การพัฒนาระบบการแจ้งเตือนการรับประทานยา
Developing the Medicine Intake Reminder System

นางสาวอัญมณี เนียมหอม
Anyamanee Niemhom
นายศรายุทธ ยินดี
Sarayut Yindi

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมชีวการแพทย์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

การพัฒนาระบบการแจ้งเตือนการรับประทานยา

Developing the Medicine Intake Reminder System

นางสาวอัญมณี เนียมหอม รหัสประจำตัว 57011520

นายศรายุทธ ยินดี รหัสประจำตัว 57011571

อาจารย์ที่ปรึกษา

รศ.ดร. ชูชาติ ปิณฑวิรุจน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมชีวการแพทย์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ปริญญานิพนธ์ สาขาวิศวกรรมชีวการแพทย์ ปีการศึกษา 2560

สาขาวิชา วิศวกรรมชีวการแพทย์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

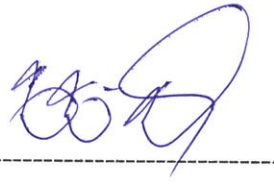
เรื่อง การพัฒนาระบบการแจ้งเตือนการรับประทานยา

Developing the Medicine Intake Reminder System

ผู้จัดทำ นางสาวอัญมณี เนียมหอม รหัสประจำตัว 57011520

นายศรายุทธ ยินดี รหัสประจำตัว 57011571

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว



(รศ.ดร.ชูชาติ ปิณฑวิรุจน์)

อาจารย์ที่ปรึกษาปริญญานิพนธ์

หัวข้อโครงงาน การพัฒนาระบบการแจ้งเตือนการรับประทานยาของผู้ป่วย

ผู้จัดทำ นางสาวอัญมณี เนียมหอม รหัสนักศึกษา 57011520

นายศรายุทธ ยินดี รหัสนักศึกษา 57011571

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมชีวการแพทย์

ปีการศึกษา 2560

อาจารย์ที่ปรึกษา รศ. ดร. ชูชาติ ปิณฑวิรุจน์

บทคัดย่อ

การพัฒนาระบบการแจ้งเตือนการรับประทานยาของผู้ป่วยมีวัตถุประสงค์เพื่อที่จะส่งเสริมความถูกต้องในการรับประทานยาของผู้ป่วยสูงอายุ เนื่องจากการรับประทานยานั้นส่งผลโดยตรงกับประสิทธิภาพในการรักษาทางการแพทย์ ทางผู้วิจัยจึงได้นำเทคโนโลยีเข้ามาพัฒนากล่องยาควบคู่กับแอพพลิเคชั่นแอนดรอยด์ กล่องยาได้ถูกพัฒนาโดยไมโครคอนโทรลเลอร์เพื่อตั้งเวลาแจ้งเตือนการรับประทานยาของผู้สูงอายุ เมื่อถึงเวลาแจ้งเตือนจะมีสัญญาณเสียงและไฟแสดงบนแต่ละช่องของกล่องยาให้ผู้ป่วยรับรู้ ในช่วงเวลามื้ออาหารเช้า กลางวัน เย็น และก่อนนอน การแจ้งเตือนจะสิ้นสุดเมื่อไมโครคอนโทรลเลอร์ตรวจจับการเปิดฝาของกล่องยา วันและเวลาการเปิดฝากล่องยาที่แสดงถึงการรับประทานยาของผู้ป่วยนี้ยังเก็บไว้เป็นประวัติสำหรับให้แพทย์ประเมินความครบถ้วนถูกต้องในการรับประทานยาของผู้ป่วยได้ ข้อมูลทั้งหมดนี้จะถูกเก็บไว้บนฐานข้อมูลออนไลน์ไฟร์เบส(Firebase) ที่เป็นตัวกลางเชื่อมต่อข้อมูลระหว่างกล่องยาและแอพพลิเคชั่น การแจ้งเตือนจึงสามารถแสดงบนแอพพลิเคชั่นแอนดรอยด์ที่ถูกพัฒนาโดยโปรแกรมแอนดรอยด์สตูดิโอ(Android Studio) ได้ด้วย นอกจากนี้แอพพลิเคชั่นยังแสดงข้อมูลต่างๆ ได้แก่ แสดงข้อมูลทั่วไปของผู้ป่วย ข้อมูลยาของผู้ป่วย ประวัติการรับประทานยาที่มาจากการตรวจจับการเปิดฝาของกล่องยา โดยข้อมูลทางการแพทย์เหล่านี้สามารถเพิ่มหรือแก้ไขผ่านฐานข้อมูลออนไลน์ไฟร์เบสโดยแพทย์หรือบุคลากรทางการแพทย์ได้

Thesis Title	Developing the Medicine Intake Reminder System	
Student	Miss Anyamanee Niemhom	**ID** 57011520
	Mr. Sarayuth Yindi	**ID** 57011571
Degree	Bachelor of Engineering	
Major	Biomedical Engineering	
Year	2017	
Thesis Advisor	Assoc. Prof. Dr. Chuchart Pintavirooj	

Abstract

Developing the Medicine Intake Reminder System has the objectives to support medicine intake in elderly patients because medicine intake directly affects with efficiency of treatment. Our project applied technology to develop the pill box with android application. The pill box has a microcontroller which can set the time at breakfast, lunch, dinner and before going to bed. This pill box can remind medicine taking time to the elderly patient by sound and LED in each channel of the pill box. The reminder would dismiss when the patient opens the pill box's cover. After that, the time and date of the opening pill box's cover are collected on history. All of the data are collected on "Firebase" real-time database which connected the android application to the pill box. So, the reminder can show on the android application which developed by Android Studio Program. Besides, the application could display the patient information — general information, medicine information and history of opening pill box's cover. The doctor can add or edit these data via Firebase.

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงได้ด้วยดี ด้วยความช่วยเหลือจาก รศ.ดร.ชูชาติ ปิณฑวิรุจน์ อาจารย์ที่ปรึกษา และนักศึกษาชั้นปริญญาโท ปริญญาตรี สาขาวิศวกรรมชีวการแพทย์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้แนวคิดในการทำวิทยานิพนธ์ ข้อเสนอแนะ และแก้ไขข้อบกพร่องต่าง ๆ ทำให้งานออกมาสมบูรณ์ ผู้วิจัยจึงขอกราบขอบพระคุณเป็นอย่างสูง นอกจากนี้ขอขอบพระคุณผู้แต่งหนังสือ บทความ คลิปวีดีโอ และงานวิจัยต่าง ๆ ที่ได้แบ่งปันเพื่อการศึกษา ทำให้ผู้วิจัยได้นำมาใช้ประโยชน์ในงานได้อย่างครบถ้วน

สุดท้ายนี้ผู้วิจัยขอขอบพระคุณบิดามารดา และครอบครัว ซึ่งส่งเสียให้ได้รับการศึกษาเล่าเรียน ตลอดจนคอยช่วยเหลือและให้กำลังใจผู้วิจัยเสมอมาจนสำเร็จการศึกษา

อัญมณี เนียมหอม

ศรายุทธ ยินดี

สารบัญ

หน้า

บทคัดย่อ.....i

Abstract.....ii

กิตติกรรมประกาศ.....iii

สารบัญ.....iv

สารบัญตาราง.....xi

สารบัญรูปภาพ.....xiii

บทที่ 1 บทนำ

1.1. ความเป็นมาและความสำคัญของปัญหา.....1

1.2. วัตถุประสงค์.....2

1.3. สมมติฐาน.....2

1.4. ขอบเขตของงานวิจัย.....2

1.5. แผนการดำเนินงาน.....3

1.6. ประโยชน์ที่คาดว่าจะได้รับ.....4

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1 งานวิจัยและบทความ.....6

2.1.1 งานวิจัยที่เกี่ยวข้อง.....6

2.1.1.1 ผลการศึกษางานวิจัย.....6

2.1.1.2 ข้อสรุปของงานวิจัย.....7

2.1.2 หลักการและหลักปฏิบัติทั่วไปของการทานยา 8

2.1.2.1 ยาก่อนอาหาร....8

2.1.2.2 ยาหลังอาหาร....8

2.1.2.3 ยาก่อนนอน....9

2.1.2.4 ยาสำหรับทานเมื่อมีอาการ....9

2.2 ระบบแอนดรอยด์ (Android)....10

2.2.1 เกี่ยวกับระบบแอนดรอยด์....10

2.2.2 โครงสร้างของระบบแอนดรอยด์....10

2.2.3 เวอร์ชั่นและ API Level ของแอนดรอยด์....10

2.3 Android Studio....11

2.3.1 การใช้งานของแถบเครื่องมือต่าง ๆ....13

2.3.2 Activity ของแอนดรอยด์....14

2.2.2.1 วงจรสถานะของ Activity....15

2.2.2.2 Callback Method ของ Activity....17

2.2.2.3 การ Override เมธอดของ Activity....17

2.3.3 พื้นฐานเกี่ยวกับ Android Application....18

2.3.4 แอตทริบิวต์พื้นฐานของ View....19

2.3.5 การแบ่งกลุ่มของ View....21

2.3.6 การแจ้งเตือน....22

2.4 Microcontroller board....32

2.4.1 Arduino Uno....32

2.4.2 Arduino Mega....36

2.4.2.1 Power 37

2.4.2.2 Memory 38

2.4.2.3 ฟังก์ชันอื่น ๆ เพิ่มเติม 38

2.4.3 Wemos D1 Mini 39

2.5 การรับค่าเวลา.....39

2.5.1 การรับค่าเวลาจาก RTC DS3231.....39

2.5.2 การรับค่าเวลา Universal Time ด้วย Node MCU 42

2.6. รหัสแทนข้อมูล.....44

2.6.1 รหัส BCD – 8421.....45

2.6.2 รหัสเลขฐานแปดและเลขฐานสิบหก.....46

2.6.3 การใช้พาริตี้ในรหัส.....46

2.6.4 รหัสเกิน 3.....47

2.6.5 รหัสเกรย์.....48

2.6.6 รหัสแทนข้อมูลในคอมพิวเตอร์.....50

2.6.7 เทคโนโลยีในการรักษาความปลอดภัยของข้อมูล.....50

2.6.8 ใบรับรองดิจิตอล.....53

2.7 การรับส่งข้อมูล.....54

2.7.1 การรับส่งข้อมูลแบบใช้สาย.....54

2.7.1.1 สายตีเกลียวคู่.....54

2.7.1.2. สายโคแอกซ์.....55

2.7.1.3 สายใยแก้วนำแสง หรือเส้นใยแก้วนำแสง.....55

2.7.2 การรับส่งข้อมูลแบบไร้สาย....56
2.7.2.1 อินฟราเรด (Infrared)....56
2.7.2.2 คลื่นวิทยุ (radio frequency)....57
2.7.2.3 ไมโครเวฟ (microwave)....57
2.7.2.4 ดาวเทียม (satellite)....58
2.8 ระบบจัดการข้อมูล....58
2.8.1 ฐานข้อมูล (Data base)....58
2.8.1.1 สาระสำคัญ....58
2.8.1.2 ความรู้พื้นฐานเกี่ยวกับระบบฐานข้อมูล....59
2.8.1.3 นิยามและคำศัพท์พื้นฐานเกี่ยวกับระบบฐานข้อมูล....59
2.8.1.4 ความสำคัญของการประมวลผลแบบระบบฐานข้อมูล....61
2.8.1.5 รูปแบบของระบบฐานข้อมูล....62
2.8.1.6 โปรแกรมฐานข้อมูลที่นิยมใช้....63
2.8.2 Server....64
2.8.2.1 Server คืออะไร....64
2.8.2.2 ประเภทของเครื่อง Server....64
2.8.2.3 ระบบปฏิบัติการที่ใช้ในเครื่อง Server....65
2.8.2.4 หน้าที่ของ Server....65
2.8.2.5 ประโยชน์ Server....66
2.8.2.6 ประเภทของ Server....66
2.9 ฐานข้อมูลออนไลน์ไฟร์เบส (Firebase)....67

2.9.1 Firebase Realtime Database....69

2.9.2 ไฟร์เบสและแอนดรอยด์สตูดิโอ....70

2.9.2.1 การเขียนข้อมูล....70

2.9.2.2 การลบข้อมูล (Delete)....74

2.9.2.3 การอ่านข้อมูล....75

2.9.2.4 การเปิดใช้งานโหมดออฟไลน์....80

2.9.3 Security & Rules....81

2.9.4 Firebase Authentication....81

2.10 โปรแกรมช่วยออกแบบ Autodesk Inventor Professional....83

บทที่ 3 ระเบียบวิธีวิจัย

3.1 กล่องยา....85

3.1.1 การออกแบบกล่องยา....85

3.1.2 การสร้างสวิตช์ที่ฝากล่องยา....89

3.2 Arduino Mega 2560 R3....89

3.2.1 การอ่านค่าสถานะ การเปิด-ปิดฝาของกล่องยา....89

3.2.2 เชื่อมต่อ Arduino Mega 2560 R3 กับ Keypad....90

3.2.3 หน้าจอแสดงผลบนกล่องยา....91

3.2.4 เชื่อมต่อ Arduino 2560 R3 กับ RTC DS3231....92

3.2.5 อุปกรณ์อื่นๆ กับ Arduino Mega 2560....93

3.2.6 การทำงานของ Arduino Mega 2560 R3 ภายในงานวิจัยนี้....93

3.3 Wemos D1 Mini....97

3.3.1 การเชื่อมต่อ Wemos D1 Mini กับสัญญาณอินเทอร์เน็ต....97

3.3.2 การรับค่าจาก Arduino Mega 2560 R3....98

3.3.3 การดึงเวลาสากลจากอินเทอร์เน็ตมาใช้....99

3.3.4 ชุดคำสั่งที่ใช้สำหรับส่งค่าไปยังฐานข้อมูลออนไลน์ไฟร์เบส....101

3.4 แอปพลิเคชันแอนดรอยด์....101

3.4.1 การสร้างโปรเจ็กต์ใหม่....101

3.4.2 การออกแบบแอปพลิเคชัน....104

3.5 ฐานข้อมูลออนไลน์....107

3.5.1 การเริ่มต้นใช้งานฐานข้อมูลออนไลน์ไฟร์เบส....108

3.5.2 การสร้างฐานข้อมูลบนฐานข้อมูลออนไลน์ไฟร์เบส....111

3.5.3 การเชื่อมต่อฐานข้อมูลออนไลน์ไฟร์เบสกับแอปพลิเคชัน....113

3.5.4 การสร้างระบบการสร้างบัญชีผู้ใช้และการเข้าสู่ระบบ....114

3.5.4.1 การสร้างบัญชีผู้ใช้....114

3.5.4.2 การเข้าสู่ระบบ....118

3.5.5 การนำข้อมูลบนฐานข้อมูลออนไลน์ไปแสดงบนแอปพลิเคชัน....119

3.5.6 การส่งข้อมูลจากNodeMCUไปยังฐานข้อมูลด้วยระบบการส่งข้อมูลแบบไร้สาย120

3.5.7 การสร้างการแจ้งเตือนบนแอปพลิเคชัน....122

บทที่ 4 ผลการทดลองและอภิปรายผล

4.1 ภาพรวมของระบบ....123

4.1.1 กล่องยาพร้อมไมโครคอนโทรลเลอร์....123

4.1.2 แอปพลิเคชันบนแอนดรอยด์สมาร์ทโฟน....126

4.2 การทดสอบ.....131

4.2.1 การทดสอบการตรวจจับการเปิดปิดฝากล่องยา.....131

4.2.2 การทดสอบการแจ้งเตือนที่กล่องยา.....134

4.2.3 การทดสอบการแจ้งเตือนด้วยไฟที่แต่ละช่องกล่องยา.....135

4.2.4 การแจ้งเตือนที่แอปพลิเคชัน.....136

4.2.5 การแจ้งเตือนซ้ำที่แอปพลิเคชัน.....136

4.2.6 ตารางสรุปการทดสอบ.....137

บทที่ 5 บทสรุป

5.1 สรุป.....138

5.2 ปัญหาและอุปสรรคในการทำงาน.....138

5.3 ข้อเสนอแนะ.....139

บรรณานุกรม

ภาคผนวก

สารบัญตาราง

หน้า

ตารางที่ 1.1 แผนการดำเนินงานภาคการศึกษาที่ 1 ปีการศึกษา 2560.....3

ตารางที่ 1.2 แผนการดำเนินงานภาคการศึกษาที่ 2 ปีการศึกษา 2560.....4

ตารางที่ 2.1 ปัญหาการใช้ยาที่พบในผู้ป่วย.....7

ตารางที่ 2.2 เวอร์ชั่นและ API Level ของแอนดรอยด์.....11

ตารางที่ 2.3 หน้าที่ของเมธอดต่าง ๆ.....17

ตารางที่ 2.4 โฟลเดอร์และไฟล์ที่สำคัญ.....18

ตารางที่ 2.5 แอตทริบิวต์พื้นฐานของ View และ Layout เบื้องต้น.....19

ตารางที่ 2.6 แอตทริบิวต์พื้นฐานของ View และ Layout เบื้องต้น (ต่อ).....20

ตารางที่ 2.7 แอตทริบิวต์พื้นฐานของ View และ Layout เบื้องต้น (ต่อ).....21

ตารางที่ 2.8 ลักษณะโดยสังเขปของ View ชนิดต่าง ๆ.....22

ตารางที่ 2.9 รหัส BCD – 8421 เปรียบเทียบกับเลขฐานสิบ.....45

ตารางที่ 2.10 พาริตี้บิตในรหัส BCD – 8421.....47

ตารางที่ 2.11 การเปรียบเทียบระหว่างรหัส BCD – 8421 กับรหัสเกิน 3.....48

ตารางที่ 2.12 การเปรียบเทียบระหว่างเลขฐานสองกับรหัสเกรย์ 0 – 15.....49

ตารางที่ 2.13 การเปรียบเทียบข้อดี-ข้อเสียของการใช้กุญแจสมมาตร.....51

ตารางที่ 3.1 แสดงการเชื่อมต่อ LCD I2C กับ Arduino Mega.....91

ตารางที่ 4.1 แสดงผลบนฐานข้อมูลออนไลน์ สำหรับการทดสอบการตรวจจับการเปิดฝากล่องยา...132

ตารางที่ 4.2 แสดงผลบนฐานข้อมูลออนไลน์ สำหรับการทดสอบการตรวจจับการปิดฝากล่องยา....133

