตน(Anonymous) เป็นต้น



รูปที่ 2.10 ตัวอย่างหน้าจัดการ Account ต่างๆ ใน Firebase Console

2.8 จาวา (Java)

Java หรือ Java programming language คือภาษาโปรแกรมเชิงวัตถุ โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับภาษาอ็อบเจกต์ทีฟซี (Objective-C) แต่เดิมภาษานี้เรียกว่า ภาษาโอ๊ก (Oak) ซึ่งตั้งชื่อตามต้นโอ๊กใกล้ที่ทำงานของ เจมส์ กอสลิง แล้วภายหลังจึงเปลี่ยนไปใช้ชื่อ "จาวา" ซึ่งเป็นชื่อกาแฟแทน จุดเด่นของภาษา Java อยู่ที่ผู้เขียนโปรแกรมสามารถใช้หลักการของ Object-Oriented Programming มาพัฒนาโปรแกรมของตนด้วย Java ได้

ภาษา Java เป็นภาษาสำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (OOP : Object-Oriented Programming) โปรแกรมที่เขียนขึ้นถูกสร้างภายในคลาส ดังนั้นคลาสคือที่เก็บเมทอด (Method) หรือพฤติกรรม (Behavior) ซึ่งมีสถานะ (State) และรูปพรรณ (Identity) ประจำพฤติกรรม (Behavior)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 กิท (Git)

Git คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจ็ค มีการ backup code ให้ สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชั่นต่างๆของโปรเจ็คที่ใด เวลาใดก็ได้ หรือดูว่าไฟล์นั้นๆผู้ใดเป็นคนเพิ่มหรือแก้ไข หรือว่าจะดูว่าไฟล์นั้นๆถูกเขียนโดยผู้

2.9.1 Git Status

สถานะของ Source Code ที่เก็บอยู่ในระบบของ Git นั้นมีดั่งนี้

- **Untracked** เป็นสถานะที่ Source Code ถูกเพิ่มเข้ามาใหม่และยังไม่ได้ถูกเก็บไว้ในระบบของ Git
- **Working Directory** เป็นสถานะที่กำลังมีการเปลี่ยนแปลงหรือแก้ไข Source Code หรืออาจจะเรียกสถานะนี้ว่า Modified
- **Staged** เป็นสถานะที่ Source Code กำลังเตรียมที่จะ Commit เพื่อยืนยันการเปลี่ยนแปลงก่อนที่จะเก็บลงในสถานะ Local Repository
- **Local Repository** เป็นสถานะที่มีการเก็บบันทึกข้อมูลการเปลี่ยนแปลงของ Source Code ลงไปที่ Git Repository ที่เป็น Local (ที่เครื่องของตน)
- **Remote Repository** เป็นสถานะที่มีการเก็บบันทึกข้อมูลการเปลี่ยนแปลงของ Source Code ลงไปที่ Git Repository ที่เป็น Hosting (ที่เครื่องเซิร์ฟเวอร์)

2.10 มาเทอเรียลดีไซน์ (Material Design)

Material Design ถูกคิดค้นหรือเป็นแนวทางการออกแบบ โดย Google และเปิดตัว Material Design ในงาน Google I/O ในปี 2014 โดย Material Design ที่ Google ได้วางแนวทางไว้นั้น เหตุเพราะ Google เป็นบริษัทเปิดหรือ แบบเสรี ที่นักพัฒนาจะสามารถหยิบเอาบางส่วนไปพัฒนาต่อได้ Google จึงสร้างแนวทางการออกแบบ Material Design และเปรียบเสมือน ภาษาของการออกแบบ (Visual Language) โดยจะรวบรวม แนวทางการออกแบบ ไว้ด้วยกัน

2.10.1 แนวทางการออกแบบหรือหลักการออกแบบ

1) **Material** วัสดุที่เสมือนจริง ทางด้านกายภาพ ทั้งเรื่องของ แสง-เงา (Light-Edge Shadow) ,พื้นผิว (Surface) การตอบสนองหรือ การเคลื่อนไหว
2) **Bold, graphic, intentional** การจงใจออกแบบส่วนสำคัญต่างๆ ได้แก่ ช่องตารางการออกแบบ (Grid) , ตัวอักษรในการการออก (Typography) , พื้นที่ว่าง (Space) , สี (Color) และ รูปภาพ (Imagery) การจัดลำดับชั้นของความหมายและบ่งบอกจุดโฟกัสที่สำคัญของสายตา เน้นการกระทำของผู้ใช้เป็นหลัก (UX)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) **Motion to Meaning** การเคลื่อนไหวหรือการแสดงให้เห็นว่า วัตถุที่เคลื่อนไหวนั้นจะให้ความหมายอย่างไรกับผู้ใช้

2.11 สัดส่วนทองคำ (Golden Ratio)

มาจากการคำนวนทางคณิตศาสตร์เพื่อทำให้งานออกแบบมีสัดส่วนที่งามตามสูตรคำนวณที่คิดค้นโดย เลโอนาร์โด ฟีโบนัชชี ที่ใช้ตัวเลขทางคณิตศาสตร์มาอธิบายความงามงดงามของธรรมชาติ อัตราส่วนของสัดส่วนทองคำจะเท่ากับ 1 : 1.618

โดยการออกแบบโดยใช้สัดส่วนทองคำสามารถนำไปประยุกต์ใช้ได้หลายแขนง ยกตัวอย่าง เช่น การออกแบบโลโก้ของแบรด์ Apple หรือ NATIONAL GEOGRAPHIC เป็นต้น

รูปที่ 2.11 ตัวอย่างโลโก้ที่ใช้สัดส่วนทองคำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการดำเนินงานสหกิจศึกษา

ในวิธีการดำเนินงานสหกิจนี้ จะกล่าวถึงขั้นตอนการพัฒนาแอปพลิเคชันสอบการยืนยันตัวบุคคลเพียงอย่างเดียว เนื่องจากงานที่ได้รับมอบหมายอื่นๆ ไม่ได้อยู่ในขอบเขตการทำงานของสหกิจศึกษานี้ โดยขั้นตอนการพัฒนาแอปพลิเคชันทดสอบการยืนยันตัวบุคคล ได้มีการวิเคราะห์ปัญหาของระบบเดิมเพื่อสร้างฟังก์ชันใหม่ และการออกแบบส่วนติดต่อผู้ใช้งานใหม่เพื่อการใช้งานที่ง่ายขึ้น มีขั้นตอนดังนี้

3.1 กระบวนการทำงานของระบบ

กระบวนการทำงานเดิมของแอปพลิเคชันทดสอบการยืนยันตัวบุคคลมีลำดับดังนี้

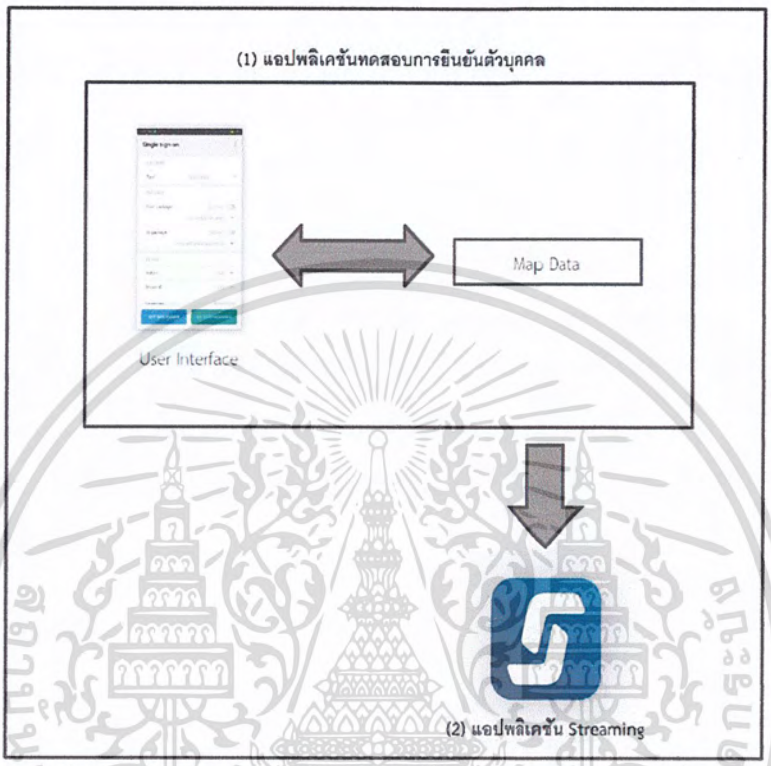
1) เทสเตอร์กรอกรายละเอียดของชุดทดสอบที่ตนต้องการผ่านฟอร์มของแอปพลิเคชันทดสอบการยืนยันตัวบุคคล
2) เมื่อได้ชุดทดสอบที่เทสเตอร์ต้องการ ทำการกดปุ่ม Go to Streaming
3) แอปพลิเคชันทดสอบการยืนยันตัวบุคคลจะทำการเข้ารหัสข้อมูลตามฟอร์มที่เทสเตอร์ได้กรอกไว้ แล้วส่งต่อข้อมูลเข้ารหัสชุดนี้ไปให้แอปพลิเคชัน Streaming
4) แอปพลิเคชัน Streaming จะทำการตรวจสอบแบบ Single Sign-On (SSO) จากข้อมูลที่เข้ารหัสแล้วส่งมา

