**โปรแกรมเช็คชื่อนักศึกษาผ่านโทรศัพท์มือถือ**

**MOBILE PHONE CLASS CHECKING**

**นายณัฐพงศ์ สุริยะฉาย**

**นายเนติวิทย์ แก้วทอง**

**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรเทคโนโลยีบัณฑิต**

**สาขาวิชาเทคโนโลยีสารสนเทศ ภาควิชาเทคโนโลยีสารสนเทศ**

**คณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ**

**ปีการศึกษา 2560**

**ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ**

โปรแกรมเช็คชื่อนักศึกษาผ่านโทรศัพท์มือถือ

MOBILE PHONE CLASS CHECKING

นายณัฐพงศ์ สุริยะฉาย

นายเนติวิทย์ แก้วทอง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรเทคโนโลยีบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ ภาควิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ปีการศึกษา 2560

ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

MOBILE PHONE CLASS CHECKING

MR.NATTAPONG SURIYACHAY

MR.NETIWIT KEAWTHONG

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE BACHELOR’S DEGREE OF TECHNOLOGY IN INFORMATION TECHNOLOGY DEPARTMENT OF INFORMATION TECHNOLOGY

FACULTY OF INDUSTRIAL TECHNOLOGY AND MANAGEMENT

KING MONGKUT’S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK

2017

COPYRIGHT OF KING MONGKUT’S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK

**  
ใบรับรองปริญญานิพนธ์**

**คณะเทคโนโลยีและการจัดการอุตสาหกรรม**

**มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ**

เรื่อง โปรแกรมเช็คชื่อผ่านโทรศัพท์มือถือ

โดย นายณัฐพงค์ สุริยะฉาย

นายเนติวิทย์ แก้วทอง

ได้รับอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรเทคโนโลยีบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณบดี (ผู้ช่วยศาสตราจารย์ ดร.อนิราช มิ่งขวัญ)

คณะกรรมการสอบปริญญานิพนธ์

ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ขนิษฐา นามี)

กรรมการ

(อาจารย์นิติการ นาคเจือทอง)

กรรมการ

(อาจารย์วัชรชัย คงศิริวัฒนา)

ชื่อ : นายณัฐพงค์ สุริยะฉาย

นายเนติวิทย์ แก้วทอง

ชื่อปริญญานิพนธ์ : โปรแกรมเช็คชื่อนักศึกษาผ่านโทรศัพท์มือถือ

ภาควิชา : เทคโนโลยีสารสนเทศ

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ที่ปรึกษาปริญญานิพนธ์ : อาจารย์วัชรชัย คงศิริวัฒนา

ปีการศึกษา : 2560

**บทคัดย่อ**

ในปัจจุบัน สถานศึกษาต่าง ๆ เช่น โรงเรียน และมหาวิทยาลัย ในประเทศไทย ยังคงมีการ เช็คชื่อนักเรียน นักศึกษา เข้าชั้นเรียน เพื่อตรวจสอบจำนวนนักศึกษาที่เข้าชั้นเรียนว่ามีการขาดเรียน เข้าชั้นเรียนสาย หรือเข้าชั้นเรียนปกติ เพื่อใช้ในการพิจารณาสถานะของนักเรียน นักศึกษา ในการเข้าสอบ และคะแนนจิตพิสัยอยู่ ด้วยเหตุนี้ทางผู้จัดทำได้เล็งเห็นแนวทางในการประยุกต์เทคโนโลยีสารสนเทศ เข้ากับรูปแบบที่ใช้กันอยู่ในปัจจุบัน พัฒนาโปรแกรมเช็คชื่อนักศึกษาผ่านโทรศัพท์มือถือ เพื่อให้ง่ายและสะดวกต่อการตรวจสอบและพิจารณาของอาจารย์ผู้สอนวิชาต่าง ๆ ลักษณะการเช็คชื่อในปัจจุบันใช้วิธีการเรียกชื่อตามใบรายชื่อนักศึกษา และใช้เวลาขึ้นอยู่กับจำนวนของนักศึกษาในชั้นเรียน เพื่อแก้ปัญหาของการเช็คชื่อให้สะดวก และรวดเร็วจึงได้มีแนวความคิดการเช็คชื่อผ่านมือถือ ระหว่างอาจารย์ และนักศึกษาโดยใช้ความสามารถของ Smart phone ในปัจจุบัน

การทำงานของโปรแกรมจะแบ่งเป็นสองส่วน คือฝั่งอาจารย์ จะทำการรอรับการเข้าเรียน จากนักศึกษา ในต้นคาบเรียน และท้ายคาบเรียน ขณะที่นักศึกษาจะต้องทำการยืนยันตัวเองเช่นกัน ในช่วงเวลาที่อาจารย์เปิดโหมดสถานะรอการเช็คชื่อจากนักศึกษาโดยมีเงื่อนไขว่าโทรศัพท์มือถือของนักศึกษาจะต้องอยู่ภายในรัศมี ถึงโทรศัพท์มือถือของอาจารย์ ภายในระยะที่เหมาะสม เช่น 50 เมตร ผ่านการระบุตำแหน่ง ของ GPS จึงถือว่านักศึกษาเช็คชื่อสำเร็จ โปรแกรมเช็คชื่อเข้าชั้นเรียนนี้ใช้เทคนิคการพิสูจน์ตัวตนแทนนักศึกษาโดยใช้คุณสมบัติหมายเลข MAC address หรือ หมายเลข IMEI ของโทรศัพท์มือถือ ซึ่งมีอยู่แล้ว ทำให้มีการเช็คชื่อ ระหว่างอาจารย์และนักศึกษา ผ่านเครือข่าย Internet ผ่าน Cloud บนฐานข้อมูล Firebase ผลจากการทดสอบโปรแกรม สามารถลดระยะเวลาการเช็คชื่อลง และสามารถสรุปสถานะของนักศึกษา การเข้าชั้นเรียน การมาสาย การขาดเรียน รวมถึงดูรายงานในแบบสัปดาห์ แบบวันได้เป็นอย่างดี (โครงงานพิเศษมีจำนวนทั้งสิ้น 81 หน้า)

อาจารย์ที่ปรึกษาปริญญานิพนธ์

Name : Mr.Nattapong Suriyachay

Mr.Netiwit Keawyhong

Project Title : Mobile Phone Class Checking

Major Field : Information Technology

King Mongkut’s University of Technology North Bangkok

Project Advisor : Mr.Watcharachai Kongsiriwattana

Academic Year : 2017

**Abstract**

Nowadays, education system in Thailand still checks student attendance in classroom by calling out name during a class, particularly in primary, secondary and under graduation. The purpose of this checking is to examine student attendance as nonattendance, attendance, and late arrival. These status is considered as affective score, which is also calculated in total scores. Therefore, it is feasible to merge information technology with current class attendance by implementing class checking via mobile phone or smartphone application. In order to achieve its objectives, this application is able to help lecturer examine and reduce time in class checking effectively.

Normally, classroom checking relies on student list from register office or system within university or school. To call out student name in classroom in short period of time, it is difficult and uncomfortable. Therefore, mobile phone class checking is proposed to deal with aforementioned problem. With capability of smartphone, it is easy to implement application being able to examine student status via mobile phone of teacher seamlessly. Class checking functions can be divided into two parts including teacher and student part. Teacher part always listens connection from student application at the start and end of class period. On the other hand, student part is required to identify their status within limited distance 100 meters via GPS. This method uses location based technique to locate distance between teacher and student. Provided that student is far from teacher over 100 meters, student status is incomplete for class attendance. Also, MAC address and IMEI number of mobile phone are used as student identification to update attendance status with teacher upon firebase cloud through the Internet.

Finally, mobile phone class checking can diminish period of time to check student status, show student report of appearance, nonattendance, and late arrival status in daily and weekly.

(Total 81 pages)

Project Advisor

**กิตติกรรมประกาศ**

การจัดทำโครงงานปริญญานิพนธ์นี้สำเร็จลุล่วงไปด้วยดี เนื่องจากได้รับความช่วยเหลือและความอนุเคราะห์จากบุคคลหลายท่าน อาจารย์วัชรชัย คงศิริวัฒนา อาจารย์ที่ปรึกษาโครงงานที่ให้ความดูแลและให้คำปรึกษาในการจัดทำทุกส่วนของโครงงานพิเศษ ตลอดจนคณาจารย์ประจำคณะเทคโนโลยีและการจัดการอุตสาหกรรมทุกท่านที่ได้ให้การสนับสนุนในทุกด้านมา ขอขอบพระคุณทุกท่านเป็นอย่างสูง

ขอรำลึกและสำนึกในพระคุณบิดามารดา และครูบาอาจารย์ที่ได้สั่งสอนข้าพเจ้า จนได้มาศึกษา ณ มหาวิทยาลัยแห่งนี้ ขอขอบคุณทุกกำลังใจจากเพื่อน ที่มีส่วนในการจัดทำโครงงานปริญญานิพนธ์นี้สำเร็จลุล่วงไปด้วยดี

คณะผู้จัดทำจะขอจดจำสิ่งที่ดี ณ ที่แห่งนี้ไว้ ขอกราบขอบพระคุณ

ณัฐพงศ์ สุริยะฉาย

เนติวิทย์ แก้วทอง

**สารบัญ**

หน้า

บทคัดย่อภาษาไทย ข

บทคัดย่อภาษาอังกฤษ ค

กิตติกรรมประกาศ จ

สารบัญภาพ ซ

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของการจัดทำโครงงานพิเศษ 1

1.3 ขอบเขตของการทำปัญหาพิเศษ 1

1.4 ผลที่คาดว่าจะได้รับ 6

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 7

2.1 การพิสูจน์ตัวตน (Authentication) 7

2.2 Mac Address Authentication 8

2.3 IMEI 8

2.4 GPS (Global Positioning System) 9

2.5 A-GPS (Assisted GPS) 10

2.6 Location Based Service (LBS) 13

2.7 การจัดเก็บข้อมูลแบบ NoSQL และการส่งข้อมูล 11

2.8 Firebase 13

2.9 Nightmare JS 14

2.10 Node.js 14

2.11 Platform as a Service (PaaS) 15

2.12 Google Gmail 15

บทที่ 3 ขั้นตอนการพัฒนาและการออกแบบ 17

3.1 การวิเคราะห์ปัญหา 17

3.2 การกำเนินงาน 17

3.3 การวิเคราะห์ถึงระบบงาน 21

3.4 Use Case Diagram 23

3.5 Class Method Java 24

**สารบัญ (ต่อ)**

หน้า

3.6 การเก็บข้อมูลเพื่อใช้เชื่อมต่อและควบคุมฐานข้อมูล 25

3.7 การออกแบบหน้าจอเพื่อติดต่อกับผู้ใช้งาน 28

3.7 Code การทำงานแต่ละส่วน 42

บทที่ 4 ผลการดำเนินงาน 50

4.1 ผลจากการดำเนินงาน 50

4.2 รูปแบบหน้าจอที่ได้ทำการออกแบบและนำไปพัฒนา 51

บทที่ 5 สรุปผลการดำเนินงาน และข้อเสนอแนะ 59

5.1 การวิเคราะห์และสรุปผล 59

5.2 ปัญหาและอุปสรรค 60

5.3 ข้อเสนอแนะ 62

บรรณานุกรม 63

ภาคผนวก 64

ประวัติผู้จัดทำโครงการ 82

**สารบัญภาพ**

ภาพที่  หน้า

2-1 ภาพกระบวนการทำงานการพิสูจน์ตัวตนผ่าน Google 7

2-2 ภาพตัวอย่างการระบุตำแหน่งผ่านดาวเทียม GPS 9

2-3 ภาพกระบวนการส่งสัญญาณจากดาวเทียมระหว่าง A-GPS และ GPS 11

2-4 ภาพกระบวนการทำงาน Location Based Service (LBS) 12

2-5 ภาพกระบวนการทำงานพิสูจน์ตัวตนผ่านแอคเค้าท์ Google 16

3-1 แผนผังกระบวนการทำงานเพิ่มรายวิชาจากฐานข้อมูล Klogic 18

3-2 แผนผังกระบวนการเก็บข้อมูลและประมวลผลออกมาในรูปแบบรีพอร์ต(Report) 19

3-3 แผนผังกระบวนการการตรวจสอบสถานะรายชื่อนักศึกษา 20

3-4 Diagram รวมการทำงานของโปรแกรม 21

3-5 แผนผังแสดงการเก็บข้อมูลเพื่อจัดทำระบบ 22

3-6 รูปแบบUse Case Diagram 23

3-7 รูปแบบClass Method Java 24

3-8 รูปแบบการเก็บข้อมูลของ Child StudentUser 25

3-9 รูปแบบการเก็บข้อมูลของ Child register1:60 26

3-10 รูปแบบการเก็บข้อมูลของ Child teacher 26

3-11 รูปแบบการเก็บข้อมูลของ Child Semeter1:60 27

3-12 ตัวอย่างหน้าจอการยืนยันตัวตน 28

3-13 ตัวอย่างหน้าจอการยืนยันตัวตน (หลังจากกดปุ่ม Sing In) 28

3-14 ตัวอย่างหน้าจอเมนู และ ข้อมูลของ User ที่เกี่ยวข้อง 29

3-15 ตัวอย่างหน้าจอเลือกรายวิชา 30

3-16 ตัวอย่างหน้าจอแสดงปุ่มเปิดการเช็คชื่อ 30

3-17 ตัวอย่างหน้าจอแสดงการเช็คชื่อ(ต้นชั่วโมง) 31

3-18 ตัวอย่างหน้าจอแสดงการเช็คชื่อ(ท้ายชั่วโมง) 32

3-19 หน้าจอตัวอย่าง Alert เมื่อนักศึกษากดเช็คเชื่อครบทุกคนแล้ว 32

3-20 หน้าจอตัวอย่าง Alert ถ้าหมดเวลาเช็คชื่อ 33

3-21 ตัวอย่างหน้าจอเลือกรายวิชาเพื่อแก้ไข 33

**สารบัญภาพ (ต่อ)**

ภาพที่  หน้า

3-22 ตัวอย่างหน้าจอแก้ไข (ต่อ) 34

3-23 ตัวอย่างหน้าจอ Alert เมื่อกดปุ่มแก้ไข 34

3-24 ตัวอย่างหน้าจอเมื่อกดปุ่มเพิ่มรายชื่อนักศึกษา 35

3-25 หน้าจอ Alert เมื่อกดปุ่มเพิ่มรายชื่อ 35

3-26ตัวอย่างหน้าจอ Report 36

3-27 ตัวอย่างหน้าจอ Login ฝั่งนักศึกษา 36

3-28 หน้าจอการยืนยันตัวตน (หลังจากด Sing in) 37

3-29 ตัวอย่างหน้าจอเลือกวิชาเช็คชื่อ 37

3-30 ตัวอย่างหน้าจอเช็คชื่อครั้งแรก 38

3-31 ตัวอย่างหน้าจอเช็คชื่อ 38

3-32ตัวอย่าง Report วันที่ 01 กันยายน 25605 39

3-33 ตัวอย่าง Report ระหว่างเทอม 40

3-34 ตัวอย่าง Report 1 ภาคกี่ศึกษา 41

4-1 แสดงหน้าจอสำหรับเข้าสู่ระบบของ Google 52

4-2 แสดงหน้าจอแสดงเมนู 52

4-3 แสดงหน้าจอสำหรับเลือกรายวิชาที่อาจารย์รับผิดชอบ 53

4-4 หน้าจอสำหรับเลือกช่วงเวลาในการเช็คชื่อ 53

4-5 หน้าจอเช็คชื่อต้นชั่วโมงแสดงรายชื่อนักศึกษาที่ลงทะเบียนทั้งหมด 54

4-6 หน้าจอเช็คชื่อท้ายชั่วโมงแสดงรายชื่อนักศึกษาที่ลงทะเบียนทั้งหมด 54

4-7 หน้าจอ Report แสดงรายชื่อนักศึกษาทุกคนที่ลงทะเบียน 55

4-8 หน้าจอสำหรับเช็คชื่อ 55

4-9 หน้าจอสำหรับ Alert แสดงข้อความเมื่อเกิดเหตุขัดข้อง 56

4-10 หน้าจอสำหรับ Alert แสดงข้อความเมื่อเช็คชื่อสำเร็จ 56

4-11 หน้าจอสำหรับ Alert แสดงข้อความเมื่อตรวจสอบ Mac Address แล้วไม่ตรงกัน 57

4-12 หน้าจอสำหรับ Alert แสดงข้อความเมื่อตรวจสอบแล้วอยู่ห่างจากระยะที่กำหนด 57

5-1 ภาพกระบวนการทำงานในการ GET ข้อมูลจาก Klogic 61

5-2 ภาพกระบวนการทำงานภายใน Middleware 62

**สารบัญภาพ (ต่อ)**

ภาพที่  หน้า

ก-1 เว็บไซต์ Firebase 65

ก-2 เว็บไซต์ Firebase (ต่อ) 65

ก-3 เริ่มการสร้างโปรเจค Firebase 66

ก-4 กรอกชื่อโปรเจคที่ต้องการ 66

ก-5 แสดงหน้าจอเริ่มต้นการใช้งานบน Firebase 67

ก-6 กรอกชื่อแพ็กเกจโปรเจคที่ต้องการผูกกับ Firebase 67

ก-7 แสดงตำแหน่งชื่อ Packeage จะอยู่บรรทัดที่ 1 เสมอ(.java) 68

ก-8 แสดงเมนูการทำงานวิธีการดู SHA-1 บน android 68

ก-9 แสดงการทำงานของเมนู singingReport 69

ก-10 แสดง Codeให้เพิ่มลง โปรเจค Android Studio ของเรา 69

ก-11 แสดงจอการเข้าสู่เมนู Export Signed APK 70

ก-12 แสดงการสร้างหรือเปิดใช้งานไฟล์ Key 70

ก-13 แสดงการสร้างหรือเปิดใช้งานไฟล์ Key (ต่อ) 71

ก-14 แสดงการสร้างหรือเปิดใช้งานไฟล์ Key (ต่อ) 71

ก-15 แสดงหน้าจอการสร้าง APK 72

ก-16 แสดงขั้นตอนสุดท้ายของการตั้งค่า APK 72

ก-17 แสดงที่อยู่ของ ไฟล์ APK ที่สร้างไว้ 73

ก-18 แสดงเว็บไซต์ Klogic 73

ก-19 แสดงการเข้าใช้งานด้วยรหัสของนักศึกษา 74

ก-20 แสดงการเข้าใช้งานด้วยรหัสของอาจารย์ 74

**บทที่ 1**

**บทนำ**

**1.1 ความเป็นมาและความสำคัญของปัญหา**

ปัจจุบันระบบการศึกษาภายในประเทศไทยยังมีการเช็คชื่อนักศึกษาเพื่อเป็นการบันทึก การเข้าเรียน วิธีที่ใช้กันอยู่ในมหาวิทยาลัยคือ การบันทึกการเข้าเรียนของนักศึกษาโดยการเรียกชื่อนักศึกษา แล้วให้นักศึกษาขานรับ แต่วิธีนี้ค่อนข้างไม่สะดวกและใช้เวลาเนิ่นนาน ซึ่งในแต่ละคลาสมีนักศึกษาเป็นจำนวนมากทำให้การใช้วิธีขานชื่อที่มีนักศึกษามากกว่า 30 คน เป็นไปอย่างล่าช้าซึ่ง ทางผู้จัดทำได้เล็งเห็นถึงปัญหาดังกล่าว จึงศึกษาปัญหาและจัดหาวิธีการทางเลือกโดยประยุกต์ใช้เทคโนโลยีสารสนเทศเพื่อจัดทำระบบเช็คชื่อนักศึกษาผ่านโทรศัพท์มือถือ ซึ่งนักศึกษาทุกคนมีติดตัวอยู่แล้ว ระบบดังกล่าวจะช่วยลดระยะเวลาการเช็คชื่อและสามารถออก รายงานสรุปสถานะของนักศึกษาเก็บลงในฐานข้อมูล และรายงานผลออกมาให้อาจารย์สามารถพิจารณาผลการเช็คชื่อและเงื่อนไขการเข้าชั้นเรียนในคลาสนั้น ๆ ได้ง่าย และเป็นระบบมากยิ่งขึ้น

* 1. **วัตถุประสงค์ของการจัดทำโครงงานพิเศษ**

1.2.1 เพื่อลดระยะเวลาในการเช็คชื่อนักศึกษาของอาจารย์

1.2.2 เพื่ออำนวยความสะดวกในการสรุปสถานะของนักศึกษาผ่านรายงานประจำปีการศึกษา

1.2.3 เพื่อให้นักศึกษาสามารถยืนยันตัวผ่านโทรศัพท์มือถือในการเข้าชั้นเรียน

1.2.4 เพื่อประยุกต์การใช้ Mac Address และ IMEI ในโทรศัพท์มือถือ (Smartphone) ในการแสดงตัวตนขอนักศึกษา

**1.3 ขอบเขตของการจัดทำปัญหาโครงงานพิเศษ**

ภาคเรียนที่ 1/2559

* + 1. โปรแกรมมีการทำงานในส่วนของผู้ใช้งานสามารถแบ่งประเภทของผู้ใช้งาน
       1. อาจารย์

1.3.1.1.1 อาจารย์สามารถเข้าสู่ระบบโดยจำเป็นต้องใช้ E-mail ของทางคณะ และ สามารถแสดงรายละเอียดที่เกี่ยวข้องกับอาจารย์ได้ เช่น ชื่อ นามสกุล

1.3.1.1.2 สามารถดูรายชื่อของนักศึกษาที่ลงทะเบียนในรายวิชาต่าง ๆ ได้

1.3.1.1.3 ฐานข้อมูลสามารถดึงข้อมูลรายวิชาที่อาจารย์รับผิดชอบจาก ระบบทะเบียนการศึกษา Klogic ได้

1.3.1.1.4 สามารถเช็คชื่อนักศึกษาในคลาสได้ว่าเข้าเรียนปกติ เข้าเรียนสาย และขาดเรียน

1.3.1.1.5 สามารถเปิด/ปิดการเช็คชื่อได้ ในกรณีที่นักศึกษาเช็คชื่อครบ ตามรายชื่อที่มีอยู่ในรายวิชานั้น โปรแกรมจะส่ง Alert แจ้งเตือนมายังอาจารย์แสดงข้อความ “นักศึกษามาเรียนครบทุกคนแล้ว” เพื่อให้อาจารย์รับรู้และกดปุ่มยืนยันเพื่อปิดการเช็คชื่อ

* + - 1. นักศึกษา

1.3.1.2.1 นักศึกษาสามารถเข้าสู่ระบบโดยใช้ E-mail ของทางคณะ และ สามาแสดงรายละเอียดที่เกี่ยวข้องกับนักศึกษาได้ เช่น ชื่อ-นามสกุลของนักศึกษา รหัสนักศึกษา

1.3.1.2.2 ในกรณีเช็คชื่อไปแล้วจะมีการแสดงสถานะผ่านข้อความว่าสำเร็จหรือไม่สำเร็จ หากสำเร็จจะมีข้อความ “ระบบบันทึกการเช็คชื่อเรียบร้อยแล้ว” หากไม่สำเร็จ จะมีข้อความ “การเช็คชื่อไม่สำเร็จ โปรดทำรายการอีกครั้ง”

1.3.1.2.3 หากอาจารย์เปิดระบบการเช็คชื่อนักศึกษาสามารถเช็คชื่อได้ แต่ในกรณีที่อาจารย์ไม่เปิดระบบการเช็คชื่อ ระบบจะไม่ทำการบันทึกการเช็คชื่อของนักศึกษา รวมถึงต้องอยู่ภายใต้เงื่อนไขของระยะทางที่กำหนด

1.3.1.2.4 ในขั้นตอนของการเช็คชื่อ จะปรากฏหน้าจอแสดงรายวิชาทั้งหมดที่ได้ลงทะเบียน หลังจากทำการเลือกรายวิชาจะมีข้อความเตือนให้ยืนยันเพื่อทำการผูก Mac Address และ IMEI ดังนั้นเมื่อทำการผูก Mac Address และ IMEI ของโทรศัพท์มือถือ เข้ากับระบบ นักศึกษาต้องทำการยืนยันตนเองผ่านปุ่มยืนยัน ซึ่งจะทำเพียงครั้งเดียวเท่านั้นในแต่ละภาคการศึกษา