ตารางที่ 4.3 แสดงผลการทดสอบการแจ้งเตือนที่กล่องยา.....135

ตารางที่ 4.4 แสดงผลการทดสอบการแจ้งเตือนด้วยไฟที่แต่ละช่องกล่องยา.....135

ตารางที่ 4.5 แสดงผลการทดสอบการแจ้งเตือนที่แอปพลิเคชัน.....136

ตารางที่ 4.6 แสดงผลการทดสอบการแจ้งเตือนซ้ำที่แอปพลิเคชัน.....136

ตารางที่ 4.7 แสดงผลสรุปของการทดสอบ.....136

สารบัญรูปภาพ

หน้า

รูปที่ 2.1 โครงสร้างลำดับชั้นของแอนดรอยด์....10
รูปที่ 2.2 Android Studio....12
รูปที่ 2.3 โฟลเดอร์ต่าง ๆ ของ Android Studio....13
รูปที่ 2.4 แถบเครื่องมือต่าง ๆ ของ Android Studio....13
รูปที่ 2.5 วงจรสถานะของ Activity....15
รูปที่ 2.6 การแบ่งกลุ่มของ View....21
รูปที่ 2.7 ออกแบบ Layout ให้มีปุ่มกดให้แจ้งเตือน....23
รูปที่ 2.8 เวอร์ชัน API 21....24
รูปที่ 2.9 ตั้งค่าการสั่นของการแจ้งเตือน....25
รูปที่ 2.10 พารามิเตอร์ int flags....30
รูปที่ 2.11 การดู id....31
รูปที่ 2.12 บอร์ด Arduino Uno....33
รูปที่ 2.13 บอร์ด Arduino และอุปกรณ์เสริม Arduino XBee Shield....33
รูปที่ 2.14 โปรแกรม Arduino IDE....34
รูปที่ 2.15 การเชื่อมต่อกันของ Arduino IDE และบอร์ด Arduino....34
รูปที่ 2.16 หน้าที่ของแต่ละ Pin ของบอร์ด Arduino....35
รูปที่ 2.17 Arduino Mega2560....36
รูปที่ 2.18 Wemos Mini....39
รูปที่ 2.19 แสดงอุปกรณ์โมดูล DS3231 RTC (มุมมองจากด้านล่าง)....40

รูปที่ 2.20 แสดงอุปกรณ์โมดูล DS3231 RTC (มุมมองจากด้านบน).....41

รูปที่ 2.21 ตัวอย่างสายตีเกลียวคู่.....54

รูปที่ 2.22 ตัวอย่างสายโคแอกซ์.....55

รูปที่ 2.23 ตัวอย่างสายใยแก้วนำแสง.....56

รูปที่ 2.24 การเชื่อมต่อคอมพิวเตอร์สองเครื่องโดยผ่านพอร์ตไออาร์ดีเอ.....56

รูปที่ 2.25 การส่งสัญญาณผ่านคลื่นวิทยุไปในอากาศ.....57

รูปที่ 2.26 การส่งข้อมูลแบบไร้สายโดยใช้คลื่นไมโครเวฟ.....57

รูปที่ 2.27 การส่งข้อมูลแบบไร้สายโดยใช้ดาวเทียมเป็นการรับส่งสัญญาณไมโครเวฟบนอวกาศ.....58

รูปที่ 2.28 Server.....64

รูปที่ 2.29 Firebase Realtime Database.....67

รูปที่ 2.30 แสดงการเขียนข้อมูลไปบนฐานข้อมูลออนไลน์.....70

รูปที่ 2.31 แต่ละชุดข้อมูลถูกสร้าง unique key.....71

รูปที่ 2.32 แสดงแต่ละสีที่บอกความหมายที่ต่างกัน.....72

รูปที่ 2.33 แสดงการอัพเดตข้อมูลพร้อมกัน.....73

รูปที่ 2.34 แสดงขั้นตอนที่หลากหลายของการพัฒนาระบบ log-in.....81

รูปที่ 2.35 แสดงการลดขั้นตอนจากเดิมด้วยบริการ Firebase Authentication.....82

รูปที่ 2.36 แสดงบริการ Firebase Authentication ที่รองรับหลายรูปแบบ.....82

รูปที่ 2.37 Autodesk Inventor Professional.....83

รูปที่ 3.1 แผนภาพรวมในการดำเนินการวิจัย.....85

รูปที่ 3.2 แสดงภาพฉายของกล่องยา.....86

รูปที่ 3.3 แสดงภาพฉายของฝากล่องยา.....86

รูปที่ 3.4 แสดงภาพฉายของฝาผิดด้านข้างฝั่งซ้ายของอุปกรณ์....87

รูปที่ 3.5 แสดงภาพฉายของฝาผิดด้านข้างฝั่งขวาของอุปกรณ์....87

รูปที่ 3.6 แสดงภาพฉายของฝาผิดด้านหลังของอุปกรณ์....88

รูปที่ 3.7 แสดงภาพฉายของฝาผิดด้านบนของอุปกรณ์....88

รูปที่ 3.8 แสดงตำแหน่งที่ติดคอนเนคเตอร์เทปบนฝากล่องยา....89

รูปที่ 3.9 แสดงการต่อกล่องยาเข้ากับ Arduino Mega 2560 R3....90

รูปที่ 3.10 แสดงการต่อ Arduino Mega กับ Keypad....90

รูปที่ 3.11 แสดงการต่อ Arduino Mega กับ LCD I2C 16X2....91

รูปที่ 3.12 แสดงการต่อ Arduino Mega กับ RTC DS3231....92

รูปที่ 3.13 แสดงแผนผังตั้งแต่เริ่มต้นไปยังเลือกมื้ออาหาร....94

รูปที่ 3.14 แสดงแผนผังหลังจากเลือกมื้ออาหารไปยังยืนยันการตั้งเวลา....95

รูปที่ 3.15 แสดงแผนผังหลังจากยืนยันการตั้งเวลาไปยังเลือกช่องที่กล่องยา....96

รูปที่ 3.16 แสดงแผนผังหลังจากเลือกช่องที่กล่องยาไปยังการกลับมาที่หน้าจอหลัก....97

รูปที่ 3.17 แสดงการเชื่อมต่อเพื่อส่งข้อมูลการแจ้งเตือนจาก Arduino Mega ไปยัง Wemos D1 Mini....98

รูปที่ 3.18 แสดงการเชื่อมต่อเพื่อส่งข้อมูลสถานะการเปิดปิดฝาของกล่องยาจาก Arduino Mega ไปยัง Wemos D1 Mini....99

รูปที่ 3.19 การสร้างโปรเจ็กต์ใหม่บนโปรแกรม Android studio....102

รูปที่ 3.20 ตั้งชื่อและกำหนดที่อยู่ให้แอปพลิเคชัน....102

รูปที่ 3.21 หน้าต่างเลือกเวอร์ชั่นของแอนดรอยด์....103

รูปที่ 3.22 หน้าต่างเลือกชนิดเริ่มต้นของงาน....103

รูปที่ 3.23 หน้าต่างตั้งชื่อ Activity name และ Layout Name....104

รูปที่ 3.24 มุมมอง Design หน้าต่าง layout เข้าสู่ระบบ....104

รูปที่ 3.25 มุมมอง Design หน้าต่าง layout สมัครแอคเค้าท์.....105

รูปที่ 3.26 มุมมอง Design หน้าต่าง layout หน้าหลัก.....105

รูปที่ 3.27 มุมมอง Design หน้าต่าง layout แสดงข้อมูลทั่วไปของผู้ป่วย.....106

รูปที่ 3.28 มุมมอง Design หน้าต่าง layout แสดงข้อมูลยาของผู้ป่วย.....106

รูปที่ 3.29 มุมมอง Design หน้าต่าง layout แสดงประวัติการรับประทานยาของผู้ป่วย.....107

รูปที่ 3.30 มุมมองdesign หน้าต่างlayout แสดงสถานการณ์แจ้งเตือนและการเปิดปิดฝากล่องยา.107

รูปที่ 3.31 หน้าเว็บฐานข้อมูลออนไลน์ Firebase (Database Realtime).....108

รูปที่ 3.32 หน้าต่างเพิ่มโครงการใหม่บน Firebase 108

รูปที่ 3.33 หน้าเว็บเริ่มต้นการใช้งานทั้งแอปแบบ Android, iOS และเว็บ.....109

รูปที่ 3.34 หน้าต่างการกรอกรายละเอียดแอปพลิเคชันที่ต้องการจะเชื่อมต่อกับ Firebase 109

รูปที่ 3.35 ขั้นตอนการโหลดไฟล์ google-services.json.....110

รูปที่ 3.36 หน้าต่างขั้นตอนที่ 3 ให้นำชุดคำสั่งไปใส่ใน Android Studio.....110

รูปที่ 3.37 แอปพลิเคชันถูกเพิ่มลงใน Firebase เรียบร้อยแล้ว.....111

รูปที่ 3.38 แถบเครื่องมือหลักของโปรเจคนี้คือ Database และ Authenication 111

รูปที่ 3.39 แสดงเข้าสู่ฐานข้อมูลบนฐานข้อมูลออนไลน์ไฟร์เบส.....112

รูปที่ 3.40 แสดงการเพิ่มข้อมูลบนฐานข้อมูลออนไลน์ไฟร์เบส.....112

รูปที่ 3.41 แสดงข้อมูลที่หลากหลายบนฐานข้อมูลออนไลน์ไฟร์เบส.....113

รูปที่ 3.42 แสดงแถบเครื่องมือช่วย Firebase.....113

รูปที่ 3.43 แสดงหน้าต่างตัวช่วย Firebase.....114

รูปที่ 3.44 แสดงการเชื่อมต่อแอปพลิเคชันและไฟร์เบสเสร็จสิ้น.....114

รูปที่ 3.45 แสดงการเปิดใช้งานอีเมล/รหัสผ่าน.....115

รูปที่ 3.46 แสดงการเพิ่มบัญชีอีเมล/รหัสผ่าน บนไฟร์เบส.....115

รูปที่ 3.47 แสดงหน้าต่าง SignUpActivity ที่เขียนฟังก์ชันคำสั่งในการเพิ่มบัญชีผู้ใช้.....118

รูปที่ 3.48 แสดง NodeMCU Wemos D1 Mini.....121

รูปที่ 4.1 กล่องยาพร้อมไมโครคอนโทรลเลอร์.....124

รูปที่ 4.2 แสดงหน้าต่างหน้าแรกของอุปกรณ์.....124

รูปที่ 4.3 แสดงหน้าต่างแสดงเวลาของอุปกรณ์.....124

รูปที่ 4.4 แสดงตัวอย่างการตั้งค่าเวลาสำหรับมื้ออาหาร.....125

รูปที่ 4.5 แสดงตัวอย่างการตั้งค่าการเลือกช่องของกล่องยา.....125

รูปที่ 4.6 อุปกรณ์ในขณะที่มีการแจ้งเตือน.....126

รูปที่ 4.7 อุปกรณ์ในขณะที่มีการเปิดฝาล่องยาหลังจากการแจ้งเตือน.....126

รูปที่ 4.8 ไอคอนแอปพลิเคชันชื่อ “Belmoz”.....127

รูปที่ 4.9 แสดงหน้าจอเข้าสู่ระบบและสมัครหน้าจอสมัครสมาชิกบนแอนดรอยด์สมาร์ทโฟน.....127

รูปที่ 4.10 แสดงหน้าจอหลังจากเข้าสู่ระบบจะเจอหน้าหลักที่มีปุ่มเข้าไปดูข้อมูลต่าง ๆ.....128

รูปที่ 4.11 หน้าจอแสดงหน้าข้อมูลทั่วไปของผู้ป่วย และข้อมูลยา.....129

รูปที่ 4.12 หน้าจอแสดงหน้าประวัติหารรับประทานยาของผู้ป่วย และข้อมูลสถานะการแจ้งเตือน.130

รูปที่ 4.13 แสดงรายชื่ออีเมล์ที่ทำการสมัครสมาชิกเพื่อใช้แอคเคาท์เข้าสู่ระบบบนฐานข้อมูลออนไลน์ไฟร์เบส.....130

รูปที่ 4.14 แสดงฐานข้อมูลที่เก็บไว้บนฐานข้อมูลออนไลน์ไฟร์เบส ข้อมูลจะอยู่ในรหัสของแต่ละแอคเคาท์.....131

บทที่ 1

บทนำ

1.1. ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันประเทศไทยได้ก้าวเข้าสู่สังคมผู้สูงอายุซึ่งเป็นผลมาจากอัตราการเกิดที่ลดลงความเจริญก้าวหน้าทางการแพทย์ทำให้ผู้สูงอายุมีอายุยืนยาวขึ้น นวัตกรรมเพื่อผู้สูงอายุ จึงเป็นสิ่งที่ควรให้ความสำคัญ ซึ่งทางผู้วิจัยได้มองเห็นถึงปัญหาทางด้านการรับประทานยาของผู้สูงอายุตัวอย่าง เช่น ผู้สูงอายุจำนวนมากไม่สามารถจดจำข้อมูลเกี่ยวกับยา ไม่สามารถรับประทานยาของตนได้อย่าถูกต้องหรือรับประทานยาไม่ตรงตามเวลาตามคำแนะนำของแพทย์ ซึ่งส่งผลให้ประสิทธิภาพในการรักษาของแพทย์ ได้ผลน้อยกว่าที่ควรจะเป็น

ในปี 2553 ได้มีการวิจัยเกี่ยวกับปัญหาการใช้ยาของผู้ป่วยในการรักษาตนเองที่บ้านในเขตชุมชนอำเภอเมือง จังหวัดขอนแก่น ซึ่งเป็นการทำงานวิจัยร่วมกันของ คุณหทัยรัตน์ สุขศรี คุณเพ็ญนภา ศรีหริ่ง จากวิทยาลัยการสาธารณสุขสิรินธร จังหวัดขอนแก่น และคุณทิพาพร กาญจนราช จากคณะเภสัชศาสตร์ มหาวิทยาลัยขอนแก่น โดยการเก็บข้อมูลโดยใช้แบบสอบถามและสัมภาษณ์ผู้ป่วยโรคเรื้อรัง จำนวน 147 คน ซึ่งมีผู้ป่วยที่อายุมากกว่า 60 ปี ประมาณร้อยละ 84.9 และอายุต่ำกว่า 60 ปี ร้อยละ 15.1 พบว่าปัญหาของการใช้ยาที่บ้านของผู้ป่วยคือการลืมทานยาร้อยละ 52.4 ทานยาผิดเวลาร้อยละ 21.8 ไม่ตั้งใจใช้ยาและมียาเหลือใช้ ซึ่งจัดเป็นการไม่ให้ความร่วมในการใช้ยาและการเก็บรักษายาไม่เหมาะสม สาเหตุที่พบเนื่องจากภาระงานที่ต้องทำ การงดทานอาหารบางมื้อ การทานยาผิดเวลา ความเครียด กังวล หลงลืม เพราะชรา เก็บยารวมกันไว้กับคู่สมรส และขาดคนคอยเตือน รวมทั้งผู้ป่วยยังมีพฤติกรรมด้านสุขภาพที่ยังไม่เหมาะสม และผู้ป่วยไม่ทราบข้อมูลรายละเอียดอื่นๆที่เกี่ยวกับยาที่ตนเองใช้อยู่

ปัญหาที่พบบ่อยครั้งนี้ นอกจากส่งผลต่อประสิทธิภาพการรักษาแล้วยังอาจส่งผลกระทบต่อการวินิจฉัยของแพทย์ อาทิเช่น การที่แพทย์ได้รับข้อมูลการรับประทานยาที่ไม่ถูกต้องจากผู้ป่วย ผู้ป่วยไม่สามารถให้ข้อมูลเกี่ยวกับยาที่ตนเองทานได้ อาจทำให้เกิดการวินิจฉัยหรือรักษาที่ผิดพลาด ซึ่ง

เป็นอันตรายต่อผู้ป่วยอย่างมาก บางกรณียังส่งผลให้การวินิจฉัยล่าช้าเนื่องจากแพทย์ต้องเปิดดูข้อมูลการจ่ายยาย้อนหลัง งานวิจัยนี้จึงเป็นการวิจัยเพื่อพัฒนากล่องยาควบคู่กับแอปพลิเคชันให้สามารถเก็บข้อมูลของผู้ป่วย ข้อมูลยาที่ผู้ป่วยต้องรับประทาน และประวัติการเปิดปิดฝากล่องยาเพื่อให้แพทย์หรือเภสัชกรสามารถอนุมานผลการทานยาของผู้ป่วยได้จากแอปพลิเคชัน

1.2. วัตถุประสงค์

1.2.1 เพื่อแจ้งเตือนเมื่อถึงเวลาการรับประทานยาแก่ผู้ป่วย บนกล่องยาและแอปพลิเคชัน

1.2.2 เพื่อให้ข้อมูลการใช้ยาที่ถูกต้องแก่ผู้ป่วย เช่น ชนิดของยา ปริมาณ เป็นต้น

1.2.3 เพื่อเก็บประวัติการรับประทานยาของผู้ป่วย

1.2.4 เพื่อช่วยจัดเก็บข้อมูลให้เป็นระเบียบ ทำให้แพทย์และผู้ป่วยเข้าถึงได้สะดวกมากขึ้น

1.3. สมมติฐาน

1.3.1 อุปกรณ์สามารถตรวจจับการเปิดปิดฝากล่องได้อย่างถูกต้องและแม่นยำ

1.3.2 อุปกรณ์สามารถแจ้งเตือนเวลาการรับประทานยาของผู้ป่วยได้อย่างถูกต้องและแม่นยำ

1.3.3 แอปพลิเคชันสามารถแจ้งเตือนเวลารับประทานยาให้แก่ผู้ป่วยได้อย่างถูกต้องและแม่นยำ

1.3.4 แอปพลิเคชันสามารถแสดงข้อมูลทั่วไปของผู้ป่วย ข้อมูลการทานยา และประวัติการเปิดปิดฝากล่องยาได้