จากกระบวนข้างต้นผู้พัฒนาจะได้พัฒนาระบบในส่วนของ ข้อ 1) ข้อ 2) และข้อ 3) เท่านั้น เนื่องจากข้อ 4) เป็นการทำงานภายในแอปพลิเคชัน Streaming ซึ่งไม่มีผลกับการพัฒนา แอปพลิเคชันทดสอบการยืนยันตัวบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 วิเคราะห์ระบบ

3.2.1 วิเคราะห์ระบบเดิม



รูปที่ 3.1 สถาปัตยกรรมของแอปพลิเคชันเดิม

จากรูป 3.1 มีองค์ประกอบดังนี้

1) **แอปพลิเคชันทดสอบการยืนยันตัวบุคคล** เทสเตอร์หรือผู้ใช้งานจะทำการสร้างชุดคำสั่งผ่าน User Interface หลังจากกดส่งข้อมูล แอปพลิเคชันจะทำการประมวลผลจัดรูปแบบของชุดคำสั่งที่ผู้ใช้กรอก แล้วส่งไปให้แอปพลิเคชัน Streaming
2) **แอปพลิเคชัน Streaming** ตัวแอปพลิเคชันจะรับข้อมูลจากแอปพลิเคชันทดสอบการยืนยันตัวบุคคลเพื่อปะมวลผลการเข้าใช้งานแบบ Single Sign-On (SSO)

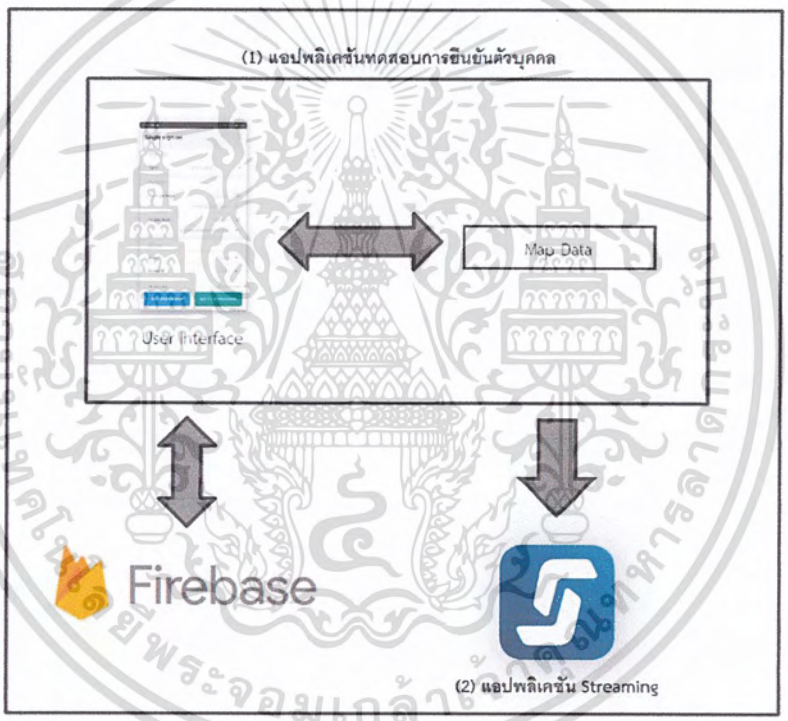
โดยการทำงานของระบบเดิมมีปัญหาดังนี้

1) เนื่องจากการเข้าใช้งานแอปพลิเคชันเดิมนั้น จะไม่มีการตรวจสอบว่าผู้ใช้งานคนนี้เป็นใคร จึงอาจเกิดปัญหาข้อมูลรั่วไหลได้ และอาจเกิดปัญหามากมายตามมาในภายหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ถ้าผู้ใช้งานทดสอบหลายครั้ง จำเป็นจะต้องกรอกฟอร์มใหม่ทุกครั้งจึงทำให้ยากต่อการใช้งาน และใช้เวลานานในการทดสอบ
3) เนื่องจากถ้าผู้ใช้งานต้องการจะเพิ่มข้อมูลหรือแก้ไขข้อมูลในฟอร์ม จำเป็นจะต้องเขียนโปรแกรมเพื่อแก้ไขส่วนนั้นๆแล้วทำการคอมไพล์แอปพลิเคชันลงบน สมาร์มโฟน Android ใหม่ทุกครั้ง จึงเกิดปัญหาเพราะผู้ใช้งานบางคนไม่มีความสามารถด้านการเขียนโปรแกรมจึงไม่สามารถทำการแก้ไขฟอร์มได้

3.2.2 วิเคราะห์ระบบใหม่



รูปที่ 3.2 สถาปัตยกรรมของแอปพลิเคชันใหม่

จากรูป 3.2 ได้นำบริการของ Firebase มาใช้เพื่อจัดการการทำงานต่างๆ โดยมีองค์ประกอบดังนี้

1) **แอปพลิเคชันทดสอบการยืนยันตัวบุคคล** เทสเตอร์หรือผู้ใช้งานจำเป็นจะต้องลงชื่อเข้าใช้งานเพื่อใช้งานฟังก์ชันต่างๆ ประกอบไปด้วยการสร้างชุดคำสั่งของตน การดูประวัติการส่งชุดคำสั่ง หรือการส่งชุดคำสั่งผ่าน User Interface หลังจากกดส่งข้อมูล แอปพลิเคชันจะทำการประมวลผลจัดรูปแบบของชุดคำสั่งที่ผู้ใช้กรอก แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

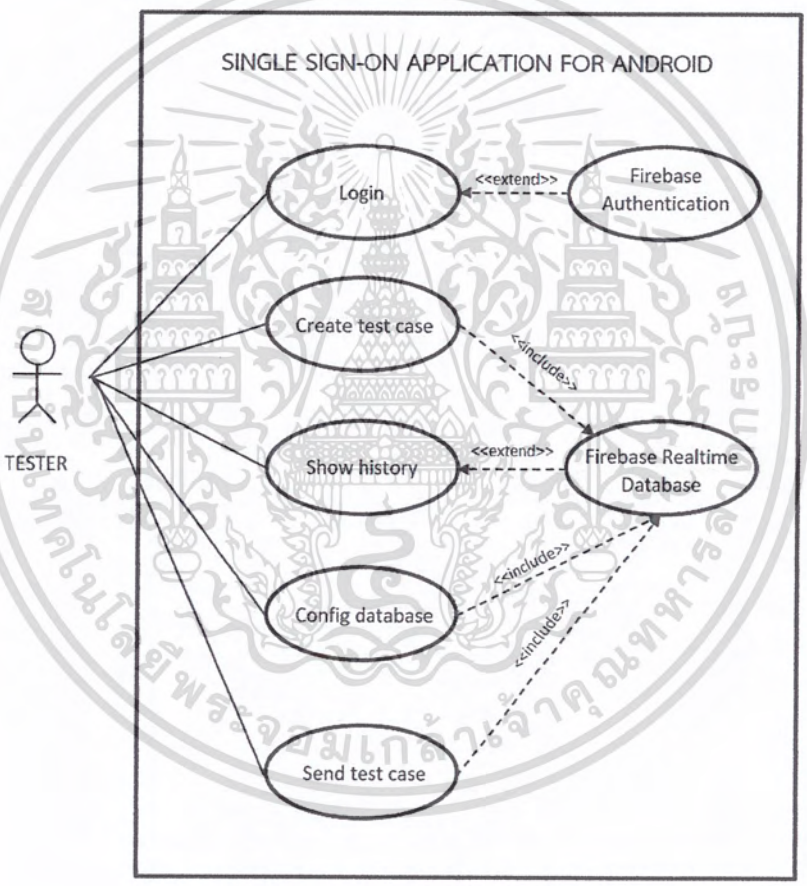
ส่งไปให้แอปพลิเคชัน Streaming พร้อมทั้งบันทึกชุดข้อมูลนั้นๆ ลงดาต้าเบสของ Firebase

2) **แอปพลิเคชัน Streaming** ตัวแอปพลิเคชันจะรับข้อมูลจากแอปพลิเคชันทดสอบการยืนยันตัวบุคคลเพื่อประมวลผลการเข้าใช้งานแบบ Single Sign-On (SSO)

3.3 การออกแบบระบบ

3.3.1 แผนภาพยูสเคส (Use Case Diagram)

การทำงานของระบบแสดงดังรูป 3.3



รูปที่ 3.3 แผนภูมิยูสเคสของแอปพลิเคชันทดสอบการยืนยันตัวบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.3 สามารถอธิบายการทำงานของแต่ละฟังก์ชันได้ดังนี้

Use Case Title : Login	**Use Case Id :** 1
Primary Actor : Tester	
Flow : การเข้าใช้งานของ Tester ทุกครั้งจำเป็นต้องมีการตรวจสอบก่อนการเข้าใช้งานแอปพลิเคชันทดสอบการยืนยันตัวบุคคลเสียก่อน โดยการพิมพ์ Username และ Password ที่ทางทีมพัฒนาตกลงกันไว้ลงไป หลังจากกด Login ระบบจะทำการตรวจสอบ Username และ Password นี้โดยใช้ข้อมูลจาก Firebase Authentication มาตรวจสอบ	

ตารางที่ 1.1 ตาราง Use case อธิบายการทำงาน Login

Use Case Title : Create test case	**Use Case Id :** 2
Primary Actor : Tester	
Flow : Tester สร้าง Test case ที่ต้องการแล้วบันทึก ระบบจะนำข้อมูล Test case ที่สร้างไว้ เก็บลงใน Firebase Realtime Database	