1.3.1.2.5 โปรแกรมสามารถแสดงรายงานผลการเข้าชั้นเรียนในรูปแบบของ Report ของการเช็คชื่อในแต่ละสัปดาห์ได้

1.3.2 ในกระบวนการเช็คชื่อ โปรแกรมมีการตรวจสอบระยะทางผ่านระบบ GPS โดยนักศึกษาจำเป็นที่จะต้องอยู่ในระยะไม่เกิน 100 เมตร หรือระยะที่กำหนด จากตำแหน่ง GPS ของอาจารย์ ระบบจึงจะบันทึกการเช็คชื่อของนักศึกษาได้ (นักศึกษาและอาจารย์ต้องเปิด GPS เสมอ)

1.3.3 ในส่วนของการเข้าสู่ระบบเข้าใช้งานแอปพลิเคชั่นจะใช้ E-mail ของทางคณะ

1.3.4 ฐานข้อมูลสามารถดึงข้อมูลจากระบบทะเบียน Klogic ได้ จากทั้งฝั่งนักศึกษา และ ฝั่งอาจารย์ โดยมีรายละเอียด ดังนี้

1.3.4.1 อาจารย์

1.3.4.1.1 ชื่อรายวิชาที่รับผิดชอบ

1.3.4.1.2 รหัสรายวิชาที่รับผิดชอบ

1.3.4.1.3 ตอนเรียนที่รับผิดชอบ

1.3.4.1.4 รายชื่อของนักศึกษาที่อยู่ในรายวิชานั้น ๆ

1.3.4.2 นักศึกษา

1.3.4.2.1 ชื่อรายวิชาที่ลงทะเบียน

1.3.4.2.2 รหัสรายวิชาที่ลงทะเบียน

1.3.4.2.3 ตอนเรียนที่ลงทะเบียน

1.3.5 จัดทำ Mobile Application บนระบบปฏิบัติการ Android

1.3.5.1 อาจารย์

1.3.5.1.1 หน้าจอ Login ก่อนเข้าใช้งาน Application

1.3.5.1.2 หน้าจอหลัก

1.3.5.1.3 แสดงรายละเอียดของอาจารย์

1.3.5.1.3.1 ชื่อ – นามสกุล

1.3.5.1.5 แสดงรายวิชาที่รับผิดชอบ

1.3.5.1.5.1 ชื่อวิชา

1.3.5.1.5.2 รหัสวิชา

1.3.5.1.5.3 ตอนเรียน

1.3.5.1.6 หน้าจอเช็คชื่อต้นชั่วโมง

1.3.5.1.7 หน้าจอเช็คชื่อท้ายชั่วโมง

1.3.5.2 นักศึกษา

1.3.5.2.1 หน้าจอ Login ก่อนเข้าใช้งาน Application

1.3.5.2.2 หน้าจอหลัก

1.3.5.2.3 แสดงรายละเอียดของนักศึกษา

ชื่อ – นามสกุล

รหัสนักศึกษา

1.3.5.2.4 แสดงรายวิชาที่ลงทะเบียน

1.3.5.2.4.1 ชื่อวิชา

1.3.5.2.4.2 รหัสวิชา

1.3.5.2.4.3 ตอนเรียน

1.3.5.2.5 แสดงสถิติการเช็คชื่อของนักศึกษา

1.3.5.2.5.1 เข้าเรียนกี่ครั้ง

1.3.5.2.5.2 สายกี่ครั้ง

1.3.5.2.5.3 ขาดเรียนกี่ครั้ง

1.3.5.2.6 หน้าจอเช็คชื่อ

ภาคเรียนที่ 2 / 2560

1.3.6 โปรแกรมมีการทำงานในส่วนของผู้ใช้งานสามารถแบ่งประเภทของผู้ใช้งาน

* + - 1. อาจารย์

1.3.6.1.1 สามารถกำหนดและแก้ไข เงื่อนไขของการเข้าเรียนสาย ตัวอย่างเช่น สาย 3 ครั้ง เท่ากับ ขาดเรียน 1 ครั้ง หรือ สาย 1 ครั้ง เท่ากับ หัก 1 คะแนน เป็นต้น

1.3.6.1.2 สามารถกำหนดและแก้ไข เงื่อนไขของการขาดเรียน ตัวอย่างเช่น ขาดเรียน 3 ครั้ง หมดสิทธิ์สอบ

1.3.6.1.3 ในกรณีที่นักศึกษาไม่สามารถเช็คชื่อได้หรือมีเหตุจำเป็นที่จะต้องขาดเรียนหรือออกจากห้องเรียนก่อนหมดเวลาอาจารย์สามารถกำหนดหมายเหตุให้นักศึกษา ดังนี้

1.3.6.1.3.1 ลาป่วย

1.3.6.1.3.2 ลืมนำโทรศัพท์มา

1.3.6.1.3.3 GPS ไม่สามารถใช้งานได้

1.3.6.1.3.4 ทำกิจกรรม

1.3.6.1.3.5 ทำธุระส่วนตัว

1.3.6.1.3.6 อื่น ๆ

1.3.7 ในกรณีที่นักศึกษาเปลี่ยนเครื่องโทรศัพท์จำเป็นจะต้องให้อาจารย์แก้ไข Mac address ของนักศึกษาคนนั้น

1.3.8 ในกรณีที่นักศึกษาไม่มีรายชื่ออยู่ในระบบของทะเบียนนักศึกษา Klogic อาจารย์สามารถเพิ่มรายชื่อของนักศึกษาลงไปในรายวิชานั้น ๆ ได้ด้วยตนเอง

1.3.9 ในกรณีมีรายวิชาทีสอนครบ 16 สัปดาห์หรือมีเหตุจำเป็นจะต้องยกเลิกการสอนวิชานั้นอาจารย์สามารถลบรายวิชาของตนเองออกจากฐานข้อมูลได้

* + 1. โปรแกรมจะอ้างอิงวันและเวลาตามสัปดาห์ที่สอน

1.3.11.1 สัปดาห์ที่

* + - 1. วัน/เดือน/ปี

1.3.12 โปรแกรมสามารถตรวจสอบสถานะนักศึกษาหลังจากวันหมดเขตการลงทะเบียนโดยจะอัพเดตทุกอาทิตย์

* + - 1. วันที่ 4 สิงหาคม (ภาคเรียนที่ 1)
      2. วันที่ 6 มกราคม (ภาคเรียนที่ 2)

1.3.13 โปรแกรมจะบันทึกใน report ในสัปดาห์ที่มีการสอบโดย แบ่งเป็นสองช่วง

* + - 1. สอบกลางภาค ในสัปดาห์ ที่ 8
      2. สอบปลายภาค ในสัปดาห์ ที่ 16

1.3.14 โปรแกรมมีตัวเลือกเช็คชื่อในวันที่สอนชดเชยเพื่อเช็คชื่อในสัปดาห์ที่เลื่อนการสอนหรือยกคลาส

1.3.15 จัดทำ Mobile Application บนระบบปฏิบัติการ Android

* + - 1. อาจารย์

1.3.13.3.1 หน้าจอสอนชดเชย

1.3.13.3.2 หน้าจอเพิ่มนักศึกษา

1.3.13.3.3 หน้าจอ Report

1.3.16 จัดทำ Mobile Application บนระบบปฏิบัติการ iOS

1.3.16.1 อาจารย์

1.3.16.1.1 หน้าจอ Login ก่อนเข้าใช้งาน Application

1.3.16.1.2 หน้าจอหลัก

1.3.16.1.3 แสดงรายละเอียดของอาจารย์

1.3.16.1.3.1 ชื่อ – นามสกุล

1.3.16.1.4 แสดงรายวิชาที่รับผิดชอบ

1.3.16.1.4.1 ชื่อวิชา

1.3.16.1.4.2 รหัสวิชา

1.3.16.1.4.3 ตอนเรียน

1.3.16.1.4.4 วันเวลา

1.3.16.1.5 หน้าจอเช็คชื่อต้นชั่วโมง

1.3.16.1.6 หน้าจอเช็คชื่อท้ายชั่วโมง

1.3.16.1.7 หน้าจอสอนชดเชย

1.3.13.1.8 หน้าจอเพิ่มนักศึกษา

1.3.16.1.9 หน้าจอ Report

1.3.16.2 นักศึกษา

1.3.16.2.1 หน้าจอ Login ก่อนเข้าใช้งาน Application

1.3.16.2.2 หน้าจอหลัก

1.3.16.2.3 แสดงรายละเอียดของนักศึกษา

1.3.16.2.3.1 ชื่อ – นามสกุล

1.3.16.2.3.2 รหัสนักศึกษา

1.3.16.2.4 แสดงรายวิชาที่ลงทะเบียน

1.3.16.2.4.1 ชื่อวิชา

1.3.16.2.4.2 รหัสวิชา

1.3.16.2.4.3 ตอนเรียน

1.3.16.2.4.4 วันเวลา

1.3.16.2.5 แสดงสถิติการเช็คชื่อของนักศึกษา

1.3.16.2.5.1 สายกี่ครั้ง

1.3.16.2.5.2 ขาดเรียนกี่ครั้ง

1.3.16.2.6 หน้าจอเช็คชื่อ

1.3.17 โปรแกรมสามารถแสดงบันทึกรายการเช็คชื่อโดยจะแสดงทั้งหมด 3 ช่วง ได้แก่ รายสัปดาห์ ระหว่างเทอม และรายเทอม มีรายละเอียด ดังนี้

* + - 1. ภาคเรียน
      2. ปีการศึกษา
      3. ตอนเรียน
      4. รหัสวิชา
      5. สาขา
      6. ชื่อนักศึกษา
      7. รหัสนักศึกษา
      8. สถานะ
      9. สถานะสรุป (ในระหว่างเทอม และ รายเทอม)
    1. . หมายเหตุ

**1.4 ผลที่คาดว่าจะได้รับ**

1.4.1 ลดการใช้เวลาในการเช็คชื่อนักศึกษาของอาจารย์

1.4.2 เพื่ออำนวยความสะดวกในการสรุปสถานะของนักศึกษาผ่านรายงานประจำปีการศึกษา

1.4.3. ได้รับ Application ที่ช่วยให้นักศึกษายืนยันตัวตนการเข้าคลาสเรียน ด้วยตนเอง

1.4.4 ได้รับความรู้การประยุกต์ใช้งาน Mac Address หรือ IMEI ภายในโทรศัพท์มือถือ (Smartphone) ในการแสดงตัวตนของนักศึกษา

**บทที่ 2**

**ทฤษฎีที่เกี่ยวข้อง**

* 1. **การพิสูจน์ตัวตน (Authentication)**

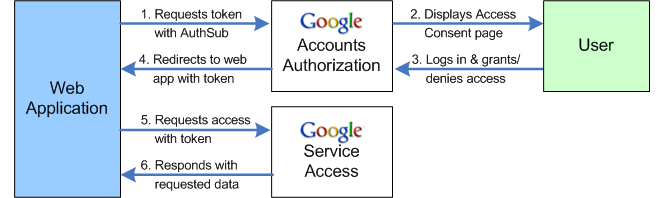
ความหมาย ขั้นตอนการยืนยันความถูกต้องของหลักฐาน (Identity) ที่แสดงว่าเป็นบุคคลที่กล่าว อ้างอิงจริง ในทางปฏิบัติจะแบ่งออกเป็น 2 ขั้นตอน คือ

2.2.1 การระบุตัวตน (Identification) คือ ขั้นตอนที่ผู้ใช้งานแสดงหลักฐานว่าตนเองคือใคร เช่น ชื่อผู้ใช้งาน (Username) และ รหัสผ่านเข้าใช้งาน (Password)

2.2.2 การพิสูจน์ตัวตน (Authentication) คือ ขั้นตอนที่ตรวจสอบหลักฐานเพื่อแสดงว่าเป็น บุคคลที่กล่าวอ้างจริง

การนำมาใช้งานในโปรแกรม

ได้นำการพิสูจน์ตัวตนมาใช้ในการยืนยันความถูกต้อง กระบวนการพิสูจน์ตัวตน ซึ่งขั้นตอนแรกผู้ใช้งานจะทำการแสดงหลักฐานที่ใช้ในการพิสูจน์ตัวตนต่อ โปรแกรม และในขั้นตอนต่อมาโปรแกรม จะทำการตรวจสอบหลักฐานที่ผู้ใช้นำมากล่าวอ้าง ซึ่งก็คือการพิสูจน์ ตัวตน หลังจากโปรแกรม ได้ทำการตรวจสอบหลักฐานเรียบร้อยแล้วถ้าหลักฐานที่นำมากล่าวอ้างถูกต้อง จึงจะอนุญาตให้ เข้าสู่โปรแกรม ได้ หากหลักฐานที่นำมากล่าวอ้างไม่ถูกต้องผู้ใช้งานจะถูกปฏิเสธจากโปรแกรม



**ภาพที่ 2-1** ภาพกระบวนการทำงานการพิสูจน์ตัวตนผ่าน Google

(ที่มา : https://developers.google.com/gdata/docs/auth/overview, 2560)

จากภาพที่ 2-1 จะเห็นว่าเริ่มแรกโปรแกรมจะร้องขอ Token จาก Google Account ที User เป็นคนส่งมาโดยการ Login ผ่านหน้า Application ตัวของ Google Auth จะส่งข้อมูลกลับไป แล้วขั้นตอนถัดไปเป็นการขอเข้าใช้ข้อมูลระหว่าง Application กับ Service ของ Google

**2.2 Mac Address Authentication**

Media Access Control คือค่าที่ใช้ในการอ้างอิงอุปกรณ์ที่ใช้ในระบบเครือข่าย ซึ่งค่าเหล่านี้จะถูกกำหนดมาตั้งแต่โรงงานที่ผลิต และในทางทฤษฎีแล้วในแต่ละอุปกรณ์จำเป็นจะต้องมีหมายเลข MAC Address ไม่ซ้ำกัน ค่า MAC Address จะประกอบด้วยตัวเลขฐาน 16 จำนวน 6 ชุด แบ่งตามเครื่องหมาย - หรือ : เช่น 01-23-45-67-89-ab หรือ 01:23:45:67:89:ab

การนำมาใช้งานในโปรแกรม

ได้นำหมายเลข Mac Address ที่อยู่ในอุปกรณ์มือถือมาผนวกเข้ากับกระบวนการพิสูจน์ตัวตนโดยกระบวนการนี้จะนำหมายเลข Mac Address จากโทรศัพท์มือถือของผู้ใช้นำมาเข้ากระบวนการขั้นตอนการเช็คชื่อเพื่อนำมาพิสูจน์ตัวตน โปรแกรมจะทำการตรวจสอบหมายเลข Mac Address ตรงกับ E-maill หรือไม่ หากถูกต้องโปรแกรมจะอนูญาติให้ผู้ใช้งานเช็คชื่อได้ แต่ถ้าหากหมายเลข ไม่ตรงกันกับ E-mail ของผู้ใช้งานโปรแกรมจะไม่อนุญาตให้เช็คชื่อได้

**2.3 IMEI**

International Mobile Equipment Identity เปรียบเสมือนเลขบัตรประชาชนของเครื่องโทรศัพท์มือถือ ถูกสร้างขึ้นโดยจะให้เป็นหมายเลขเฉพาะตัวของโทรศัพท์ในระบบ GSM เพื่อนำไปใช้ในการแก้ปัญหาเครื่องถูกขโมย โดยปกติโทรศัพท์มือถือทุกเครื่องจะมีหมายเลขอีมี่ ที่แตกต่างกันออกไปโดยไม่ซ้ำเลขกัน ซึ่งตัวเลขที่มีนั้นจะถูกเก็บอยู่ในหน่วยความจำของเครื่องโทรศัพท์แต่ละเครื่องโดยมีแบบตัวเลขฐาน 10 และสามารถทำการเช็กได้โดยกด \*#06# แล้วโทรออกจากนั้นหน้าจอโทรศัพท์มือถือของเราก็จะปรากฏหมายเลขอีมี่นี้ขึ้นมาเป็นจำนวน 15 หลักด้วยกัน ซึ่งโปรแกรมจะทำการคำนวณผ่านทางระบบคอมพิวเตอร์ แล้วจะทำการส่งข้อมูลเหล่านี้ไปยังตัวเครื่องโทรศัพท์มือถือเครื่องนั้น ๆ แล้วภายในเครื่องก็จะกำหนดข้อมูลภายในเอง ซึ่งจะอ่านเป็นตัวเลขอีมี่ 15 หลักนั่นเอง

การนำมาใช้งานในโปรแกรม

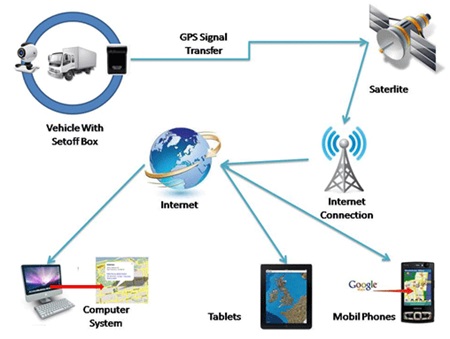
ได้นำหมายเลข IMEI ที่อยู่ในอุปกรณ์มือถือมาผนวกเข้ากับกระบวนการพิสูจน์ตัวตนโดยกระบวนการนี้จะนำหมายเลข IMEI จากโทรศัพท์มือถือของผู้ใช้นำมาเข้ากระบวนการขั้นตอนการเช็คชื่อเพื่อนำมาพิสูจน์ตัวตน โปรแกรมจะทำการตรวจสอบหมายเลข IMEI ตรงกับ E-maill และ หมายเลข Mac Address หรือไม่ หากถูกต้องโปรแกรมจะอนูญาติให้ผู้ใช้งานเช็คชื่อได้ แต่ถ้าหากหมายเลขไม่ตรงกันกับ E-mail และ Mac Address ของผู้ใช้งานโปรแกรมจะไม่อนุญาตให้เช็คชื่อได้

**2.4 GPS (Global Positioning System)**

ระบบระบุตำแหน่งบนพื้นโลก โดยการอ้างอิงจากดาวเทียมสามารถหาตำแหน่งได้ทุก ๆ แห่ง บนโลกซึ่งทำงานร่วมกับดาวเทียมบอกตำแหน่งทั้งหมด 24 ดวง ทำงานโดยการรับสัญญาณจากดาวเทียมแต่ละดวง โดยสัญญาณดาวเทียมนี้ประกอบไปด้วยข้อมูลที่ระบุตำแหน่งและเวลาขณะ ส่งสัญญาณ ตัวเครื่องรับสัญญาณ GPS จะต้องประมวลผลความแตกต่างของเวลาในการรับสัญญาณเทียบกับเวลาจริง ณ ปัจจุบันเพื่อแปรเป็นระยะทางระหว่างเครื่องรับสัญญาณกับดาวเทียมแต่ละดวง ซึ่งได้ระบุมีตำแหน่งของมันมากับสัญญาณดังกล่าวข้างต้น เพื่อให้เกิดความแม่นยำในการค้นหาตำแหน่งด้วยดาวเทียม ต้องมีดาวเทียมอย่างน้อย 4 ดวง เพื่อบอกตำแหน่งบนผิวโลก ซึ่งระยะห่างจากดาวเทียมทั้ง 3 กับเครื่อง GPS (ที่จุดสีแดง) จะสามารถระบุตำแหน่งบนผิวโลกได้ หากพื้นโลกอยู่ในแนวระนาบแต่ในความเป็นจริงพื้นโลกมีความโค้งเนื่องจากสัณฐานของโลกมีลักษณะกลม ดังนั้นดาวเทียมดวงที่ 4 จะทำให้สามารถคำนวณเรื่องความสูงเพื่อทำให้ได้ตำแหน่งที่ถูกต้องมากขึ้น

การนำมาใช้งานในโปรแกรม

กระบวนการทำงานของโปรแกรมจะดึงค่าตำแหน่ง ละติจูด ลองติจูด จากอุปกรณ์โทรศัพท์มือถือระบุตำแหน่งที่ผู้ใช้งานอยู่เพื่อนำไปเข้าสูตรคำนวณค้นหาระยะทางระหว่าง 2 ฝั่ง (ฝั่งอาจารย์และนักศึกษา) เพื่อเข้าสู่กระบวนการตรวจสอบเงื่อนไขการเช็คชื่อระยะไม่เกิน 100 เมตร หรือระยะที่กำหนด



**ภาพที่ 2-2** ภาพตัวอย่างการระบุตำแหน่งผ่านดาวเทียม GPS

(ที่มา : http://www.xn--12cg1cxchd0a2gzc1c5d5a.net/gps/, 2560)

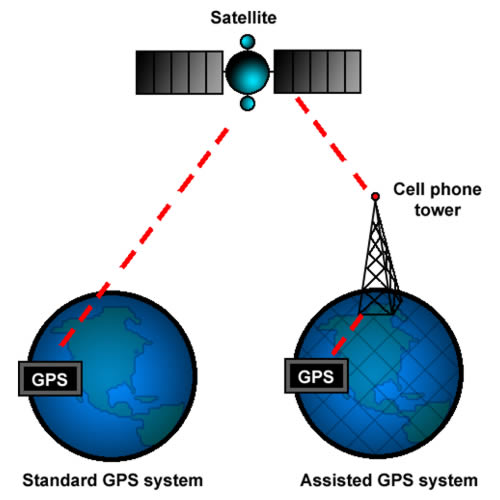
ภาพที่2-2 เป็นการสงสัญญาณจากดาวเทียมลงมายังต้นเสาสัญญาณและกระจายต่อไปยังอินเตอร์เน็ตยกตัวอย่างเมื่อมีอุปกรณ์เข้าไปใช้งานจะส่งช้อมูล GPS ของตำแหน่งที่อุปกรณ์นั้นใช้งานอยู่

**2.5 A-GPS (Assisted GPS)**

เป็นระบบที่ช่วยในการเพิ่มสมรรถภาพของ[จีพีเอส](https://th.wikipedia.org/wiki/%E0%B8%88%E0%B8%B5%E0%B8%9E%E0%B8%B5%E0%B9%80%E0%B8%AD%E0%B8%AA)ในช่วงการเริ่มต้นการใช้งาน เอจีพีเอสนิยมพบได้ใน[โทรศัพท์มือถือ](https://th.wikipedia.org/wiki/%E0%B9%82%E0%B8%97%E0%B8%A3%E0%B8%A8%E0%B8%B1%E0%B8%9E%E0%B8%97%E0%B9%8C%E0%B8%A1%E0%B8%B7%E0%B8%AD%E0%B8%96%E0%B8%B7%E0%B8%AD)และ[สมาร์ตโฟน](https://th.wikipedia.org/wiki/%E0%B8%AA%E0%B8%A1%E0%B8%B2%E0%B8%A3%E0%B9%8C%E0%B8%95%E0%B9%82%E0%B8%9F%E0%B8%99)ที่มีความสามารถจีพีเอสการทำงานทั่วไปของจีพีเอส จะใช้เวลาประมาณ 30-40 วินาทีในการเริ่มต้นระบุตำแหน่ง ซึ่งใช้ข้อมูลการโคจรของ[ดาวเทียม](https://th.wikipedia.org/wiki/%E0%B8%94%E0%B8%B2%E0%B8%A7%E0%B9%80%E0%B8%97%E0%B8%B5%E0%B8%A2%E0%B8%A1)ในการคำนวณตำแหน่งปัจจุบัน โดยมีอัตราการส่งสัญญาณอยู่ที่ 50 บิต/วินาที ซึ่งการดาวน์โหลดข้อมูลโคจรโดยตรงหลายครั้งจะกินเวลานานมากกว่านั้น ในระบบเอจีพีเอสจึงถูกนำมาใช้ โดยการทำงานผ่านทาง[เซิร์ฟเวอร์](https://th.wikipedia.org/wiki/%E0%B9%80%E0%B8%8B%E0%B8%B4%E0%B8%A3%E0%B9%8C%E0%B8%9F%E0%B9%80%E0%B8%A7%E0%B8%AD%E0%B8%A3%E0%B9%8C)ของระบบ โดยข้อมูลที่อยู่ในเซิร์ฟเวอร์เอจีพีเอสเป็นข้อมูลจากวงโคจรที่ถูกนำมาเก็บไว้ล่วงหน้าใน[ฐานข้อมูล](https://th.wikipedia.org/wiki/%E0%B8%90%E0%B8%B2%E0%B8%99%E0%B8%82%E0%B9%89%E0%B8%AD%E0%B8%A1%E0%B8%B9%E0%B8%A5) และเมื่อเครื่องมือเช่นโทรศัพท์มือถือที่ต้องการเชื่อมต่อ สามารถดาวน์โหลดข้อมูลเหล่านี้จากเซิร์ฟเวอร์โดยตรงผ่านทางคลื่นสัญญาโทรศัพท์มือถือ เช่น [GSM](https://th.wikipedia.org/wiki/GSM), [CDMA](https://th.wikipedia.org/w/index.php?title=CDMA&action=edit&redlink=1) หรือแม้แต่สัญญาณวิทยุอย่าง [Wi-Fi](https://th.wikipedia.org/wiki/Wi-Fi) ซึ่งข้อมูลเหล่านี้ปกติจะมีการดาวน์โหลดด้วยความเร็วที่สูงกว่า ทำให้ใช้เวลาในการรับข้อมูลได้เร็วยิ่งขึ้น