1.4. ขอบเขตของงานวิจัย

1.4.1 ของเขตการวิจัยด้านฮาร์ดแวร์

1.4.1.1 การออกแบบกล่องยาสำหรับใช้ในบ้าน

1.4.1.2 การออกแบบวงจรและกล่องยาเพื่อให้สามารถตรวจจับการเปิด-ปิดฝากล่องยาได้

1.4.1.3 การตั้งค่าเวลาการแจ้งเตือนการรับประทานยาบนกล่องยา

1.4.1.4 การแจ้งเตือนการรับประทานยาบนกล่องยา

1.4.1.5 การส่งข้อมูลผ่านเทคโนโลยีแบบไร้สาย

1.4.2 ของเขตการวิจัยด้านซอฟแวร์

1.4.2.1 การออกแบบแอปพลิเคชันด้วยโปรแกรม Android studio

1.4.2.2 การเก็บฐานข้อมูลยาของผู้ป่วยไว้ในฐานข้อมูลออนไลน์

1.4.2.3 การแสดงข้อมูลยาบนแอปพลิเคชัน

1.4.2.4 การแจ้งเตือนการรับประทานยาของผู้ป่วยบนแอปพลิเคชัน

1.4.2.5 การเก็บประวัติการรับประทานยาของผู้ป่วย โดยการรับสัญญาณที่ฝากล่องยาผ่านเทคโนโลยีแบบไร้สาย

1.4.3 ขอบเขตระยะเวลาการทำวิจัย

เดือนสิงหาคม พ.ศ.2560 ถึง เดือนเมษายน พ.ศ.2561

1.5. แผนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงานภาคการศึกษาที่ 1 ปีการศึกษา 2560

เดือน	สิงหาคม	กันยายน	ตุลาคม	พฤศจิกายน
ศึกษาปัญหาการใช้ยาของผู้สูงอายุ				
ศึกษาการเขียนโปรแกรมด้วยภาษาซีและภาษาจาวา เบื้องต้น				
ศึกษาการนำวงจรอิเล็กทรอนิกส์มาประยุกต์ใช้				
ออกแบบการทำงานของแอปพลิเคชันและกล่องยา				
ทดลองการทำงานของกล่องยาและแอปพลิเคชัน				
วิเคราะห์ผลที่ได้จากการทดลองในภาคการศึกษาที่ 1 ประเมินข้อดีข้อเสียของอุปกรณ์				
นำเสนอความก้าวหน้าของงานวิจัย				

ตารางที่ 1.2 แผนการดำเนินงานภาคการศึกษาที่ 2 ปีการศึกษา 2560

เดือน	ธันวาคม	มกราคม	กุมภาพันธ์	มีนาคม	เมษายน
ออกแบบอุปกรณ์					
พิมพ์กล่องยาด้วยเครื่อง 3D printer และทดสอบการเปิดปิดฝาของกล่องยา					
พัฒนาฟังก์ชันในแอปพลิเคชันและทดสอบการทำงาน					
พัฒนาฟังก์ชันของกล่องยาทดสอบการทำงาน					
ทดสอบการทำงานร่วมกันของแอปพลิเคชันกับกล่องยา					
จัดทำรูปเล่มงานวิจัย					
ออกแบบ พิมพ์และทดลองประกอบสำหรับอุปกรณ์ประกอบ					
นำเสนองานวิจัย					

1.6. ประโยชน์ที่คาดว่าจะได้รับ

1.3.1 ผู้ป่วยรับประทานยาได้ตรงตามเวลาที่แพทย์กำหนด ส่งผลต่อประสิทธิภาพในการรับการรักษาด้วยการรับประทานยาของผู้ป่วยเพิ่มขึ้น

1.3.2 ผู้ป่วยและญาติเข้าถึงข้อมูลยาที่ถูกต้องได้ง่ายขึ้น

1.3.3 แพทย์เข้าถึงข้อมูลยาที่ถูกต้องของผู้ป่วยได้สะดวกและรวดเร็วขึ้น

นิยามศัพท์

ผู้ป่วย หมายถึง ผู้สูงอายุ อายุระหว่าง 60-90 ปี ที่รับประทานยาอยู่เป็นประจำและต้องการความช่วยเหลือในการแจ้งเตือนการรับประทานยา โดยผู้สูงอายุคนนั้น ๆ สามารถมองเห็นแสงไฟหรือได้ยินเสียงบนกล่องยาและสามารถเปิดปิดฝากล่องยาเองได้

กล่องยา หมายถึง อุปกรณ์ที่ใช้สำหรับเก็บยา

แอปพลิเคชัน หมายถึง โปรแกรมประยุกต์บนสมาร์ทโฟนในระบบปฏิบัติการแอนดรอยด์ ใช้งานร่วมกับกล่องยาในงานวิจัยนี้

ฐานข้อมูลออนไลน์ หมายถึง ฐานข้อมูลที่ให้บริการผ่านทางระบบเครือข่าย คอมพิวเตอร์และให้บริการผ่านทางอินเทอร์เน็ตผู้จัดการฐานข้อมูลสามารถปรับปรุง ฐานข้อมูลให้ทันสมัย และผู้ใช้สามารถเข้าถึงข้อมูลได้ตลอดเวลา

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 งานวิจัยและบทความ

2.1.1 งานวิจัยที่เกี่ยวข้อง

จากงานวิจัยเรื่อง “ปัญหาการใช้ยาของผู้ป่วยในการรักษาตนเองที่บ้าน ในเขตชุมชนอำเภอเมือง จังหวัดขอนแก่น” การศึกษานี้มีวัตถุประสงค์เพื่อสำรวจปัญหาการใช้ยาที่บ้านของผู้ป่วย โดยการเก็บข้อมูลโดยใช้แบบสอบถามและสัมภาษณ์ผู้ป่วยโรคเรื้อรังจำนวน 147 คน จากจำนวนผู้ป่วยทั้งหมด 267 คน ในระหว่างวันที่ 15 มีนาคม - 30 เมษายน 2553 ผลการศึกษาพบว่าผู้ป่วยส่วนใหญ่ป่วยเป็นโรคความดันโลหิตสูงและเบาหวาน ยาที่มีผู้ใช้มาก 3 ลำ ดับแรกคือ ยากลัยเบนคลาไมด์ รองลงมาคือวิตามินบี 1-6-12 และยาไฮโดรคลอโรไทอะไซด์ ปัญหาการใช้ยาที่พบมากคือ การลืมรับประทานยา รับประทานยาผิดเวลา เก็บรักษายาไม่เหมาะสม ไม่ตั้งใจใช้ยาและมียาเหลือใช้ สาเหตุที่พบเนื่องจากภาระงานที่ต้องทำ การงดรับประทานอาหารบางมื้อ การรับประทานยาผิดเวลาความเครียดกังวล หลงลืมเพราะชรา เก็บยารวมกันไว้กับคู่สมรส และขาดคนคอยเตือน การปฏิบัติตนหลังการลืมรับประทานยาได้แก่ รับประทานยาทันทีที่นึกได้ รับประทานยาควบกับยามื้อถัดไป งดรับประทานยามื้อนั้นเลย รวมทั้งมีการเพิ่มขนาดยาเป็นสองเท่าในมื้อถัดมา ส่วนใหญ่ไม่ทราบรายละเอียดอื่นๆเกี่ยวกับยาที่ตนใช้ บางรายยอมรับว่าไม่ได้ออกกำลังกาย ผู้ป่วยชายยังคงสูบบุหรี่และดื่มสุรา ผู้ป่วยเบาหวานควบคุมชนิดของอาหารยังไม่ได้ สรุปปัญหาการใช้ยาที่บ้านของผู้ป่วยคือการไม่ให้ความร่วมในการใช้ยา จึงควรให้ความรู้เพิ่มความตระหนักในการใช้ยา และเพิ่มบริการการติดตามการใช้ยาของผู้ป่วยที่บ้าน

2.1.1.1 ผลการศึกษางานวิจัย

ผลการศึกษาพบว่าจำนวนปัญหาการใช้ยาที่พบในผู้ป่วยในแต่ละรายมีตั้งแต่ 1 ถึง 7 ปัญหา รายละเอียดปัญหาแสดงในตารางที่ 2.1

ตารางที่ 2.1 ปัญหาการใช้ยาที่พบในผู้ป่วย

ปัญหา	จำนวน (คน)	ร้อยละ
ลืมรับประทานยา	77	52.4
รับประทานยาผิดเวลา	32	21.8
มีอาการแทรกซ้อนของโรค	22	15.0
เก็บรักษายาไม่เหมาะสม	20	13.6
ความไม่ตั้งใจในการใช้ยา	16	10.9
รับประทานอาหารไม่เหมาะสม	14	9.5
มียาเหลือใช้	12	8.2
จำนวนครั้งต่อวันมากเกินไป	11	7.5
การเกิดอาการไม่พึงประสงค์	11	7.5
จำนวนครั้งต่อวันน้อยเกินไป	11	7.5
การใช้ยาอื่นๆที่หามาเองร่วมด้วย	11	7.5
ไม่มารับยาตามนัด และยาไม่พอใช้	10	6.8
ดื่มเครื่องดื่มผสมคาเฟอีน	5	3.4
ผู้อื่นจัดยาให้รับประทาน	4	2.7
ความเครียดกังวลเกี่ยวกับโรค	4	2.7

2.1.1.2 ข้อสรุปของงานวิจัย

ปัญหาของการใช้ยาที่บ้านของผู้ป่วยคือการลืมรับประทานยา รับประทานยาผิดเวลา ไม่ตั้งใจใช้ยาและมียาเหลือใช้ซึ่งจัดเป็นการไม่ให้ความร่วมในการใช้ยาและการเก็บรักษายาไม่

เหมาะสม สาเหตุที่พบเนื่องจากภาระงานที่ต้องทำ การงดรับประทานอาหารบางมื้อ การรับประทานยาผิดเวลา ความเครียดกังวล หลงลืมเพราะชรา เก็บยารวมกันไว้กับคู่สมรส และขาดคนคอยเตือน รวมทั้งผู้ป่วยยังมีพฤติกรรมด้านสุขภาพที่ยังไม่เหมาะสม และผู้ป่วยไม่ทราบข้อมูลรายละเอียดอื่นๆ เกี่ยวกับยาที่ตนใช้อยู่ จึงเสนอว่าควรมีกิจกรรมให้ความรู้และเพิ่มความตระหนักถึงความสำคัญของการให้ความร่วมมือในการรับประทานยา และควรเพิ่มกิจกรรมการติดตามการใช้ยาของผู้ป่วยที่บ้านโดยเภสัชกร เพื่อให้ใช้ยาของผู้ป่วยที่บ้านและการรักษามีประสิทธิภาพมากยิ่งขึ้น

2.1.2 หลักการและหลักปฏิบัติทั่วไปของการทานยา

2.1.2.1 ยาก่อนอาหาร

ยาก่อนอาหารควรทานในขณะที่ท้องว่าง ควรทานก่อนอาหารอย่างน้อย 30 นาที เนื่องจากยาอาจถูกทำลายและเสียประสิทธิภาพในการรักษาเมื่อพบกับกรดปริมาณมากที่กระเพาะอาหารจะหลั่งออกมาหลังมื้ออาหาร การทานยาในขณะที่ท้องว่างทำให้ยาไม่ถูกทำลายและประสิทธิภาพของยาไม่ลดลง นอกจากนี้อาหารและส่วนประกอบของอาหารอาจลดการดูดซึมของยาเข้าสู่ร่างกาย จึงไม่สามารถรับประทานยาพร้อมหรือหลังอาหารได้ สำหรับยาบางชนิด เช่น ยาที่ออกฤทธิ์เพิ่มการเคลื่อนไหวของระบบทางเดินอาหาร ยาลดอาการคลื่นไส้อาเจียน รวมทั้งยาที่ออกฤทธิ์เพิ่มการหลั่งอินซูลิน จะใช้เวลาประมาณ 30 นาทีก่อนที่จะออกฤทธิ์ การทานยาก่อนอาหารจึงเป็นเสมือนการเตรียมพร้อมให้ระบบทางเดินอาหาร ก่อนจะทานอาหาร

หากลืมทานยาก่อนอาหาร หรือรับประทานยาก่อนที่จะทานอาหารไม่ถึงครึ่งชั่วโมง อาจทำให้ยาถูกทำลายหรืออาหารอาจลดการดูดซึมของยา ควรเว้นการรับประทานยามื้อนั้นไปหรืออาจรอให้กระเพาะอาหารว่างก่อนแล้วค่อยรับประทานยาประมาณ 2 ชั่วโมงหลังทานอาหาร แต่ถ้าหากเป็นยาที่ต้องทานในมื้อถัดไปอยู่แล้ว ให้รับประทานยาก่อนอาหารมื้อถัดไปแทนได้เลย ไม่ต้องรับประทานยาซ้ำ

2.1.2.2 ยาหลังอาหาร

ยาหลังอาหารควรทานหลังอาหารทันที หรืออาจทานพร้อมอาหารหรือก่อนรับประทานอาหารคำแรกเพราะไม่ว่าจะกรณีใด ยาจะเข้าสู่กระเพาะอาหารพร้อมกับอาหารที่ทานเข้าไป ยาบางชนิดต้องทานหลังอาหาร เนื่องจากยามีผลข้างเคียงที่สำคัญคือ ระคายเคืองต่อระบบทางเดิน

อาหาร ทำให้เกิดอาการคลื่นไส้อาเจียน การทานพร้อมหรือหลังอาหารทันทีจะช่วยลดอาการเหล่านี้ได้ นอกจากนี้กรดในกระเพาะอาหารจะช่วยในการดูดซึมยาเข้าสู่ร่างกาย ซึ่งกรดในกระเพาะอาหารจะหลั่งสูงสุดในระหว่างที่ทานอาหารเท่านั้น

หากลืมรับประทานยาหลังอาหาร สามารถรับประทานยาได้ทันทีที่นึกได้และไม่เกิน 15 นาที หากนึกได้หลังจากทานอาหารมากกว่า 15 นาทีแล้ว ควรรอทานหลังอาหารในมื้อถัดไปแทน หรืออาจทานอาหารมื้อย่อยแทนมื้อหลักก่อนรับประทานยาในกรณีที่ยานั้นมีความสำคัญมาก

2.1.2.3 ยาก่อนนอน

ยาที่แนะนำให้รับประทานก่อนนอนมีหลายประเภท แต่โดยทั่วไป ควรรับประทานก่อนนอน 15 – 30 นาที เนื่องจากยามีผลข้างเคียงสำคัญคือทำให้ง่วงนอนหรือวิงเวียนศีรษะมาก หากทานก่อนนอนนานเกินไป อาจส่งผลให้ผู้ที่รับประทานยาทำงานได้ไม่เต็มประสิทธิภาพ ยาที่ช่วยให้นอนหลับ มักใช้เวลาประมาณ 15 – 30 นาทีก่อนที่จะออกฤทธิ์ช่วยให้นอนหลับ หากลืมรับประทานยาก่อนนอน ควรรอให้ถึงเวลาก่อนเข้านอนในคืนถัดไปจึงค่อยรับประทานยา

2.1.2.4 ยาสำหรับทานเมื่อมีอาการ

ยาในกลุ่มนี้ มักระบุในฉลากว่าควรทานทุก 4 – 6 ชั่วโมง ทุก 8 ชั่วโมง หรือทุก 12 ชั่วโมง เมื่อมีอาการสามารถรับประทานยาได้เลยไม่ต้องคำนึงถึงมื้ออาหาร เนื่องจากการทานอาหารหรือไม่ทานอาหารไม่ส่งผลต่อการออกฤทธิ์ของยา หลังรับประทานยาแล้วถ้ายังมีอาการอยู่สามารถรับประทานยาซ้ำได้ ตามระยะเวลาที่ระบุไว้ ไม่ควรทานบ่อยกว่าที่ระบุไว้บนฉลาก เมื่อหายแล้วสามารถหยุดยาได้เลย

หมายเหตุ ยาบางประเภท อาจมีวิธีรับประทานยานอกเหนือไปจากยาโดยทั่ว ๆ ไปข้างต้น รวมทั้งยาบางประเภทอาจรับประทานก่อนหรือหลังอาหารก็ได้ เนื่องจากยาอาจมีการออกฤทธิ์ที่พิเศษหรือมีผลข้างเคียงอื่นๆ ซึ่งผู้ทำหน้าที่จ่ายยาเหล่านี้จะอธิบายวิธีการรับประทานเป็นกรณีๆ ไป

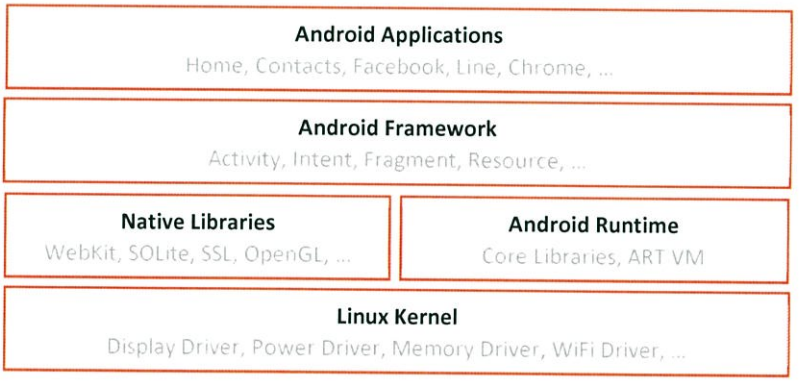
2.2 ระบบแอนดรอยด์ (Android)

2.2.1 เกี่ยวกับระบบแอนดรอยด์

Android เป็นระบบปฏิบัติการแบบเปิดเผยซอร์สโค้ด (Open Source) ภายใต้ลิขสิทธิ์ของ Google ซึ่งบริษัทต่างๆ สามารถนำไปพัฒนาเพิ่มเติม หรือปรับแต่งให้ใช้ร่วมกับผลิตภัณฑ์ของตนได้ จึงทำให้แอนดรอยด์ไม่ยึดติดกับอุปกรณ์ หรือผู้ผลิตรายใดรายหนึ่ง เราจึงสามารถพบเห็นระบบแอนดรอยด์ได้อย่างหลากหลาย ทั้งในอุปกรณ์เคลื่อนที่ (Mobile Device) ชนิดต่างๆ เช่น โทรศัพท์มือถือ, แท็บเล็ต, นาฬิกา หรือกล้องดิจิตอล รวมถึงเครื่องใช้ไฟฟ้าบางชนิด เช่น โทรทัศน์ นอกจากนี้รถยนต์บางรุ่น ก็เริ่มนำระบบแอนดรอยด์มาใช้กันแล้ว