ตารางที่ 1.2 ตาราง Use case อธิบายการทำงาน Create test case

Use Case Title : Show history	**Use Case Id :** 3
Primary Actor : Tester	
Flow : ระบบจะนำข้อมูลของชุดคำสั่งที่ Tester ได้กดส่งไปยังแอปพลิเคชัน Streaming มาแสดงในส่วนติดต่อผู้ใช้งานตามช่วงเวลาที่กดส่งชุดคำสั่ง	

ตารางที่ 1.3 ตาราง Use case อธิบายการทำงาน Show history

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Use Case Title : Config database	**Use Case Id :** 4
Primary Actor : Tester	
Flow : Tester ทำการเพิ่ม ลบ หรือเปลี่ยนแปลงตัวเลือกของฟอร์ม ระบบจะนำตัวเลือกที่เพิ่ม ลบ หรือเปลี่ยนแปลง บันทึกลงใน Firebase Realtime Database และทำการอัพเดทส่วนติดต่อผู้ใช้งานตามข้อมูลที่เปลี่ยนแปลงแบบ Real-time	

ตารางที่ 1.4 ตาราง Use case อธิบายการทำงาน Config database

Use Case Title : Send test case	**Use Case Id :** 5
Primary Actor : Tester	
Flow : Tester ทำการกดส่งชุดคำสั่งไปยังแอปพลิเคชัน Streaming ระบบจะทำการบันทึกข้อมูลทั้งของชุดคำสั่งนั้น รวมถึงเวลาที่กดส่งชุดคำสั่งนั้น ลงใน Firebase Realtime Database	

ตารางที่ 1.5 ตาราง Use case อธิบายการทำงาน Send test case

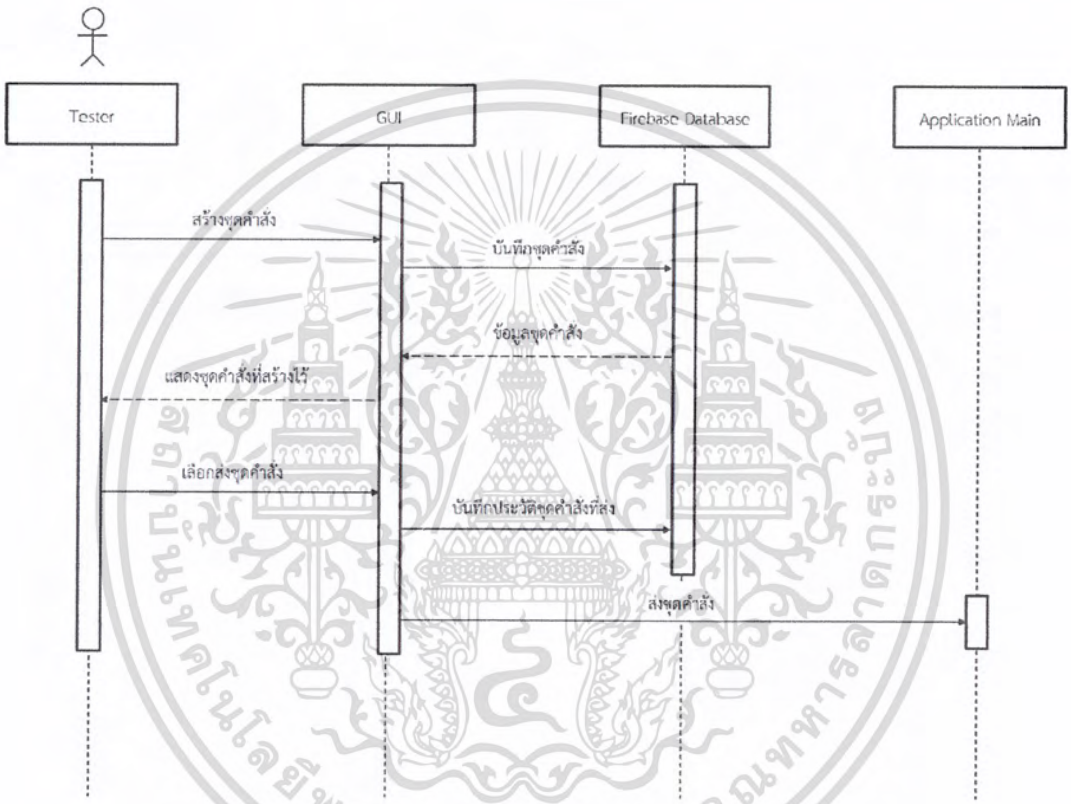
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 ขั้นตอนการทำงานของระบบ (Sequence Diagram)

ขั้นตอนการทำงานของระบบแบ่งได้ 3 ฟังก์ชัน คือ

3.3.2.1 ฟังก์ชันการสร้างและบันทึกชุดคำสั่ง (Profile)

การสร้างและบันทึกชุดคำสั่ง มีขั้นตอนการทำงานแสดงใน Sequence Diagram ดังรูป 3.4



รูปที่ 3.4 Sequence Diagram ฟังก์ชันการสร้างและบันทึกชุดคำสั่ง (Profile)

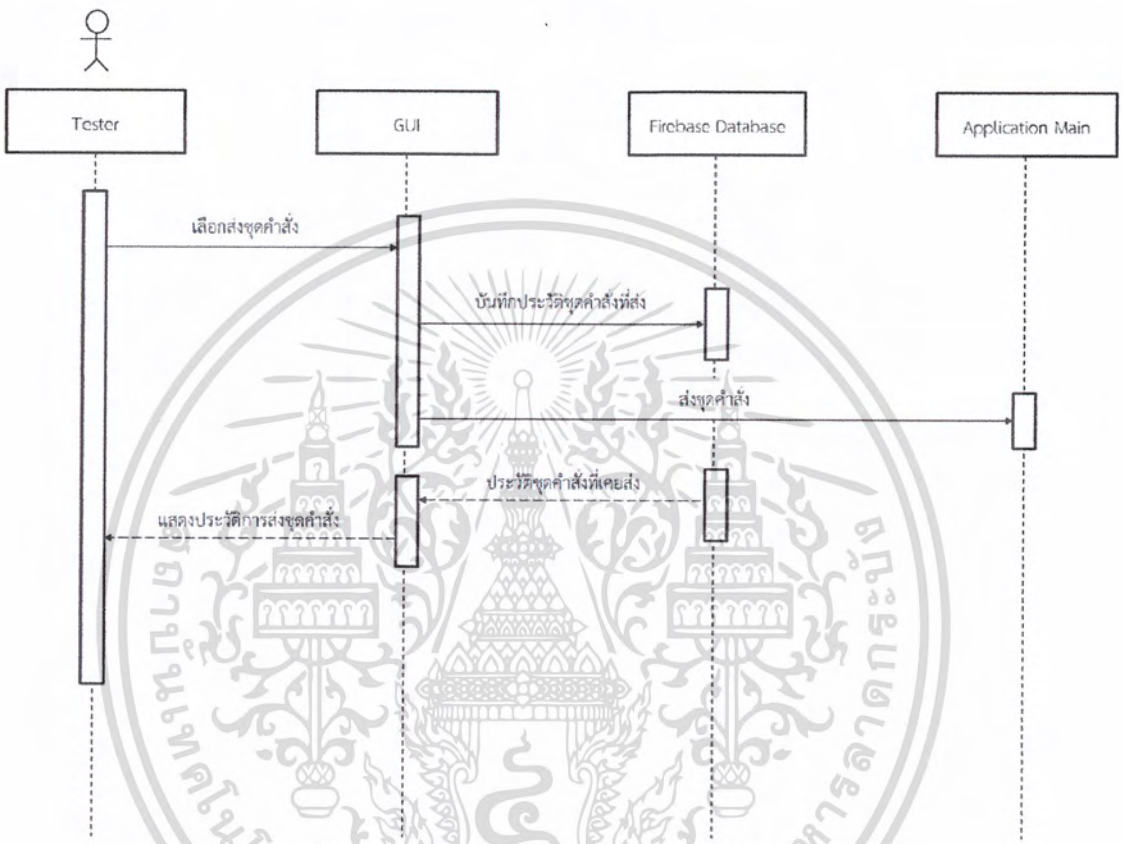
จากรูป 3.4 มีฟังก์ชันการทำงาน ดังนี้ gf

1) สร้างชุดคำสั่ง: เมื่อผู้ใช้เข้าเมนู Profile และกดเพิ่มชุดคำสั่ง ในหน้า GUI พร้อมกรอกฟอร์มให้ แล้วกดบันทึก ระบบจำทำการส่งข้อมูลในฟอร์มไปบันทึกในดาต้าเบสของ Firebase
2) เลือกส่งชุดคำสั่ง: เมื่อผู้ใช้เข้าเมนู Profile และกดเลือกชุดคำสั่งที่ต้องการจะส่ง ในหน้า GUI ระบบจำทำการบันทึกประวัติชุดคำสั่งที่ส่งไปบันทึกในดาต้าเบสของ Firebase พร้อมทั้งส่งชุดคำสั่งไปยังแอปพลิเคชันหลักที่ต้องการจะทดสอบ Single Sign-On (SSO)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2.2 ฟังก์ชันบันทึกประวัติชุดคำสั่ง (History)

การบันทึกประวัติชุดคำสั่ง มีขั้นตอนการทำงานแสดงใน Sequence Diagram ดังรูป 3.5



รูปที่ 3.5 Sequence Diagram ฟังก์ชันการบันทึกประวัติชุดคำสั่ง (History)