ตารางที่ 1 ความต่างระหว่าง A-GPS และ GPS

|  |  |  |
| --- | --- | --- |
|  | **A-GPS** | **GPS** |
| **Speed** | A-GPS จะกำหนดตำแหน่งพิกัดได้เร็วขึ้นเนื่องจากมีการเชื่อมต่อกับไซต์เซลล์ได้ดีกว่าการใช้ดาวเทียมโดยตรง | GPS อาจใช้เวลาหลายนาทีในการระบุตำแหน่งของพวกเขาเนื่องจากต้องใช้เวลานานในการสร้างการเชื่อมต่อกับดาวเทียม 4 ดวง |
| **Reliability** | ตำแหน่งที่กำหนดผ่านทาง A-GPS มีความแม่นยำน้อยกว่า GPS เล็กน้อย | อุปกรณ์ GPS สามารถกำหนดตำแหน่งพิกัดได้ไม่เกิน 1 เมตร |
| **Cost** | มีค่าใช้จ่าย เนื่องจากใช้อินเตอร์เน็ตในการค้นหาตำแหน่ง | GPS ไม่มีค่าใช้จ่ายสำหรับการเชื่อมต่อผ่านดาวเทียม (ฟรี) |
| **Usage** | โทรศัพท์มือถือ | รถยนต์, เครื่องบินเรือ, เรือ |



**ภาพที่ 2-3** ภาพกระบวนการส่งสัญญาณจากดาวเทียมระหว่าง A-GPS และ GPS

(ที่มา : http://news.siamphone.com/news-02905.html, 2560)

ภาพที่2-3 เป็นกระบวนการส่งข้อมูลรหะว่างดาวเทียมไป GPS และ A-GPS จะเห็นความแตกต่างอย่างชัดเจนโดย A-GPS จะมีเสาโทรศัพท์รับสัญญาณอีกทีก่อนที่จะส่งไปยัง อุปกรณ์แต่ GPS เป็นการรับตำแหน่งข้อมูลมาตรง ๆ

**2.6 Location Based Service (LBS)**

การบริการการบอกตำแหน่งทางภูมิศาสตร์ โดยใช้อุปกรณ์พกพา เช่น โทรศัพท์มือถือ PDA หรืออุปกรณ์ต่าง ๆ ผ่านสัญญาณเครือข่ายของผู้ให้บริการต่าง ๆ การให้บริการตำแหน่งที่อยู่นั้น ต้องอาศัยอุปกรณ์เฉพาะในการเชื่อมต่อกับดาวเทียม เช่น เครื่องรับสัญญาณ GPS สามารถแบ่งออกเป็นสองกลุ่มใหญ่ คือ

2.6.1 Pull services เป็นลักษณะบริการโดยแบ่งย่อยได้เป็น functional services เช่น การเรียกแท็กซี่ผ่านแอพพลิเคชั่น Grab Taxi ที่พนักงานขับรถแท็กซี่ และผู้โดยสารสามารถเห็นตำแหน่งของกันแลกัน เพื่อความสะดวกในการวางแผนการให้บริการ และการรับบริการได้ตามที่ต้องการ

2.6.2 Push services เป็นลักษณะที่ข้อมูลต่าง ๆ จะถูกส่งโดยมีการร้องขอ หรือ ไม่มีการ ร้องขอก็ตามจากผู้ใช้บริการ โดยปกติบริการจะเริ่มทำงานเมื่อผู้ใช้เข้าสู่บริเวณที่กำหนด หรือ ตามเวลาที่ตั้งไว้ ตัวอย่างที่เห็นง่าย ๆ

การนำมาใช้งานในโปรแกรม

นำมาใช้กับการเช็คระยะห่างของผู้ใช้งานฝั่งอาจารย์กับนักศึกษา เมื่อเริ่มกระบวนการเช็คชื่อโปรแกรมจะส่งพิกัดของอาจารย์และนักศึกษาคำนวณระยะห่างจากจุดที่ยืนวัดระยะห่างจากพิกัดของอาจารย์ นำไปคำนวณถ้าหากไม่เกิน 100 เมตรโปรแกรมจะอนุญาตให้บันทึกการเช็คชื่อได้ แต่ถ้าเกิน 100 เมตร โปรแกรมจะไม่บันทึกรายการเช็คชื่อของนักศึกษาที่อยู่ห่างจากอาจารย์คนนั้น



**ภาพที่ 2-4** ภาพกระบวนการทำงาน Location Based Service (LBS)

(ที่มา : https://www.it24hrs.com/2011/location-based, 2560)

ภาพที่ 2-4 เป็นกระบวนการของ อุปกรณ์ที่เชื่อมต่อกับเสาสัญญาณแต่ละเสาโดยในรูปเป็นการต่อกับเสาที่ 2 จากนั้นไปเรียกใช้บริการของ Map point location service ผ่าน internet

**2.7 การจัดเก็บข้อมูลแบบ NoSQL และการส่งข้อมูล**

หมายถึง การจัดเก็บข้อมูลโปรแกรมฐานข้อมูล ซึ่งมีหน้าที่ในการเก็บข้อมูลแบบใหม่ซึ่งสามารถเก็บข้อมูลเป็น Object เพื่อแสดงผลและส่งต่อให้ไปเพื่อเก็บหรือพัฒนาต่อ และสามารถ รองรับความต้องการของผู้ใช้งานระบบ NoSQL และการส่งข้อมูล

การนำมาใช้งานในโปรแกรม

ได้นำ NoSQL มาพัฒนาให้เข้ากับโปรแกรมซึ่งเป็นการติดต่อระหว่างฐานข้อมูลกับ ผู้ใช้งานและการติดต่อระหว่างฐานข้อมูลของโปรแกรมและฐานข้อมูล Klogic เป็นหลัก โดยเลือกใช้งาน Firebase เป็นที่เก็บข้อมูล

**2.8 Firebase**

คือบริการ backend และ แพลตฟอร์ม ครบวงจรสำหรับนักพัฒนาแอพ และโปรแกรมประยุกต์บนเว็บแพลตฟอร์มที่มีเครื่องมือและโครงสร้างพื้นฐานที่ได้รับการออกแบบมาเพื่อช่วยให้นักพัฒนาสามารถสร้างแอพพลิเคที่มีคุณภาพสูง Firebase (ไฟร์เบส )เป็นฐานข้อมูลเรียลไทม์ซึ่งมี API ที่ช่วยให้นักพัฒนาในการจัดเก็บและซิงค์ข้อมูลและมีการพัฒนาให้สามารถ จากบริการ backend เก็บข้อมูลอย่างเดียว มาเป็น แพลตฟอร์ม ครบวงจรสำหรับนักพัฒนาแอพ (รองรับ iOS, Android, Web) รองรับบริการแทบทุกอย่างที่นักพัฒนาแอพต้องใช้งานโดยมีบริการหลัก ๆ ดังนี้

2.8.1 Hosting บริการ Host Static File นั้นเองโดยสามารถทำเว็บง่าย ๆ และเอาขึ้นไป Online ได้อย่างง่ายดายและสามารถใช้กับ Frontend Framerwork ยกตัวอย่าง Angular2, Vue.js และ React

2.8.2 Realtime Database บริการฐานข้อมูลแบบ NoSQL ซึ่งการออกแบบนั้นอาจจะ ต้องต่างจาก SQL พอสมควรแต่ข้อดีคือมันเป็น Real-time นั้นเอง และมี SDK ครบทุก Platform ทำให้ดึงเอาไปใช้งานได้อย่างง่ายดายและประหยัดเวลาอีกด้วย

2.8.3 Storage บริการเก็บข้อมูล Static Asset อย่างเช่นรูปภาพ และสามารถกำหนด Rules ของตัว Firebase ได้ด้วยทำให้สามารถตั้ง Permission การเข้าถึงไฟล์ต่าง ๆ ได้อีกด้วย

2.8.4 Authentication คือระบบ Login ด้วย Username ที่สร้างขึ้นเองหรือจะผูกกับบัญชี Provider ต่าง ๆ เช่น Google Facebook เป็นต้น

2.8.5 Firebase Test Lab เป็นบริการสำหรับทดสอบเสมือนกับการทดสอบอยู่บนเครื่องจริง ๆ มีการเก็บ Screenshot ระหว่างทำการ Test และมีการสร้างแผนภาพความสำคัญอีกด้วย

2.8.6 Firebase Cloud Messaing เป็นบริการที่ช่วยให้เราทำ Push Notification ไปยังอุปกรณ์ที่เชื่อมต่ออยู่ไม่ว่าจะเป็น Mobile ไปจนถึง Push บนเว็บได้ด้วย

การนำมาใช้งานกับโปรแกรม

นำ Firebase มาใช้งานในส่วนของการเก็บช้อมูลต่าง ๆ ลงฐานข้อมูลและการพิสูจน์ตัวตนด้วยการผูกกับบัญชี Provider ของ Google

**2.9 Nightmare JS**

เป็นไลบรารีที่ใช้เรียกหน้าเว็บไซต์อัตโนมัติยกตัวอย่างเช่นเมื่อต้องการเข้าหน้าเว็บไซต์จะใช้คำสั่ง goto แล้วตามด้วยชื่อ url ของเว็บไซต์ ระบบจะนำพาเราไปยังเว็บไซต์ที่กำหนด url นั้นไว้เป็นต้น มีความสามารถในการจัดการกับผลลัพธ์ของหน้าเว็บได้ สามารถนำไปใช้ในงานต่าง ๆ ดังนี้

**2.9.1 Testing**

สำหรับทดสอบระบบ จำลองการเข้าใช้งานของ User ที่เข้ามาใช้งานยกตัวอย่างเช่น จำลองการเข้าใช้งานของ User ที่เข้ามาเลือกซื้อสินค้าออนไลน์ผ่านหน้าเว็บไซต์ หรือจำลองการเข้าสู่ระบบ Login มีการกำหนดสถานการณ์จำลองที่จะต้องเกิด ซึ่งต้องครอบคลุมถึงกรณีที่เป็น Negative หรือกรณีที่กรอกข้อมูลผิดพลาดเพื่อทดสอบ Response ที่ระบบจะแจ้งเตือนแก่ user เป็นต้น

2.9.2 Page Automation

สามารถใช้งานควบคู่ไปกับ DOM Manipulation กับหน้าเว็บหลังจากที่โหลดมาโดยดึงผลลัพธ์จาก Tag HTML

การนำมาใช้งานในโปรแกรม

นำมาใช้เก็บข้อมูลหน้าเว็บไซต์ Klogic ลงฐานข้อมูลของโปรแกรม เพื่อนำข้อมูลที่เก็บมาแสดงผ่านโปรแกรมอัตโนมัติ กระบวนการทำงานใช้ควบคู่กับ ภาษา DOM getค่าจากTag HTML ในส่วนหน้าเว็บแล้วส่งข้อมูลลงฐานข้อมูล

**2.10 Node.js**

Node.js คือ Cross Platform Runtime Environment สำหรับฝั่ง Server และเป็น Open Source ซึ่งเขียนด้วยภาษา JavaScript เป็น Platform ตัวหนึ่งที่เขียนด้วย JavaScript สำหรับเป็น Web Server นั่นเองสามารถรันได้บนทุกระบบปฏิบัติการยอดนิยมมาพร้อมเทคโนโลยีที่เรียกว่า Non - Blocking I/O ปัจจุบัน NodeJS ถูกนำมาทำเป็น Web Server, Mobile Hybrid, IOT, Webkit , TVOS ,OS อื่น ๆ อีกมาก เรียกได้ว่าเข้าถึงได้หลากหลายเทคโนโลยี

การนำมาใช้งานในโปรแกรม

เป็นตัวกลางระหว่างการส่งข้อมูลจาก Applition ลงฐานข้อมูล และเป็นตัวกลางการรับส่งข้อมูลระหว่าง Firebase กับ Klogic

* 1. **Platform as a Services (PaaS)**

บริการด้าน Platformสำหรับการพัฒนา Software และ Application โดยผู้ให้บริการจะจัดเตรียมสิ่งที่จำเป็นต้องใช้ในการพัฒนาโปรแกรม Software และ Application ต่าง ๆ เช่น Web Application, Database Server รวมถึงระบบประมวลผลกลางสำหรับองค์กรขนาดใหญ่ และ Middleware อื่น ๆ เป็นต้น โดยบริการทั้งหมดทำงานภายใต้ระบบรักษาความปลอดภัยเครือข่าย และสามารถเรียกใช้งานได้ผ่าน Web Application ซึ่งผู้ใช้บริการสามารถนำไปใช้ในการปรับใช้และจัดการได้เอง ระบบ PaaS นั้นประกอบด้วยระบบปฏิบัติการ ระบบฐานข้อมูล และระบบมิดเดิ้ลแวร์ตัวอย่างเช่น Window Server, Linux, Oracle Database เป็นต้น

การนำมาใช้งานในระบบ

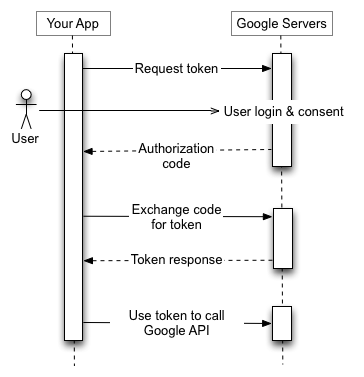
ในการทำงานของโปรแกรมมีการใช้บริการ Cloud Server เพื่อฝากพื้นที่เก็บข้อมูล ทำให้ช่วย ลดต้นทุน การทำงานและลดการเสียเวลาการติดตั้ง Server

* 1. **Google Gmail**

Google Mail  หรือ Gmail  เป็นหนึ่งในบริการดี ๆ จาก Google เว็บไซต์ที่ให้บริการฟรีอีเมล์ที่จะทำให้สามารถมี Email (จดหมายอิเล็กทรอนิกส์) เป็นของตัวเองได้ โดยสามารถเขียนจดหมาย อีเมล์ ไปยังผู้รับที่ต้องการได้มากมาย ไม่ว่าจะเป็น Hotmail , yahoo และอื่น ๆ อีกมากมาย โดยวิธีการใช้สามารถใช้ได้ผ่านทางระบบเว็บเมล์ POP และ IMAP  ในปัจจุบัน Google Mail เป็นบริการ Email บนเว็บไซต์ที่มีผู้ใช้มากที่สุดในโลกโดยมีผู้ใช้ทั่วโลกมากกว่า 425 ล้านคน  Google Mail ในปัจจุบันนี้มีการรองรับภาษาทั่วโลกได้ 54 ภาษาแล้วภาษารวมถึงภาษาไทยด้วย

การนำมาใช้งานในโปรแกรม

นำ Gmail มาประยุกต์ใช้เข้ากับการพิสูจน์ตัวตนของผู้ใช้งาน โดยจะใช้ E-mail ของทางคณะ (@fitm.kmutnb.ac.th)



**ภาพที่ 2-5** ภาพกระบวนการทำงานพิสูจน์ตัวตนผ่านแอคเค้าท์ Google

(ที่มา : https://developers.google.com/identity/protocols/OAuth2, 2560)

ภาพที่ 2-5 คือขั้นตอนของการนำ Gmail มาใช้พิสูจน์ตัวตนโดย User เป็นคนกรอกข้อมูลผ่าน Application จากนั้นตัว Application จะได้ Token มาไว้เช็คเวลา เรียกข้อมูลมาใช้จาก Google ว่าตรงกันหรือไม่ ถ้าตรงก็จะได้ข้อมูลนั้นมา

**บทที่ 3**

**ขั้นตอนการพัฒนาและการออกแบบ**

**3.1 การวิเคราะห์ปัญหา**

ปัจจุบันโรงเรียน และมหาวิทยาลัยยังมีการบันทึกการเข้าชั้นเรียนอยู่จึงมีความจำเป็นต้อง ทำการวิเคราะห์และศึกษาถึงระบบงานเดิมที่มีอยู่ ในการจัดทำโปรแกรมเช็คชื่อผ่านโทรศัพท์มือถือ เพื่อให้ทราบถึงปัญหาและโครงสร้างของระบบเก่า และจุดบกพร่องที่อาจจะเกิดขึ้นในด้านต่าง ๆ เพื่อให้สามารถพัฒนาโปรแกรมให้ตรงกับเป้าหมายที่วางไว้ให้เกิดประสิทธิภาพมากยิ่งขึ้นในการวางแผนการออกแบบการสื่อสารต่าง ๆ ให้เหมาะสม ซึ่งสามารถใช้งานได้จริง และมีความถูกต้องตามที่ได้รับกลับมาซึ่งการวิเคราะห์ถึงปัญหาเดิมทำให้ทราบได้ว่า หากห้องเรียนมีมากกว่า 30 คนทำให้เกิดข้อจำกัดในเรื่องระยะเวลาการบันทึกการเข้าเรียนที่ค่อนช้านาน โครงงานพิเศษนี้จึงนำแนวคิดและเทคโนโลยีที่มีอยู่ร่วมกับการวิเคราะห์ระบบ แบบใหม่มาพัฒนาให้ดียิ่งขึ้นจากปัจจุบัน ทุกคนต้องมีโทรศัพท์ติดตัวอยู่แล้ว จึงนำมาประยุกต์ใช้ให้เข้ากับฐานข้อมูล ของมหาวิทยาลัยโดยลดเวลาและเพิ่มความรวดเร็วให้มากยิ่งขึ้น

**3.2 การดำเนินงาน**

3.2.1 ศึกษาค้นคว้าทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับระบบ

3.2.1.2 ศึกษาว่าตัวโปรแกรมมีหน้าที่อะไร

โปรแกรมเช็คชื่อที่จะพัฒนาขึ้นมานี้ จะทำหน้าที่เป็นตัวกลางในการเช็คชื่อแทนกระดาษเพื่อช่วยลดการสูญหายของข้อมูลที่อยู่บนกระดาษมาเก็บลงไว้ในฐานข้อมูลแทน อีกทั้งสิ่งสำคัญคือช่วยในเรื่องความรวดเร็วของการบันทึกการเข้าชั้นเรียน และประมวลผลเป็น รีพอร์ตคำนวนแสดงเป็นรายการของนักศึกษาแต่ละคน

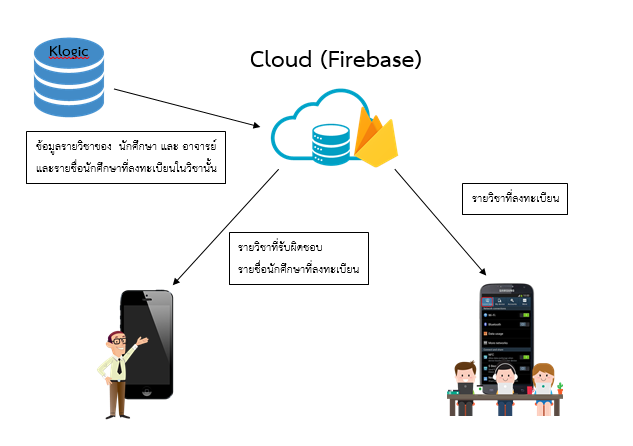
3.2.1.3 ศึกษาการทำงานของโปรแกรม

การทำงานของโปรแกรม ถูกแบ่งออกเป็นส่วนต่าง ๆ ได้แก่

3.2.1.3.1 การเพิ่มรายวิชาและการเข้าคลาสเรียน

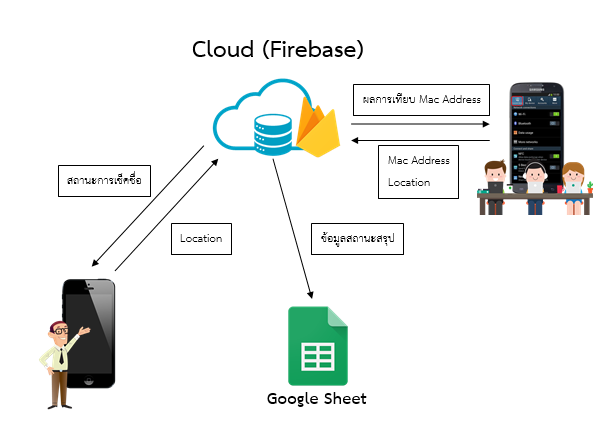
3.2.1.3.2 การเช็คชื่อและการออก Report

3.2.1.3.3 การตรวจสอบสถานะรายชื่อนักศึกษา



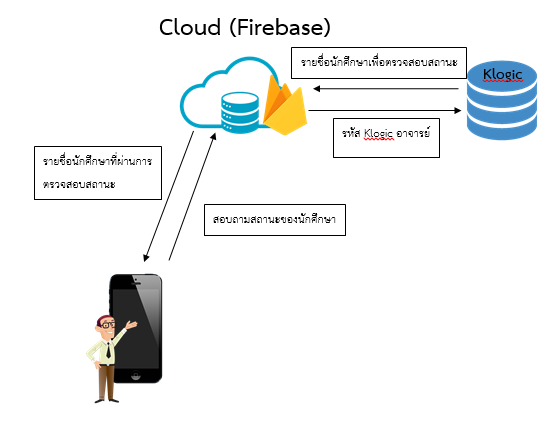
**ภาพที่ 3-1** แผนผังกระบวนการทำงานเพิ่มรายวิชาจากฐานข้อมูล Klogic

จากภาพที่ 3-1 อาจารย์และนักศึกษาสามารถเห็นวิชาที่ตัวเองรับผิดชอบได้โดยมีตัวกลางที่จะไปดึงข้อมูลจาก ฐานข้อมูล Klogic จะดึงข้อมูลมาและจะส่งกลับมาทางฝั่งนักศึกษา คือ รายวิชาที่นักศึกษาลงทะเบียนเรียนแล้ว และฝั่งอาจารย์ คือ รายวิชาที่รับผิดชอบ และรายชื่อนักศึกษาที่ละทะเบียนในรายวิชานั้น ๆ



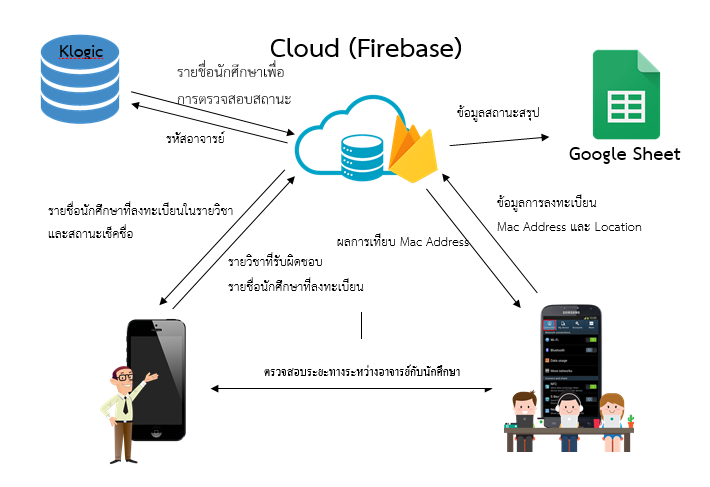
**ภาพที่ 3-2** แผนผังกระบวนการเก็บข้อมูลและประมวลผลออกมาในรูปแบบรีพอร์ต(Report)