2.2.2 โครงสร้างของระบบแอนดรอยด์

Android เป็นระบบที่เกิดจากการผสมผสานเทคโนโลยีที่หลากหลายเข้าด้วยกัน ซึ่งบางส่วนก็เป็นการพัฒนาโดย Google เอง แต่บางส่วนก็นำมาจากผู้สร้างรายอื่น รวมถึงการใช้พื้นฐานบางอย่างจากระบบปฏิบัติการ Linux อีกด้วย โดยเราสามารถแบ่งโครงสร้างของแอนดรอยด์ออกเป็นลำดับชั้นได้ดังนี้



รูปที่ 2.1 โครงสร้างลำดับชั้นของแอนดรอยด์

2.2.3 เวอร์ชั่นและ API Level ของแอนดรอยด์

ระบบแอนดรอยด์นั้นมีการปรับปรุงเปลี่ยนแปลงอยู่ตลอดและมีการพัฒนามาแล้วหลายเวอร์ชั่น โดยแต่ละเวอร์ชั่นจะสัมพันธ์ระดับ API ซึ่งเราต้องนำไปพิจารณาในการสร้างแอปพลิเคชั่นในลำดับต่อไปด้วย เช่น ถ้าเราต้องการให้แอปนั้นรองรับตั้งแต่เวอร์ชั่น 4.4 (KitKat) ขึ้นไป ก็จะต้องเลือก

Minimum API เป็น Level 19 เป็นต้น โดยข้อมูลเวอร์ชั่นและ API ในบาง Level รวมถึงสัดส่วนการติดตั้งในอุปกรณ์แอนดรอยด์

ตารางที่ 2.2 เวอร์ชั่นและ API Level ของแอนดรอยด์

Code Name	Version	API Level
Oreo	8.0	26
Nougat	7.1	25
	7.0	24
Marshmallow	6.0	23
Lollipop	5.1	22
	5.0	21
KitKat (Watch)	4.4w	20
KitKat	4.4-4.4.4	19
Jelly Bean	4.3.x	18
	4.2.x	17
	4.1.x	16
Ice Cream Sandwich	4.0.3-4.0.4	15
	4.0.1-4.0.2	14

2.3 Android Studio

Android Studio เป็นเครื่องมือพัฒนา (IDE: Integrated Development Environment) จาก Google ที่ถูกสร้างขึ้นมาเพื่อการพัฒนาแอนดรอยด์แอปพลิเคชัน บนพื้นฐานของแนวคิด IntelliJ คล้ายกับการทำงานของ Eclipse หรือ Netbean และ Android ADT Plugin วัตถุประสงค์ของ Android Studio คือต้องการพัฒนาเครื่องมือ IDE ที่สามารถพัฒนาแอปพลิเคชั่นบนแอนดรอยด์ โดยเฉพาะให้มีประสิทธิภาพมากขึ้น ทั้งด้านการออกแบบ GUI ที่ช่วยให้สามารถดูภาพตัวอย่างของตัวแอปพลิเคชัน ในมุมองที่แตกต่างกันบนสมาร์ทโฟน แต่ละรุ่นสามารถแสดงผลบางอย่าง

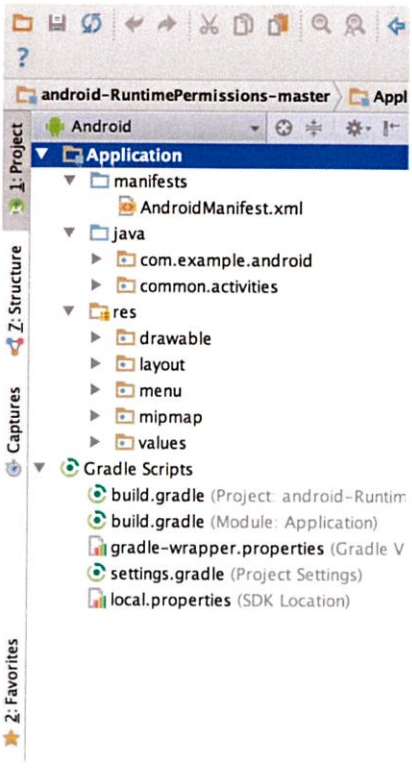
ได้ทันทีโดยไม่ต้องทำการรันแอปพลิเคชันบนสมาร์ทโฟนจำลอง (Emulator) รวมทั้งยังแก้ไขปรับปรุงในเรื่องของความเร็วของ Emulator ที่ยังเจอปัญหากันอยู่ในปัจจุบัน

รูปที่ 2.2 Android Studio

โปรแกรม Android Studio จะแสดงโปรเจ็คไฟล์ในมุมมอง Android Project ซึ่งจะแตกต่างจาก Eclipse หรือ Netbeans โดยเน้นถึงการใช้งานง่ายและรวดเร็ว โดยในทุก ๆ Android Project จะมีไฟล์ 2 ส่วนเกินออกมา คือ โฟลเดอร์ Application จัดเก็บ Source Code โดยภายในจะแบ่งเป็นพื้นที่จัดเก็บ 3 ส่วน ดังนี้

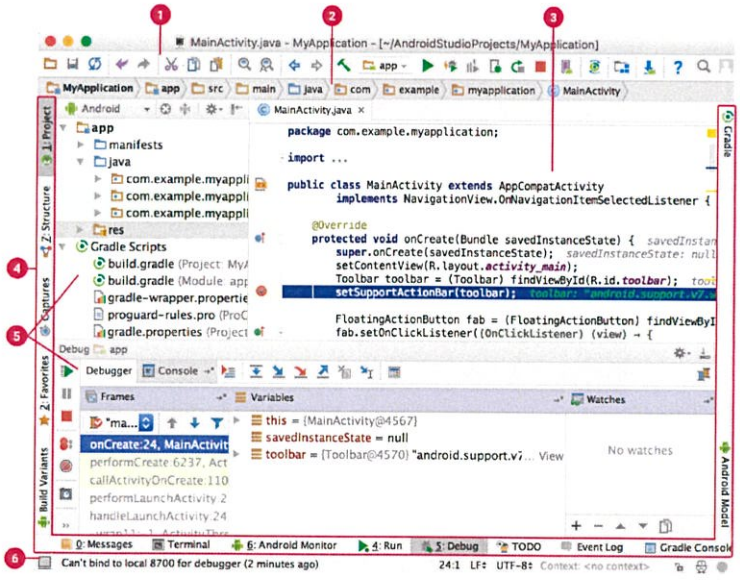
1. manifests จัดเก็บไฟล์ AndroidManifest.xml
2. java จัดเก็บไฟล์ Java source code และ JUnit test code
3. res จัดเก็บ non-code resources เช่น XML layouts, UI strings, and bitmap images

และส่วนของ Gradle Scripts จะทำหน้าที่จัดการเรื่องการ Build Android Application ดังรูป



รูปที่ 2.3 โฟลเดอร์ต่าง ๆ ของ Android Studio

2.3.1 การใช้งานของแถบเครื่องมือต่าง ๆ



รูปที่ 2.4 แถบเครื่องมือต่าง ๆ ของ Android Studio

1) toolbar คือแถบเครื่องมือที่มีหลากหลายการทำงานให้เลือก รวมถึงการรันแอปพลิเคชันและเข้าถึงอุปกรณ์

2) navigation bar คือแถบเครื่องมือที่บอกที่อยู่ของไฟล์ต่าง ๆ และทำให้เข้าถึงไฟล์ได้ง่ายขึ้น

3) editor window คือพื้นที่แสดงโค้ด และกราฟิกต่าง ๆ

4) tool window bar คือแถบหน้าต่างที่วางอยู่รอบ ๆ หน้าต่าง IDE ซึ่งจะประกอบด้วยปุ่มที่สามารถกดขยายหรือเก็บเข้าไปได้

5) tool windows คือหน้าต่างที่ใช้จัดการ project เช่น แสดงไฟล์ในโปรเจ็ค แสดงการ debug แสดง console ต่าง ๆ ซึ่งจะวางอยู่รอบ ๆ editor

6) status bar คือแถบแสดงสถานะของ project เช่น error หรือ warning

2.3.2 Activity ของแอนดรอยด์

ในระบบแอนดรอยด์ จะเรียกการแสดงผลแต่ละหน้าของแอปพลิเคชันว่า Activity ซึ่งเทียบได้กับแต่ละหน้าของเว็บเพจนั่นเอง โดยเราสามารถเลื่อนไปมาระหว่าง Activity ได้ เช่นดียวกับเว็บเพจ และในแต่ละ Activity นั้นเราสามารถจัดวางองค์ประกอบต่าง ๆ เพื่อสร้าง UI ได้ตามต้องการ ซึ่งภายในแอปพลิเคชันหนึ่งๆ เราจะมีกี่ Activity ก็ได้ โดยเมื่อเราเปิดแอป Activity แรกที่จะนำขึ้นมาแสดงเรียกว่า Main Activity ซึ่งก็เหมือนกับหน้า Homepage หรือ index ของเว็บไซค์นั่นเอง

ลักษณะของคลาส Activity ในมุมมองของการเขียนโค้ด Activity คือคลาสอันดับหนึ่งของแอนดรอยด์ (อยู่ในแพ็คเกจ android.support.v7.app) ซึ่งเมื่อเราจะสร้างแต่ละหน้าของแอปพลิเคชัน ก็ให้สร้างคลาสสำหรับหน้านั้น โดยให้คลาสสืบทอด (extends) มาจากคลาส Activity เช่น

```
import android.support.v7.app.Activity;
class MainActivity extends Activity {
. . .
}
```

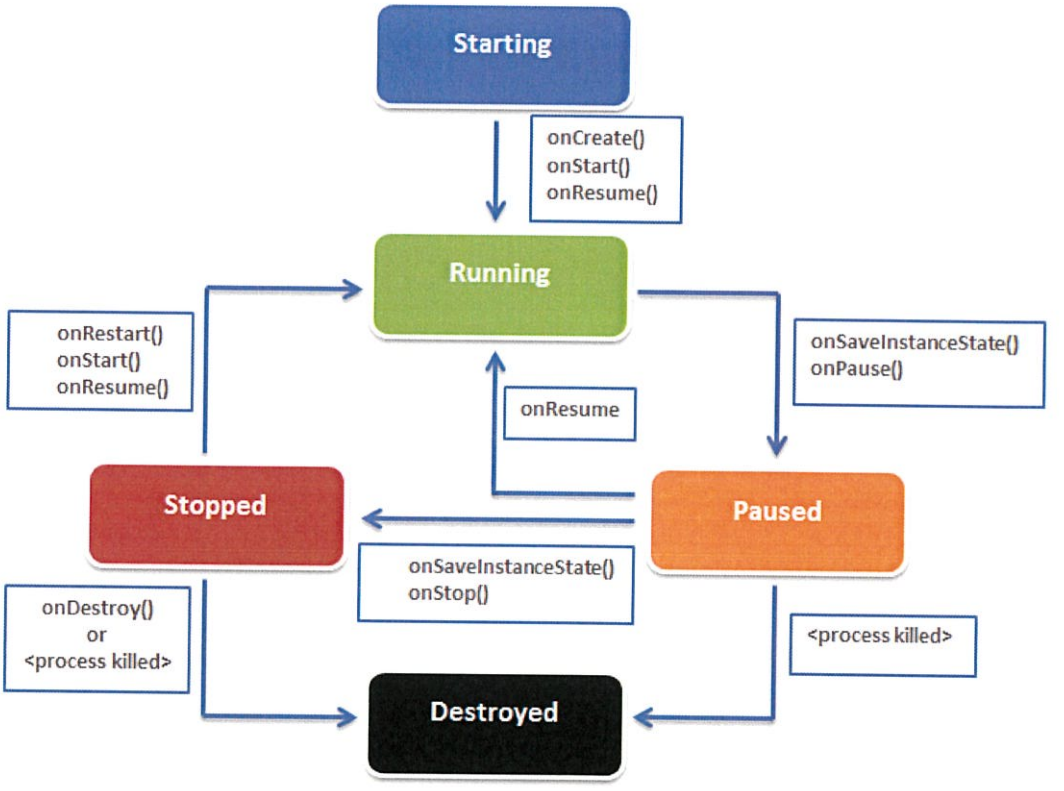
ถ้าเราจะสร้างหน้า Activity อื่น ๆ ก็ทำแบบเดียวกันทั้งหมด อย่างไรก็ตาม ในแอนดรอยด์ยังมีซับคลาสของ Activity อีกบางส่วน เช่น ActionBarActivity, AppCompatActivity ซึ่งคลาส

เหล่านี้ก็เพียงแต่ใส่องค์ประกอบบางอย่างเข้าไปด้วย และใช้งานได้เลย จึงสะดวกรวดเร็วกว่าการที่เราจะเพิ่มเข้าไปเอง เช่น ถ้าเราต้องการให้ Activity นั้นมี ActionBar ด้วย ก็ให้สืบทอดจากคลาส ActionBarActivity เป็นต้น เช่น โค้ดต่อไปนี้

```
class MainActivity extends ActionBarActivity {
. . .
 }
```

2.3.2.1 วงจรสถานะของ Activity

Activity จะมีการเปลี่ยนแปลงสถานะไปตามลักษณะและการกระทำที่เกิดขึ้นในแต่ละช่วงเวลา หรือที่เรียกว่า Lifecycle State ซึ่งถูกกำหนดและควบคุมโดย Activity Manager ทั้งนี้เราสามารถแบ่งสถานะที่เกิดกับ Activity ได้ดังภาพต่อไปนี้



รูปที่ 2.5 วงจรสถานะของ Activity

- Launched เป็นสถานะที่ Activity เริ่มทำงานเมื่อเปิดแอพพลิเคชั่น แต่ยังไม่ได้ปรากฏบนหน้าจอ โดยอยู่ในระหว่างการโหลดเข้าสู่หน่วยความจำ
- Running เป็นสถานะที่ Activity ปรากฏบนหน้าจอ และกำลังถูกใช้งานโดยผู้ใช้ หรืออาจกล่าวได้ว่าเป็นสถานะที่ Activity กำลัง Active อยู่นั่นเอง ดังนั้นจึงมีเพียง Activity เดียวเท่านั้นที่อยู่ในสถานะ Running
- Paused เป็นสถานะที่ Activity ยังปรากฏบนหน้าจอ แต่ไม่ได้ถูกใช้งานใด ๆ ในขณะนั้น หรือไปอยู่เบื้องหลัง เช่น กรณีการแสดงไดอะล็อกซึ่งจะปรากฏอยู่ข้างหน้า Activity เพื่อรอการตอบสนองจากผู้ใช้ก่อน เป็นต้น
- Stopped เป็นสถานะที่ Activity ไม่ปรากฏบนหน้าจอ แต่ยังคงอยู่ในหน่วยความจำ เช่น กรณีที่เรากดปุ่ม Home ซึ่งจะทำให้ Activity นั้นหายไปจากจอ (โดยที่ยังไม่ถูกปิด) ทั้งนี้ Activity ที่อยู่ในสถานะดังกล่าวนี้ สามารถกลับไปสู่สถานะ Running ได้ใหม่ ดังจะเห็นได้ว่า ถ้าเราเปิดแอปเดิมที่เคยเปิดค้างเอาไว้ มันจะปรากฏขึ้นมาเร็วมาก และสามารถใช้งานต่อจากเดิมได้เลย นอกจากนี้แล้ว Activity ที่อยู่ในสถานะ Stopped อาจจะถูกทำลายโดยระบบก็ได้ เช่น กรณีหน่วยความจำเหลือน้อย
- Destroyed เป็นสถานะที่ Activity ถูกทำลาย และหายไปจากหน่วยความจำ จึงไม่สารถกลับไปสู่สถานะอื่น ๆ ได้อีก เช่น การกดปุ่ม Back เป็นต้น แต่ก่อนที่ Activity จะถูกทำลาย เราสามารถกระทำการบางอย่างได้ เช่น การบันทึกข้อมูล เป็นต้น อย่างไรก็ตามทางผู้พัฒนา Android แนะนำว่า Activity อาจถูกทำลายขณะที่อยู่ในสถานะ Stopped ได้เช่นกัน ดังนั้นหากจะมีการบันทึกข้อมูลต่าง ๆ ควรจะทำเมื่อเข้าสถานะ Stopped จะปลอดภัยกว่า

การเปลี่ยนสถานะต่าง ๆ ดังที่กล่าสวมานี้ เราไม่ได้เป็นผู้สั่งการเอง แต่มัดจะเป็นผลสืบเนื่องจากการกระทำต่าง ๆ เช่น การกดปุ่ม Home, ปุ่ม Back หรือการเปิด-ปิดแอปต่าง ๆ เป็นต้น โดยในแอนดรอยด์จะมี Activity Manager ซึ่งทำหน้าที่เป็นตัวควบคุมการเปลี่ยนแปลงสถานะของ Activity ให้เป็นไปโดยอัตโนมัติ

2.3.2.2 Callback Method ของ Activity

ในทางการเขียนโปรแกรม เมธอดที่กำหนดให้ถูกเรียกขึ้นมาทำงานเมื่อมีบางอย่างเกิดขึ้น เราเรียกว่าเป็น Callback Method ทั้งนี้จากภาพวงจรสถานะของ Activity จะพบว่ามีการระบุชื่อเมธอดบางส่วนเอาไว้ระหว่างการเปลี่ยนสถานะต่าง ๆ ซึ่งก็คือเมธอดที่ Activity Manager จะเรียกขึ้นมาทำงานนั่นเอง โดยเมธอดเหล่านี้เป็นของคลาส Activity และจากการที่เราสร้างคลาสโดยให้สืบทอดมาจากคลาส Activity แสดงว่าเราสามารถโอเวอร์ไรด์เพื่อกำหนดการกระทำที่ต้องการไว้ในเมธอดเหล่านั้นได้ เช่น ถ้าเราต้องการให้บันทึกข้อมูลเมื่อเข้าสู่สถานะ Stopped ก็ให้กำหนดการกระทำไว้ในเมธอด onStop() เป็นต้น ลักษณะโดยสังเขปของเมธอดเหล่านั้นมีดังนี้