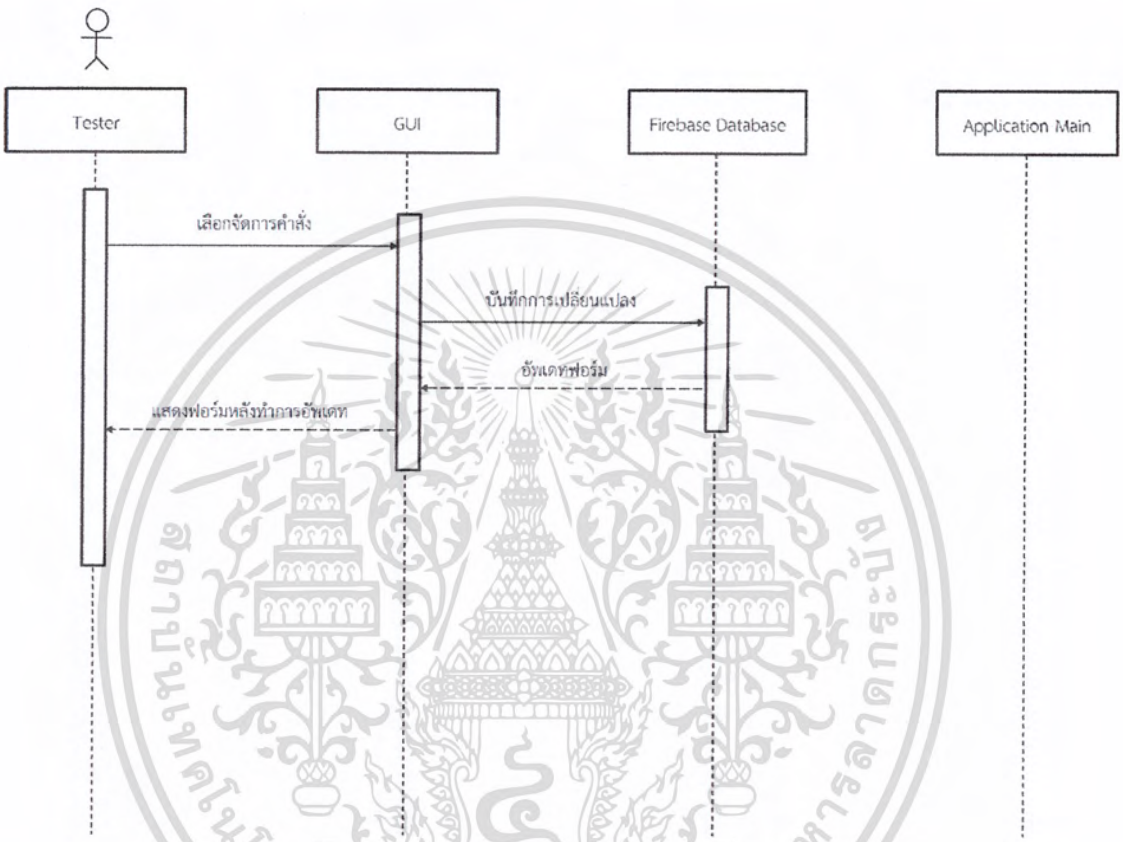
จากรูป 3.5 มีฟังก์ชันการทำงาน ดังนี้

1) เลือกส่งชุดคำสั่ง: เมื่อผู้ใช้กรอกฟอร์มคำสั่งเสร็จ แล้วกดส่งชุดคำสั่ง ในหน้า GUI ระบบจะทำการบันทึกประวัติชุดคำสั่งที่ส่งไปบันทึกในดาต้าเบสของ Firebase พร้อมทั้งส่งชุดคำสั่งไปยังแอปพลิเคชันหลักที่ต้องการจะทดสอบ Single Sign-On (SSO) หลังจากนั้นดาต้าเบสจะทำการส่งประวัติชุดคำสั่งที่กดส่งไปทั้งหมดกลับมาแสดงในหน้า GUI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2.3 ฟังก์ชันจัดการตัวเลือกในฟอร์มบนดาต้าเบส (Config Database)

การจัดการคำสั่งในดาต้าเบส มีขั้นตอนการทำงานแสดงใน Sequence Diagram ดังรูป 3.6



รูปที่ 3.6 Sequence Diagram ฟังก์ชันจัดการตัวเลือกในฟอร์มบนดาต้าเบส (Config Database)

จากรูป 3.6 มีฟังก์ชันการทำงาน ดังนี้

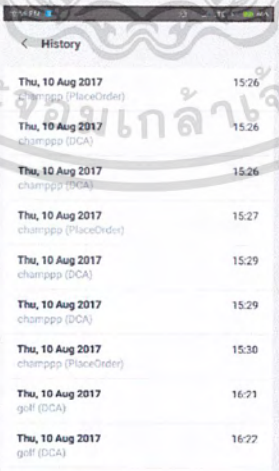
เลือกจัดการคำสั่ง: เมื่อผู้ใช้ทำการเพิ่ม ลบ หรือเปลี่ยนแปลงกับตัวเลือกของฟอร์ม ในหน้า GUI แล้วกดบันทึก ระบบจะทำการบันทึกการเปลี่ยนแปลงหรืออัพเดทข้อมูลบนดาต้าเบส แล้วจะทำการอัพเดทตัวฟอร์มในแอปพลิเคชันด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ส่วนติดต่อผู้ใช้งาน (User Interface)

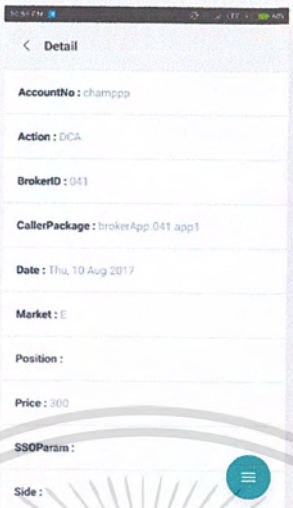
ในการออกแบบส่วนติดต่อผู้ใช้งาย (User Interface) ผู้จัดทำได้นำวิธีการของ Material Design มาช่วยในการออกแบบเพราะจะช่วยให้เทสเตอร์อ่านข้อมูลในแอปพลิเคชันได้ง่ายขึ้น โดยจะใช้สีที่เรียบง่าย มีการใช้ลักษณะตัวหนาที่หัวข้อและตัวบางที่ตัวเลือกของแต่ละหัวข้อ มีการเว้นช่องไฟของแต่ละส่วนที่พอเหมาะ โดยจะยกตัวอย่างส่วนติดต่อผู้ใช้งานที่ผ่านการออกแบบแบบ Material Design ดังรูปที่ 3.7 ถึงรูปที่ 3.13

รูปที่ 3.7 หน้าจอเริ่มต้นเมื่อเข้าแอปพลิเคชัน

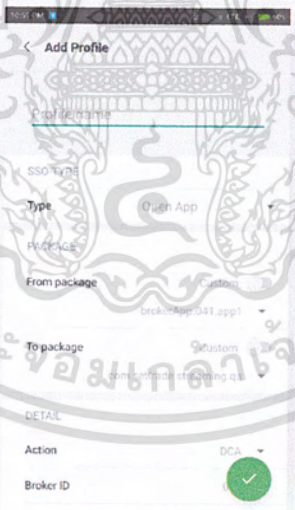


รูปที่ 3.8 หน้าจอแสดงรายการประวัติชุดคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

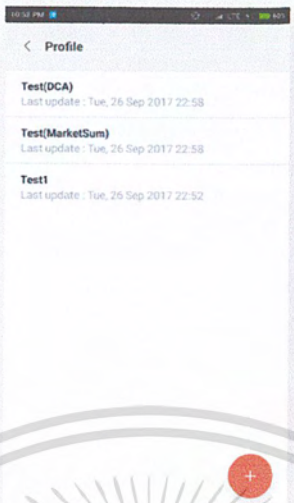


รูปที่ 3.9 หน้าจอแสดงข้อมูลของรายการประวัติชุดคำสั่ง



รูปที่ 3.10 หน้าจอแสดงฟังก์ชัน Profile

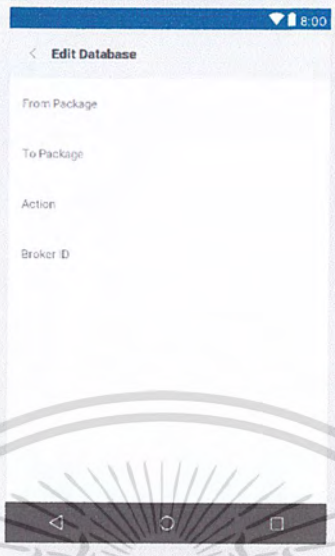
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 หน้าจอแสดงรายการชุดคำสั่งที่เคยสร้างไว้จากฟังก์ชัน Profile

รูปที่ 3.12 หน้าจอแสดงข้อมูลของรายการของชุดคำสั่งที่เคยสร้างไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 หน้าจอแสดงรายการที่สามารถจัดการได้

3.5 การออกแบบโลโก้

การออกแบบโลโก้แอปพลิเคชันผู้จัดทำได้นำวิธีการสัดส่วนทองคำ (Golden Ratio) มาใช้และสีที่ใช้เป็นสีโทนเข้มตามหลัก Material Design และแนวคิดที่ใช้รวมกับการออกแบบคือคำว่า Single Sign-On (SSO) และรูปกุญแจ เพื่อสื่อถึงความปลอดภัย โดยตัวอย่างโลโก้ของแอปพลิเคชันดังรูปที่ 3.14

รูปที่ 3.14 โลโก้แอปพลิเคชันทดสอบการทดสอบการยืนยันตัวบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 ตัวอย่างโค้ดโปรแกรม