จากภาพที่ 3-2 โปรแกรมจะทำการเช็คชื่อโดย ให้อาจารย์เปิดการเช็คชื่อ แล้วให้นักศึกษา ทำการกดปุ่มเช็คชื่อที่หน้าโปรแกรมของตนเอง หลังจากนั้นโปรแกรมจะทำการส่ง Mac Address และ IMEI ไปยัง ฐานข้อมูลเพื่อทำการประมวลผล และส่งแจ้งเตือนมายังนักศึกษาว่าตนนั้น ได้รับการเช็คชื่อแล้ว รวมถึงส่งผลการตรวจสอบ Mac Address และ IMEI มาให้ทางโปรแกรมฝั่งอาจารย์เพื่อสรุปสถานนะการเช็คชื่อโดยในการเช็คชื่อนั้นจะมีเงื่อนไขอยู่ว่าจำเป็นต้องอยู่ห่างจากระยะของอาจารย์ไม่เกิน 100 เมตร หากเกินจะไม่ทำการเช็คชื่อให้ และ ไม่บันทึกลงไปใน Report

****

**ภาพที่ 3-3** แผนผังกระบวนการการตรวจสอบสถานะรายชื่อนักศึกษา

จากภาพที่ 3-3 โปรแกรมจะเช็คสถานะรายชื่อของนักศึกษาเมื่ออาจารย์ Login โปรแกรม จะนำรายชื่อ นักศึกษาในรายวิชาต่าง ๆ มาเพื่อทำการตรวจสอบรายชื่อนักศึกษาที่มีอยู่ในฐานข้อมูลเทียบกับรายชื่อนักศึกษาในระบบ Klogic หากไม่มีรายชื่อในระบบ Klogic แล้วจะทำการตัดรายชื่อของนักศึกษาคนนั้นออกจากฐานข้อมูล และ Update ข้อมูลลงโปรแกรมของอาจารย์โดยทันที



**ภาพที่ 3-4** Diagram รวมการทำงานของโปรแกรม

**3.3 การวิเคราะห์ถึงระบบงาน**

3.3.1 การออกแบบตารางจัดเก็บข้อมูล (Table Layout Design) มีการใช้ฐานข้อมูล แยกออกจากกันด้วยมีในส่วนของการบันทึกเป็นสมัครสมาชิกอัตโนมัติและเช้าสู่ระบบ และอีกส่วนเป็นส่วนจัดการระบบฐานข้อมูลของนักศึกษาเพื่อเก็บข้อมูลสำคัญ เช่น

3.3.1.1 ฐานข้อมูลเกรดนักศึกษาในแต่ละปีการศึกษา โดยเริ่มจากปี 60 เป็นต้นมา

3.3.1.2 ฐานข้อมูลรายวิชาของนักศึกษาที่ได้ลงทะเบียนในแต่ละปีการศึกษา

3.3.1.3 ฐานข้อมูลชื่อของนักศึกษาที่ได้ลงทะเบียนทั้งหมด

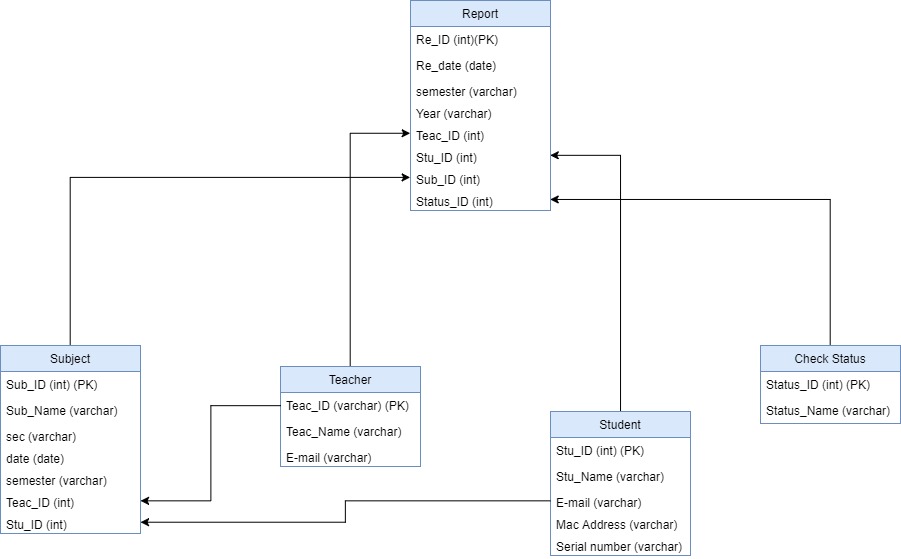
3.3.1.4 ฐานข้อมูลชื่ออาจารย์ผู้สอนในภาควิชาเทคโนโลยีสารสนเทศ

3.3.1.5 ฐานข้อมูลตารางสอนของอาจารย์ซึ่งเป็นข้อมูลแบบเดียวกับ Klogic

3.3.1.6 ฐานข้อมูล Status การเข้าชั้นเรียน

3.3.1.7 ฐานข้อมูล Report ของแต่ละวิชา

การออกแบบตารางจัดเก็บข้อมูล (Table Layout Design)

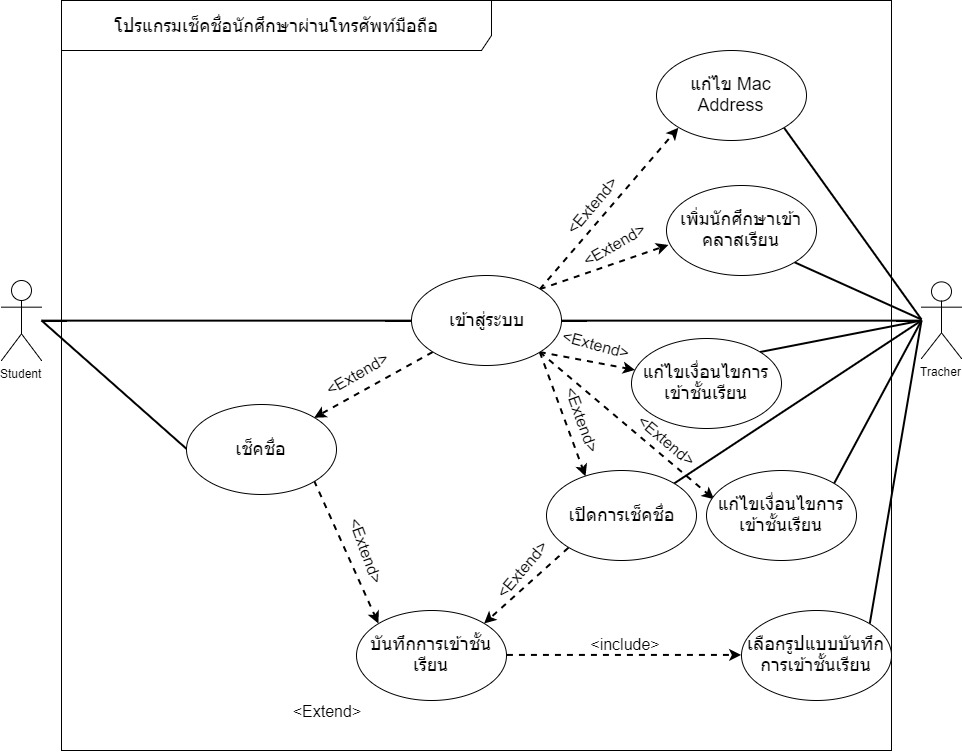


**ภาพที่ 3-5** แผนผังแสดงการเก็บข้อมูลเพื่อจัดทำระบบ

ภาพที่ 3-5 คือการวาดระบบโดยการออกแบบส่วนของข้อมูลต่าง ๆ ที่เรานำมาใช้งานในระบบโดยในแต่ละตารางจะมีข้อมูลไม่เหมือนกัน ลูกศรที่เชื่อมโยงกันนั้นได้แสดงถึงความสัมพันธ์ ของกันและกันดังนี้

* Subject สัมพันธ์กับ Report ผ่านตัว Sub\_ID หรือรหัสวิชา
* Teacher สัมพันธ์กับ Report ผ่านตัว Teac\_ID หรือรหัสอาจารย์
* Teacher สัมพันธ์กับ Subject ผ่านตัว Teac\_ID หรือรหัสอาจารย์
* Student สัมพันธ์กับ Subject ผ่านตัว Stu\_ID หรือรหัสนักศึกษา
* Student สัมพันธ์กับ Report ผ่านตัว Stu\_ID หรือรหัสนักศึกษา
* Check Status สัมพันธ์กับ Report ผ่านตัว Status\_ID หรือรหัสการเช็คชื่อ

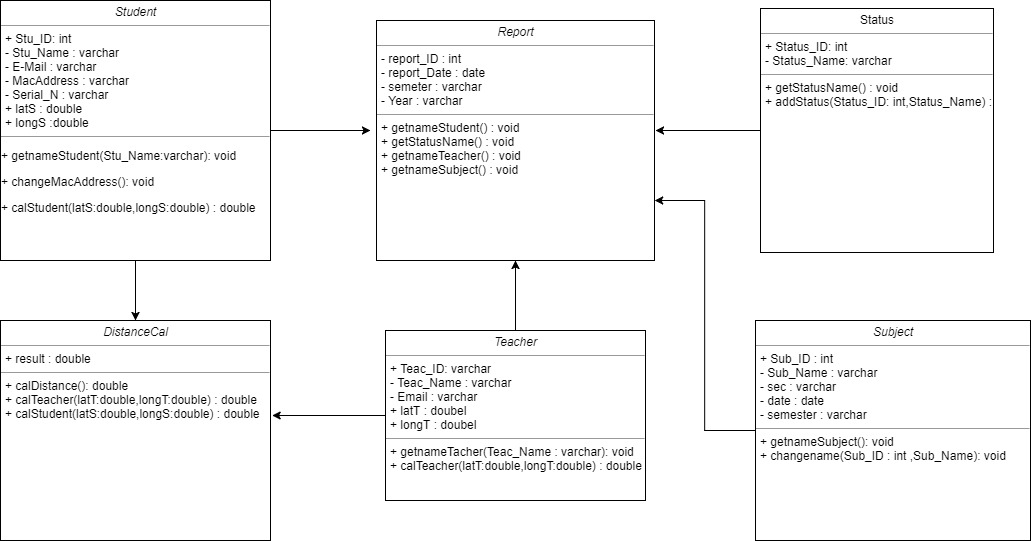
**3.4 Use Case Diagram**

****

**ภาพที่ 3-6** รูปแบบUse Case Diagram

ภาพที่ 3-6 จากภาพจะเห็นว่ามี Actor 2 คนคือ Student และ Teacher เริ่มจาก Teacher เมื่อ Teacher เข้าสู่ระบบ จะสามารถเข้าทำงานตาม Extend ที่มีความสัมพันธ์กับการเข้าสู่ระบบได้เช่นเมื่อเข้าสู่ระบบสามารถเปิดการเช็คชื่อนักศึกษาและเมื่อเปิดการเช็คชื่อจะก่อให้เกิดการ Extend การบันทึกข้อมูล เมื่อเกิดการบันทึกการเช้าชั้นเรียนทำให้ Teacher สามารถเลือกรูปแบบกาบันทึกการเข้าชั้นเรียนได้แต่จะเกิดขึ้นได้ก็ต่อเมื่อมี Student เข้ามาเกี่ยวข้องด้วยถึงจะเป็นการทำงานได้สมบูรณ์

**3.5 Class Method Java**

****

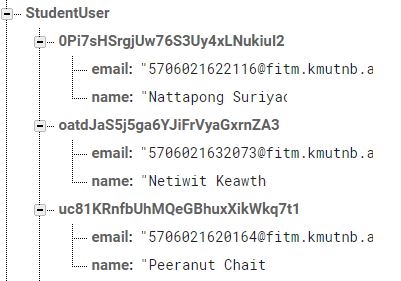
**ภาพที่ 3-7** รูปแบบClass Method Java

จากภาพที่ 3-7มี Class หลักทั้งหมด 6 Class ประกอบไปด้วย Student ข้อมูลของนักศึกษา Teacher ข้อมูลข้องอาจารย์ Subject ข้อมูลรายวิชาเรียน Status ข้อมูลของสถานะต่าง ๆ DistanceCal การคำนวณระยะทางของอาจารย์กับนักศึกษา สุดท้าย Report ข้อมูลแสดงผล การบันทึกการเข้าชั้นเรียน โดยการทำงานทุกคลาสจะส่งข้อมูลไปยัง Report ทั้งหมดยกตัวอย่าง การเช็คชื่อจะเริ่มจากอาจารย์ส่งชื่อตัวเองไปเก็บไว้ใน Report และส่งข้อมูลตำแหน่งไปคำนวณ ที่ DistanceCal จากนั้นส่งข้อมูลวิชาที่อาจารย์คนนั้นต้องการเช็คชื่อไปที่ Report เมื่อนักศึกษา เข้ามาทำการเช็คชื่อ จะส่งข้อมูลตำแหน่งไปที่ DistanceCal เช่นกันเมื่อได้ผลลัพธ์แล้วจะส่งไปที่ Report จากนั้น Status จะส่ง Status Name ไปเทียบลงใน Report เป็นการเสร็จสิ้นการทำงาน

**3.6 การเก็บข้อมูลเพื่อใช้เชื่อมต่อและควบคุมฐานข้อมูล**

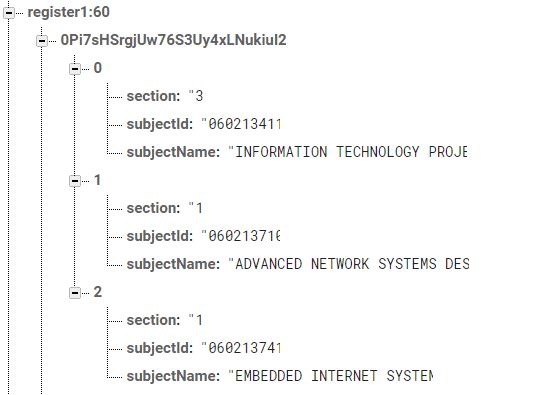
หมายถึง การเพิ่มช่องทางในการเชื่อมต่อระหว่างเว็บไซต์หนึ่งไปอีกที่หนึ่ง หรือเป็นการเชื่อมต่อระหว่างผู้ใช้งานกับ Server หรือ Server เชื่อมต่อไปยัง Server เองก็สามารถทำงานได้เช่นกัน เปรียบได้กับภาษาคอมพิวเตอร์ที่ทำให้คอมพิวเตอร์สามารถสื่อสารและแลกเปลี่ยนข้อมูลกัน อย่างอิสระ โดยส่วนมากแล้วเราจะเห็น APIs ถูกใช้งานกันอย่างแพร่หลายที่เห็นกันได้อย่างชัดเจน โดยได้นำมาประยุกต์ให้เข้ากับระบบเพื่อช่วยให้เข้าถึงข้อมูลฐานข้อมูลได้สะดวกยิ่งขึ้น

3.6.1 ตารางการเก็บข้อมูล StudenUser ทำหน้าที่ในการเก็บข้อมูลนักศึกษาที่เข้าสู่ระบบเพื่อใช้งาน



**ภาพที่ 3-8** รูปแบบการเก็บข้อมูลของ Child StudentUser

3.6.2 ตารางการเก็บข้อมูลรายวิชาที่นักศึกษาลงทะเบียน ปีการศึกษา 1/2560



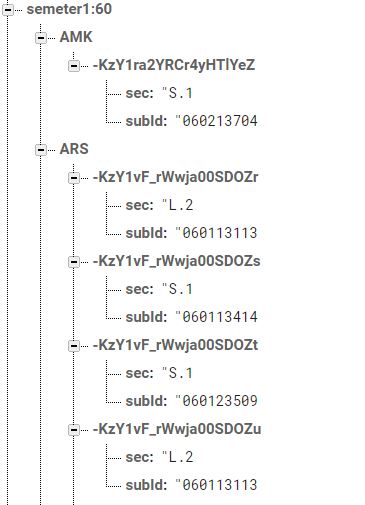
**ภาพที่ 3-9** รูปแบบการเก็บข้อมูลของ Child register1:60

3.6.3 ตารางการเก็บข้อมูล teacher ชองอาจารย์ผู้สอนภายในคณะเทคโนโลยีและ การจัดการอุตสาหกรรม ภาควิชาเทคโนโลยีสารสนเทศ



**ภาพที่ 3-10** รูปแบบการเก็บข้อมูลของ Child teacher

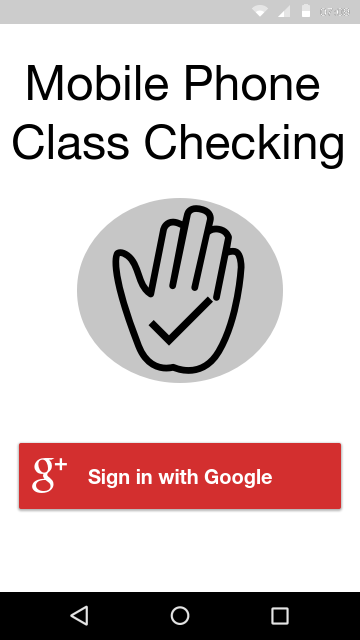
3.6.4 ตารางข้อมูลเก็บข้อมูล Subjec ที่ทำหน้าที่เก็บข้อมูล ตารางสอนของอาจารย์ผู้สอนในคณะเทคโนโลยีและการจัดการอุตสาหกรรม ภาควิชาเทคโนโลยีสารสนเทศหรือที่เกี่ยวข้องกับหลักสูตรปีการศึกษา 1/2560



**ภาพที่ 3-11** รูปแบบการเก็บข้อมูลของ Child Semeter1:60

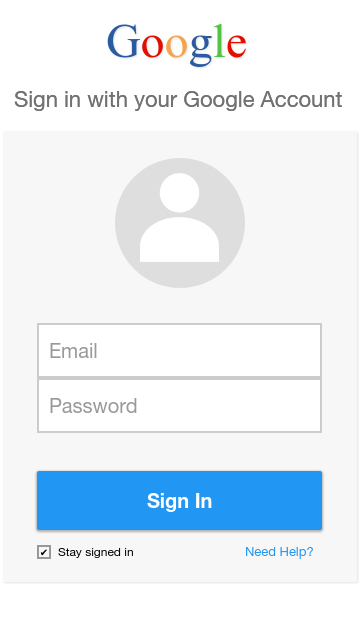
**3.7 การออกแบบหน้าจอเพื่อติดต่อกับผู้ใช้งาน**

3.7.1 ฝั่งอาจารย์



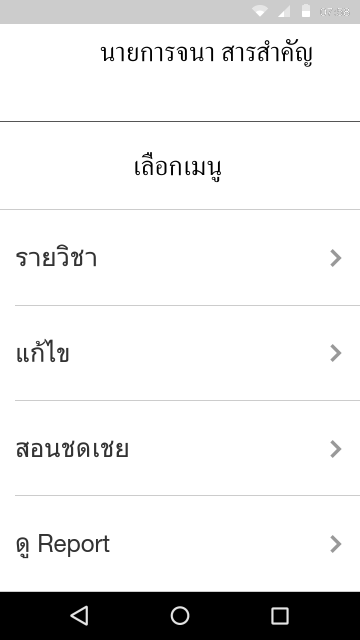
**ภาพที่ 3-12** ตัวอย่างหน้าจอการยืนยันตัวตน

จากภาพที่ 3-12 หน้าจอการยืนยันตัวตนเป็นการ Login ของอาจารย์โดยใช้ gmail ของคณะเพื่อทำการยืนยันตัวตนอาจารย์



**ภาพที่ 3-13** ตัวอย่างหน้าจอการยืนยันตัวตน (หลังจากกดปุ่ม Sing In)

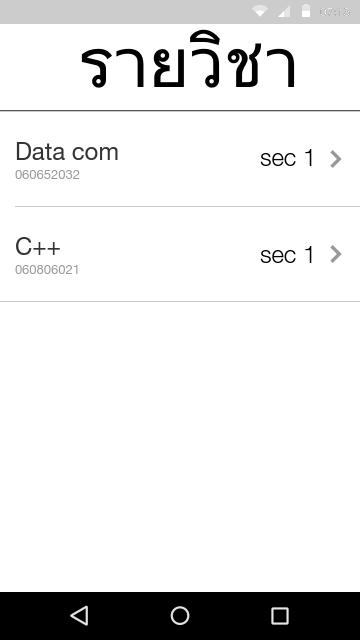
จากภาพที่ 3-13 หน้าจอการยืนยันตัวตน (หลังจากด Sing in) เป็นหน้าต่อจากหน้าจอยืนยันตนตัวโดยจะให้อาจารย์ กรอก E-mail ของคณะโดยจะลงท้ายด้วย @fitm.kmutnb.ac.th และใส่รหัสผ่านของ E-mail



**ภาพที่ 3-14** ตัวอย่างหน้าจอเมนู และ ข้อมูลของ User ที่เกี่ยวข้อง

จากภาพที่ 3-14 หน้าจอเมนู และ ข้อมูลของ User ที่เกี่ยวข้องภายในหน้าจอนี้จะประกอบไปด้วย 4 ตัวเลือก

* รายวิชา คือ ให้เลือกรายวิชาที่จะทำการเช็คชื่อนักศึกษา
* แก้ไข คือ การแก้ไขหมายเลข Mac Address ในกรณีที่นักศึกษาต้องการจะเปลี่ยนเครื่องหรือเปลี่ยนเครื่องโทรศัพท์
* สอนชดเชย คือ การเช็คชื่อในสัปดาห์ที่ทำการสอนชดเชยเพื่อชดเชยในสัปดาห์ที่ยกคลาสหรือ ไม่สามารถมาทำการสอนได้
* Report คือ การสรุปผลการเช็คชื่อแต่ละสัปดาห์ ระหว่างเทอม และ ทั้งเทอม



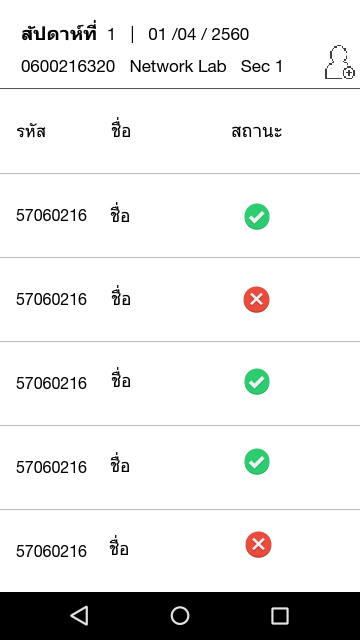
**ภาพที่ 3-15** ตัวอย่างหน้าจอเลือกรายวิชา

จากภาพที่ 3-15 หน้าจอเลือกรายวิชาเป็นการเลือกรายวิชาที่อาจารย์ต้องการจะทำการเช็คชื่อโดยรายวิชาที่แสดงบนหน้าจอนั้น แสดงตามรายวิชาที่อาจารย์เป็นคนสอนในเทอมนั้น



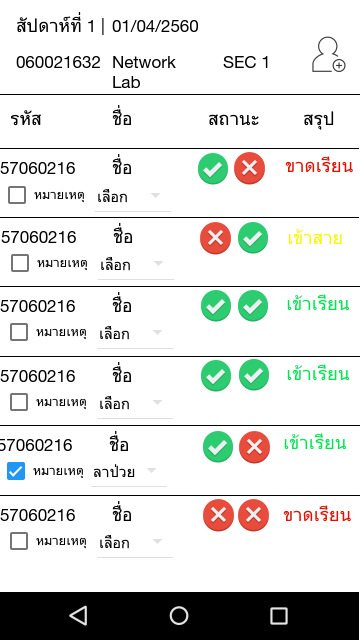
**ภาพที่ 3-16** ตัวอย่างหน้าจอแสดงปุ่มเปิดการเช็คชื่อ

จากภาพที่ 3-16 หน้าจอแสดงปุ่มเปิดการเช็คชื่อหน้าจอแสดงปุ่มเปิดการเช็คชื่อเพื่อเป็นสัญญาณในการให้นักศึกษากดปุ่มเช็คชื่อได้ (ถ้าอาจารย์ไม่เปิดการเช็คชื่อนักศึกษาก็จะเช็คชื่อไม่ได้) แยกเป็นต้นชั่วโมงและท้ายชั่วโมง