ตารางที่ 2.3 หน้าที่ของเมธอดต่าง ๆ

onCreate(Bundle)	จะถูกเรียกเพียงครั้งเดียวเมื่อ Activity ถูกสร้างขึ้น เมธอดนี้จะมีคลาส Bundle เป็นพารามิเตอร์ ซึ่งเป็นเมธอดที่เราต้องใช้งานกันไปตลอด
onStart()	ถูกเรียกเมื่อ Activity เริ่มปรากฏบนหน้าจอ
onResume()	จะถูกเรียกเมื่อ Activity เริ่มการติดต่อ (Interacting) กับผู้ใช้ ซึ่งเรานิยมกำหนดภาพเคลื่อนไหว หรือเริ่มเล่นไฟล์มัลติมิเดียในเมธอดนี้
onPause()	จะถูกเรียกเมื่อ Activity ไปอยู่ข้างหลัง (Background) เพื่อให้ Activity อื่นขึ้นมาอยู่ข้างหน้าแทน เรานิยมบันทึกข้อมูลสถานะต่างๆของ Activity ไว้ในเมธอดนี้
onStop()	ถูกเรียกเมื่อ Activity ไม่ปรากฏบนหน้าจอ (แต่ยังอยู่ใน Memory) คล้ายกับการ Minimixed บนระบบ Windows นั่นเอง
onRestart()	ถูกเรียกเมื่อ Activity กลับมาปรากฏบนหน้าจออีกครั้ง
onDestroy()	ถูกเรียกเมื่อ Activity กำลังจะถูกทำลาย

2.3.2.3 การ Override เมธอดของ Activity

ในการเขียนโค้ด หากเราต้องการใช้เมธอดใดๆ ก็ให้โอเวอร์ไรด์เมธอดนั้นในคลาสของ Activity ในลักษณะดังนี้

```
class MyActivity extends Activity {
        @Override
protected void onCreate(Bundle saveInstantState)  {
   super.onCreate(saveInstantState);
                . . .
         }
         @Override
        protected void onStart()
         . . .
         }
         . . .
}
```

ส่วนใหญ่จะใช้เมธอด onCreate() เป็นหลัก เพื่อกำหนดองค์ประกอบต่างๆ ของ Activity เพราะเมธอดนี้ถูกเรียกเพียงครั้งเดียวเมื่อสร้าง Activity

2.3.3 พื้นฐานเกี่ยวกับ Android Application

ไฟล์และโฟลเดอร์ต่างๆในโปรเจ็กต์แต่ละโปรเจ็กต์ของการพัฒนา Android Application นั้นจะประกอบด้วยโฟลเดอร์และไฟล์ต่างๆ มากมาย ซึ่งส่วนใหญ่โปรแกรมจะสร้างให้โดยอัตโนมัติเมื่อเราสร้างโปรเจ็กต์ใหม่ สำหรับโฟลเดอร์และไฟล์สำคัญที่เราควรรู้จักในเบื้องต้นมีดังนี้คือ

ตารางที่ 2.4 โฟลเดอร์และไฟล์ที่สำคัญ

App	เป็นโฟลเดอร์หลักที่เก็บไฟล์ทั้งหมดของแอปพลิเคชัน
manifests	เก็บไฟล์ AndroidManifest.xml ซึ่งงเป็นไฟล์ที่แสดงการปรับตั้งค่าของแอปพลิเคชัน หรืออาจเทียบได้กับไฟล์ Configuration หรือ Setting ในโปรแกรมทั่ว ๆ ไปนั่นเอง ซึ่งอาจกล่าวได้ว่า ถ้าเราจะตั้งค่าอะไรก็ตามของแอปพลิเคชัน ก็ให้มาที่ไฟล์ AndroidManifest.xml เป็นอันดับแรก
Java	โฟลเดอร์ src ใช้ในการเก็บไฟล์ต่าง ๆ ที่เป็นโค้ด java เช่น Activity.java
Res	โฟลเดอร์ res ใช้ในการเก็บไฟล์ทรัพยากรต่าง ๆ ที่นำมาใช้ในแอป เช่น ค่าสตริง, สี, ขนาด, รูปภาพ หรือมัลติมิเดียต่าง ๆ โดยภายในโฟลเดอร์ res จะแบ่งออกเป็นโฟลเดอร์ย่อย สำหรับแยกจัดเก็บข้อมูลที่ต่างกัน เช่น value, menu, layout เป็นต้น

2.3.4 แอตทริบิวต์พื้นฐานของ View

ถึงแม้ว่าใน Android Studio จะมีเครื่องมือมาช่วยในการสร้าง GUI โดยการลากแล้ววางในแบบ Visual รวมถึงการใช้เม้าส์ในการเคลื่อนย้ายตำแหน่งได้ แต่อย่างไรก็ตาม เราอาจใช้เม้าส์ได้เพียงบางกรณีเท่านั้น แต่ส่วนใหญ่แล้วเราไม่สามารถใช้เม้าส์ในการเปลี่ยนมุมมอง Drsign ได้ เช่น การเปลี่ยนขนาด (ความกว้างหรือสูง), การแก้ไขขเกี่ยวกับฟอนต์ หรือการปรับแต่งอื่นๆ ซึ่งกรณีเหล่านี้ เราต้องเข้าไปแก้ไขในมุมมอง Text เอาอง หรือไม่ก็แก้ไขผ่านช่องรายการ Properties อย่างไรก็ตาม การฺเราจะแก้ไขสิ่งใดนั้น ประการแรกต้องทราบก่อนว่า สิ่งที่จะแก้ไขนั้น กำหนดด้วยแอตทริบิวต์หรือพร็อปเพอร์ตี้อะไร เช่น ถ้าเราจะแก้ไข “ข้อความ” บนวิวนั้นก็ต้องแก้ที่แอตทริบิวต์ text หรือถ้าจะเปลี่ยนความกว้างก็ต้องไปแก้ที่แอตทริบิวต์ layout_width เป็นต้น ทั้งนี้ แอตทริบิวต์พื้นฐานของ View และ Layout ที่เราควรรู้จักในเบื้องต้นมีดังนี้คือ

ตารางที่ 2.5 แอตทริบิวต์พื้นฐานของ View และ Layout เบื้องต้น

Id	วิวทุกตัวที่เรานำมาสร้าง UI ของแอปพลิเคชัน จะต้องกำหนดค่า id ให้กับมันเสมอ ใช้ในการอ้างอิงทั้งในส่วนของโค่ดจาวา รวมถึงในไฟล์ xml ด้วย ทั้งนี้เมื่อเรานำวิวมาวางบนหน้าจอในมุมมอง Design โปรแกรมจะสร้างชื่ออ้างอิงให้ในแบบ @+id/ชื่ออ้างอิง เช่น @+id/button_ok หรือ @+id/text_login เป็นต้น โดยที่ - @ เป็นสัญลักษณ์ Annotation ที่ใช้อ้างอิงข้อมูลใน Resource (ไฟล์ R.java) - + เป็นเครื่องหมายบวก หมายถึง ถ้ายังไม่มี id ชื่อนี้ในไฟล์ Resource ให้ใช้โปรแกรมสร้าง id ชื่อนี้ขึ้นมาใหม่ แล้วเพิ่ม id ลงใน Resource ให้ด้วย
Text	ข้อความที่ปรากฏบนวิวนั้นๆ
textAllCaps	ถ้ากำหนดเป็น true ข้อความบนวิวจะเป็นตัวพิมพ์ใหญ่ทั้งหมด (วิวบางชนิดมีค่าดีฟอลต์เป็น true) มิฉะนั้นให้กำหนดเป็น false เช่น android:textAllCaps= “false”

ตารางที่ 2.6 แอตทริบิวต์พื้นฐานของ View และ Layout เบื้องต้น (ต่อ)

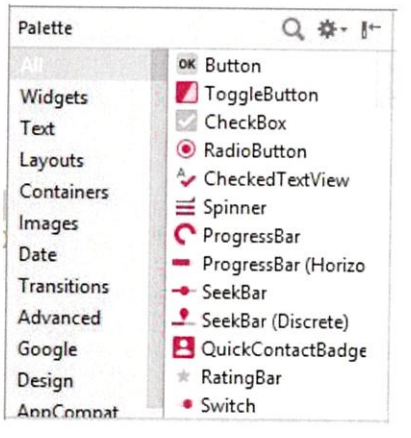
textSize	ขนาดตัวอักษรของข้อความ ซึ่งในแอนดรอยด์นิยมกำหนดขนาดของตัวฎักษรในหน่วย sp
textColor	สีของข้อความของวิว โดยกำหนดชื่อสีเป็นเลข hexadecimal ในแบบ #rgb และต้องเขียนเป็นตัวพิมพ์เล็กทั้งหมด เช่น android:textColor= “#00ff9e”
textStyle	รูปแบบข้อความ โดยค่าที่กำหนดได้คือ normal, bold, italic หรือถ้าจะใช้หลายรูปแบบร่วมกันให้เชื่อมด้วยเครื่องหมาย I เช่น boldlitalic (ต้องเขียนติดกัน)
background	สีพื้นหลังของวิว หลักการกำหนดสีเช่นเดียวกับ textColor
layout_width	ความกว้างของวิว สำหรับขนาดระยะต่างๆ ในระบบแอนดรอยด์นิยมใช้หน่วย dp (หรือ dip) เช่น 100 dp หรือกำหนดเป็นค่าคงที่อย่างใดอย่างหนึ่งคือ - match_parent หมายถึงให้กระจายเต็มพื้นที่ความกว้างของวิวที่เป็นคอนเท็นเนอร์ของมัน หรือถ้าวางบน Layout โดยตรงก็จะกว้างเท่ากับความกว้างของ Layout เป็นต้น (เริ่มใช้ใน API Level 8 โดย API 1-7 จะใช้คำว่า fill_parent) - wrap_content หมายถึงให้ความกว้างพอดีกับเนื้อหา หรือห่อหุ้มเนื้อหาได้พอดีนั่นเอง เช่น android:layout_width= “wrap_content”
layout_height	ความสูงของวิวส่ วนค่าที่กำหนดเช่นเดียวกันกับ layout_width
layout_margin	คำว่า margin ในแอนดรอยด์ เป็นการกำหนดระยะห่างจากแนวขอบทั้ง 4 ด้านของวิวกับวิวตัวอื่นที่อยู่รอบ ๆ หรืออาจจะเป็นแนวขอบของ Layout ก็ได้
layout_marginTop/ Right/Bottom/Left	เป็นการกำหนดระยะ margin เพียงด้านใดด้านหนึ่งตามชื่อของแอตทริบิวต์ เช่น layout_marginTop คือระยะห่างจากแนวขอบด้านบนของวิวอันนั้น กับวิวอื่นๆ ที่อยู่ด้านบน ดังนั้น แอตทริบิวต์ต่างๆ เหล่านี้จึงมีผลกับด้านใดด้านหนึ่งเท่านั้น

ตารางที่ 2.7 แอตทริบิวต์พื้นฐานของ View และ Layout เบื้องต้น (ต่อ)

layout_padding	เป็นการจัดระยะระหว่างเนื้อหากับแนวขอบทั้ง 4 ด้านของวิว ซึ่งแอตทริบิวต์นี้จะมีผลให้ทุกด้านมีระยะห่างเท่ากันหมด
layout_paddingTop/ Right/Bottom/Left	เป็นการกำหนดระยะ padding เพียงด้านใดด้านหนึ่งหนึ่งตามชื่อของแอตทริบิวต์ เช่น layout_paddingTop คือระยะห่างเนื้อหากับแนวขอบด้านบนของวิวนั้น
gravity	ใช้กับการจัดตำแหน่งเนื้อหาของวิว ว่าจะให้เลื่อนไปอยู่ด้านใด โดยค่าที่กำหนดให้แก่แอตทริบิวต์นี้ได้คือ (ที่น่าสนใจ) - left, right, top, bottom ให้อยู่ด้านซ้าย, ขวา, บน และล่าง ตามลำดับ - center, center_horizontal, center_vertical ให้อยู่กึ่งกลางของทั้งสองด้าน, อยู่กึ่งกลางในแนวนอน (ซ้ายกับขวา) และอยู่กึ่งกลางในแนวตั้ง (บนและล่าง) ตามลำดับ

2.3.5 การแบ่งกลุ่มของ View

ในระบบแอนดรอยด์ คอมโพเนนต์ที่ใช้ในการสร้าง User Interface (UI) จะเรียกว่า View โดยจะแบ่งออกเป็นกลุ่มต่างๆ ตามความเกี่ยวข้องกันดังนี้



รูปที่ 2.6 การแบ่งกลุ่มของ View

ตารางที่ 2.8 ลักษณะโดยสังเขปของ View ชนิดต่าง ๆ

Layouts	ใช้ในการจัดโครงร่างของ UI เช่น Relative Layout
Widgets	สำหรับที่เราจะกล่าวถึงในบทนี้คือ View Widget ซึ่งเป็นกลุ่มวิวพื้นฐานในการสร้าง UI เช่น Button, TextView ทั้งนี้ คำว่า Widget ในแอนดรอยด์ยังถูกใช้ในอีกความหมายหนึ่งคือ หมายถึงแอปพลิเคชันขนาดเล็กหรือที่เรียกว่า App Widget นั่นเอง
Text Fields	เป็นวิวสำหรับข้อมูลแบบข้อความจากผู้ใช้ แต่แบ่งออกเป็นชนิดย่อย ๆ ตามลักษณะการใช้งาน เช่น PlainText, Password, Email, Number เป็นต้น
Containers	เป็นวิวที่ใช้จัดวาง หรือเป็นตัวบรรจุวิวชนิดอื่นๆ เช่น ListView, GridView
Date & Time	เป็นกลุ่มวิวที่ใช้แสดงผลเกี่ยวกับวันและเวลา เช่น DatePicker, TimePicker
Expert	เป็นวิวสำหรับการสร้าง UI ขั้นสูงในระดับผู้เชี่ยวชาญ เช่น AutoCompleteTextView

2.3.6 การแจ้งเตือน

Notification คือข้อความแจ้งเตือนรูปแบบหนึ่งเท่านั้น ซึ่งจะแตกต่างจากพวก Toast หรือ Snackbar ตรงที่ไม่จำเป็นที่แอพพลิเคชันต้องทำงานอยู่ ก็สามารถแสดงการแจ้งเตือนข้อความ Notification ได้ นอกจากข้อความแจ้งเตือนแล้ว Notification ยังสั่น หรือมีเสียงได้อีกด้วย เราจะใช้ NotificationCompat ซึ่งอยู่ใน support library v4 โดยเริ่มต้นการทำการแจ้งเตือน โดยการออกแบบ layout ก่อน เช่น จะมีปุ่มสักปุ่มนึง กดแล้วจะให้เด้งแจ้งเตือนขึ้นมา

activity_main.xml

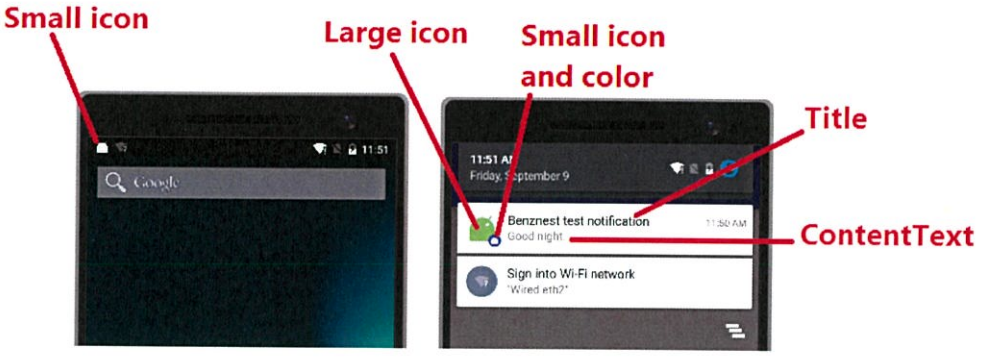
รูปที่ 2.7 ออกแบบ Layout ให้มีปุ่มกดให้แจ้งเตือน

ส่วนที่ MainActivity.java ให้ findViewById ของปุ่มให้เรียบร้อย พร้อมกำหนด onClick ของปุ่ม

```
public class MainActivity extends AppCompatActivity {

    Button btnShowNotification;


    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        btnShowNotification = (Button)
findViewById(R.id.btn_show_notification);

        btnShowNotification.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                showNotification();

            }

        });


}


   public void showNotification(){
```

```
        // do about noti.
    }
```

จะได้หน้าตา Notification จะแตกต่างไปตาม android เวอร์ชัน API



รูปที่ 2.8 เวอร์ชัน API 21

implement ส่วนของการแสดง notification กัน โดยหลักๆ คือต้อง set พวก icon และเนื้อหา ไอคอนเล็ก สามารถใช้จาก resource ได้ โดยมันคือ ไอคอนที่แสดงบน notification bar จะถูกแปลงเป็นสีขาวอัตโนมัติ ส่วนไอคอนใหญ่ต้องใช้ Bitmap ก็สามารถแปลง drawble resource มาเป็น Bitmap ได้

```
Bitmap bitmap = BitmapFactory.decodeResource(context.getResources(),
                R.mipmap.ic_launcher);
```

ทำการกำหนดค่าให้กับ Builder ซึ่งตัวนี้คือตัวสำคัญของเรื่องนี้ จากนั้นเรียกคำสั่ง .build() จะได้ object ของ Notification

```
Notification notification =
                new NotificationCompat.Builder(context)
                        .setSmallIcon(drawable)
                        .setLargeIcon(bitmap)
                        .setContentTitle("Benznest test notification")
                        .setContentText("Good night.")
                        .setAutoCancel(false)
                        .setColor(color)
                        .build();
```

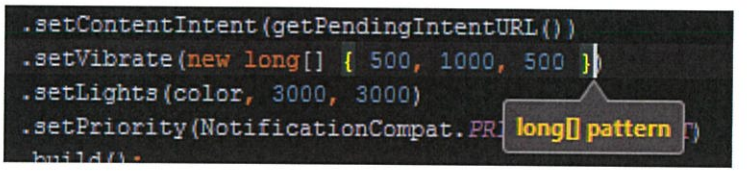
การใส่ AutoCancel คือ ถ้าผู้ใช้กด notification นั้นแล้ว ให้ลบอันที่กดออกไป

สามารถกำหนดการสั่นได้ และยังตั้ง รูปแบบการสั่นได้ จังหวะการสั่น โดยก่อนอื่นเพิ่ม permission ใน AndroidManifest.xml

```
<uses-permission android:name="android.permission.VIBRATE" />
```