ตัวอย่างโค้ดในหัวข้อนี้จะนำมาแสดงและอธิบายเพียงบางส่วนเท่านั้น

3.6.1 ตัวอย่างโค้ดเพื่อตรวจสอบการเข้าใช้งาน

ตรวจสอบการเข้าใช้งานของ Tester โดยใช้ Firebase Authentication ดังรูปที่ 3.15 มีการทำงานดังนี้ บรรทัดที่ 1 สร้างตัวแปร FirebaseAuth ชื่อ mAuth เพื่อเป็นการเรียกใช้งาน Firebase Authentication เมื่อฟังก์ชัน validateLogin ถูกเรียกใช้งานจะมีตรวจสอบ username และ password จากหน้าจัดการ Account ใน Firebase Console ถ้ามี username นั้นอยู่และ password ถูกต้อง จะทำการเปิดหน้าถัดไปของแอปพลิเคชัน ถ้าไม่มี user นั้น หรือมี user นั้นแต่ password ไม่ถูกต้อง จะทำการแจ้งเตือน Authentication failed และอยู่ที่หน้าปัจจุบัน

```
private FirebaseAuth mAuth = FirebaseAuth.getInstance();

private void validateLogin(String username, String password) {
    mAuth.signInWithEmailAndPassword(username, password)
            .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful()) {
                        Intent intent = new Intent(getApplication(), PAGE_2.class);
                        startActivity(intent);
                        finish();
                    } else {
                        Toast.makeText(getApplication(), "Authentication failed.",
                                Toast.LENGTH_SHORT).show();
                    }
                }
            });
}
```

รูปที่ 3.15 ตัวอย่างโค้ดการบันทึกข้อมูลลง Firebase Realtime Database

โดยหลักการเข้าสู่ระบบของ Firebase Authentication ที่เลือกใช้คือ เข้าสู่ระบบโดยใช้ email และ password ผ่านฟังก์ชัน signInWithEmailAndPassword มีขั้นตอนการใช้งาน ดังนี้

1. ประกาศตัวแปรเพื่อรับค่า Instance จาก Firebase

```
FirebaseAuth mAuth = FirebaseAuth.getInstance();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ส่ง email และ password ให้กับฟังก์ชัน

```
mAuth.signInWithEmailAndPassword(username,
password)...
```

3. ตรวจสอบการเข้าสู่ระบบ

```
if (task.isSuccessful()) {
        FirebaseUser user = mAuth.getCurrentUser();
        updateUI(user);
    } else {
        Toast.LENGTH_SHORT).show();
        updateUI(null);
    }
```

จากข้อ 3 ถ้าเข้าสู่ระบบสำเร็จ สามารถใช้ FirebaseUser เพื่อดึงข้อมูลของ Tester มาใช้ตามต้องการได้

3.6.2 ตัวอย่างโค้ดเพื่ออ่านข้อมูล

การอ่านข้อมูลใน Main_Field จาก Firebase Realtime Database ดังรูปที่ 3.16 มีการทำงานดังนี้ บรรทัดที่ 1 กำหนดให้ myRef เป็นตัวอ้างอิงที่อยู่ของข้อมูลในดาต้าเบสที่ชื่อ Main_Field บรรทัดที่ 2 สร้างตัวแปรอาร์เรย์ลิสต์ชื่อ data_1 บรรทัดที่ 6 จะทำงานในกรณีที่มีการเพิ่มข้อมูลใน myRef เมื่อมีการเพิ่มข้อมูลเข้ามา บรรทัดที่ 8 จะทำการตรวจสอบว่า key ตรงกับคำว่า Field_1 หรือไม่ ถ้าตรงจะนำค่าที่ได้มาเก็บใน data_1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
private myRef = database.getReference().child("Main_Field");
private List<String> data_1= new ArrayList<String>();

myRef.addChildEventListener(new ChildEventListener() {
    @Override
    public void onChildAdded(DataSnapshot dataSnapshot, String s) {
        for (DataSnapshot children : dataSnapshot.getChildren()) {
            if(children.getKey().toString().equals("Field_1")){
                data_1.add(children.getValue().toString());
            }
        }
    }
});
```

รูปที่ 3.16 ตัวอย่างโค้ดการอ่านข้อมูลจาก Firebase Realtime Database

โดยหลักการดึงข้อมูลของ Firebase Realtime Database ดังรูปที่ 3.16 เรียกว่า ChildEventListener ซึ่งจะคอยรับข้อมูลเมื่อมีการเพิ่ม การเปลี่ยนแปลง การลบ และการย้ายจาก child ที่อ้างอิง โดยจะมี callback 5 แบบ คือ

- onChildAdded() ทำงานเมื่อมีการเพิ่มชุดข้อมูลเข้ามาใน child
- onChildChanged() ทำงานเมื่อข้อมูลใน child มีการเปลี่ยนแปลง
- onChildRemoved() ทำงานเมื่อข้อมูลใน child ถูกลบ
- onChildMoved() ทำงานเมื่อมีการเรียงลำดับของข้อมูลใน child เกิดขึ้น
- onCancelled() ทำงานเมื่อโหลดข้อมูลจาก child ไม่สำเร็จ

แต่ภายรูปที่ 3.16 ใช้เพียง onChildAdded() เท่านั้นจึงละ callback อื่นออก และขอยกตัวอย่างการใช้งานทั้ง 5 callback ดังโค้ดต่อไปนี้

```
ChildEventListener childEventListener = new ChildEventListener() {
    @Override
    public void onChildAdded(DataSnapshot dataSnapshot, String
previousChildName) {

        Comment comment = dataSnapshot.getValue(Comment.class);
        // ...
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    @Override
    public void onChildChanged(DataSnapshot dataSnapshot, String
previousChildName) {

        Comment newComment = dataSnapshot.getValue(Comment.class);
        String commentKey = dataSnapshot.getKey();
        // ...
    }

    @Override
    public void onChildRemoved(DataSnapshot dataSnapshot) {

        String commentKey = dataSnapshot.getKey();
        // ...
    }

    @Override
    public void onChildMoved(DataSnapshot dataSnapshot, String
previousChildName) {
        Comment movedComment =
dataSnapshot.getValue(Comment.class);
        String commentKey = dataSnapshot.getKey();
        // ...
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {
        Toast.makeText(mContext, "Failed to load comments.",
                Toast.LENGTH_SHORT).show();
    }
};
ref.addChildEventListener(childEventListener);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.3 ตัวอย่างโค้ดเพื่อเขียนข้อมูล

การเขียนข้อมูลลงใน Firebase Realtime Database ดังรูปที่ 3.17 มีการทำงานดังนี้ บรรทัดที่ 1 กำหนดให้ myRef เป็นตัวอ้างอิงที่อยู่ของข้อมูลในดาต้าเบสที่ชื่อ Main_Field บรรทัดที่ 2 สร้างตัวแปรแฮชแมพชื่อ map บรรทัดที่ 5 และ 6 เป็นการนำข้อมูลใส่ใน map โดยมี key และ value ดังนี้ (key_child_1, data1) และ (key_child_2, data2) บรรทัดที่ 8 เป็นการนำตัวแปร map เขียนลงไปใน myRef

```
1  private myRef = database.getReference().child("Main_Field");
2  private Map<String, String> map = new HashMap<String, String>();
3
4  private void saveDataToMainField() {
5      map.put("key_child_1", data1);
6      map.put("key_child_2", data2);
7
8      myRef.push().setValue(map);
9  }
```

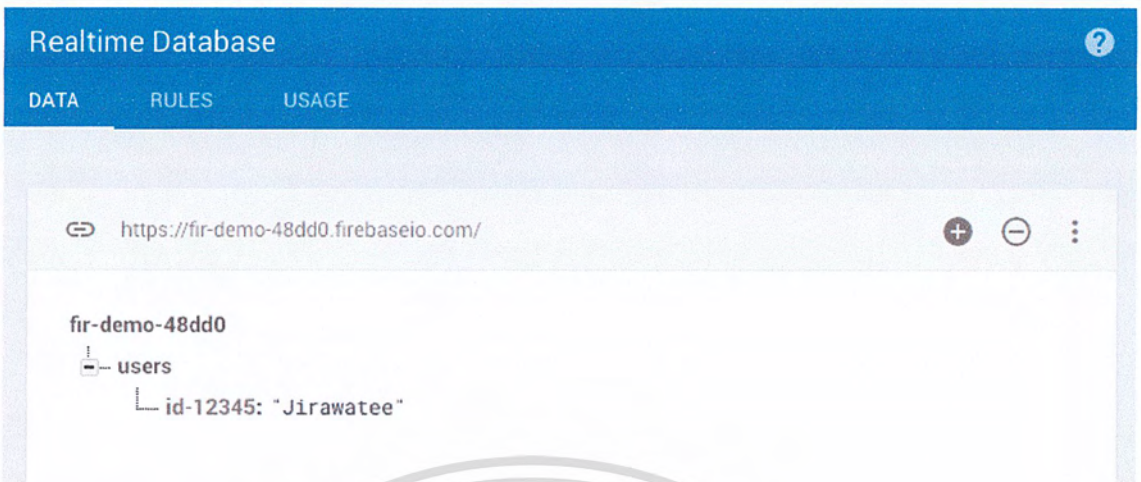
รูปที่ 3.17 ตัวอย่างโค้ดการเขียนข้อมูลลง Firebase Realtime Database

โดยหลักการเขียนข้อมูลลง Firebase Realtime Database ลองรับค่าได้หลายประเภท เช่น String, Long, Double, Boolean, Map<String, Object> และ List<Object> เป็นต้น โดยฟังก์ชันที่ใช้ในการเขียนข้อมูลคือ setValue เป็นการเขียนข้อมูลหรืออัพเดทข้อมูลไปยัง path ที่อ้างอิง เช่น