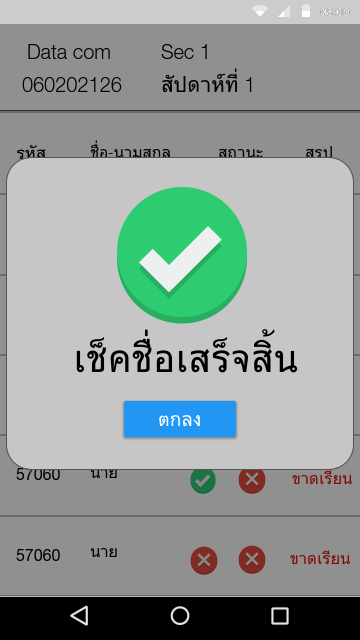


**ภาพที่ 3-17** ตัวอย่างหน้าจอแสดงการเช็คชื่อ(ต้นชั่วโมง)

จากภาพที่ 3-17 หน้าจอแสดงการเช็คชื่อหน้าจอ Report การเช็คชื่อ และ แสดงรายชื่อนักศึกษาที่ได้ลงทะเบียนและสถานะของนักศึกษาในคาบนั้น ซึ่งจะแบ่งการเช็คชื่อเป็นสองช่วงคือในช่วงต้นชั่วโมงและในช่วง ท้ายชั่วโมงเพื่อทำการเช็คชื่อ โดยรายละเอียดสถานะจะแบ่งได้ดังนี้

* + เข้าเรียน คือ นักศึกษาได้ทำการเช็คชื่อในช่วงต้นชั่วโมงและท้ายชั่วโมง
  + สาย คือ นักศึกษาได้ทำการเช็คชื่อเพียงท้ายคาบ
  + ขาดเรียนคือนักศึกษาไม่ได้ทำการเช็คชื่อทั้งสองช่วง หรือ เข้าเช็คชื่อช่วงต้นชั่วโมง แต่ท้ายชั่วโมงไม่ได้ทำการเช็คชื่อหรือหายไปจากคาบเรียน โดยไม่ได้ทำการแจ้งอาจารย์ก่อนซึ่งทางโปรแกรมจะถือว่านักศึกษามีเจตนา โดดเรียน
  + หากมีหมายเหตุ ไม่ว่าสถานะของนักศึกษคนนั้นจะขึ้นอย่างไร โปรแกรมจะถือว่านักศึกษามาเข้าเรียน
  + 

**ภาพที่ 3-18** ตัวอย่างหน้าจอแสดงการเช็คชื่อ(ท้ายชั่วโมง)



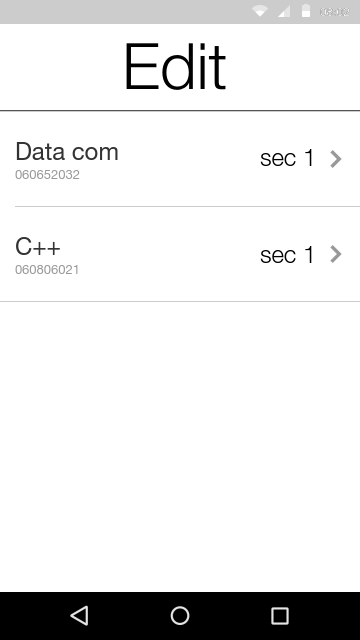
**ภาพที่ 3-19** หน้าจอตัวอย่าง Alert เมื่อนักศึกษากดเช็คเชื่อครบทุกคนแล้ว

จากภาพที่ 3-19 หน้าจอตัวอย่าง Alert เมื่อนักศึกษากดเช็คเชื่อครบ ทุกคนแล้วจะทำงานต่อเมื่อนักศึกษาในวิชานั้นกดเช็คชื่อครบทุกคน



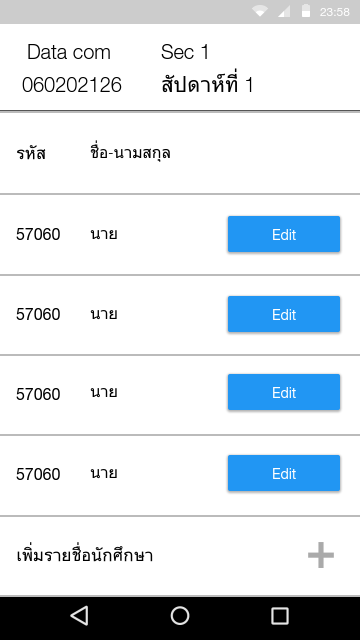
**ภาพที่ 3-20** หน้าจอตัวอย่าง Alert ถ้าหมดเวลาเช็คชื่อ

จากภาพที่ 3-20 หน้าจอตัวอย่าง Alert หมดเวลาเช็คชื่อทำงานต่อเมื่อหมดเวลาเช็คชื่อภายในระยะเวลาที่กำหนด



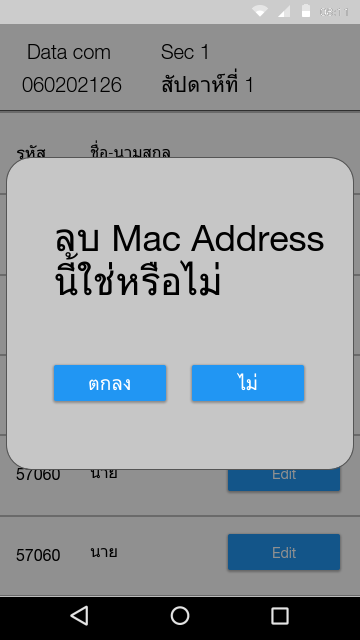
**ภาพที่ 3-21** ตัวอย่างหน้าจอเลือกรายวิชาเพื่อแก้ไข

จากภาพที่ 3-21 หน้าจอแก้ไขเมื่อเลือกรายการแก้ไขจะเลือกว่าต้องการที่จะแก้ไขรายวิชาไหน ๆ ก็ตาม



**ภาพที่ 3-22** ตัวอย่างหน้าจอแก้ไข (ต่อ)

จากภาพที่ 3-22 หน้าจอแก้ไข (ต่อ)เมื่อเลือกได้แล้วจะเข้าไปยังรายชื่อของนักศึกษาในวิชานั้นจะมีปุ่มให้กด Edit เพื่อแก้ไข Mac Address ของนักศึกษา



**ภาพที่ 3-23** ตัวอย่างหน้าจอ Alert เมื่อกดปุ่มแก้ไข

จากภาพที่ 3-23 หน้าจอ Alert เมื่อกดปุ่มแก้ไขเมื่อผู้ใช้งานกดปุ่มแก้ไข จะมี Alert ขึ้นมาถามว่าต้องการลบ Mac Address นี้ใช้หรือไหม



**ภาพที่ 3-24** ตัวอย่างหน้าจอเมื่อกดปุ่มเพิ่มรายชื่อนักศึกษา

จากภาพที่ 3-24 หน้าจอเมื่อกดปุ่มเพิ่มรายชื่อนักศึกษาในกรณีที่นักศึกษาไม่มีรายชื่อ สามารถกรอกข้อมูลของนักศึกษาคนนั้นลงไปยังรายวิชานั้นได้ โดยจะกรอกรหัสนักศึกษา



**ภาพที่ 3-25** หน้าจอ Alert เมื่อกดปุ่มเพิ่มรายชื่อ

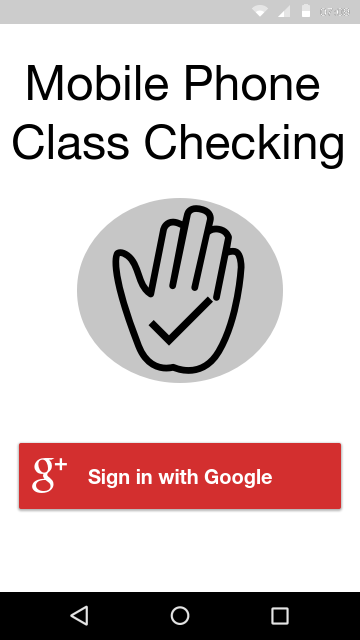
จากภาพที่ 3-25 หน้าจอ Alert เมื่อกดปุ่มเพิ่มรายชื่อเมื่อผู้ใช้งานกดปุ่มแก้ไข จะมี Alert แสดงข้อความ “เพิ่มรายชื่อเสร็จเสร็จสิ้น”



**ภาพที่ 3-26** ตัวอย่างหน้าจอ Report

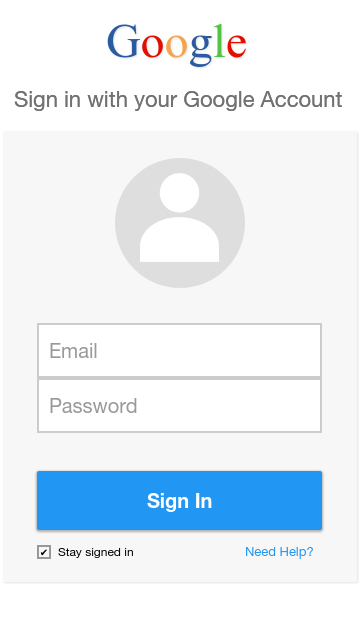
จากภาพที่ 3-25 หน้าจอ Report หน้าจอ Report ให้อาจารย์เลือกการแสดงผล 2 แบบ ไฟล์.xls และไฟล์ .pdf หรือจะเลือก ส่งไปหา E-mail มีให้เลือก 3 รูปแบบ คือ รายสัปดาห์ ระหว่างเทอม และ 1 ภาคการศึกษา

3.7.2 ฝั่งนักศึกษา



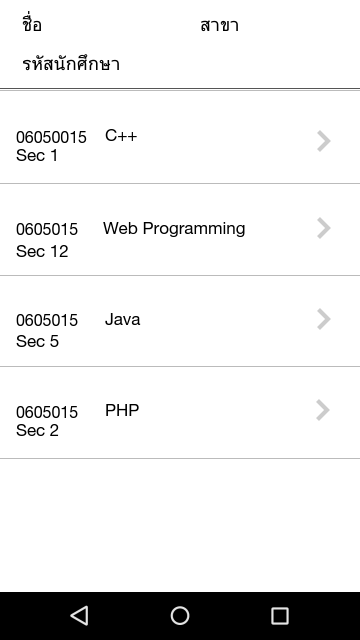
**ภาพที่ 3-27** ตัวอย่างหน้าจอ Login ฝั่งนักศึกษา

จากภาพที่ 3-25 หน้าจอ Login ฝั่งนักศึกษาเป็นการ Login ของนักศึกษาโดยใช้ E-mail เพื่อทำการยืนยันตัวตนนักศึกษา



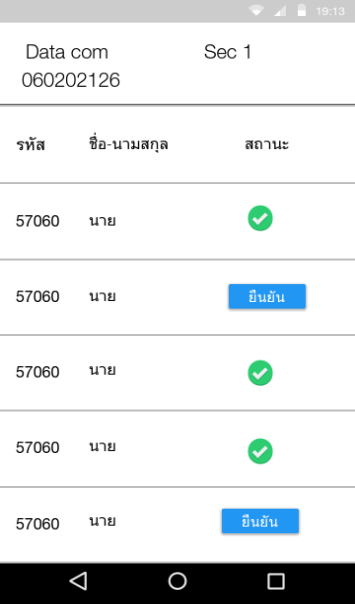
**ภาพที่ 3-28** หน้าจอการยืนยันตัวตน (หลังจากด Sing in)

จากภาพที่ 3-28 หน้าจอการยืนยันตัวตน (หลังจากด Sing in) เป็นหน้าต่อจากหน้าจอยืนยันตนตัวโดยจะให้กรอกE-mail ของคณะ



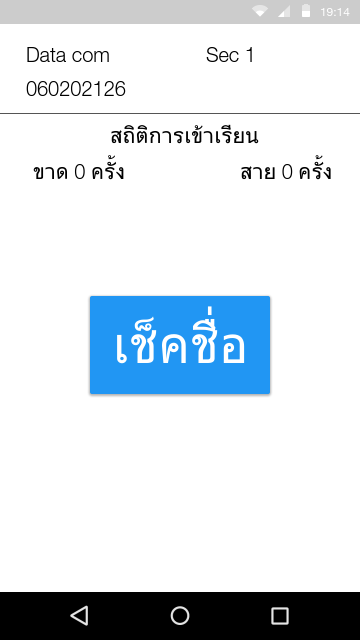
**ภาพที่ 3-29** ตัวอย่างหน้าจอเลือกวิชาเช็คชื่อ

จากภาพที่ 3-29 หน้าจอ เลือกวิชาเช็คชื่อหน้าจอที่แสดงผลถัดมาจากหน้า Login จะเป็น หน้าจอหลักแสดงรายวิชาที่นักศึกษาเรียนในเทอมนั้น (รายวิชาทั้งหมดจะมาจาก klogic ที่นักศึกษาคนนั้นลงทะเบียน)



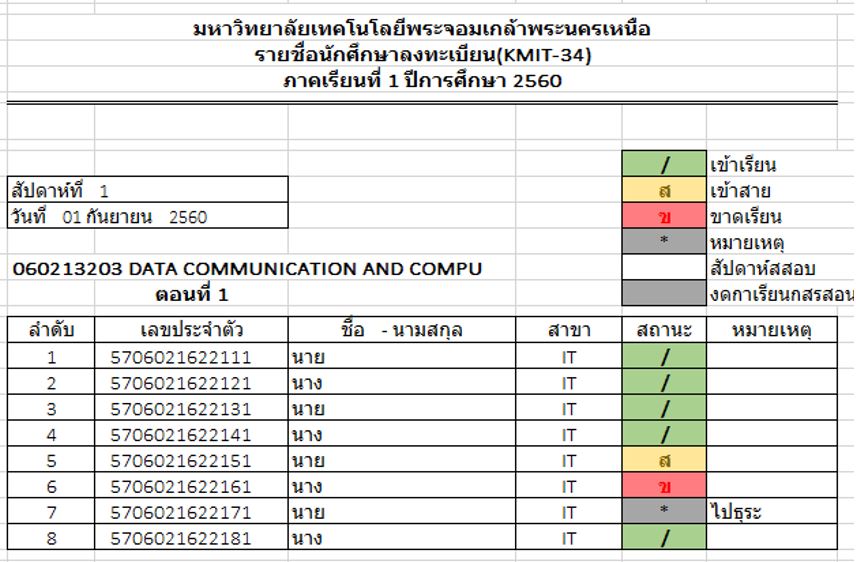
**ภาพที่ 3-30** ตัวอย่างหน้าจอเช็คชื่อครั้งแรก

จากภาพที่ 3-30 หน้าจอเช็คชื่อครั้งแรกในการเช็คชื่อครั้งแรกให้นักศึกษาเลือกชื่อของตนเองเพื่อส่งค่า Mac Address ของโทรศัพท์ตนเองเพื่อเป็นการยืนยันตัวตนว่านี้เป็นเครื่องของนักศึกษาสามารถใช้เครื่องที่ยืนยันเดียวในการเช็คชื่อ



**ภาพที่ 3-31** ตัวอย่างหน้าจอเช็คชื่อ

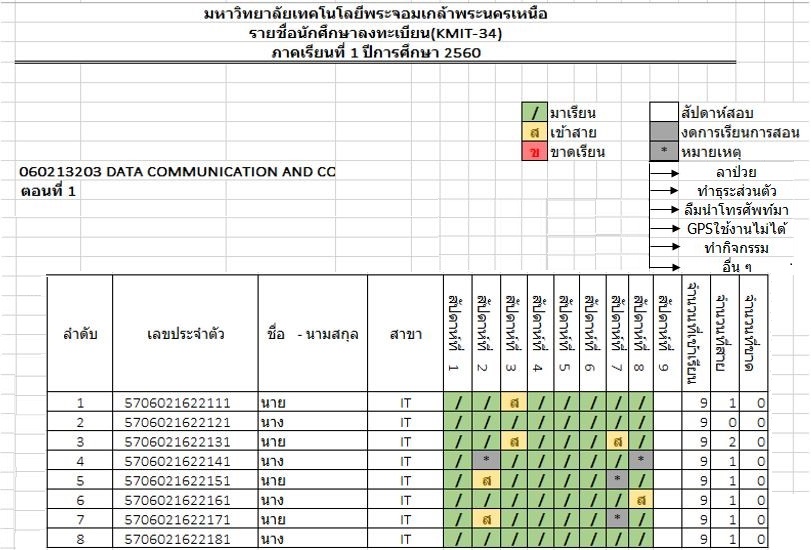
3.7.3 Report

****

**ภาพที่ 3-32** ตัวอย่าง Report วันที่ 01 กันยายน 2560

จากภาพที่ 3-32 เป็น Report ที่ออกเป็นรายสัปดาห์โดยจะมีข้อมูลรายละเอียด ดังนี้

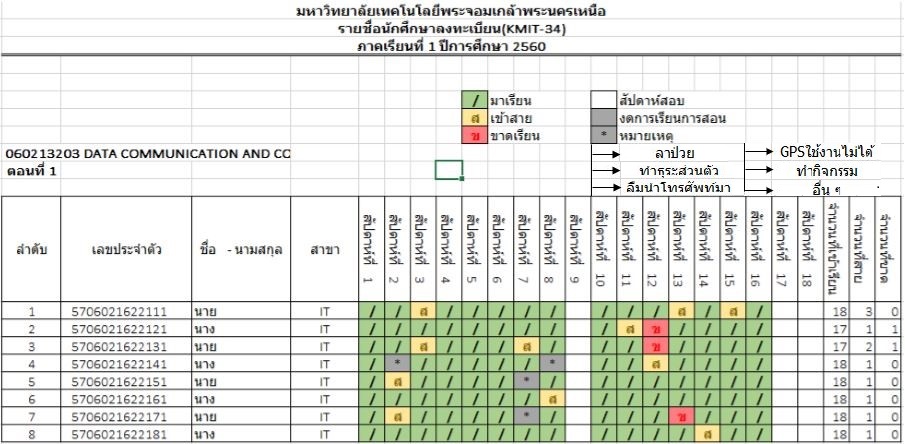
* ลำดับ
* เลขประจำตัว
* ชื่อ – นามสกุล
* สาขา
* สถานะ
* หมายเหตุ

****

**ภาพที่ 3-33** ตัวอย่าง Report ระหว่างเทอม

จากภาพที่ 3-33 เป็น Report ที่ออกเป็นระหว่างเทอมโดยจะมีข้อมูลรายละเอียด ดังนี้

* ลำดับ
* เลขประจำตัว
* ชื่อ – นามสกุล
* สาขา
* สถานะในแต่ละสัปดาห์
* สรุปยอดทั้งหมด

****

**ภาพที่ 3-34** ตัวอย่าง Report 1 ภาคการศึกษา

จากภาพที่ 3-34 เป็น Report ที่ออกเป็น 1 ภาคการศึกษาโดยจะมีข้อมูลรายละเอียด ดังนี้

* ลำดับ
* เลขประจำตัว
* ชื่อ – นามสกุล
* สาขา
* สถานะในแต่ละสัปดาห์
* สรุปยอดทั้งหมด

|  |
| --- |
| **3.8 Code การทำงานแต่ละส่วน** |
| 3.8.1 Google Authentication |
| public class MainActivity extends AppCompatActivity implements GoogleApiClient.OnConnectionFailedListener {  private GoogleApiClient mgoogleApiClient; //ประกาศ GoogleApiClient เป็น private  private SignInButton signInButton;  //ประกาศ สร้างปุ่มเพื่อล็อกอินเข้าใช้งาน ชื่อว่า signInButton  private static final int SIGN\_IN\_CODE = 1; //กำหนดตัวแปรแบบไม่ให้ค่าที่อยู่ข้างในเปลี่ยน  private FirebaseAuth firebaseAuth; //ประกาศเรียกใช้ FirebaseAuth  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  // สร้างตัวแปร Google Sign In เพื่อเรียกใช้ฟังก์ชั่นของ Google ชื่อว่า gso  GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT\_SIGN\_IN)  .requestIdToken(getString(R.string.default\_web\_client\_id))  .requestEmail()  .build();  signInButton = (SignInButton) findViewById(R.id.signInButton);  //สร้างตัวแปรที่เชื่อมกับปุ่มที่อยู่ใน xml  signInButton.setSize(SignInButton.SIZE\_WIDE); //กำหนดSize  signInButton.setColorScheme(SignInButton.COLOR\_DARK); //กำหนดสี  signInButton.setOnClickListener(new View.OnClickListener() { //ตั้งค่าเวลากดปุ่มที่นี่  @Override  public void onClick(View v) {  // เมื่อคลิกจะเข้าไปเช็คตัวตนของ User  Intent intent = Auth.GoogleSignInApi.getSignInIntent(mgoogleApiClient);  startActivityForResult(intent, SIGN\_IN\_CODE);  }  });  firebaseAuth = FirebaseAuth.getInstance(); //สั่งเริ่มการทำงาน Authen  firebaseAuthListener = new FirebaseAuth.AuthStateListener() {  //เช็คเงื่อนไข หาก User ไม่มีค่าว่างจะสั่งไปทำงานที่ goMainScreen()  @Override  public void onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {  FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();  if (user != null) {  goMainScreen();  }  }  };  }  @Override  protected void onStart() {  //ตรวจสอบผู้ใช้ที่ Sign In เข้ามาโดยจะเรียกฟังก์ชั่น firebaseAuthListener  super.onStart();  firebaseAuth.addAuthStateListener(firebaseAuthListener);  }  @Override  protected void onActivityResult(int requestCode, int resultCode, Intent data) {  super.onActivityResult(requestCode, resultCode, data);  //ฟังชั่นที่เช็คผลการล็อกอินสำเร็จหรือไม่หากสำเร็จ จะเรียก handleSignInResult(result);  if (requestCode == SIGN\_IN\_CODE) {  GoogleSignInResult result = Auth.GoogleSignInApi.getSignInResultFromIntent(data);  handleSignInResult(result);  }  }  }  private void firebaseAuthWithGoogle(GoogleSignInAccount signInAccount) {  AuthCredential credential = GoogleAuthProvider.getCredential(signInAccount.getIdToken(),null);  firebaseAuth.signInWithCredential(credential).addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  @Override  public void onComplete(@NonNull Task<AuthResult> task) {  if(!task.isSuccessful()){ //ถ้าหาก Login ไม่สำเร็จจะแสดง Alert ตามเงื่อน  AlertDialog alertDialog = new AlertDialog.Builder(MainActivity.this).create();  alertDialog.setTitle("Warning message");  alertDialog.setMessage("กรุณาใช้ E-mail ที่ลงท้ายด้วย @fitm.kmutnb.ac.th");  alertDialog.setButton(AlertDialog.BUTTON\_NEUTRAL, "OK",  new DialogInterface.OnClickListener() {  public void onClick(DialogInterface dialog, int which) {  dialog.dismiss();  }  });  alertDialog.show();  }  }  });  }  private void goMainScreen() { //สั่งให้ไปที่หน้า MenuActivity.class  Intent intent = new Intent(this, MenuActivity.class);  intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_NEW\_TASK);  startActivity(intent);  }  @Override  protected void onStop() {  //หยุดการทำงานเมื่อ ค่าที่อยู่ใน firebaseAuthListener มีค่าเป็น Null  super.onStop();  if (firebaseAuthListener != null){  firebaseAuth.removeAuthStateListener(firebaseAuthListener);  }  }  } |

3.8.2 GPS

public void CheckGPS(){

if (!isGPS && !isNetwork) {

Log.d(TAG, "Connection off");

showSettingsAlert();

getLastLocation();

} else {

Log.d(TAG, "Connection on");

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.M) {

if (permissionsToRequest.size() > 0) {

requestPermissions(permissionsToRequest.toArray(new String[permissionsToRequest.size()]),

ALL\_PERMISSIONS\_RESULT);

Log.d(TAG, "Permission requests");

canGetLocation = false;

}

}

getLocation();

}

เช็ค GPS บนโทรศัพท์ทำงานหรีอไม่ ถ้าเปิดการทำงานเรียบร้อยแล้วจะเข้าสู่ฟังก์ชั่นของ getLocation แต่ถ้าไม่เปิด จะเข้าสู่ฟังก์ชั่นของ ShowSettingAlert() กับ getLastLocation()

3.8.3 Location

3.8.3.1 สั่ง Get

private void getLocation() {

try {

if (canGetLocation) {

Log.d(TAG, "Can get location");

if (isGPS) {

// from GPS

Log.d(TAG, "GPS on");

locationManager.requestLocationUpdates(

LocationManager.GPS\_PROVIDER,

MIN\_TIME\_BW\_UPDATES,

MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES, this);

if (locationManager != null) {

loc = locationManager.getLastKnownLocation(LocationManager.GPS\_PROVIDER);

//Log.d("TAG",loc.toString());

if (loc != null)

updateUI(loc);

}

} else if (isNetwork) {

// from Network Provider

Log.d(TAG, "NETWORK\_PROVIDER on");

locationManager.requestLocationUpdates(

LocationManager.NETWORK\_PROVIDER,

MIN\_TIME\_BW\_UPDATES,

MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES, this);

if (locationManager != null) {

loc=locationManager.getLastKnownLocation(LocationManager.NETWORK\_PROVIDER);

if (loc != null)

updateUI(loc);

} else {

loc.setLatitude(0);

loc.setLongitude(0);

updateUI(loc);

}

} else {

Log.d(TAG, "Can't get location");

}

} catch (SecurityException e) {

e.printStackTrace();

}

}

Get location โดยใช้ locationManger โดยมีหน้าที่ get latitude และ longitude

3.7.3.2 คำนวณระยะทาง

double lat1 =loc.getLatitude() ;

double long1 =loc.getLongitude() ;

double R = 6371 ;

double dlat = degreesToRadians(lat1 - lat2);

double dlon = degreesToRadians(long1 - long2)

lat1 = degreesToRadians(lat1);

lat2 = degreesToRadians(lat2);

double a = Math.sin(dlat/2) \* Math.sin(dlat/2) + Math.sin(dlon/2) \* Math.sin(dlon/2) \* Math.cos(lat1) \* Math.cos(lat2);

double c = 2 \* Math.atan2(Math.sqrt(a), Math.sqrt(1-a));

Dist = R \* c ;

Dist = Dist\*1000;

Dist = Dist/1000;

ส่วนของการคำนวณ ระยะห่างระหว่างโทรศัพท์นักศึกษาและอาจารย์ และนำข้อมูล ระยะทาง และ latitude, longitude เก็บลงดาต้าเบส

3.7.4 Get Mac address

public static String getMacAddr() {

try {

List<NetworkInterface> all=Collections.list(NetworkInterface.getNetworkInterfaces());

for (NetworkInterface nif : all) {

if (!nif.getName().equalsIgnoreCase("wlan0")) continue;

byte[] macBytes = nif.getHardwareAddress();

if (macBytes == null) {

return "";

}

StringBuilder res1 = new StringBuilder();

for (byte b : macBytes) {

res1.append(Integer.toHexString(b & 0xFF) + ":");

}

if (res1.length() > 0) {

res1.deleteCharAt(res1.length() - 1);

}

return res1.toString();

}

} catch (Exception ex) {

}

return "00000";

}

ส่วนของการดึงค่า Mac Address ของ โทรศัพท์มือถือ ถ้าไม่สามารถ Get ออกมาได้จะได้ค่าเป็น “00000”

3.7.5 Get IMEI

public String getIMEI() {

TelephonyManager tel = (TelephonyManager) getSystemService(Context.TELEPHONY\_SERVICE);

if(ActivityCompat.checkSelfPermission(this, Manifest.permission.READ\_PHONE\_STATE) != PackageManager.PERMISSION\_GRANTED) {

// TODO: Consider calling

// ActivityCompat#requestPermissions

// here to request the missing permissions, and then overriding

// public void onRequestPermissionsResult(int requestCode, String[] permissions,

// int[] grantResults)

// to handle the case where the user grants the permission. See the documentation

// for ActivityCompat#requestPermissions for more details.