ใช้คำสั่ง setVibrate() โดย array of long ก็คือ จังหวะการสั่น ตัวเลขคือสั่นเป็นหน่วย ms

```
.setVibrate(new long[] { 500, 1000, 500 })
```



รูปที่ 2.9 ตั้งค่าการสั่นของการแจ้งเตือน

เราสามารถกำหนดความสำคัญ ของ Notification ได้โดยใช้คำสั่งนี้

```
.setPriority(NotificationCompat.PRIORITY_DEFAULT)
```

หลังจากตั้งค่าไว้แล้วก็ต้องทำการส่งการแจ้งเตือนออกไป โดยจะทำโดย NotificationManager แล้วเรียก notify() ก็จะส่งออกไปแล้ว โผล่ให้ user เห็น

```
NotificationManager notificationManager =
(NotificationManager) context.getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.notify(1000, notification);
```

สรุปรวม

```
  private void showNotification() {
        Context context = MainActivity.this;
        int color = ContextCompat.getColor(context, R.color.colorPrimary);
```

```
        Bitmap bitmap = BitmapFactory.decodeResource(context.getResources(),
                R.mipmap.ic_launcher);

        Notification notification =
                new NotificationCompat.Builder(context)
                        .setSmallIcon(R.mipmap.ic_launcher)
                        .setLargeIcon(bitmap)
                        .setContentTitle("Benznest test notification")
                        .setContentText("Good night.")
                        .setAutoCancel(false)
                        .setColor(color)
                        .setVibrate(new long[] { 500, 1000, 500 })
                        .build();
        NotificationManager notificationManager =
                (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.notify(1000, notification);
}
```

บางกรณีก็ต้องการมีข้อความหลายบรรทัดใน Notification ทำการสร้าง InBoxStyle ขึ้นมาแล้ว add ข้อความลงไป แทนพวก textContent ได้เลย

```
        NotificationCompat.InboxStyle inboxStyle =
                new NotificationCompat.InboxStyle();
        inboxStyle.setBigContentTitle("Event tracker details:");
        for (int i = 0; i < 6; i++) {
            inboxStyle.addLine("Hello "+i); }
```

เอา inboxStyle ใส่ลงไปใน Notification.Builder เหมือนตัวอื่นๆ

```
.setStyle(inboxStyle)
```

ต่อมาก่อนจะสร้าง action ต้องสร้างอีกหน้ารอก่อน หลังนจากกดที่ Notification แล้วให้มาหน้าที่สองนี้ พร้อมส่งค่ามาด้วย และนำมาแสดง ทำการสร้าง activity ใหม่ให้เรียบร้อยชื่อ activity_main2.xmlกลับมาที่ ActivityMain.java ใส่คำสั่งเพิ่มเข้าไป

```
.setContentIntent( PendingIntent )
```

PendingIntent คือ intent ที่แอปพลิเคชันอื่นสามารถเอาไปใช้ได้ โดยการทำ pendingIntent ในกรณีนี้ มีขั้นตอนเล็กน้อย คือ ต้องไปสร้าง TaskStackBuilder ก่อน แล้ว add Intent ที่เราต้องการเปิดลงไป จากนั้นค่อยดึงมันออกมาจึงจะได้ PendingIntent โดยขั้นตอนนี้ เราจะ put ค่าที่เตรียมจะส่งไปด้วยเลยกับ Intent ในที่นี้คือ message

```
    private PendingIntent getPendingIntent(Context context,String message) {

        Intent intent = new Intent(context, Main2Activity.class);

        intent.putExtra("MESSAGE", message);

        TaskStackBuilder stackBuilder = TaskStackBuilder.create(context);

        stackBuilder.addParentStack(Main2Activity.class);

        stackBuilder.addNextIntent(intent);

        return stackBuilder.getPendingIntent(0,
PendingIntent.FLAG_UPDATE_CURRENT);

    }
```

รวมทั้งหมด ตั้งแต่สร้าง Notification แล้ว set ค่าให้ เพิ่ม action แล้วส่งออกไป

```
private void showNotification() {

        Context context = MainActivity.this;

        int color = ContextCompat.getColor(context, R.color.colorPrimary);

        Bitmap bitmap = BitmapFactory.decodeResource(context.getResources(),

                R.mipmap.ic_launcher);

        Notification notification =
```

```
                new NotificationCompat.Builder(context) // this is context
                        .setSmallIcon(R.mipmap.ic_launcher)
                        .setLargeIcon(bitmap)
                        .setContentTitle("Benznest test notification")
                        .setContentText("Good night.")
                        .setAutoCancel(false)
                        .setColor(color)
                        .setContentIntent(getPendingIntent(context , "Hello
World!!"))
                        .setVibrate(new long[] { 500, 1000, 500 })
                        .setLights(color, 3000, 3000)
                        .build();



        NotificationManager notificationManager =
                (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.notify(1000, notification);
    }


    private PendingIntent getPendingIntent(Context context,String message) {
        Intent intent = new Intent(context, Main2Activity.class);
        intent.putExtra("MESSAGE", message);
        TaskStackBuilder stackBuilder = TaskStackBuilder.create(context);
        stackBuilder.addParentStack(Main2Activity.class);
        stackBuilder.addNextIntent(intent);
        return stackBuilder.getPendingIntent(0,
PendingIntent.FLAG_UPDATE_CURRENT);
    }
```

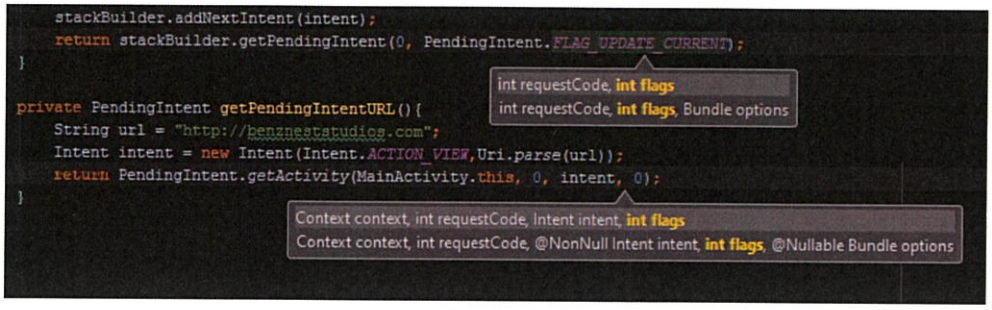
ในส่วนของการรับค่าที่ส่งมาใน หน้าที่สอง ทำการรับค่าจาก Intent แล้วใส่ให้แสดงใน textView

Main2Activity.java

```
public class Main2Activity extends AppCompatActivity {


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);


        TextView tv = (TextView) findViewById(R.id.tv_answer);
        Bundle bundle = getIntent().getExtras();
        if(bundle != null){
            tv.setText(bundle.getString("MESSAGE"));
        }
    }
}
```

เมื่อกดบน Notification จะเด้งไป activity นั้น ๆ

จะเห็นพารามิเตอร์นึงที่น่าสนใจ ของ getPendingIntent , getActivity คือ `int flags` หน้าที่ของมันก็คือกำหนด behavior ของ pendingIntent นั่นเอง



รูปที่ 2.10 พารามิเตอร์ `int flags`

โดยที่

FLAG_ONE_SHOT คือ ใช้ได้ครั้งเดียว ถ้าเรียกซ้ำมันจะไม่ทำงาน

FLAG_NO_CREATE คือ ไปเช็คก่อนว่า มีอยู่มัยถ้า มีอยู่แล้วมันจะไม่สร้างใหม่ พร้อมกับ return null

FLAG_CANCEL_CURRENT คือ อันก่อนหน้า ถ้ามีอยู่แล้วจะถูกยกเลิก แล้วสร้างอันใหม่

FLAG_UPDATE_CURRENT คือ ถ้ามีอยู่แล้ว จะทำการไปอัพเดท

ในความเป็นจริง เมื่อผู้ใช้กดดูแอปจาก notification แล้ว ควรลบมันออกจาก Notification center ถ้าตั้ง autoCancel ใน builder พอผู้ใช้กด มันจะลบให้อัตโนมัติ

```
.setAutoCancel(true)
```

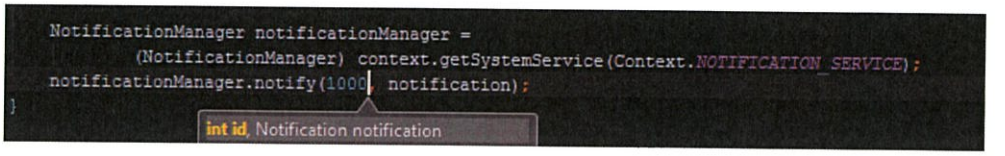
แต่บางครั้งอาจจะมีหลาย notification ที่ต้องการลบ คำสั่งลบทั้งหมด สามารถใช้ cancelAll() ได้เลย

```
NotificationManager notificationManager =

                (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);

notificationManager.cancelAll();
```

หรือจะลบเฉพาะ notifaication ก็ได้เช่นกันโดยใช้ id

```
notificationManager.cancel(id);
```

โดย id ก็มาจากตอนที่เราใส่ไปตอน notify นั่นเอง



รูปที่ 2.11 การดู id

การอัพเดท เพียงแค่ใช้ id เดิมกับ notification ที่แสดงอยู่ก็จะอัพเดททันที

```
NotificationManager notificationManager =

(NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);

notificationManager.notify(id, notification);
```

การกำหนด Visibility ของ Notification ที่ Builder ก็ให้ใช้คำสั่ง setVisibility() มี Visibility ให้เลือก 3 ตัว

VISIBILITY_PUBLIC แสดงตามปกติ ทั้ง notification center, lock screen

VISIBILITY_SECRET ไม่แสดงบน lock screen

VISIBILITY_PRIVATE แสดงแต่หัวข้อและไอคอน ไม่แสดงเนื้อหา

นอกจากนี้ยังใส่รูปภาพใน Notification ได้ด้วย เพิ่มความน่าสนใจให้กับ notification คือ ถ้าใช้แบบแสดงรูป .setContentText() จะใช้ไม่ได้ต้องไปใส่ body ใน notiStyle.setSummaryText(body); ของ style แทน

```
NotificationCompat.BigPictureStyle notiStyle = new
NotificationCompat.BigPictureStyle();

notiStyle.setSummaryText(body);

Bitmap picture = null;
```

```
        try {

            picture = BitmapFactory.decodeStream((InputStream) new
URL(url_image).getContent());

            if (picture != null) {

                notiStyle.bigPicture(picture );

            }

        } catch (IOException e) {

            e.printStackTrace();

        }



        NotificationCompat.Builder notificationBuilder = new
NotificationCompat.Builder(context)

                .setSmallIcon(R.mipmap.ic_launcher)

                .setContentTitle(title)

                .setContentText(body)

                .setAutoCancel(true)

                .setContentIntent(pendingIntent)

                .setStyle(notiStyle);



        NotificationManager notificationManager = (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);

        notificationManager.notify(100, notificationBuilder.build());
```

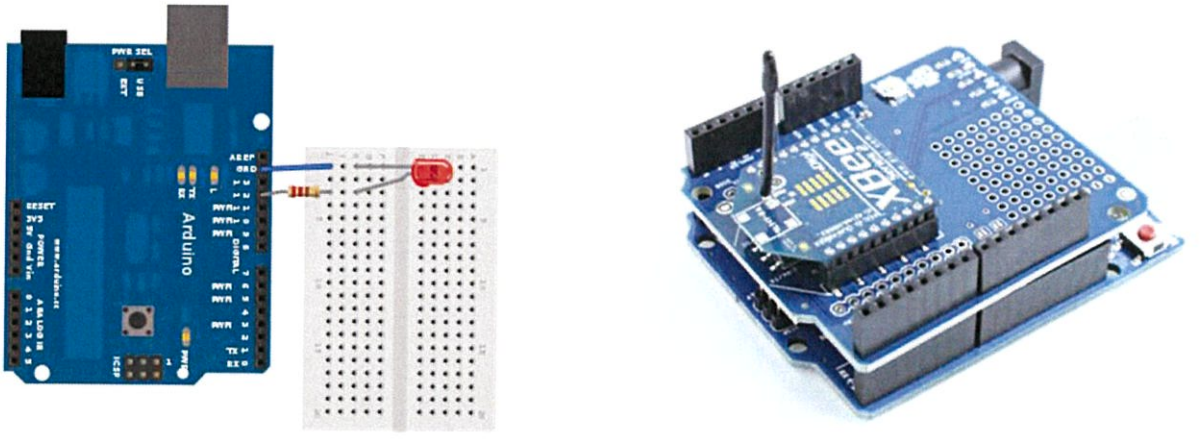
2.4 Microcontroller board

2.4.1 Arduino Uno

รูปที่ 2.12 บอร์ด Arduino Uno

เป็นบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ที่มีการพัฒนาแบบOpen Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้น จึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย

บอร์ด Arduino ใช้ในการต่ออุปกรณ์เสริมต่างๆ ได้อย่างง่ายดาย คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ รูปที่ 2.13 เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย

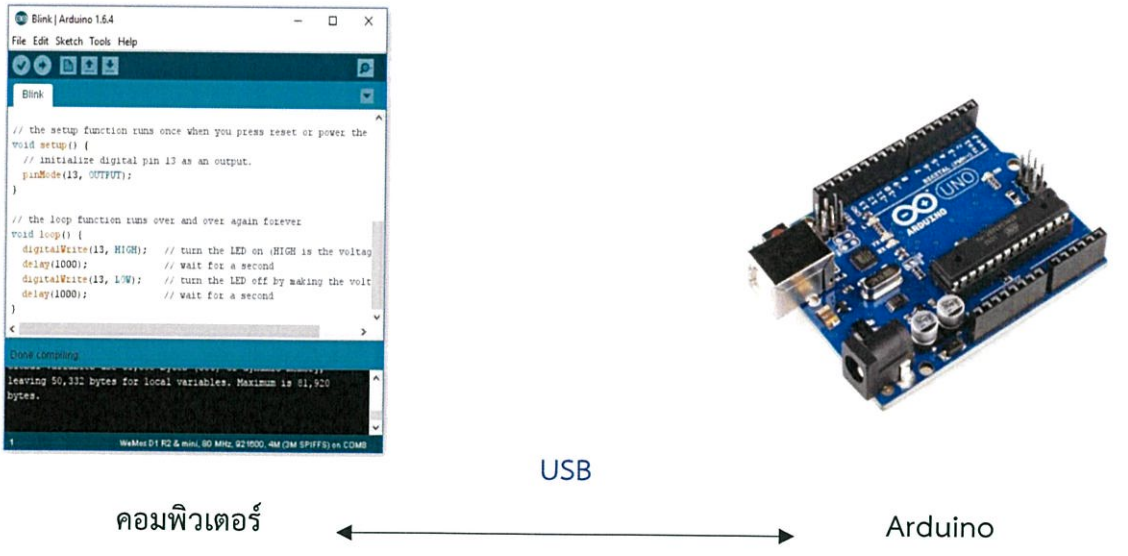


รูปที่ 2.13 บอร์ด Arduino และอุปกรณ์เสริม Arduino XBee Shield

โดยจุดเด่นของ Arduino คือเป็น Open Hardware ทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้าน ตัวบอร์ดราคาไม่แพง สามารถพัฒนาโปรแกรมบน OS ใดก็ได้ และยังมีกลุ่มคนที่ร่วมกันพัฒนาที่แข็งแรงอีกด้วย

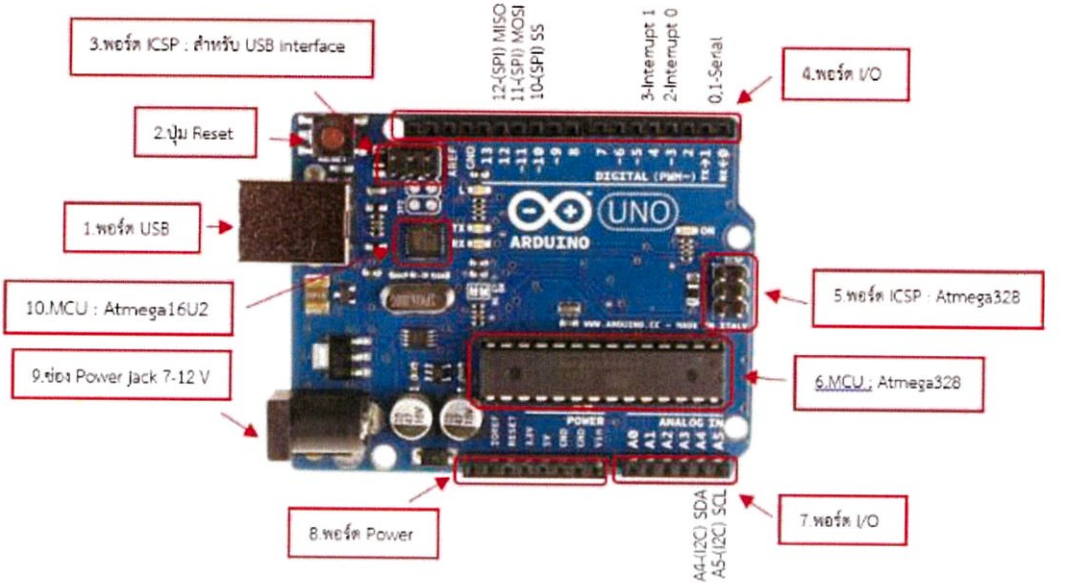
รูปที่ 2.14 โปรแกรม Arduino IDE

การเขียนโปรแกรมบน Arduino จะเขียนโปรแกรมบนคอมพิวเตอร์ด้วยภาษาซี ผ่านทางโปรแกรม Arduino IDE (Arduino Integrated Development Environment) โดยทั้งสองเชื่อมต่อกันผ่าน USB



รูปที่ 2.15 การเชื่อมต่อกันของ Arduino IDE และบอร์ด Arduino

โดย Layout และ Pin out ของบอร์ด Arduino มีรายละเอียด ดังนี้



รูปที่ 2.16 หน้าที่ของแต่ละ Pin ของบอร์ด Arduino

1) USB Port ใช้สำหรับต่อกับ Computer เพื่ออัปโหลดโปรแกรมเข้า MCU และจ่ายไฟให้บอร์ด