```
mUsersRef.child("id-12345").setValue("Jirawatee");
```

ผลลัพธ์ที่ได้สามารถดูได้ที่ Firebase Console ดังรูปที่ 3.18 จะเห็นได้ว่ามีข้อมูลถูกเขียนลงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 หน้าแสดงข้อมูลของ Firebase Realtime Database บน Firebase Console

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

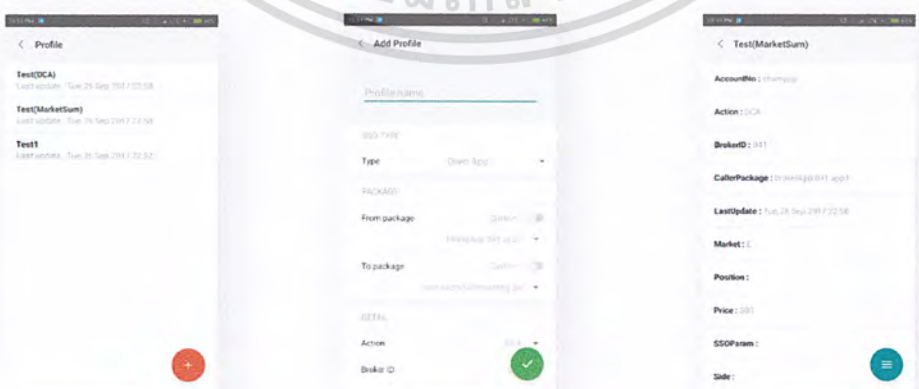
บทที่ 4
ผลการพัฒนาแอปพลิเคชัน

แอปพลิเคชันนี้พัฒนาโดยใช้ Android Studio ซึ่งใช้สำหรับสร้างแอปพลิเคชันบระบบปฏิบัติการแอนดรอยด์ โดยการพัฒนาจะใช้ภาษา Java ในการควบคุมการทำงานต่างๆภายในแอปพลิเคชัน และใช้ภาษา XML ในการออกแบบส่วนติดต่อผู้ใช้ (User Interface)

แอปพลิเคชันทดสอบการยืนยันตัวบุคคลบนระบบปฏิบัติการแอนดรอยด์ แบ่งออกเป็น 3 ฟังก์ชันหลัก ได้แก่ Profile, Save History และ Config Database โดยแต่ละฟังก์ชันมีการทำงาน ดังนี้

4.1 ฟังก์ชัน Profile

ฟังก์ชัน Profile เป็นฟังก์ชันสำหรับสร้างชุดคำสั่ง Test case พร้อมตั้งชื่อ Test case นั้นแล้วบันทึกลง Firebase Real-time Database โดยลำดับการทำงานของฟังก์ชันนี้คือ เมื่อผู้ใช้งานเลือกเมนู Profile จากเมนูด้านล่าง จะปรากฏหน้าต่างรายการชุดคำสั่ง Test case ต่างๆ ที่ได้สร้างไว้(รูปซ้าย) ถ้าผู้ใช้งานต้องการจะสร้างชุดคำสั่ง Test case ให้กดปุ่มบวกสีแดง แล้วจะปรากฏหน้าต่างฟอร์มกรอกชุดคำสั่ง Test case ขึ้นมา(รูปกลาง) หลังจากผู้ใช้งานกรอกฟอร์มเสร็จเรียบร้อย ถ้าต้องการบันทึกให้กดปุ่มเครื่องหมายถูกสีเขียว แล้วระบบจะทำการบันทึกชุดคำสั่งนี้ แล้วชุดคำสั่งนี้จะไปแสดงในหน้าต่างรายการชุดคำสั่ง Test case ถ้าผู้ใช้ต้องการจะดูรายละเอียดของชุดคำสั่งที่ตนสร้างไว้ สามารถกดเข้าไปดูได้จากหน้ารายการชุดคำสั่ง Test case จะปรากฏหน้าต่างรายละเอียดของชุดคำสั่ง Test case นั้นๆ (รูปขวา)

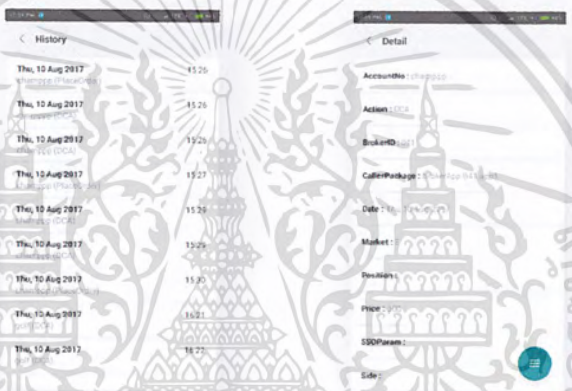


รูปที่ 4.1 หน้าต่างการทำงานของฟังก์ชัน Profile

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ฟังก์ชัน Save History

ฟังก์ชัน Profile เป็นฟังก์ชันสำหรับดูประวัติการส่งชุดคำสั่ง Test case ว่ามีการส่งชุดคำสั่งชื่ออะไร วันที่เท่าไร และเวลาเท่าไร โดยลำดับการทำงานของฟังก์ชันนี้คือ เมื่อผู้ใช้มีการกดส่งคำสั่งจากหน้าต่างหลัก หรือหน้าต่าง Profile ระบบจะทำการบันทึกข้อมูลการส่งไว้ ได้แก่ ข้อมูลทั้งหมดของชุดคำสั่งที่ส่งไป วันที่ที่ส่ง และเวลาที่ส่ง เมื่อผู้ใช้งานกดปุ่มเมนู History จะปรากฏรายการประวัติการส่งชุดคำสั่ง(รูปซ้าย) โดยเรียงตามเวลาล่าสุดอยู่ด้านบน ถ้าผู้ใช้ต้องการดูรายละเอียดของรายการประวัตินั้นๆ สามารถกดรายการประวัติที่ต้องการดู จะปรากฏหน้าต่างรายละเอียดข้อมูลต่างๆ ของชุดคำสั่ง Test case นั้นๆ (รูปขวา)

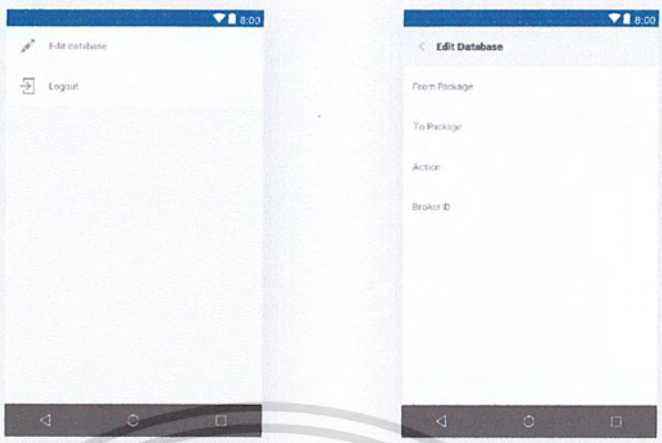


รูปที่ 4.2 หน้าต่างการทำงานของฟังก์ชัน Save

4.3 ฟังก์ชัน Config Database

ฟังก์ชัน Config Database เป็นฟังก์ชันสำหรับแก้ไขฟอร์มกรอกชุดคำสั่ง Test case เช่น เพิ่มตัวเลือกหรือลบตัวเลือกได้ โดยลำดับการทำงานของฟังก์ชันนี้คือ เมื่อผู้ใช้กดเมนู Setting ด้านล่าง จะปรากฏหน้าต่างรายการตั้งค่า(รูปซ้าย) โดยรายการในหน้านี้จะประกอบไปด้วย Edit Database และ Logout เมื่อผู้ใช้กด Edit Database จะเข้าสู่หน้ารายการคำสั่งที่สามารถเพิ่มตัวเลือกได้(รูปขวา) เมื่อผู้ใช้กดไปที่รายการที่ต้องการจะปรากฏหน้าต่างรายการตัวเลือกของคำสั่งนั้นๆ และถ้าผู้ใช้ต้องการเพิ่มตัวเลือกต้องกดเครื่องหมายบวกสีแดง จะปรากฏไดอะล็อคขึ้นมาให้กรอกตัวเลือกที่ต้องการเพิ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 หน้าต่างการทำงานของฟังก์ชัน Config