}

return tel.getDeviceId();

}

ส่วนของการดึงค่า IMEI ของ โทรศัพท์มือถือเพื่อเก็บลงฐานข้อมูล

**บทที่ 4**

**ผลการดำเนินงาน**

ในการทำโครงงานพิเศษได้มีการวิเคราะห์และวางแผนการดำเนินงานเพื่อให้การดำเนินงานสำเร็จตรงตามระยะเวลาที่กำหนดไว้ และตรงตามวัตถุประสงค์ของโครงงานพิเศษที่ได้กำหนดไว้ โดยมีดังต่อไปนี้

**4.1 ผลจากการดำเนินงาน**

4.1.1 แบ่งเป็น 2 ส่วน

4.1.1.1 หน้าจอ Login ประกอบด้วยปุ่มที่จำนำเข้าสู่หน้ากรอกข้อมูลของ Google เพื่อทำการตรวจสอบข้อมูล ได้แก่ E-mail ที่จะเป็นได้เฉพาะอีเมล์ของบุคลากรของทางคณะเทคโนโลยีและการจัดการอุตสาหกรรม ภาควิชาเทคโนโลยีสารสนเทศ และตรวจสอบข้อมูลในส่วนของรหัสผ่าน

4.1.1.1 หน้าจอแสดงผลการเชื่อมต่อข้อมูลของนักศึกษาในภาควิชาเทคโนโลยีสารสนเทศ

4.1.1.2.1 แสดงข้อมูลของนักศึกษา

4.1.1.2.2 แสดงรายวิชาที่นักศึกษาลงทะเบียน

4.1.1.2 หน้าจอการแสดงผลของอาจารย์ผู้สอนในภาควิชาเทคโนโลยีสารสนเทศ ดังนี้

4.1.1.3.1 เมนูสำหรับเลือกรายวิชาเพื่อเริ่มการเช็คชื่อ

4.1.1.3.2 เมนูสำหรับแก้ไขและเพิ่มข้อมูลนักศึกษา

4.1.1.3.3 เมนูสำหรับสอนชดเชย

4.1.1.3.4 เมนูสำหรับ Report

4.1.1.3 หน้าจอสำหรับ Alert เมื่อเช็คชื่อครบทุกคนจะแสดงข้อความแจ้งเตือนส่งมาที่อาจารย์ “การเช็คชื่อสำเร็จแล้ว”

4.1.1.4 หน้าจอสำหรับนักศึกษาจะแสดงรายวิชาที่นักศึกษาข้อมูลและรายวิชาที่นักศึกษาลงทะเบียน

4.1.1.5 หน้าจอสำหรับแสดงเมื่อนักศึกษาเลือกรายวิชาที่ต้องการจะเช็คชื่อจะแสดงข้อมูลการเข้าเรียนของนักศึกษาคนนั้น และจะแสดงปุ่มเช็คชื่อเพื่อบันทึกการเข้าชั้นเรียน

4.1.1.6 หน้าจอสำหรับ Alert เมื่อนักศึกษากดปุ่มเช็คชื่อหากสำเร็จจะแสดงข้อความ “การเช็คชื่อสำเร็จแล้ว” และถ้าหากเกิดอาการขัดข้องจะแสดงเป็นข้อความ “การเช็คชื่อไม่สำเร็จโปรดลองใหม่อีกครั้ง” หรือถ้าเกิดเหตุการณ์นักศึกษาใช้โทรศัพท์เครื่องอื่นแทนโทรศัพท์มือถือเครื่องของตัวเองจะแสดงข้อความ “หมายเลข Mac Address ไม่ตรงกันกับฐานข้อมูล”

4.1.1.7 หน้าจอสำหรับเช็คชื่อจะแสดงรายชื่อนักศึกษาที่ลงทะเบียนในรายวิชาและเมนูเพื่อปิดการเช็คชื่อมีรายละเอียด ดังนี้

4.1.1.7.1 รหัสนักศึกษา

4.1.1.7.2 ชื่อนักศึกษา

4.1.1.7.3 สถานะการเช็คชื่อต้นชั่วโมง

4.1.1.7.4 สถานะการเช็คชื่อท้ายชั่วโมง

4.1.1.7.5 สรุปสถานะการเข้าชั้นเรียนของวัน

4.1.1.8 หน้าจอสำหรับการเปิดการเช็คชื่อเมื่อเลือกรายวิชาแล้วจะแสดงเมนู ดังนี้

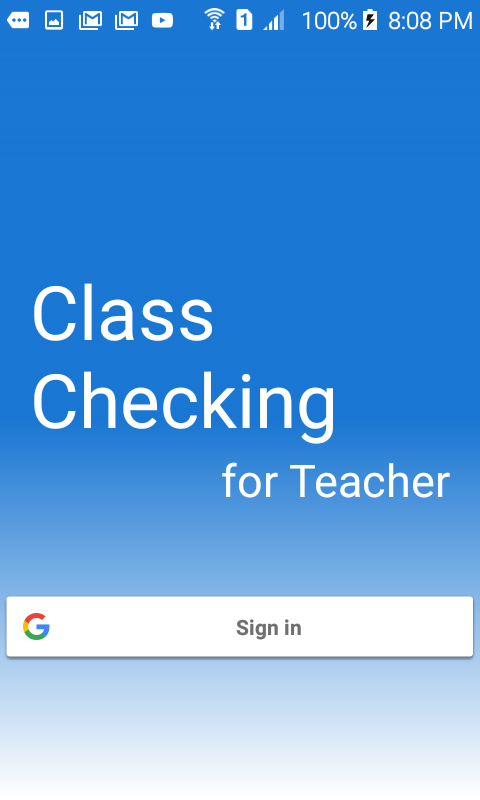
4.1.1.8.1 เมนูต้นชั่วโมง

4.1.1.8.2 เมนูท้ายชั่วโมง

4.1.1.9 หน้าจอบันทึกรายการเช็คชื่อของแต่ละวิชาที่อาจารย์ได้ทำการเช็คเสร็จเรียบร้อยแล้ว

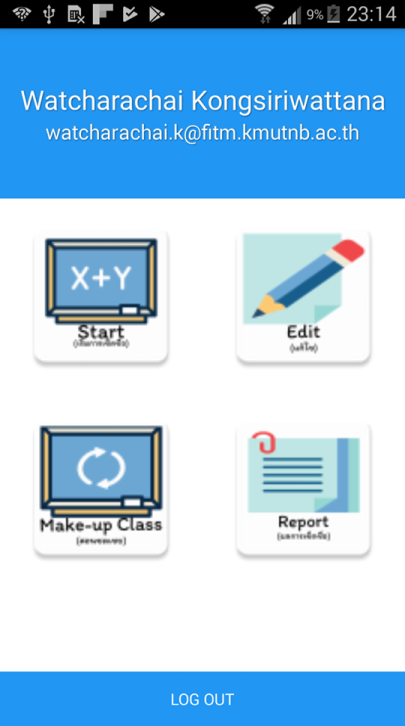
**4.2 รูปแบบหน้าจอที่ได้ทำการออกแบบและนำไปพัฒนา**

4.2.1 หน้าจอ Login ประกอบด้วยปุ่มที่จำนำเข้าสู่หน้ากรอกข้อมูลของ Google เพื่อทำการตรวจสอบข้อมูล ได้แก่ E-mail ที่จะเป็นได้เฉพาะอีเมลของบุคลากรของทางคณะเทคโนโลยีและการจัดการอุตสาหกรรม ภาควิชาเทคโนโลยีสารสนเทศ และตรวจสอบข้อมูลในส่วนของรหัสผ่าน



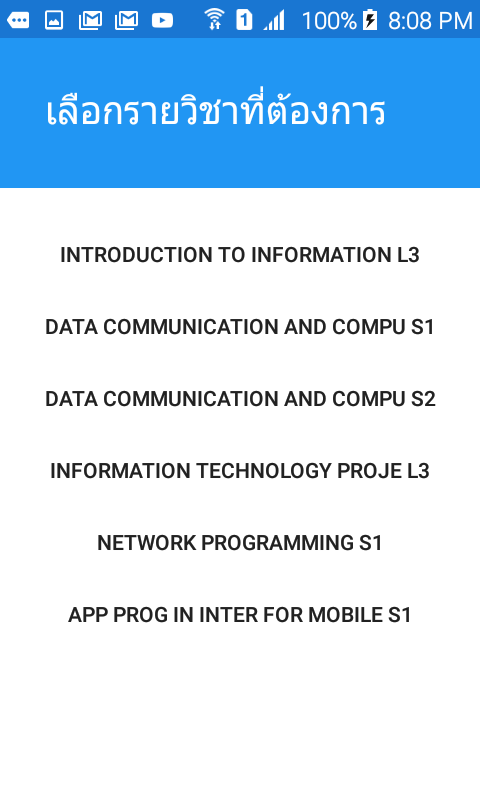
**ภาพที่ 4-1** ภาพแสดงหน้าจอสำหรับเข้าสู่ระบบของ Google

4.2.2 หน้าจอเริ่มต้นเมื่อล็อกอินเรียบร้อยแล้ว



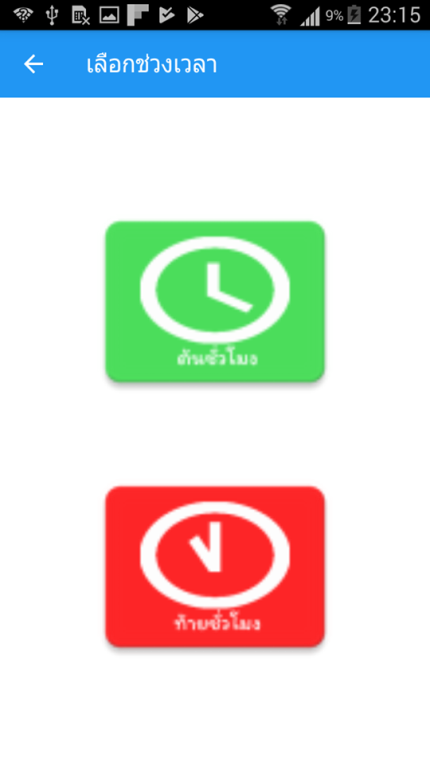
**ภาพที่ 4-2** แสดงหน้าจอแสดงเมนู

4.2.3 หน้าจอการแสดงผลรายวิชาที่อาจารย์รับผิดชอบของอาจารย์ผู้สอนในภาควิชาเทคโนโลยีสารสนเทศ



**ภาพที่ 4-3** แสดงหน้าจอสำหรับเลือกรายวิชาที่อาจารย์รับผิดชอบ

4.2.4 หน้าจอสำหรับการเลือกช่วงเวลาที่ต้องการเช็คชื่อ



**ภาพที่ 4-4** หน้าจอสำหรับเลือกช่วงเวลาในการเช็คชื่อ

4.2.5 หน้าจอแสดงผลเมื่อเลือกเมนูต้นชั่วโมง



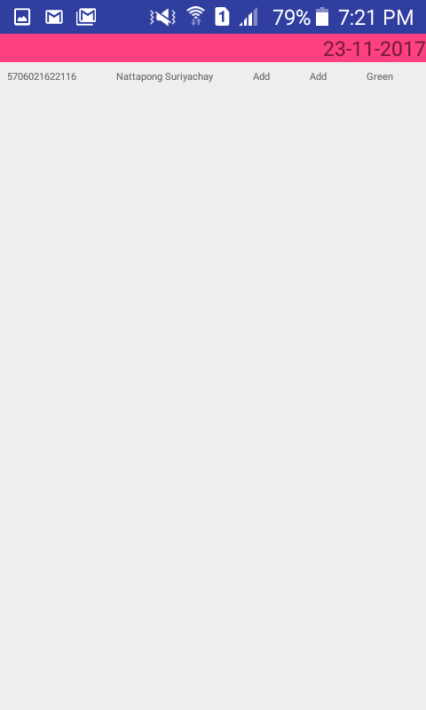
**ภาพที่ 4-5** หน้าจอเช็คชื่อต้นชั่วโมงแสดงรายชื่อนักศึกษาที่ลงทะเบียนทั้งหมด

4.2.6 หน้าจอแสดงผลเมื่อเลือกเมนูต้นท้ายชั่วโมง



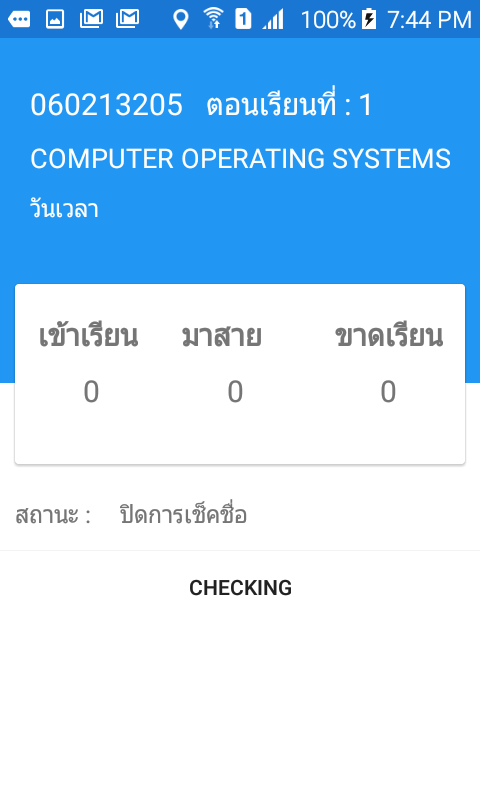
**ภาพที่ 4-6** หน้าจอเช็คชื่อท้ายชั่วโมงแสดงรายชื่อนักศึกษาที่ลงทะเบียนทั้งหมด

4.2.7 หน้าจอแสดงผลเมื่อเลือกเมนู Report แสดงข้อมูลรายวิชาที่อาจารย์เช็คชื่อเสร็จเรียบร้อยแล้ว



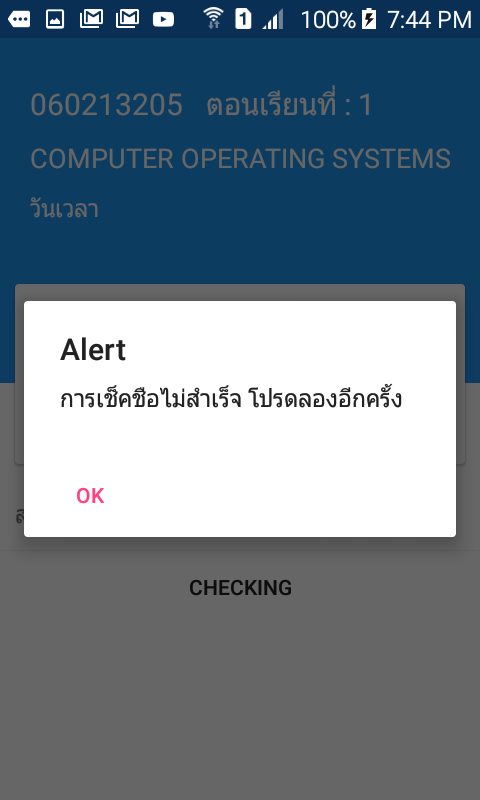
**ภาพที่ 4-7** หน้าจอ Report แสดงรายชื่อนักศึกษาทุกคนที่ลงทะเบียน

4.2.8 หน้าจอสำหรับเช็คชื่อ

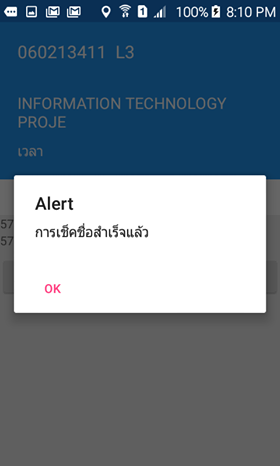


**ภาพที่ 4-8** หน้าจอสำหรับเช็คชื่อ

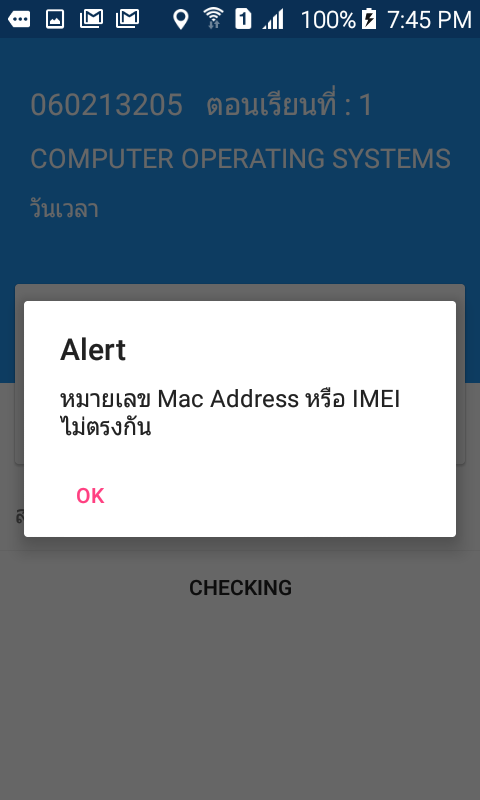
4.2.9 หน้าจอ Alert



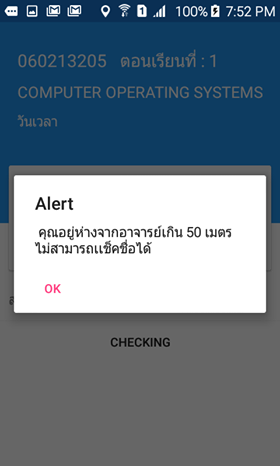
**ภาพที่ 4-9** หน้าจอสำหรับ Alert แสดงข้อความเมื่อเกิดเหตุขัดข้อง



**ภาพที่ 4-10** หน้าจอสำหรับ Alert แสดงข้อความเมื่อเช็คชื่อสำเร็จ



**ภาพที่ 4-11** หน้าจอสำหรับ Alert แสดงข้อความเมื่อตรวจสอบ Mac Address แล้วไม่ตรงกัน



**ภาพที่ 4-12** หน้าจอสำหรับ Alert แสดงข้อความเมื่อตรวจสอบแล้วอยู่ห่างจากระยะที่กำหนด

**บทที่ 5**

**สรุปผลการดำเนินงาน และข้อเสนอแนะ**

**5.1 การวิเคราะห์และสรุปผล**

5.1.1 ปีการศึกษา 1/2560

จากการผลการทำงาน จากโปรแกรมเช็คชื่อนักศึกษาผ่านโทรศัพท์มือถือ โดยมีผู้ใช้งานทั้งหมด 2 กลุ่ม ซึ่งประกอบไปด้วย นักศึกษา อาจารย์ โดยสามารถทำงานได้ดังนี้

5.1.1.1 อาจารย์

5.1.1.1.1 สามารถเปิดการเช็คชื่อตามรายวิชาที่อาจารย์รับผิดชอบ

5.1.1.1.2 สามารถดูข้อมูลชองตัวเองได้ เช่น ชื่อ – นามสกุล รายวิชาที่อาจารย์รับผิดชอบ

5.1.1.1.3 สามารถดูรายการนักเรียนที่เข้าเรียน ไม่เข้าเรียน มาสายได้

5.1.1.2 นักศึกษา

5.1.1.2.1 สามารถดูข้อมูลของตัวเอง เช่น ชื่อ - นามสกุล รายวิชาที่ลงทะเบียนแล้ว

5.1.1.2.1 สามารถบันทึกการชั้นเข้าเรียนในขั้นตอนการเช็คชื่อ

5.1.1.3 โปรแกรม

5.1.1.3.1 สามารถดึงข้อมูลจาก Klogic ได้แล้ว ยกตัวอย่างเช่น รายวิชา ที่ นักศึกษาลงทะเบียนเรียน

5.1.1.3.2 สามารถ Get ข้อมูล Mac Address หรือ IMEI ของโทรศัพท์มือถือ เก็บลงฐานข้อมูลได้ และนำไปเทียบในขั้นตอนการเช็คชื่อ

5.1.1.3.3 สามารถ Get Location ของ User ลงฐานข้อมูลและนำไปเข้า สูตรคำนวณหาระยะห่างไม่เกิน 100 เมตรได้

5.1.1.4 หน้าจอโปรแกรม

5.1.1.4.1 หน้าจอสำหรับ Login โดยใช้ Gmail ในการเข้าใช้งานโปรแกรม (ทั้งอาจารย์และนักศึกษา)

5.1.1.4.2 หน้าจอสำหรับดูรายวิชาที่อาจารย์รับผิดชอบ (ฝั่งอาจารย์)

5.1.1.4.3 หน้าจอสำหรับดูรายวิชาที่นักศึกษาลงทะเบียนโดยข้อมูลมาจาก Klogic ( ฝั่งนักศึกษา )

5.1.1.4.4 หน้าจอสำหรับเปิด/ปิดการบักทึกการเข้าชั้นเรียน (ฝั่งอาจารย์)

5.1.1.4.5 หน้าจอสำหรับกดเช็คชื่อ (ฝั่งนักศึกษา)

**5.2 ปัญหาและอุปสรรค**

5.2.1 ปีการศึกษา 1/2560

5.2.1.1 ไม่มีความเข้าใจในการทำงานของ Firebase ทำให้ไม่สามารถนำข้อมูลที่ได้มาเก็บลงฐานข้อมูลได้