2) Reset Button เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่

3)ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2

4) Input/Output Port: Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วย เช่น Pin 0,1 เป็นขา Tx, Rx Serial, Pin3,5,6,9,10 และ 11 เป็นขา PWM

5) ICSP Port Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader

6) MCU Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino

7) Input/Output Port: นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็น ช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5

8) Power Port: ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วยขาไฟเลี้ยง +3.3 V, +5V, GND, V_{IN}

9) Power Jack: รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V

10) MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2

2.4.2 Arduino Mega

รูปที่ 2.17 Arduino Mega2560

Mega 2560 R3 เป็นบอร์ดไมโครคอนโทรลเลอร์ที่ใช้ชิพ ATmaga2560 ซึ่งมี 54 ดิจิตอลอินพุต/เอาท์พุต โดยในขาเหล่านั้นสามารถใช้งานเป็น PWM ได้ 15 ขา, อนาล็อคอินพุต 16 ขา, UART 4 ชุด โดยความถี่คริสตรัลบนบอร์ดคือ 16 MHz เชื่อมต่อข้อมูลระหว่างคอมพิวเตอร์ผ่านพอร์ต USB บนบอร์ดได้โดยตรง อีกทั้งรูปแบบการออกแบบยังออกแบบให้รองรับการสวมกับ Shield ต่าง ๆ ได้โดยตรง ทำให้สามารถพัฒนาระบบต่าง ๆ ได้อย่างรวดเร็วและเรียบร้อยสวยงาม โดยรองรับการพัฒนาโปรแกรมบนแพลตฟอร์ม Arduino อย่างเต็มรูปแบบโดยบอร์ด Arduino Leonardo นี้เป็นไมโครคอนโทรลเลอร์ที่สามารถต่อไฟเลี้ยงได้ทั้งการเชื่อมต่อเข้ากับ USB cable หรือ จ่ายไฟด้วย AC-DC adapter หรือ การใช้แบตเตอรี่ ซึ่ง Mega เป็นบอร์ดที่เข้ากันได้กับ shield ที่ออกแบบมาเพื่อ Arduino Duemilanove หรือ Diecimila

Mega 2560 นี้มีความแตกต่างจากบอร์ดก่อนหน้าตรงที่ไม่ใช้ FTDI USB-to-serial driver chip แต่จะมี ATmega16U2 เข้ามาเป็นโปรแกรมแปลง USB-to-serial

Arduino Mega2560 Revision 2 มี ATmega8U2 ทำให้อัพเดท firmware ผ่าน USB protocal ที่เรียกว่า DFU (Device Firmware Update) ได้ง่ายขึ้น

Arduino Mega Revision 3 มี featureใหม่ ๆ เพิ่มขึ้นมา ได้แก่ 1.0 pinout เพิ่ม SDA และ SCL (อยู่ใกล้กับ AREF pin) และอีกสอง pins ใหม่คือ IOREF เป็น pin ที่ใช้ในการเชื่อมต่อกับ shields เพื่อแปลงเป็นแรงดันที่ได้จากบอร์ด ส่วนอีก 1 pin ที่เหลือมีไว้สำหรับใช้ร่วมกับ AVR ในอนาคต มีวงจร Reset ที่ดีขึ้นและใช้ ATmega 16U2 แทน 8U2

2.4.2.1 Power

Arduino Mega สามารถเชื่อมรับพลังงานโดยการเชื่อมต่อ micro USB connector หรือ จาก power supply จากภายนอกได้ โดยแหล่งพลังงานจะถูกเลือกโดยอัตโนมัติ แหล่งจ่ายจากภายนอกสามารถมาได้จาก AC-to-DC adapter หรือจากแบตเตอรี่ โดยต่อเข้ากับ 2.1mm center-positive plug ไปยังช่องเสียบแหล่งจ่าย และการต่อเข้ากับแบตเตอรี่สามารถทำได้โดยการต่อเข้ากับ GND และ Vin pin header ของ power connecter

บอร์ดสามารถทำงานได้ในช่วงแรงดัน 6 ถึง 20 volts ถ้า แหล่งจ่ายมีค่าต่ำกว่า 7 V อาจส่งผลให้ 5 V pin มีแรงดันที่ต่ำกว่า 5V และ บอร์ดอาจจะไม่เสถียร แต่ถ้าหากแรงดันมีค่าสูงกว่า 12 V อาจส่งผลให้บอร์ด Overheat และอาจทำให้บอร์ดเสียหายได้ ดังนั้นช่วงแรงดันที่เหมาะสมกับบอร์ดคือ 7 V ถึง 12 V

- **VIN** เป็น input voltage ของบอร์ด Arduino โดยใช้แหล่งจ่ายจากภายนอก
- **5V** เป็น output pin ที่ควบคุม 5 V จากบอร์ด
- **3V3** เป็น 3.3 volt supply ที่สร้างจาก regulator บนบอร์ด ให้กระแสได้สูงสุด 50 mA
- **GND** เป็น ground pin
- **IOREF** เป็น pin ที่ให้ voltage reference กับไมโครคอนโทรลเลอร์ เพื่อเลือกค่าแรงดันให้กับ shield ที่มาเชื่อมต่อกับบอร์ด

2.4.2.2 Memory

ATmega2560 มีหน่วยความจำ 256 KB (8 KB ใช้สำหรับ bootloader) นอกจากนี้ยังมีอีก 8 KB สำหรับ SRAM และ 4 KB สำหรับ EEPROM

2.4.2.3 ฟังก์ชันอื่น ๆ เพิ่มเติม

- **Serial: 0 (Rx) และ 1(Tx); Serial 1: 19(Rx) และ 18 (Tx); Serial 2: 17 (Rx) และ 16(Tx); Serial 3:15 (Rx) และ 14 (Tx)** ใช้สำหรับรับ (Rx) และส่ง(Tx) TTL serial data โดย pin 0 และ 1 จะถูกเชื่อมต่อไปยัง corresponding pins ของ ATmega16U2 USB-to-TTL serial chip
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), 21 (interrupt 2).** Pins เหล่านี้สามารถที่จะกำหนดค่าที่เรียก interrupt ในค่าต่ำๆ, ขอบขาขึ้นและลง หรือเปลี่ยนแปลงค่า
- **PWM: 2 ถึง 13 และ 44 ถึง 46** ให้ output PWM output 8-bits
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS)** ใช้สำหรับรองรับการสื่อสารแบบ SPI โดยที่ไม่เกี่ยวข้องกันกับ ICSP header ซึ่งจะมีลักษณะคล้ายกับ Uno, Duemilanove และ Diecimila
- **LED 13:** เป็น build-in LED ที่เชื่อมต่อกับ digital pin 13 เมื่อ pin มีค่าเป็น HIGH LED จะติด, แต่เมื่อ pin เป็น LOW LED จะดับ
- **TWI: 20 (SDA) and 21 (SCL).** รองรับการเชื่อมต่อแบบ TWI(I2C)
- บอร์ด Mega2560 มี 16 analog inputs แต่ละ pins ให้ความละเอียด 10 bits
- **AREF.** แรงดันอ้างอิง สำหรับ analog input
- **Reset** ใช้ในการ reset ไมโครคอนโทรลเลอร์ โดยทั่วไปจะใช้โดยการเพิ่มปุ่ม reset ไว้บน shield เพื่อป้องกันปุ่มที่อยู่บนบอร์ด

2.4.3 Wemos D1 Mini

รูปที่ 2.18 Wemos Mini

WeMos D1 mini Pro V1.1.0 บอร์ดพัฒนา ESP8266 ที่มีจุดเด่นคือพื้นที่โปรแกรมขนาด 16MB ทำให้สามรถเก็บไฟล์ได้มากกว่าเดิม รองรับการเขียนโปรแกรมได้มากกว่าเดิม และในรุ่นนี้ได้เปลี่ยนชิป USB to UART เป็น CP2104 ทำให้สามารถใช้งานกับ MacOS ได้แล้ว

WeMos D1 mini Pro ได้มีการปรับปรุงฮารแวร์หลายอย่างจากในรุ่นเดิม ทั้งการเปลี่ยนอุปกรณ์ตัวต้านทาน ตัวเก็บประจุ ไดโอดมาใช้ขนาด 402 เปลี่ยนทรานซิสเตอร์มาใช้เบอร์ที่ภายในบรรจุ 2 ตัว เปลี่ยนแผ่นปริ้นไปใช้แบบ 3 เลเยอร์ และเปลี่ยนโรงงานผลิต ทำให้คุณภาพในการผลิตสูงขึ้น

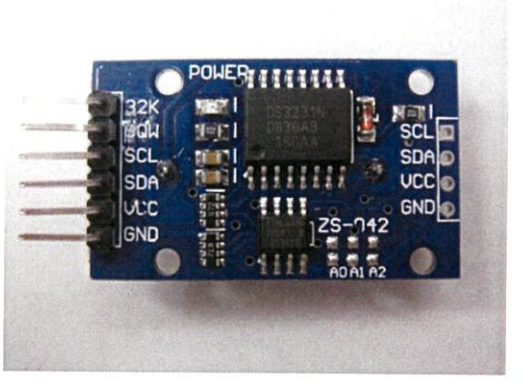
2.5 การรับค่าเวลา

2.5.1 การรับค่าเวลาจาก RTC DS3231

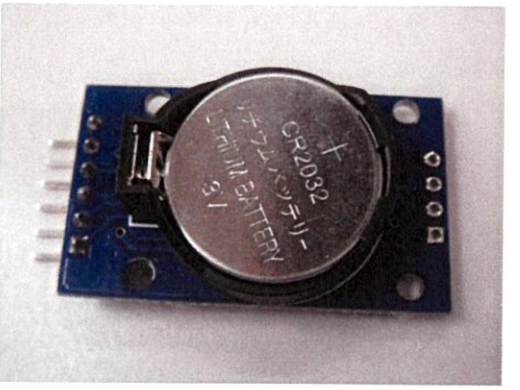
DS3231 เป็นไอซีประเภท RTC (Real-Time Clock) ของบริษัท Dallas Semiconductor / Maxim ทำหน้าที่เป็นระบบฐานเวลา (ทำหน้าที่เป็นเสมือนนาฬิกาของระบบ) เก็บข้อมูลอย่างเช่น วินาที นาที ชั่วโมง (แบบ 12 หรือ 24) วันเดือนและปีในปัจจุบัน เชื่อมต่อสื่อสารแบบบัส I2C ได้

ข้อมูลเชิงเทคนิคที่สำคัญของไอซี DS3231

- ใช้แรงดันไฟเลี้ยง (VCC) ในช่วง +2.5V .. +5.5V (+3.3V typ.)
- ใช้แบตเตอรี่สำรองได้ แรงดันในช่วง (VBAT) +2.5V .. +5.5V (+3V typ.)
- ใช้พลังงานต่ำ (Low-Power Consumption) ดังนั้นเมื่อปิดแรงดันไฟเลี้ยง VCC สามารถทำงานต่อเนื่องได้โดยใช้แรงดันไฟเลี้ยง VBAT ได้โดยอัตโนมัติ
- ใช้ตัวถังแบบ SO (Small Outline) จำนวน 16 ขา
- เชื่อมต่อแบบบัส I2C (สัญญาณ SDA และ SCL) และใช้ความเร็วได้ถึง 400kHz
- ภายในมีวงจรสร้างสัญญาณ clock (crystal oscillator) ความถี่ 32kHz
- มีความแม่นยำ (Accuracy) ±2ppm ในช่วงอุณหภูมิ 0°C..+40°C และ ±3.5ppm สำหรับ -40°C.+85°C
- สามารถตั้งค่าการแจ้งเตือนหรือ Alarm เลือกได้จาก 2 ชุด และสร้างสัญญาณอินเทอร์รัพท์ได้ (Interrupt)
- สามารถเลือกสร้างสัญญาณเอาต์พุตได้ (Programmable Square-Wave Output) ที่ขา #INT/SQW
- สามารถวัดค่าอุณหภูมิได้ ให้ข้อมูลดิจิทัลแบบ 10 บิต (2's complement) ความละเอียด 0.25°C แต่มีความแม่นยำ ±3°C



รูปที่ 2.19 แสดงอุปกรณ์โมดูล DS3231 RTC (มุมมองจากด้านล่าง)



รูปที่ 2.20 แสดงอุปกรณ์โมดูล DS3231 RTC (มุมมองจากด้านบน)

ขั้นตอนการรับค่าเวลาจาก RTC DS 3231

1) การต่อสาย

SDA = A4

SCL = A5

Vcc = 5V

GND = GND

สำหรับ RTC รุ่น DS3231 มี Library มาให้พร้อมเรียบร้อยแล้ว

2) ทำการตั้งค่าเวลาให้กับ RTC DS3231 บนโปรแกรม Arduino IDE ซึ่งการตั้งค่านี้จะตั้งค่าเพียงครั้งเดียว จากนั้นสามารถดึงเวลาจาก RTC DS3231 ได้เลย

```
#include <DS3231.h>
DS3231  rtc(SDA, SCL);
void setup()    {
        Serial.begin(9600);             // Setup Serial connection
        rtc.begin();                // Initialize the rtc object
        rtc.setDOW(MONDAY);         // Set Day-of-Week to MONDAY
        rtc.setTime(12, 0, 0);          // Set the time to 12:00:00 (24hr format)
        rtc.setDate(01, 01, 2014);              // Set the date to January 1st, 2014
}
```

```
void loop()    {
        Serial.print(rtc.getDOWStr());          // Send Day-of-Week
        Serial.print(" ");
        Serial.print(rtc.getDateStr());         // Send date
        Serial.print(" -- ");
        Serial.println(rtc.getTimeStr());       // Send time
        delay (1000);   }
```

3) การนำค่ามาใช้

ใช้คำสั่ง rtc.getTimeStr() สำหรับรับค่าเวลามาจาก RTC DS3231 และสามารถแยกเวลาออกเป็นชั่วโมง นาที วินาที ด้วยคำสั่ง

```
t = rtc.getTime();
Hor = t.hour;
Min = t.min;
Sec = t.sec;
```

โดยที่ต้องประกาศตัวแปร t ให้เป็นตัวแปรด้านเวลาด้วยคำสั่ง Time t; ก่อน

ใช้คำสั่ง rtc.getDateStr() สำหรับรับค่าวันเดือนปีจาก RTC DS3231

2.5.2 การรับค่าเวลา Universal Time ด้วย Node MCU

สำหรับนักพัฒนางานที่ต้องการเสถียรภาพของเวลา เช่น การตั้งเวลาเพื่อควบคุมอุปกรณ์ต่างๆ หรือการบันทึกข้อมูลจากฐานเวลานั้นจะใช้อุปกรณ์ที่ใช้นับเวลา โดยทั่วไปคือ โมดูลนาฬิกา (Real Time Clock Module) ซึ่งมีให้เลือกอยู่หลายชนิดด้วยกัน ตั้งแต่ราคาถูกไปจนถึงราคาแพง แต่ในปัจจุบันอุปกรณ์ Microcontroller ได้เข้าสู่ยุคของ Internet of things บทความนี้จะเป็นการนำ Node MCU หรือ ESP8266 มาตั้งเวลา โดยดึงเอาเวลาสากลจาก Server มาใช้งานแทน

ขั้นตอนการดึงเวลาสากลมาใช้

1) ติดตั้ง ESP8266 บน Arduino IDE

2) เมื่อติดตั้งเสร็จแล้ว ให้เขียน Code ตามตัวอย่าง

```
#include <ESP8266WiFi.h>
#include <time.h>
const char* ssid = "ssid";                    //ใส่ชื่อ SSID Wifi
const char* password = "password";            //ใส่รหัสผ่าน
int timezone = 7 * 3600;                      //ตั้งค่า TimeZone ตามเวลาประเทศไทย
int dst = 0;                                  //กำหนดค่า Date Swing Time
void setup()    {
        Serial.begin(115200);
        Serial.setDebugOutput(true);
        WiFi.mode(WIFI_STA);                                    //เชื่อมต่อ Wifi
        WiFi.begin(ssid, password);
        Serial.println("\nConnecting to WiFi");
        while (WiFi.status() != WL_CONNECTED) {
                Serial.print(",");
                delay(1000);    }
        configTime(timezone, dst, "pool.ntp.org", "time.nist.gov");     //ดึงเวลาจาก Server
        Serial.println("\nWaiting for time");
        while (!time(nullptr)) {
                Serial.print(".");
                delay(1000);    }
        Serial.println("");          }
void loop()     {
        configTime(timezone, dst, "pool.ntp.org", "time.nist.gov"); //ดึงเวลาปัจจุบันจาก
Server
        time_t now = time(nullptr);
        struct tm* p_tm = localtime(&now);
        delay(1000);    }
```

3) ทำการ Upload Program ให้เรียบร้อย แล้วเปิด Serial Monitor โปรแกรมก็จะแสดงวัน และเวลาปัจจุบันทาง Serial Monitor โดยที่จะได้เวลาปัจจุบันโดยไม่ต้องต่อ RTC

4) การนำค่ามาใช้ให้สังเกตที่บรรทัด struct tm* p_tm = localtime(&now) ซึ่งเป็น Struct เก็บค่าตัวแปรต่างๆไว้ ดังนี้

```
struct tm {
int tm_sec;         /* วินาที,  range 0 to 59          */
int tm_min;         /* นาที, range 0 to 59             */
int tm_hour;        /* ชั่วโมง, range 0 to 23          */
int tm_mday;        /* วันที่, range 1 to 31  */
int tm_mon;         /* เดือน, range 0 to 11            */
int tm_year;        /* ปีคริสตศักราช ตั้งแต่ 1900   */
int tm_wday;        /* วัน, range 0 to 6    */
int tm_yday;        /* วันใน 1 ปี, range 0 to 365  */
int tm_isdst;       /* daylight saving time            */
};
```

ซึ่งสามารถนำค่า Parameter ต่างๆ ไปใช้งานโดยใช้ตัวแปร p_tm->tm_hour (ตัวแปรเก็บค่าชั่วโมง)