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงานสหกิจศึกษา

การพัฒนาแอปพลิเคชันทดสอบการยืนยันตัวบุคคลบนระบบปฏิบัติการแอนดรอยด์นั้น ประสบผลสำเร็จตามที่ได้รับมอบหมาย โดยสามารถอำนวยความสะดวกให้กับผู้ใช้งานซึ่งก็คือเทสเตอร์ของกลุ่มนักพัฒนาที่ทำการทดสอบระบบ Single Sign-On (SSO) ได้เป็นอย่างดี โดยแอปพลิเคชันใหม่นี้มีการใช้งานที่ง่ายและรวดเร็วเมื่อเทียบกับแอปพลิเคชันเดิม เช่น การออกแบบส่วนติดต่อผู้ใช้แบบใหม่นั้นมีความสวยงามมากขึ้น มองแล้วรู้สึกสบายตา, ฟังก์ชัน Profile ที่ช่วยลดระยะเวลาที่ต้องไปกรอกฟอร์มใหม่ทุกครั้ง, ฟังก์ชัน Save History ที่ช่วยให้ทราบว่าผู้ใช้ได้ส่งคำสั่ง Test case อะไรไปแล้วบ้าง และฟังก์ชัน Config Database ที่ช่วยให้ผู้ใช้งานทุกคนสามารถแก้ไขข้อมูลฟอร์มได้สะดวก ทำให้การใช้งานโดยรวมดีขึ้นมากถ้าเทียบกับแอปพลิเคชันเดิม

5.2 ข้อเสนอแนะในการพัฒนา

1) เพื่อความสะดวกยิ่งขึ้นในการใช้งานกรณีหน้าต่างแต่ละหน้ามีรูปแบบคล้ายกันมาก ในบางครั้งจำเป็นจะต้องกลับไปอ่านหัวข้อในหน้าต่างใหม่อีกครั้ง เพราะฉะนั้นแต่ละหน้าต่างแต่ละฟังก์ชันควรมีรูปแบบที่ต่างกันและเข้าใจง่าย

2) หน้ารายการ Save History ควรมีปุ่มเลือกเรียงลำดับ เช่น เรียงตามวันจากล่าสุดไปนานสุด หรือเรียงตามชื่อของชุดคำสั่ง Test case เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] Ice Suntisuk. **Single Sign On คืออะไร.** [Online].
Available : http://msgconsultant.com/articledetail.php?cid=25&id=917.
สืบค้นเมื่อ 3 กรกฎาคม 2560

[2] TiMeFF. **Progressive Web App คืออะไร.** [Online].
Available : https://medium.com/@timeff/โอ้เย-progressive-web-app-มาลองแบบง่ายๆ-วันเดียวเสร็จ-b7a2be52ae5b
สืบค้นเมื่อ 18 ธันวาคม 2560

[3] นายทศพล ต้นสมบัติ. **ระบบปฏิบัติการ ANDROID.** [Online].
Available : https://beerkung.wordpress.com/ระบบปฏิบัติการรุ่นล่าสระบบปฏิบัติการ-android/
สืบค้นเมื่อ 18 ธันวาคม 2560

[4] ArcTech. **ทำความรู้จักกับ Android Studio.** [Online].
Available : https://www.arctech.in.th/articles/125
สืบค้นเมื่อ 18 ธันวาคม 2560

[5] Jirawatee. **รู้จัก Firebase Realtime Database ตั้งแต่ Zero จนเป็น Hero.** [Online].
Available : https://medium.com/@jirawatee/รู้ จั ก -firebase-realtime-database-ตั้งแต่-zero-จนเป็น-hero-5d09210e6fd6
สืบค้นเมื่อ 10 มิถุนายน 2560

[6] Jirawatee. **รู้จัก Firebase Authentication ตั้งแต่ Zero จนเป็น Hero.** [Online].
Available : https://medium.com/@jirawatee/รู้จัก-firebase-authentication-ตั้งแต่-zero-จนเป็น-hero- 7dd5839d3588
สืบค้นเมื่อ 18 มิถุนายน 2560

[7] ไม่ปรากฏชื่อผู้แต่ง. **Java คืออะไร.** [Online].
Available : http://nwannika.tripod.com/java/Chapter1.htm
สืบค้นเมื่อ 18 ธันวาคม 2560

[8] Pakin Phuhinkong. **Git คืออะไร ... Git is your friend.** [Online].
Available : https://medium.com/@pakin/git-คืออะไร-git-is-your-friend-609c5f8efea
สืบค้นเมื่อ 5 มิถุนายน 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง (ต่อ)