5.2.1.2 เรื่องการเข้าถึงข้อมูลของนักศึกษาที่ทำได้ยาก จึงจำเป็นต้องเขียนโปรแกรมเพื่อดึงข้อมูลเข้าถึง Klogic อีกขั้นหนึ่งเพื่อเข้าถึงข้อมูลของทั้งระบบ

5.2.1.3 เกิดปัญหา Program ไม่รองรับ SDK เวอร์ชั่นที่ใช้งานอยู่และตัวโปรแกรมนั้น

ไม่สามารถรันไลบรารีได้

5.2.1.4 เกิดปัญหาการเข้าใช้งาน Firebase Authentication ไม่สามารถ Login โดยใช้ G-mail ได้

5.2.1.5 เกิดปัญหาเรื่องตัวกลางการรับส่งข้อมูลไม่สามารถรับข้อมูลเข้ามายังตัวกลางได้

5.2.2 การแก้ไขปัญหาที่เกิดขึ้น

5.2.2.1ศึกษาและทำความเข้าใจโดยวาดวิธีการส่งข้อมูลลง Firebase รวมถึงทดลองการเขียนข้อมูลลง Firebase ง่าย ๆ ก่อนแล้วจึงเริ่มใหม่

5.2.2.2 สามารถแก้ไขปัญหาได้ 2 วิธี คือ

5.2.2.2.1 ติดต่อสำนักคอมพิวเตอร์ของมหาวิทยาลัยเขียนคำร้องขออนุญาตให้ทำให้ API ให้ดึงข้อมูล

5.2.2.2.2 ดึงข้อมูลเองโดยใช้ไลบรารีที่ชื่อ ว่า Nightmare JS พร้อมทั้งศึกษาเกี่ยวกับ DOM เพื่อให้ดึงข้อมูลผ่านหน้าเว็บไซต์โดยระบุ Tag HTML ที่เราต้องการลง Firebase

5.2.2.2.3 สรุป

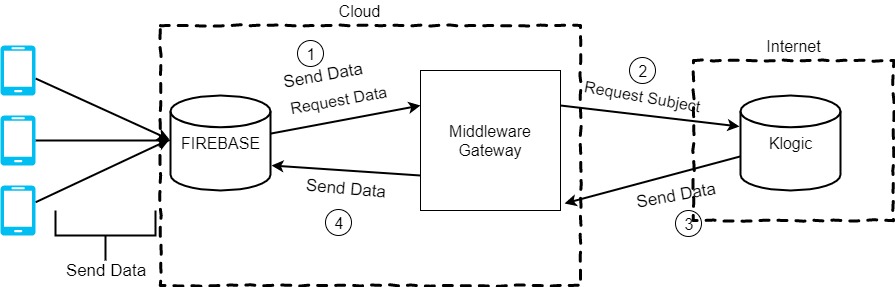
ศึกษา Nightmare JS และ DOM เนื่องจากการเขียนคำร้องขอ API ของKlogic นั้นมีความยุ่งยากแล้วอาจใช้ระยะเวลานานในการทำเรื่องอนุมัติและอาจจะเกิดปัญหาภายหลังได้จึงเลือกใช้วิธีที่ 2

5.2.2.3 เริ่มจากให้ลองใช้งานจาก Google OAuth ก่อนเมื่อทำสำเร็จแล้วจึงปรับและประยุกต์เพื่อให้สามารถเชื่อมต่อเข้ากับ Service Firebase Authentication ได้

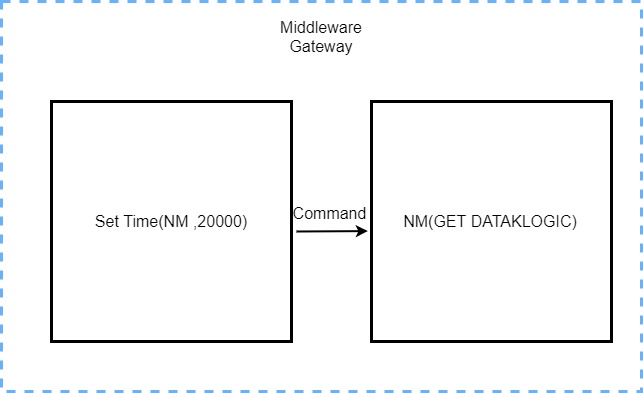
5.2.2.4 ลองมองปัญหาที่เกิดขึ้นใหม่ในที่นี้คือความสำคัญในการส่งข้อมูลกลับมายังตัวกลาง ไม่สามารถทำได้เราจะมองปัญหานี้ลึกลงไปอีกว่า เพราะอะไรถึงเกิดปัญหานี้ เนื่องจาก ขั้นแรกต้องการดึงข้อมูลนักศึกษาที่ลงทะเบียนเรียนแล้วพุชขึ้น Firebase แต่ไม่สามารถทำได้เพราะปัญหาเรื่องการส่งข้อมูลของตัวกลางไม่รู้จักที่เก็บข้อมูลที่ประกาศ จึงหาวิธีแล้วคิดได้ 2 วิธี คือ

5.2.2.4.1 เช็คตำแหน่งที่เก็บ data และ ตำแหน่งที่ส่งข้อมูล

5.2.2.4.2 ปรับกระบวนการคิดใหม่โดยเริ่มแรกต้องการทำทุกคนรอบเดียวแต่เกิดปัญหาจึงเปลี่ยนแนวความคิดให้ทำทีละครั้งครั้งละ 20 วินาที ( เนื่องจากการเรียกข้อมูลจาก Firebase ค่อนข้างใช้เวลา 1-10 วินาทีจากการทดสอบ และการเพิ่มข้อมูลลงไปในฐานข้อมูลของแต่ละคนก็ใช้เวลาเช่นกัน )



**ภาพที่ 5-1** ภาพกระบวนการทำงานในการ GET ข้อมูลจาก Klogic



**ภาพที่ 5-2** ภาพกระบวนการทำงานภายใน Middleware

เลือกวิธีที่ 2 เนื่องจากกระบวนการนี้ยังไม่เคยลองทำจึงลองมองปัญหาใน วิธีที่ 1 แล้วมองหาส่วนที่จะต้องแก้ไม่เจอจึงลองใช้วิธีการเซตเวลาให้ทำงานทุก 20 วินาที แล้วจึงจะรันข้อมูล User ทีละคน ภายใน Middleware ดังภาพที่ 5-2 ก่อนจะจบการทำงาน 1 คนให้เก็บค่าลงฐานข้อมูลดังภาพที่ 5-1

**5.3 ข้อเสนอแนะ**

5.3.1 การทำงานของของโปรแกรมต้องมีการพัฒนาและสามารถยืดหยุ่นได้โดย ผู้พัฒนาต้องเข้าใจถึงโครงสร้างการทำงานของแต่ละส่วนงานรวมถึงเข้าใจหลักการของการเขียนโปรแกรมให้

มากที่สุด

5.3.2 ข้อจำกัดของ GPS การร้องขอตำแหน่งของเครื่องผู้ใช้งานภายในอาคารค่อนข้าง มีข้อจำกัดในเรื่องของสัญญาณการเชื่อมต่อผ่าน Access Point ทำให้เกิดปัญหาความแม่นยำของ GPS แต่ถ้าหากใช้งานสัญญาณ A-GPS (3G,4G) จะมีความแม่นยำมากขึ้น

**บรรณานุกรม**

การตั้งค่า Firebase บน Android. [ออนไลน์][สืบค้นวันที่ 29 สิงหาคม 2560].

จาก https://firebase.google.com/docs/android/setup

การตั้งค่า Real-Time Database บน Android [ออนไลน์][สืบค้นวันที่ 29 สิงหาคม 2560].

จาก https://firebase.google.com/docs/database/android/start/

atkinchric. การเริ่มต้นใช้งาน Nightmare พร้อมคำสั่งการเรียกใช้งาน [ออนไลน์][สืบค้นวันที่ 29 สิงหาคม 2560].จาก https://github.com/segmentio/nightmare

การตั้งค่า Firebase Authentication Google SignIn [ออนไลน์][สืบค้นวันที่ 04 กรกฎาคม 2560].

จาก https://firebase.google.com/docs/auth/android/google-signin

เริ่มต้นการ Get Location Android [ออนไลน์][สืบค้นวันที่ 04 กรกฎาคม 2560].

จาก https://developer.android.com/training/location/retrieve-current.html

เริ่มต้นการใช้งาน Android Studio [ออนไลน์][สืบค้นวันที่ 20 กรกฎาคม 2560].

จาก https://developer.android.com/studio/intro/index.html

การเรียกกลับการเชื่อมโยงบัญชีผู้ใช้. [ออนไลน์][สืบค้นวันที่ 29 สิงหาคม 2559].

จาก https://developers.facebook.com/docs/messenger-platform/webhook-reference/account-linking

Chai Phonbopit.(31 กรกฎาคม 2558) [ออนไลน์] Node คืออะไร + เริ่มต้นการใช้งาน Node Js

[สืบค้นวันที่ 20 กรกฎาคม 2560].จาก https://developers.facebook.com/docs/messenger-platform

Get Mac Address. [ออนไลน์][สืบค้นวันที่ 20 กรกฏาคม 2559].

จาก https://developer.android.com/reference/android/net/wifi/WifiInfo.html

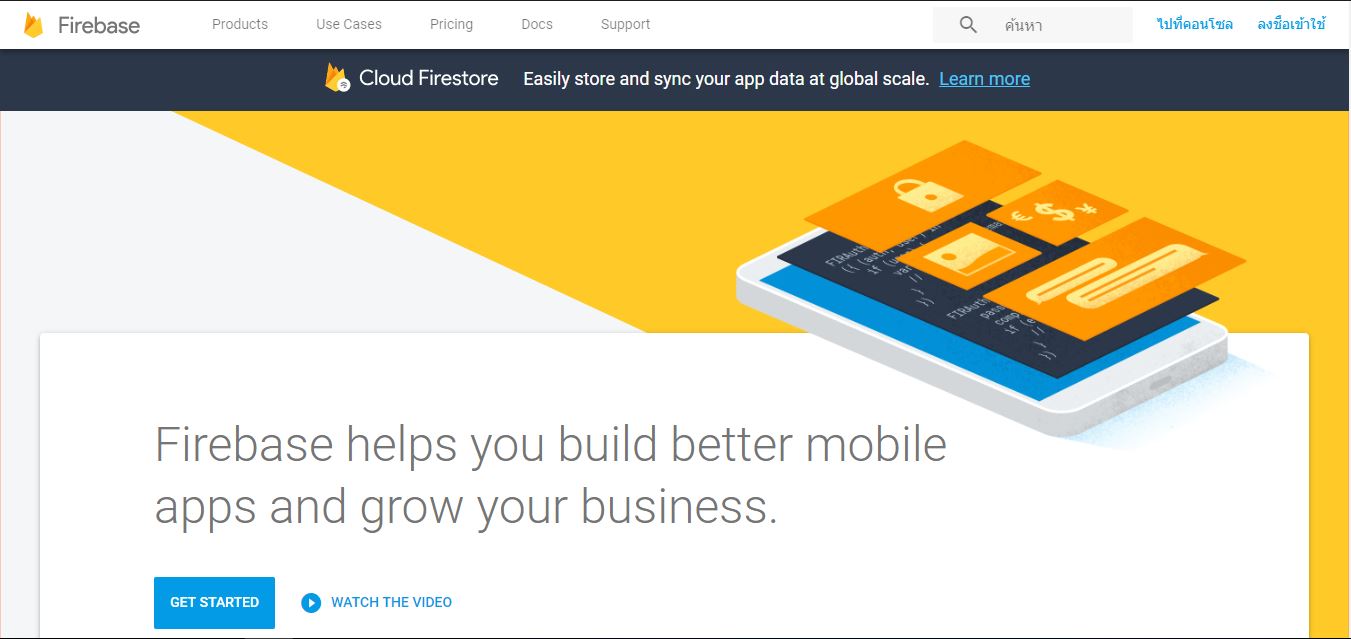
Get Serial Number Android [ออนไลน์][สืบค้นวันที่ 20 กรกฎาคม 2560].จาก

https://developer.android.com/reference/android/provider/Settings.Secure.html#ANDROID\_ID

ภาคผนวก ก

วิธีใช้งาน Firebase เริ่มต้น และ เชื่อมต่อเข้ากับ Android Studio

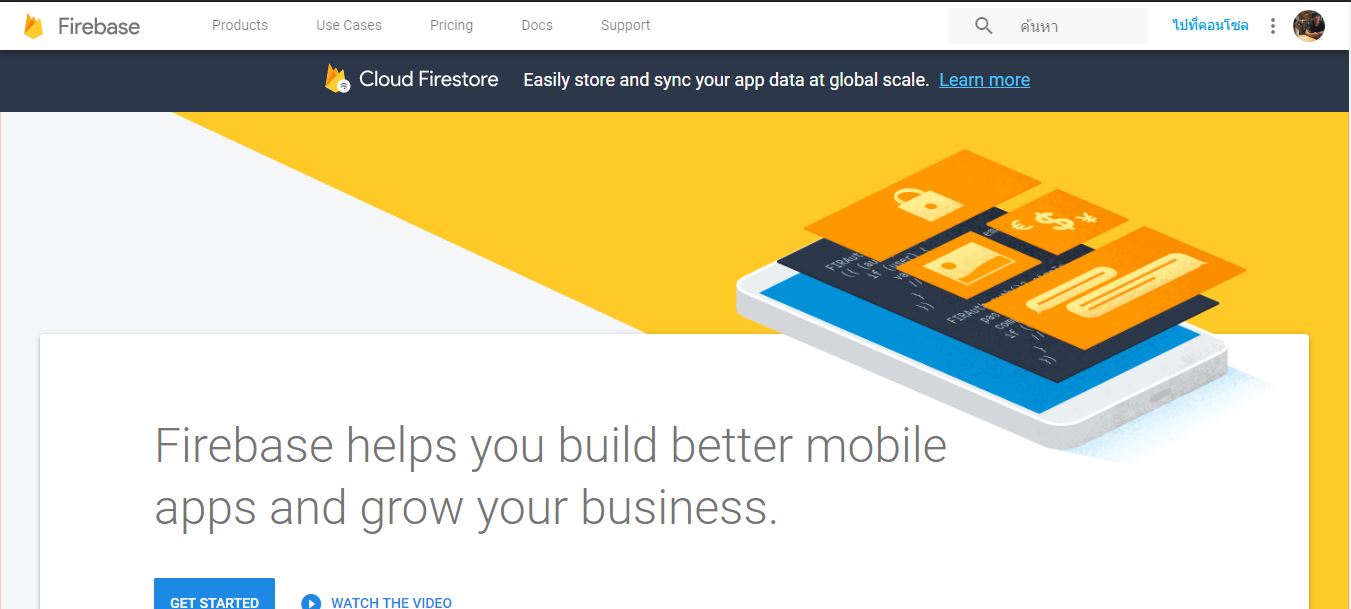
**วิธีใช้งาน Firebase เริ่มต้น และ เชื่อมต่อเข้ากับ Android Studio**

****

**ภาพที่ ก-1** เว็บไซต์ Firebase

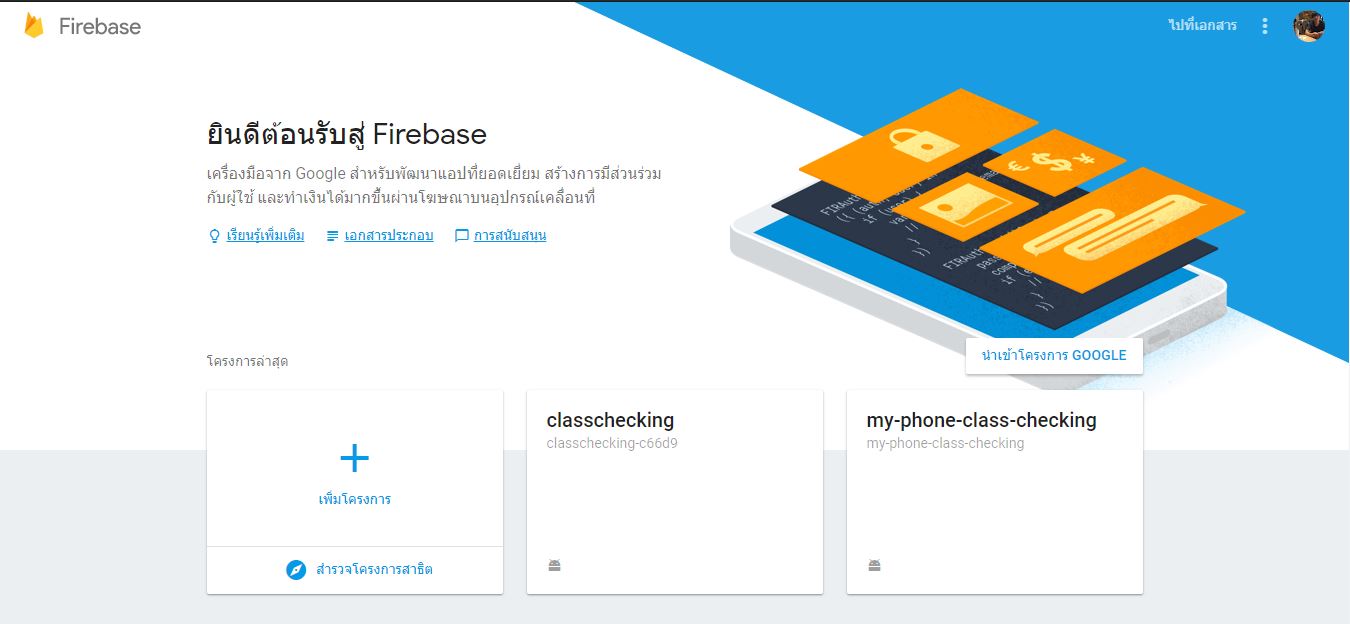
จากภาพที่ ก-1เริ่มต้นการใช้งาน Firebase โดยเข้าไปที่ https://firebase.google.com

คลิก ลงชื่อเข้าใช้ ( ระบบจะให้เข้าใช้งานด้วย Gmail )



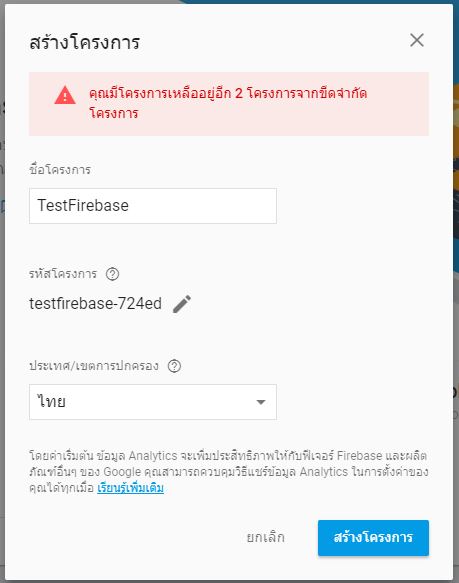
**ภาพที่ ก-2** เว็บไซต์ Firebase

จากภาพที่ ก-2หลักจากที่เข้าสู่ระบบเรียบร้อยแล้ว จะเข้าสู่หน้าจอแบบภาพ ก-2 เลือก ไปที่คอนโซล ( Go to console )



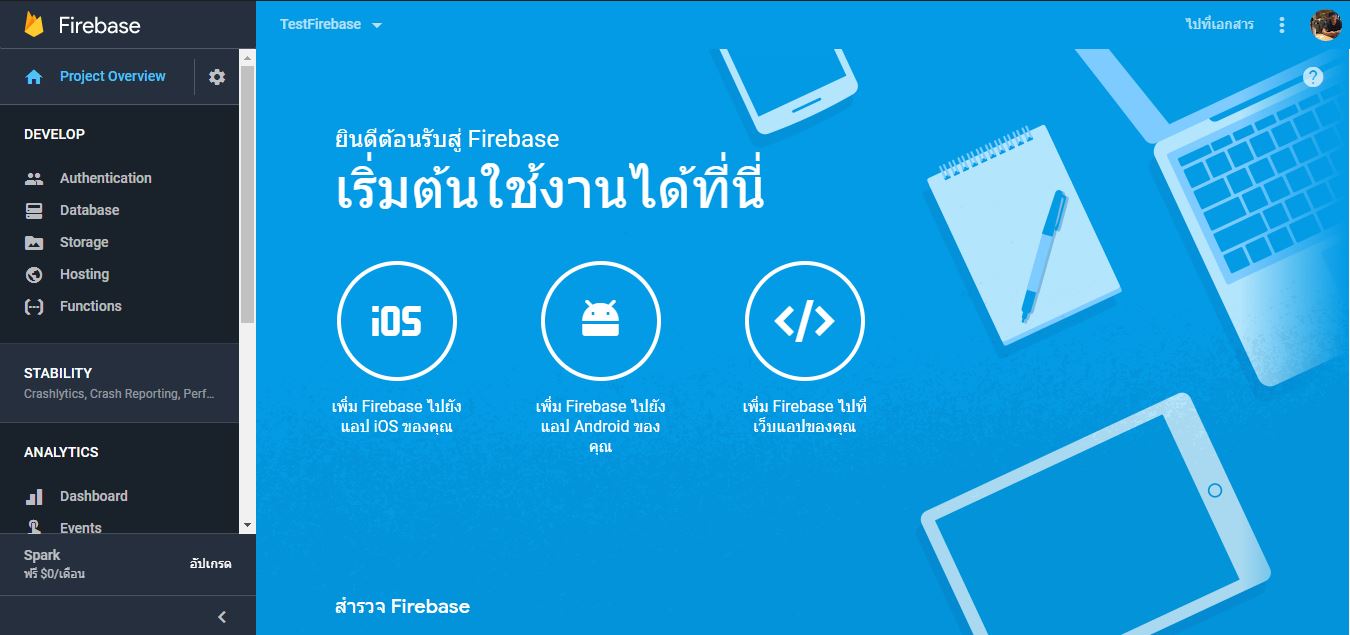
**ภาพที่ ก-3** เริ่มการสร้างฐานข้อมูล Firebase

จากภาพที่ ก-3ในส่วนนี้จะมีทางให้เลือก 2 ทาง คือ นำเข้าโครงการ และสร้างโครงการใหม่ ในส่วนนี้จะสร้างโครงการใหม่ คลิกที่ เพิ่มโครงการ

****

**ภาพที่ ก-4** กรอกชื่อโปรเจคที่ต้องการ

จากภาพที่ ก-4เมื่อคลิกเพิ่มโครงการแล้วจะเข้าสู่หน้าจอรูปที่ ก-4 ใส่ชื่อโครงการตามที่ต้องการ จะเลือกประเทศหรือไม่ก็ได้ แล้วคลิกที่สร้างโครงการ



**ภาพที่ ก-5** แสดงหน้าจอเริ่มต้นการใช้งานบน Firebase

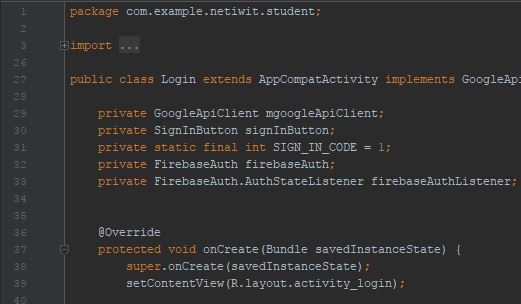
จากภาพที่ ก-5เมื่อสร้างโครงการสำเร็จจะเข้าสู่หน้าจอเริ่มต้นใช้งาน สามารถเลือกใช้งานได้ตามที่ต้องการ

**เริ่มต้นการเชื่อมต่อเข้ากับ Android Studio**

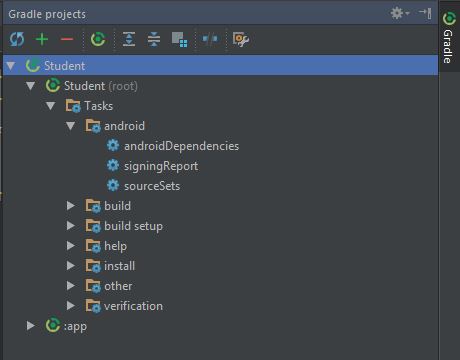
****

**ภาพที่ ก-6** กรอกชื่อแพ็กเกจโปรเจคที่ต้องการผูกกับ Firebase

จากภาพที่ ก-6เมื่อคลิกที่ เพิ่ม Firebase ไปยังแอป Android จะปรากฏหน้าต่างหน้าบน ให้ใส่ชื่อแพ็กเกจจากโปรเจคที่เราสร้างไว้บน Android และ ส่วนอื่นจะใส่หรือไม่ก็ได้ ( หากต้องการใช้ Google Authentication ให้ใส่ มิฉะนั้นจะไม่สามารถใช้งานได้ )