[9] ไม่ปรากฏชื่อผู้แต่ง. **รู้จัก Material Design แนวทางการออกแบบ UI ของกูเกิลในยุคต่อไป.** [Online].

Available : https://www.blognone.com/node/57820

สืบค้นเมื่อ 18 ธันวาคม 2560

[10] Krishdiphong Krz. **Golden Ratio หรือ สัดส่วนทอง คืออะไรหยอ?.** [Online].

Available : https://medium.com/imkrz/golden-ratio-หรือ-สัดส่วนทอง-คืออะไรหยอ-994fde7da56a

สืบค้นเมื่อ 21 กันยายน 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก วิธีการติดตั้งซอฟต์แวร์ในการพัฒนาระบบ

การติดตั้งซอฟต์แวร์ในการพัฒนาแอปพลิเคชันทดสอบการยืนยันตัวบุคคลมีขั้นตอนดังต่อไปนี้

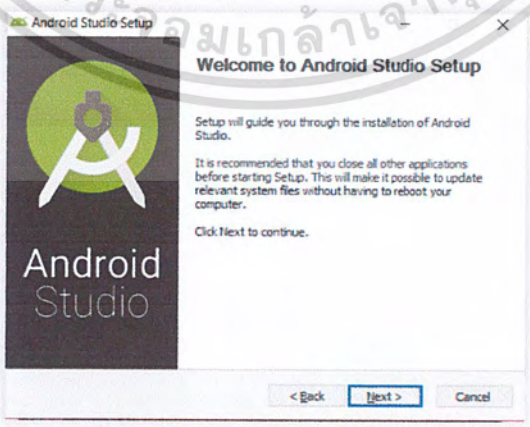
ก.1 การติดตั้ง Android Studio

เป็นโปรแกรมที่ใช้สำหรับพัฒนาแอปพลิเคชันแอนดรอย์ มีขั้นตอนการติดตั้งดังนี้

1. เข้าเว็บ https://developer.android.com/studio/

รูปที่ ก.1 หน้าเว็บโหลดโปรแกรม Android Studio

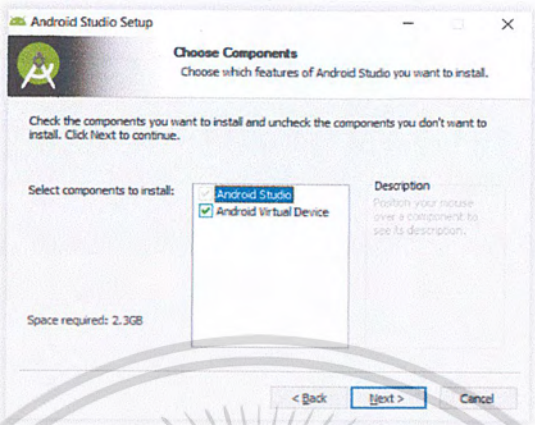
2. หลังจากอ่านรายละเอียดการติดตั้ง กด Next เพื่อเข้าสู่ขั้นตอนถัดไป



รูปที่ ก.2 หน้าแสดงรายละเอียดก่อนติดตั้ง

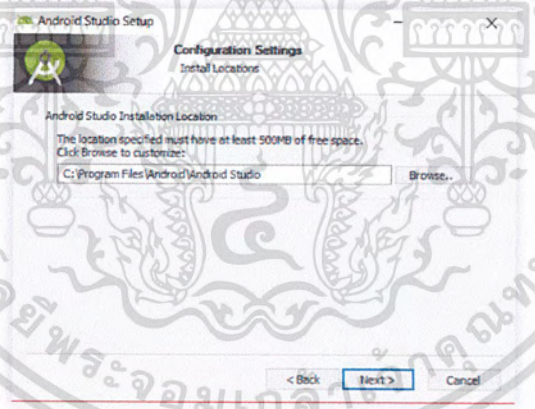
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลือกส่วนประกอบที่ต้องการ หลังจากนั้น กด Next เพื่อเข้าสู่ขั้นตอนถัดไป



รูปที่ ก.3 หน้าแสดงตัวเลือกส่วนประกอบที่จะติดตั้ง

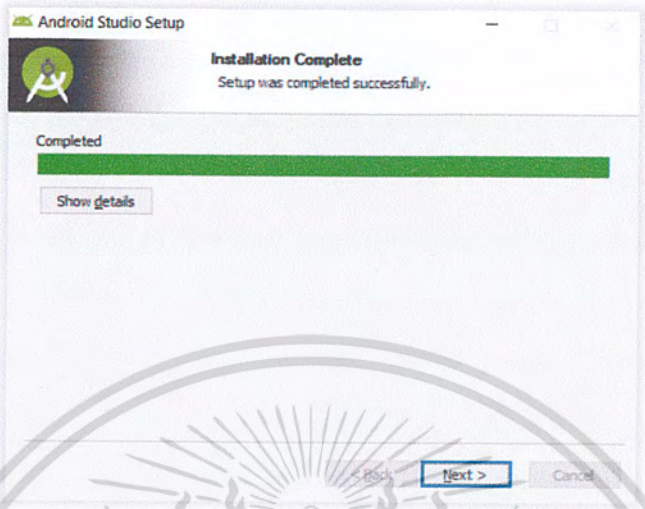
4. เลือกตำแหน่งที่ต้องการจะติดตั้งไฟล์โปรแกรม หลังจากนั้น กด Next เพื่อเข้าสู่ขั้นตอนติดตั้ง



รูปที่ ก.4 หน้าแสดงตำแหน่งการติดตั้งไฟล์โปรแกรม

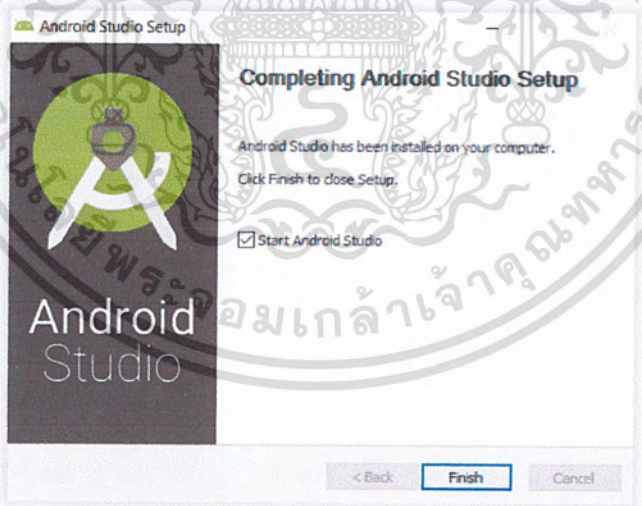
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. รอจนโปรแกรมติดตั้งเสร็จ หลังจากนั้น กด Next เพื่อเข้าสู่ขั้นตอนถัดไป



รูปที่ ก.5 หน้าแสดงการติดตั้งโปรแกรม

6. กด Finish หรือ ติ๊กเลือก Start Android Studio เพื่อเริ่มสร้างโปรเจคทันที



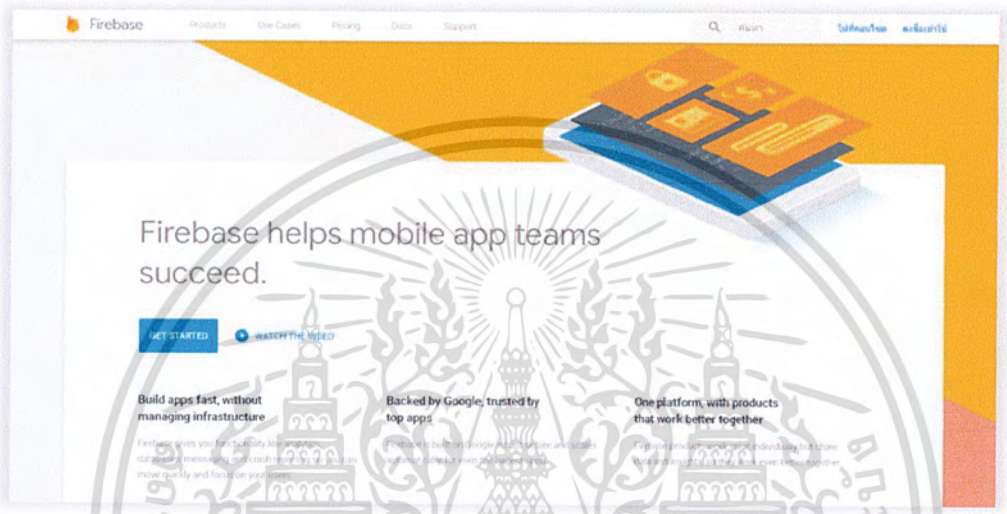
รูปที่ ก.6 หน้าแสดงขั้นตอนสุดท้ายของการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.2 การสร้างโปรเจคใน Firebase

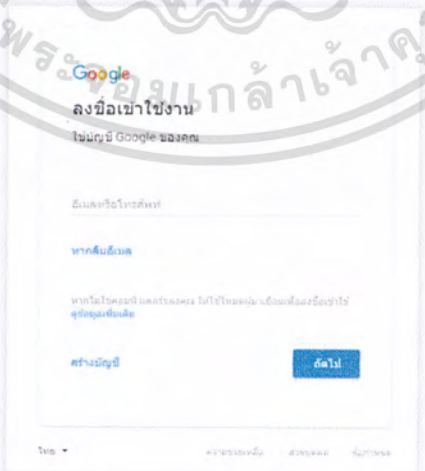
เป็นเทคโนโลยีที่ทางบริษัท Google เปิดเป็นเซอร์วิสให้ใช้บริการฟังก์ชันต่างๆ โดยมีขั้นตอนการสร้างโปรเจคดังนี้

1. เข้าเว็บ https://firebase.google.com/ และกด GET STARTED เพื่อเข้าใช้งาน



รูปที่ ก.7 หน้าเว็บของ Firebase

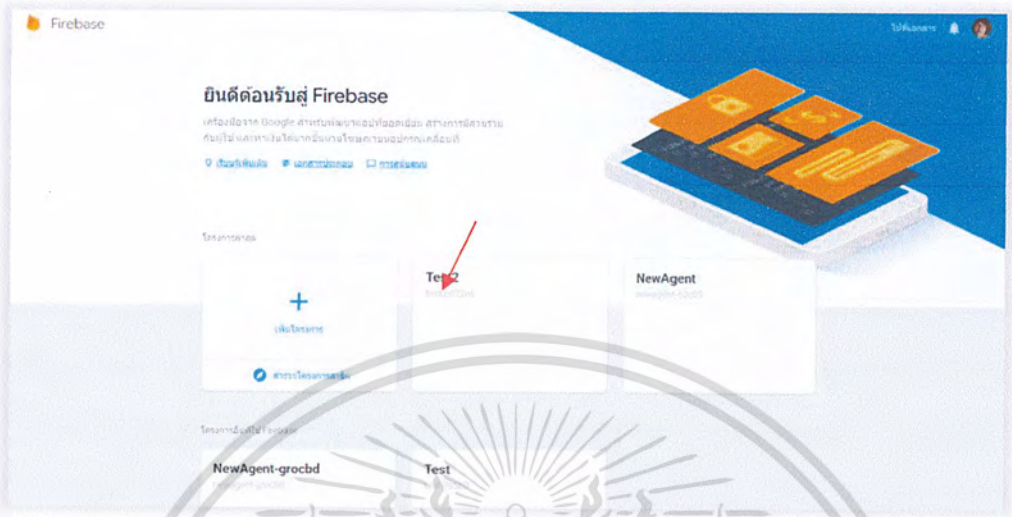
2. ผู้ใช้งานต้องเข้าสู่ระบบด้วย Gmail



รูปที่ ก.8 หน้าเว็บแสดงฟอร์มการเข้าสู่ระบบ

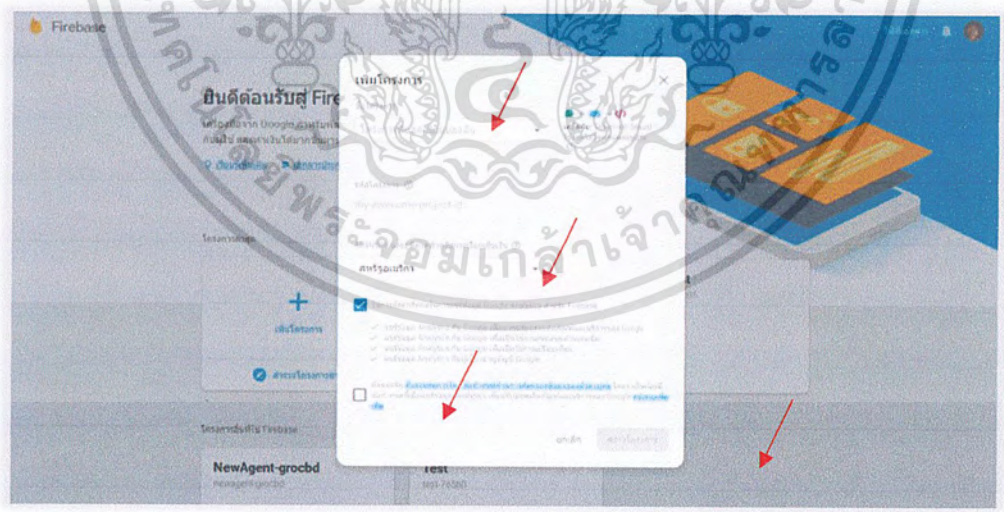
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. กด เพิ่มโครงการ เพื่อเริ่มต้นสร้างโปรเจค



รูปที่ ก.9 หน้าเว็บแสดงโปรเจคทั้งหมดของผู้ใช้งาน

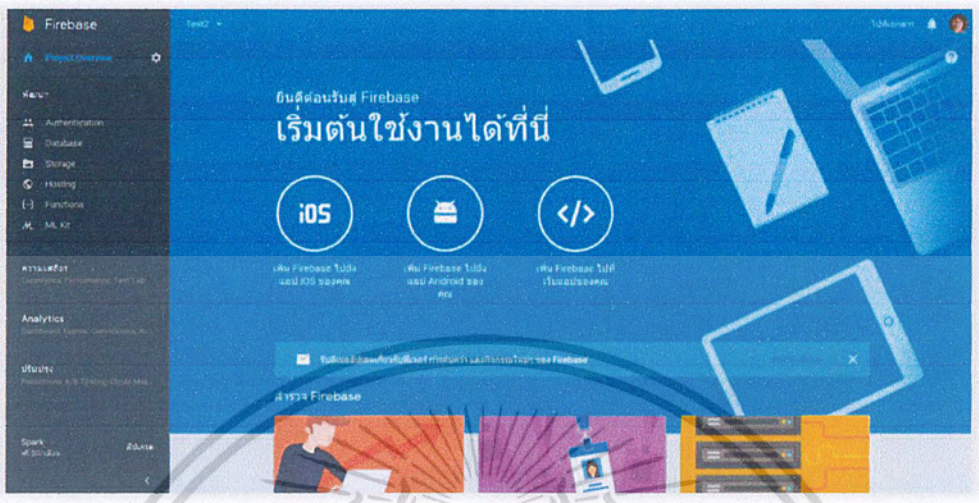
4. กรอกรายละเอียดของโปรเจคทั้งหมดในฟอร์ม พร้อมทั้งติ๊กเครื่องหมายถูกด้านล่าง และกด สร้างโครงการ



รูปที่ ก.10 หน้าเว็บแสดงฟอร์มกรอกข้อมูลของโปรเจค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. จะเข้าสู่หน้าการใช้งานเป็นที่เรียบร้อย โดยมีฟังก์ชันของ Firebase อยู่แถบเมนูด้านซ้าย



รูปที่ ก.11 หน้าเว็บแสดงหน้าหลัก Firebase Console

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้