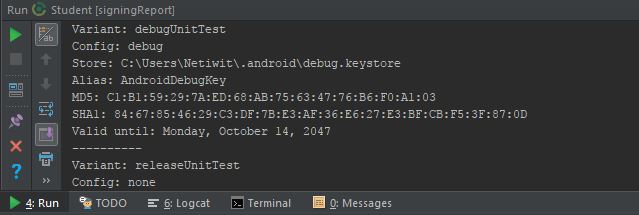


**ภาพที่ ก-7** แสดงตำแหน่งชื่อ Package จะอยู่บรรทัดที่ 1 เสมอ( .java )

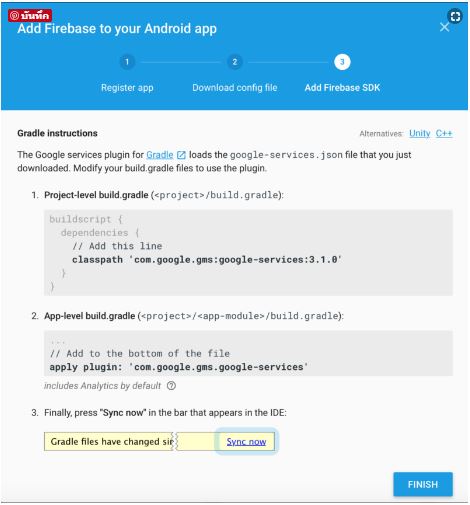
****

**ภาพที่ ก-8** แสดงเมนูการทำงานวิธีการดู SHA-1 บน android

จากภาพที่ ก-8 เมนูดังภาพจะอยู่ซ้ายสุดของโปรแกรม Android Studio ( หากเปิดแล้ว ไม่เป็นดังรูปให้กด ลูกศรสีฟ้าด้านบน ) จากนั้น เลือกเมนูที่มีคำว่า signingReport



**ภาพที่ ก-9** แสดงการทำงานของเมนู signingReport

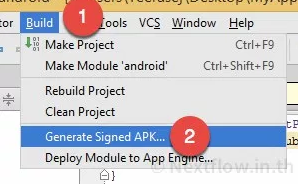


**ภาพที่ ก-10** แสดง Codeให้เพิ่มลง โปรเจค Android Studio

จากภาพที่ ก-10ให้นำโค้ดด้านบนเพิ่มตามตำแหน่งตามที่ Firebase บอกไว้ จากนั้นกดที่ Sync Now เป็นอันเสร็จสิ้นการเชื่อม Firebase เข้ากับ Android Studio

**การทำ Compile Android เพื่อให้ได้ APK**

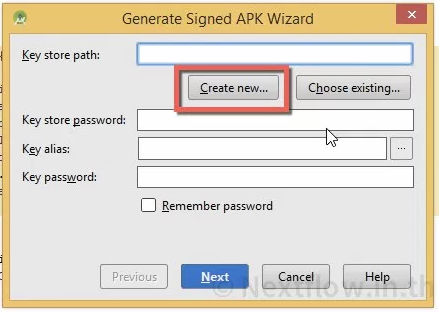
1. ไป Build > Export Signed APK



**ภาพที่ ก-11** แสดงจอการเข้าสู่เมนู Export Signed APK

1. สร้างหรือเปิดใช้งานไฟล์ Key Stone

คลิกที่ Create new…

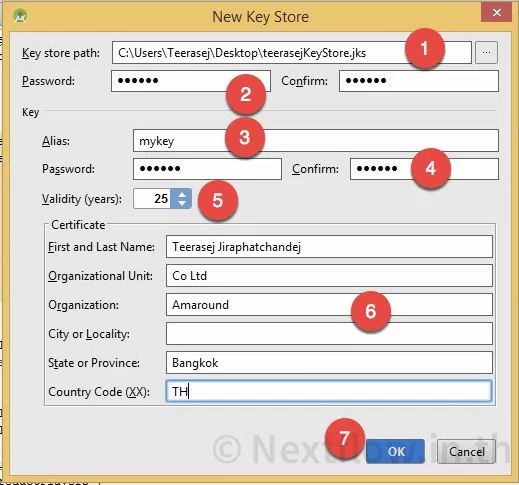


**ภาพที่ ก-12** แสดงการสร้างหรือเปิดใช้งานไฟล์ Key

จะพบกับหน้าต่างของ New Key ให้กดปุ่ม browse เพื่อเลือก folder ไว้เก็บไฟล์ Key Store ที่กำลังจะสร้าง



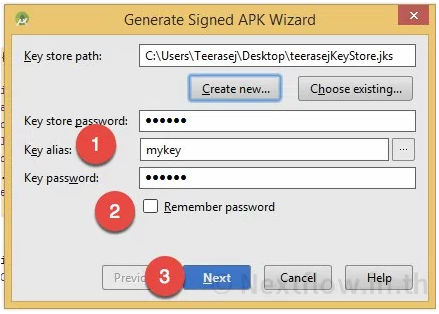
**ภาพที่ ก-13** แสดงการสร้างหรือเปิดใช้งานไฟล์ Key (ต่อ)



**ภาพที่ ก-14** แสดงการสร้างหรือเปิดใช้งานไฟล์ Key (ต่อ)

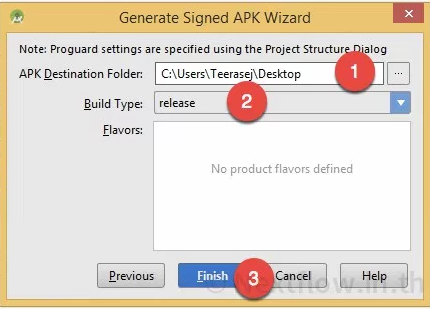
* 1. ที่อยู่ของไฟล์ Key Store ที่เราเลือกก่อนหน้านี้
  2. ตั้งรหัสให้ไฟล์ Key Store
  3. สร้าง Key Alias
  4. ตั้งรหัสให้ Key Alias
  5. อายุการใช้งาน
  6. กรอกข้อมูลที่เกี่ยวข้อง (ไม่จำเป็นต้องกรอกหมด)
  7. กดปุ่ม OK

1. เมื่อเสร็จแล้วจะกลับมาที่หน้า Generate Signed APK Wizard ซึ่งข้อมูลที่สร้างเสร็จ ทางด้านบนจะขึ้นมาให้อัตโนมัติ



**ภาพที่ ก-15** แสดงหน้าจอการสร้าง APK

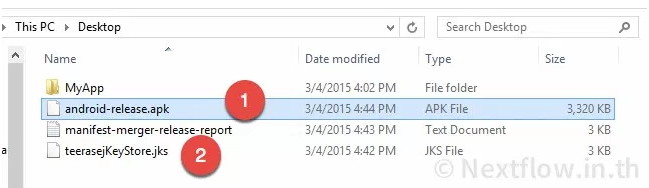
1. เมื่อทำการคลิกที่ปุ่ม Next โปรแกรมจะพามายังขั้นตอนสุดท้าย



**ภาพที่ ก-16** แสดงขั้นตอนสุดท้ายของการตั้งค่า APK

* 1. ตำแหน่งของไฟล์ APK ที่เลือกไว้จะไปอยู่ในนั้น
  2. เลือก Build Type เป็น release
  3. กดปุ่ม Finish

1. เมื่อเข้าไปที่ตำแหน่งที่เรากำหนดไว้จะพบกับไฟล์ APK ที่เราสร้างเก็บไว้ในนี้เป็นอันเสร็จสิ้น

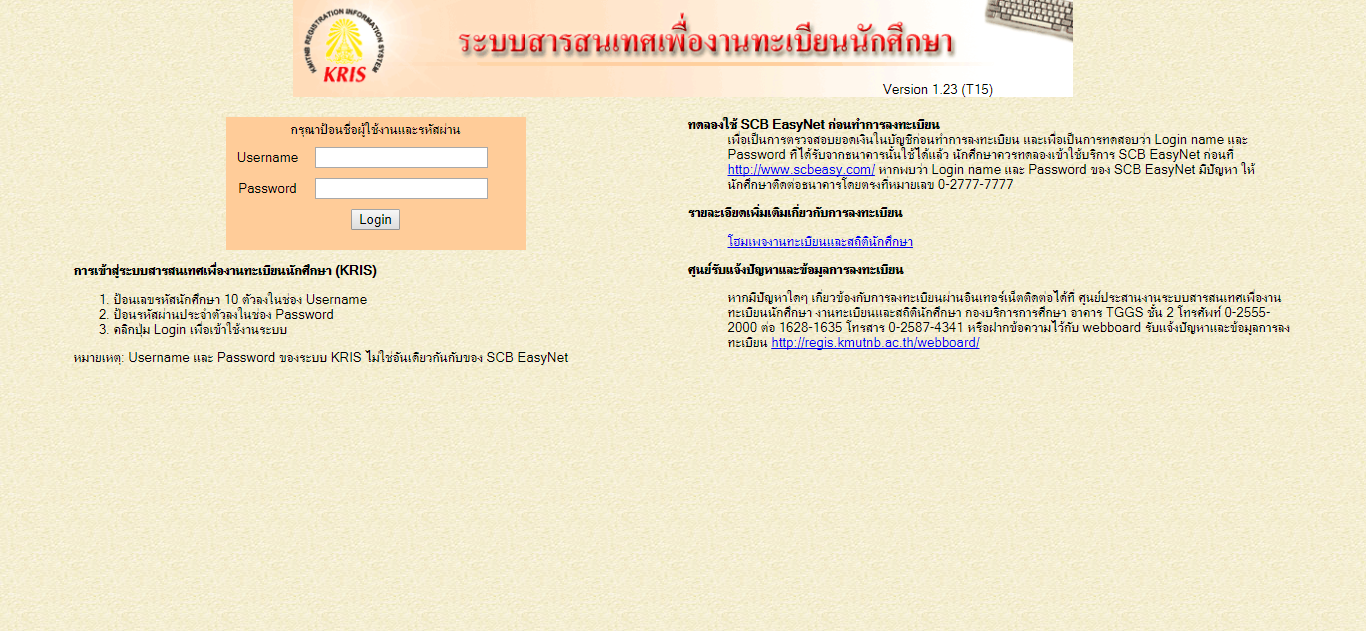


**ภาพที่ ก-17** แสดงที่อยู่ของ ไฟล์ APK ที่สร้างไว้

* 1. ไฟล์ Android Application (APK)
  2. ไฟล์ Key Store

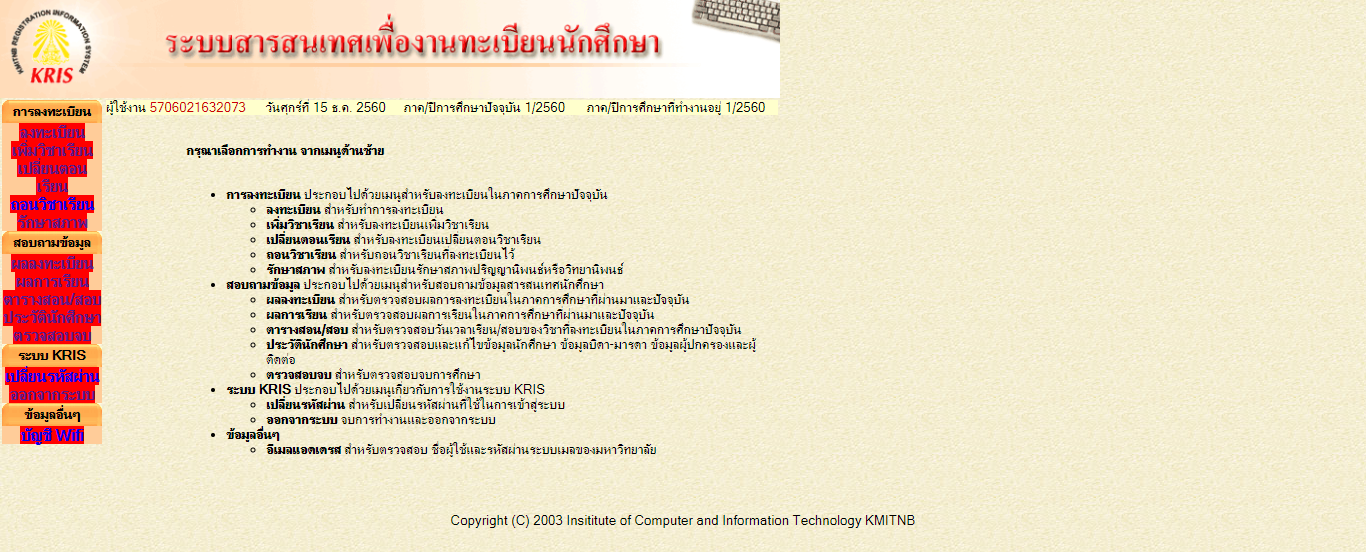
**การเข้าใช้งาน Klogic**

Klogic คือ ระบบสารสนเทศเพื่องานทะเบียนนักศึกษา ไว้สำหรับลงทะเบียนเรียน ดูข้อมูล การลงทะเบียน ดูตารางสอน จะสามารถเข้าใช้งานได้ต่อเมื่อเป็นนักศึกษาของมหาวิทยาลัยนี้ก่อน



**ภาพที่ ก-18** แสดงเว็บไซต์ Klogic http://klogic.kmutnb.ac.th:8080/kris/index.jsp

สำหรับนักศึกษา Username จะใช้เป็นรหัสนักศึกษา Password เริ่มต้นจะเป็น รหัสบัตรประชาชนของนักศึกษาแต่ละคน



**ภาพที่ ก-19** แสดงการเข้าใช้งานด้วยรหัสของนักศึกษา

เมื่อเข้าใช้สำหรับนักศึกษา จะมีเมนูต่าง ๆ ให้เลือกทางด้านซ้าย ดังนี้

1. ลงทะเบียน สำหรับทำการลงทะเบียน
   1. เพิ่มวิชาเรียน สำหรับลงทะเบียนเพิ่มวิชาเรียน
   2. เปลี่ยนตอนเรียน สำหรับลงทะเบียนเปลี่ยนตอนวิชาเรียน
   3. ถอนวิชาเรียน สำหรับถอนวิชาเรียนที่ลงทะเบียนไว้
   4. รักษาสภาพ สำหรับลงทะเบียนรักษาสภาพปริญญานิพนธ์หรือวิทยานิพนธ์
2. สอบถามข้อมูล ประกอบไปด้วยเมนูสำหรับสอบถามข้อมูลสารสนเทศนักศึกษา
   1. ผลลงทะเบียน สำหรับตรวจสอบผลการลงทะเบียนในภาคการศึกษาที่ผ่านมาและ ปัจจุบัน
   2. ผลการเรียน สำหรับตรวจสอบผลการเรียนในภาคการศึกษาที่ผ่านมาและปัจจุบัน
   3. ตารางสอน/สอบ สำหรับตรวจสอบวันเวลาเรียน/สอบของวิชาที่ลงทะเบียนในภาค การศึกษาปัจจุบัน
   4. ประวัตินักศึกษา สำหรับตรวจสอบและแก้ไขข้อมูลนักศึกษา ข้อมูลบิดา-มารดา ข้อมูลผู้ปกครองและผู้ติดต่อ
   5. ตรวจสอบจบ สำหรับตรวจสอบจบการศึกษา
3. ระบบ KRIS ประกอบไปด้วยเมนูเกี่ยวกับการใช้งานระบบ KRIS
   1. เปลี่ยนรหัสผ่าน สำหรับเปลี่ยนรหัสผ่านที่ใช้ในการเข้าสู่ระบบ
   2. ออกจากระบบ จบการทำงานและออกจากระบบ
4. ข้อมูลอื่น ๆ
   1. อีเมลแอดเดรส สำหรับตรวจสอบ ชื่อผู้ใช้และรหัสผ่านระบบเมลของมหาวิทยาลัย



**ภาพที่ ก-20** แสดงการเข้าใช้งานด้วยรหัสของอาจารย์

เมื่อเข้าใช้สำหรับอาจารย์ จะมีเมนูต่าง ๆ ให้เลือกทางด้านซ้าย ดังนี้

1. สอบถามข้อมูล ประกอบไปด้วยเมนูสำหรับสอบถามข้อมูลสารสนเทศ
   1. ผลลงทะเบียน สำหรับตรวจสอบผลการลงทะเบียนในภาคการศึกษาปัจจุบัน
   2. ผลการเรียน สำหรับตรวจสอบผลการเรียนในภาคการศึกษาที่ผ่านมาและปัจจุบันของ นักศึกษา
   3. ตารางสอน/สอบ สำหรับตรวจสอบวันเวลาเรียน/สอบของวิชาที่ลงทะเบียนใน ภาคการศึกษาปัจจุบันของนักศึกษา
   4. ประวัตินักศึกษา สำหรับตรวจสอบและแก้ไขข้อมูลนักศึกษา ข้อมูลบิดา-มารดา ข้อมูลผู้ปกครองและผู้ติดต่อของนักศึกษา
   5. ตรวจสอบจบ สำหรับตรวจสอบจบการศึกษาของนักศึกษา
   6. KMIT-21 รายงานสรุปจำนวนหน่วยกิตที่นักศึกษาลงทะเบียน KMIT-21
   7. KMIT-22 รายงานสรุปจำนวนหน่วยกิตที่นักศึกษาลงทะเบียน KMIT-22
   8. KMIT-34 รายงานสรุปจำนวนหน่วยกิตที่นักศึกษาลงทะเบียน KMIT-34
   9. KMIT-35 รายงานสรุปจำนวนหน่วยกิตที่นักศึกษาลงทะเบียน KMIT-35
   10. KMIT-35e รายงานสรุปจำนวนหน่วยกิตที่นักศึกษาลงทะเบียน KMIT-35e
   11. KMIT-36 รายงานสรุปจำนวนหน่วยกิตที่นักศึกษาลงทะเบียน KMIT-36
2. ระบบ KRIS ประกอบไปด้วยเมนูเกี่ยวกับการใช้งานระบบ KRIS
   1. เปลี่ยนเทอม สำหรับเปลี่ยนภาคการศึกษา
   2. เปลี่ยนรหัสผ่าน สำหรับเปลี่ยนรหัสผ่านที่ใช้ในการเข้าสู่ระบบ
   3. ออกจากระบบ จบการทำงานและออกจากระบบ

ภาคผนวก ข

การทดสอบความแม่นยำของ GPS

**การทดสอบความแม่นยำของ GPS**

จากการเขียนโปรแกรมจึงทำให้ต้องมีการทดสอบระยะของ GPS ซึ่งผู้พัฒนาโปรแกรมได้ทดสอบระยะห่างในแต่ละช่วง และได้ผลสรุปออกมาเป็นตารางเปรียบเทียบความแม่นยำของ GPS ในแต่ละช่วง โดยให้ตัวเครื่องอาจารย์อยู่ที่ชั้น 1 ตลอดการทดสอบ ดังนี้

**ตารางที่ 1** *ตารางจำลองตำแหน่งของการทดสอบของอาจารย์และนักศึกษา*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ชั้น 4 |  | นักศึกษาห่าง 10 เมตร | นักศึกษาห่าง 20เมตร | นักศึกษาห่าง 30 เมตร |
| ชั้น 3 |  | นักศึกษาห่าง 10 เมตร | นักศึกษาห่าง 20 เมตร | นักศึกษาห่าง 30 เมตร |
| ชั้น 2 |  | นักศึกษาห่าง 10 เมตร | นักศึกษาห่าง 20 เมตร | นักศึกษาห่าง 30 เมตร |
| ชั้น 1 | อาจารย์ | นักศึกษาห่าง 10 เมตร | นักศึกษาห่าง 20 เมตร | นักศึกษาห่าง 30 เมตร |

Wi-Fi

**ตารางที่ 2** *ตารางเปรียบเทียบการทดสอบความแม่นยำของ GPS ผ่าน Wi-F*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ระยะใกล้  ( 10 เมตร ) | ระยะกลาง  ( 30 เมตร ) | ระยะไกล  ( 50 เมตร ) | หมายเหตุ |
| ชั้นที่ 1 | ผ่าน | ผ่าน | ผ่าน | - |
| ชั้นที่ 2 | ผ่าน | ไม่ผ่าน | ไม่ผ่าน | - |
| ชั้นที่ 3 | ผ่าน | ไม่ผ่าน | ไม่ผ่าน | - |
| ชั้นที่ 4 | ไม่ผ่าน | ไม่ผ่าน | ไม่ผ่าน | - |

3G/4G

**ตารางที่ 3** *ตารางเปรียบเทียบการทดสอบความแม่นยำของ GPS ผ่าน 3G/4G*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ระยะใกล้  ( 10 เมตร ) | ระยะกลาง  ( 30 เมตร ) | ระยะไกล  ( 50 เมตร ) | หมายเหตุ |
| ชั้นที่ 1 | ผ่าน | ผ่าน | ไม่ผ่าน | - |
| ชั้นที่ 2 | ผ่าน | ผ่าน | ไม่ผ่าน | - |
| ชั้นที่ 3 | ผ่าน | ไม่ผ่าน | ไม่ผ่าน | - |
| ชั้นที่ 4 | ผ่าน | ไม่ผ่าน | ไม่ผ่าน | - |

**หมายเหตุ:** ทดสอบที่หอพักชาย มหาวิทยาลัยพระจอมเกล้าพระนครเหนือ วิทยาเขตปราจีนบุรี

ตารางเปรียบเทียบละติจูด และ ลองจิจูด ตามลำดับบรรทัดในแต่ละขั้นในแต่ละช่วง เช่น ชั้นที่ 1 ระยะ 10 เมตร ละติจูด 14.1656333114 และ ลองจิจูด 101.345345321

Wi-Fi

**ตารางที่ 4** *ตารางเปรียบเทียบละติจูด และ ลองจิจูด ผ่าน Wi-Fi*

|  |  |  |  |
| --- | --- | --- | --- |
|  | ระยะใกล้  ( 10 เมตร ) | ระยะกลาง  ( 30 เมตร ) | ระยะไกล  ( 50 เมตร ) |
| ชั้นที่ 1 | 14.1656333114 101.345345321 | 14.16562125 101.34534836 | 14.165836743 101.345392185 |
| ชั้นที่ 2 | 14.165633351 101.34534323 | 14.165881421 101.345383241 | 14.165887982 101.345394865 |
| ชั้นที่ 3 | 14.165633352 101.34534991 | 14.165871325 101.345383851 | 14.165882314 101.345395678 |
| ชั้นที่ 4 | 14.16588152 101.345393251 | 14.165891521 101.34538256 | 14.165885861 101.345395637 |

3G/4G1

**ตารางที่ 5** *ตารางเปรียบเทียบละติจูด และ ลองจิจูด ผ่าน 3G/4G*

|  |  |  |  |
| --- | --- | --- | --- |
|  | ระยะใกล้  ( 10 เมตร ) | ระยะกลาง  ( 30 เมตร ) | ระยะไกล  ( 50 เมตร ) |
| ชั้นที่ 1 | 14.16563311 101.34534393 | 14.16562321 101.34534871 | 14.165882840 101.345395683 |
| ชั้นที่ 2 | 14.165633453 101.34534478 | 14.165633432 101.34534154 | 14.165882313 101.345395670 |
| ชั้นที่ 3 | 14.165633435 101.34534127 | 14.165871325 101.345383851 | 14.165882313 101.345395680 |
| ชั้นที่ 4 | 14.165633342 101.34534928 | 14.165891521 101.34538256 | 14.165885861 101.345395637 |

**หมายเหตุ:** ทดสอบที่หอพักชาย มหาวิทยาลัยพระจอมเกล้าพระนครเหนือ วิทยาเขตปราจีนบุรี

**สูตรคำนวณระยะห่าง**

lat1\*pi/180

lat2\*pi/180

lon1\*pi/180

lon2\*pi/180

dlon = lon2 - lon1

dlat = lat2 - lat1

a = (sin(dlat/2))^2 + cos(lat1) \* cos(lat2) \* (sin(dlon/2))^2

c = 2 \* atan2( sqrt(a), sqrt(1-a) )

d = (R \* c )\*1000 (R = 6371)