2.6 รหัสแทนข้อมูล

รหัสหมายถึงสัญลักษณ์ที่ใช้แทนข้อมูล ขณะเมื่อต้องการสื่อสารกันระหว่างผู้รับสารกับผู้ส่งสาร ทั้งนี้เพื่อความปลอดภัยของข่าวสารที่ต้องมีการป้องกันมิให้ผู้อื่นรับรู้ข่าวสารนั้น ได้สำหรับในเครื่องคอมพิวเตอร์จะหมายถึงการเปลี่ยน ฐานของ ตัวเลขฐานสิบให้เป็นกลุ่มของเลขฐานสอง เนื่องจากคอมพิวเตอร์ต้องสื่อสารกับผู้ใช้ซึ่งเป็นมนุษย์ โดยคอมพิวเตอร์จะปฏิบัติตามคำสั่งได้ดี และรวดเร็วได้นั้น คำสั่งต้องมีลักษณะเป็นตัวเลขฐานสองคือมี 0 และ 1 เรียงต่อกันเป็นชุด ๆ ต่อ 1 คำสั่ง สำหรับมนุษย์ก็จะคุ้นเคยกับเลขฐานสิบ ถ้าต้องการสื่อสารให้เข้าใจซึ่งกันและกัน จำเป็นต้องมีการเปลี่ยนฐานของตัวเลขสลับกันไปมา ซึ่งมีวิธีการเปลี่ยนฐานตัวเลขอยู่หลายดังต่อไปนี้

2.6.1 รหัส BCD – 8421

รหัส BCD – 8421 (Binary Code Decimal) เป็นรหัสที่ใช้กันบ่อย มีตัวเลขทั้งสิ้น 10 ตัว เริ่ม ตั้งแต่ 0000 - 1001 สังเกตว่าเลขแต่ละตัวจะประกอบด้วยเลขฐาน สองจำนวน 4 ตัว เรียกว่า 4 บิต เริ่มตั้งแต่บิต 0 จนถึงบิต 3 แต่ละบิตมีลักษณะไม่เหมือนกันดังแสดงในตารางที่ 2.9

ตารางที่ 2.9 รหัส BCD – 8421 เปรียบเทียบกับเลขฐานสิบ

เลขฐานสิบ	BCD – 8421
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

2.6.2 รหัสเลขฐานแปดและเลขฐานสิบหก

รหัสนี้ถูกพัฒนามาจากรหัส BCD–8421 เพราะว่าในกรณีที่มีการนำเอาเลขมาบวกกัน เช่น 7+5 = 12 หรือ 0111+0101 =1100 ผลลัพธ์ 1100 นี้ไม่มีในรหัส BCD – 8421 จึงได้มีการคิดหารหัสเลขฐานแปดและเลขฐานสิบหกนี้มาใช้แทน ซึ่งถ้ามีการพิจารณาทีละ 3 บิตจะหมายถึงรหัสเลขฐานแปด และถ้าลองทำการพิจารณาทีละ 4 บิต จะเป็นรหัสเลขฐานสิบหกตัวอย่างเช่นถ้าหากต้องการเปลี่ยนเลข $(101001010010011010)_2$ ให้เป็นเลขฐานแปดและฐานสิบหกให้ดำเนินการดังนี้

การเข้ารหัส (1010010100100110 10)2 ให้เป็นรหัสเลขฐานแปด

101	001	010	010	011	010	
↓	↓	↓	↓	↓	↓	
5	1	2	2	3	2	$= (512232)_8$

$(101001010010011010)_2$ เมื่อเข้ารหัสแล้วจะมีค่า = $(512232)_8$

2.6.3 การใช้พาริตี้ในรหัส

ในการส่งข่าวสาร จากต้นทางไปยังปลายทางในระบบสัญญาณดิจิตอล (Digital) นั้น ผู้ให้บริการ สามารถตรวจสอบความถูกต้องของข่าวสารที่ส่งออกไปได้ ว่าผู้รับสามารถรับสารได้ถูกต้องหรือไม่ซึ่งวิธีที่นิยมใช้กันมากที่สุดก็คือการใช้พาริตี้บิต (Parity bit)

พาริตี้บิตคือบิต (เลข 0 หรือ 1) ที่เติมเข้าไปใน Code word ใด ๆ ก็ตามแล้วทำให้ Code word นั้น ๆ มีจำนวนของเลข 1 เป็นจำนวนคู่ (Even) หรือคี่ (Odd) ก็ได้ตามต้องการ การใช้พาริตี้ ในรหัสจึงแบ่งออกเป็น 2 แบบดังนี้

1) พาริตี้เลขจำนวนคู่ (EvenParity) เช่น เลขจำนวนคู่ = 0111 1

2) พาริตี้เลขจำนวนคี่ (OddParity) เช่น เลขจำนวนคี่ = 0101 1

ตารางที่ 2.10 พาริตี้บิตในรหัส BCD – 8421

เลขฐานสิบ	รหัส BCD – 8421	BCD with Odd Parity	BCD with Even Parity
0	0000	0000 1 หรือ 00001	0000 0 หรือ 00000
1	0001	0001 0 หรือ 00010	0001 1 หรือ 00011
2	0010	0010 0 หรือ 00100	0010 1 หรือ 00101
3	0011	0011 1 หรือ 00111	0011 0 หรือ 00110
4	0100	01000 หรือ 01000	01001 หรือ 01001
5	0101	0101 1 หรือ 01011	0101 0 หรือ 01001
6	0110	0110 1 หรือ 01101	0110 0 หรือ 01100
7	0111	0111 0 หรือ 01110	0111 0 หรือ 01111
8	1000	1000 0 หรือ 10000	1000 1 หรือ 10001
9	1001	1001 1 หรือ 10011	1001 0 หรือ 10010

2.6.4 รหัสเกิน 3

รหัสเกิน 3 ดัดแปลงมาจากรหัส BCD – 8421 เมื่อเปรียบเทียบรหัสเกิน 3 กับรหัส BCD – 8421 ตามตารางที่ 2.11 จะเห็นได้ว่ารหัสเกิน 3 จะมีค่ามากกว่ารหัส BCD – 8421 อยู่ 3

ตารางที่ 2.11 การเปรียบเทียบระหว่างรหัส BCD – 8421 กับรหัสเกิน 3

เลขฐานสิบ	BCD – 8421	รหัสเกิน 3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

2.6.5 รหัสเกรย์

รหัสเกรย์ (Gray Code) ใช้กันมากในระบบการตรวจจับสัญญาณด้วยแสง หรือระบบที่ทำด้วยแกนหมุนทางกลไกล เพื่อบอกตำแหน่งของเพลาหมุน รหัสแบบนี้เป็นแบบ Non Weighted ในระหว่างกลุ่มรหัส (Code Group) ที่เรียงลำดับกันไปจะมีการเปลี่ยนแปลงของรหัสครั้งละ 1 บิตเท่านั้น ทำให้โอกาสความผิดพลาดในการรับรหัสเป็นไปได้น้อยมาก การเปลี่ยนรหัสเลขฐานสองให้เป็นรหัสเกรย์ สามารถทำได้โดยนำบิตที่ 2 ดึงลงมาเป็นคำตอบต่อจากนั้นนำเอาบิตที่ 2 และบิตที่ 1 มาเปรียบเทียบกัน ถ้าบิตทั้งสอง ต่างกันผลลัพธ์ที่ได้จะเป็น “1” เสมอ แต้ถ้าเปรียบเทียบกันแล้ว บิตทั้งสองเหมือนกัน ผลลัพธ์ที่ได้จะเป็น “0” จากนั้น ทำเช่นนี้ไปเรื่อย ๆ จนถึงบิตสุดท้าย

ตารางที่ 2.12 การเปรียบเทียบระหว่างเลขฐานสองกับรหัสเกรย์ 0 – 15

เลขฐานสิบ	BCD – 8421	รหัสเกรย์
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

2.6.6 รหัสแทนข้อมูลในคอมพิวเตอร์

คอมพิวเตอร์ทำงานด้วยหลักการทางอิเล็กทรอนิกส์ ที่แทนสัญญาณทางไฟฟ้าด้วยตัวเลขศูนย์ และหนึ่ง (0 และ1)ซึ่งเป็นตัวเลขในระบบเลขฐานสองแต่ละหลักเรียกว่า บิต (Binary Digit : Bit) และ เมื่อนำตัวเลขหลายๆบิตมาเรียงกันจะหมายถึงการสร้างรหัสแทนความหมายของเลขจำนวน หรือตัวอักษร หรือสัญลักษณ์ทั้งภาษาอังกฤษและภาษาไทย และเพื่อให้การสื่อสารสามารถแลกเปลี่ยนข้อความระหว่างมนุษย์กับคอมพิวเตอร์เป็นไปในแนวทางเดียวกันจึงมีการกำหนดมาตรฐานรหัสตัวเลข ในระบบฐานสองสำหรับแทนสัญลักษณ์เหล่านี้ รหัสมาตรฐานที่นิยมใช้กันอยู่มากมีสองกลุ่ม ได้แก่ รหัสแอสกี รหัสเอ็บซีดิก

1) รหัสแอสกี

รหัสแอสกี (ASCII) เป็นมาตรฐานที่นิยมใช้กันมากในระบบคอมพิวเตอร์ส่วนใหญ่คำว่า ASSCII ย่อมาจาก American Standard Code for Information Interchange เป็นหัส 8 บิต แทนสัญลักษณ์ต่าง ๆ ได้ 256 ตัวเมื่อใช้แทนตัวอักษรภาษาอังกฤษ สำนักงานมาตรฐานผลิตภัณฑ์อุตสาหกรรม หรือ สมอ. ได้กำหนดรหัสภาษาไทยเพิ่มลงไปเพื่อให้ใช้งานร่วมกันได้

2) รหัสเอ็บซีดิก

รหัสเอ็บซีดิก (EBCDIC) ย่อมาจาก (Extended Binary Coded Decimal Interchange Code) มีการกำหนดรหัสที่ใช้แทนข้อมูลขนาด 8 บิต เหมือนกันกับรหัสแอสกี รูปแบบของรหัสจะมีความแตกต่างกัน ทั้งรหัสแอสกีและเอ็บซีดิกจะใช้เลขฐาน 2 (0 หรือ 1) จำนวน 8 หลักแทนข้อมูลหนึ่งตัวหรือ 1 Byte ดังนั้นรหัสแทนข้อมูลทั้งแอสกีและเอ็บซีดิกจะสามารถแทนข้อมูลที่แตกต่างกันได้ทั้งหมด 256 ตัว

2.6.7 เทคโนโลยีในการรักษาความปลอดภัยของข้อมูล

การรักษาความปลอดภัยของข้อมูลในการทำธุรกรรมอิเล็กทรอนิกส์ ต้องครอบคลุมในเรื่องของการระบุตัวบุคคล การควบคุมการเข้าถึงการรักษาความลับ ความถูกต้องครบถ้วนของข้อมูล และ

การป้องกันการปฏิเสธความรับผิด นั้น จำเป็นต้องอาศัยเทคโนโลยีเข้ามาช่วยในการรักษาความปลอดภัย ซึ่งเทคโนโลยีที่นิยมในปัจจุบัน ได้แก่ เทคโนโลยีการใช้รหัส และเทคโนโลยีลายมือชื่อดิจิตอล

1) เทคโนโลยีการใช้รหัส

เทคโนโลยีการใช้รหัส (Cryptography) หมายถึง การทำให้ข้อมูลที่จะนำส่งผ่านไปทางเครือข่ายอยู่ในรูปแบบที่ไม่สามารถอ่านออกได้ด้วย การเข้ารหัส ซึ่งผู้มีสิทธิ์จริงเท่านั้นจะสามารถอ่านข้อมูลได้ด้วยการถอดรหัส ซึ่งการเข้าและถอดรหัสนั้นจะอาศัยสมการทางคณิตศาสตร์ ที่ซับซ้อน และต้องอาศัยกุญแจซึ่งอยู่ในรูปของพารามิเตอร์ที่กำหนดไว้ ในการเข้าและถอดรหัสสามารถแบ่งเป็น 2 ประเภท ดังนี้

การใช้รหัสแบบกุญแจสมมาตร เป็นการเข้าและถอดรหัสโดย ใช้กุญแจส่วนตัวที่เหมือนกันซึ่งจะต้องเป็นที่รู้กันเพียงผู้ส่งและผู้รับเท่านั้น การใช้รหัสแบบกุญแจอสมมาตร เป็นการเข้าและถอดรหัสด้วยกุญแจต่างกัน โดยจะเน้นที่ผู้รับเป็นหลัก คือ จะใช้กุญแจสาธารณะของผู้รับซึ่งเป็นที่เปิดเผยในการเข้ารหัส และจะใช้กุญแจส่วนตัวของผู้รับในการถอดรหัส การใช้รหัสแบบกุญแจสมมาตร และกุญแจอสมมาตร มีข้อดี/ข้อเสียที่แตกต่างกันดังนี้

ตารางที่ 2.13 การเปรียบเทียบข้อดี-ข้อเสียของการใช้กุญแจสมมาตร

ข้อดี	ข้อเสีย
- มีความรวดเร็วเพราะใช้การคำนวณที่ น้อยกว่า - สามารถสร้างได้ง่ายโดยใช้ฮาร์ดแวร์ - การบริหารจัดการกุญแจทำได้ง่ายกว่า เพราะ ใช้กุญแจในการเข้ารหัส และถอดรหัสต่างกัน - สามารถระบุผู้ใช้โดยการใช้ร่วมกับลายมือชื่อ อย่างมากอิเล็กทรอนิกส์	- การบริหารจัดการกุญแจทำได้ยาก เพราะกุญแจในการเข้ารหัสและถอดรหัสเหมือนกัน - ใช้เวลาในการเข้าและถอดรหัสค่อนข้างนาน เพราะต้องใช้การคำนวณ

ในการส่งข้อมูลผ่านเครือข่ายนั้น นอกจากจะทำให้ข้อมูลที่ส่งมาเป็นความลับสำหรับผู้ไม่มีสิทธิ์โดยการใช้เทคโนโลยีการใช้รหัสแล้ว สำหรับการทำนิติกรรมสัญญาโดยทั่วไป ลายมือชื่อจะเป็นสิ่งที่ใช้ในการระบุตัวบุคคล และยังมีการแสดงถึงเจตนาในการยอมรับสาระในสัญญานั้นๆ ซึ่งสัมพันธ์กับการป้องกันการปฏิเสธความรับผิดสำหรับในธุรกรรมอิเล็กทรอนิกส์จะใช้ลายมือชื่ออิเล็กทรอนิกส์ซึ่งมีรูปแบบต่างๆ แต่ที่ได้รับการยอมรับกันมากที่สุด คือ ลายมือชื่อดิจิตอล ซึ่งเป็นองค์ประกอบหนึ่งในโครงสร้างพื้นฐานกุญแจสาธารณะ

2) ลายมือชื่อดิจิตอล

ลายมือชื่อ ดิจิตอล เป็นลายมือชื่ออิเล็กทรอนิกส์ ที่สร้างจากเทคโนโลยีเข้ารหัสด้วยกุญแจสาธารณะ ในการลงลายมือชื่อ ดิจิตอลกำกับข้อความที่ต้องการ ส่งผ่านเครือข่าย ผู้ส่งข้อความจะใช้กุญแจส่วนตัวของตนในการลงลายมือชื่อโดยผ่านกระบวนการทางคณิตศาสตร์ ผู้รับจะสามารถตรวจสอบความถูกต้องของลายมือชื่อดังกล่าวโดยใช้กุญแจสาธารณะของผู้ส่ง ซึ่งลายมือชื่อของผู้ส่งจะถูกรับรองด้วยองค์กรออกใบรับรอง โดยแสดงอยู่ในรูปของ "ใบรับรองดิจิตอล" ประโยชน์ของลายมือชื่อดิจิตอลนั้น นอกจากจะช่วยระบุตัวผู้ส่งข้อมูลแล้ว ยังช่วยป้องกันข้อมูลให้มีความถูกต้องไม่ได้ผ่านการแก้ไข หรือหากมีการแก้ไขมาก่อนก็สามารถตรวจสอบได้ กระบวนการสร้างและลงลายมือชื่อดิจิตอล มี3 ขั้นตอนดังนี้

ขั้นตอนที่ 1 : นำข้อมูลอิเล็กทรอนิกส์ต้นฉบับที่จะส่งไปนั้นมาผ่านกระบวนการทางคณิตศาสตร์ที่ เรียกว่า ฟังก์ชันย่อยข้อมูล เพื่อให้ได้ข้อมูลที่สั้น ๆ ที่เรียกว่า ข้อมูลที่ย่อยแล้ว ก่อนที่จะทำการเข้ารหัส

ขั้นตอนที่ 2 : ทำการเข้ารหัสด้วยกุญแจส่วนตัวของผู้ส่งเอง ซึ่งเปรียบเสมือนการลงลายมือชื่อของผู้ส่ง เพราะผู้ส่งเท่านั้นที่มีกุญแจส่วนตัวของผู้ส่งเอง และจะได้ข้อมูลที่เข้ารหัสแล้วเรียกว่า ลายมือชื่อดิจิตอล

ขั้นตอนที่ 3 : ขั้นตอนของการส่งลายมือชื่อไปพร้อมกับข้อมูลต้นฉบับ ไปยังผู้รับ ผู้รับก็จะทำการตรวจสอบว่าข้อมูลที่ได้รับถูกแก้ไขระหว่างทางหรือไม่ โดยการนำข้อมูลต้นฉบับที่ได้รับมาอ่านกระบวนการย่อย ฟังก์ชั่นย่อยข้อมูล ก็จะได้ข้อมูลที่ย่อยแล้ว และ นำลายมือชื่อดิจิตอลมาทำการถอด รหัสด้วย กุญแจสาธารณะของผู้ส่ง ก็จะได้ข้อมูลที่ย่อยแล้วอีกอันหนึ่ง แล้วทำการเปรียบเทียบ

ข้อมูลที่ย่อยแล้วทั้ง 2 อัน ถ้าหากว่าเหมือนกัน แสดงว่า ข้อมูลที่ได้รับนั้นไม่ได้ถูกแก้ไขแต่ถ้าข้อมูลที่ย่อยแล้วแตกต่างกัน แสดงว่า ข้อมูลที่ได้รับถูกเปลี่ยนแปลงระหว่างทาง

2.6.8 ใบรับรองดิจิตอล

การเข้ารหัส และ ลายมือชื่อดิจิตอล ในการทำธุรกรรม ทำให้สามารถรักษาความลับของข้อมูล สามารถรักษาความถูกต้องของข้อมูล และสามารถระบุตัวบุคคลได้ระดับหนึ่ง เพื่อเพิ่มระดับความปลอดภัยในการระบุตัวบุคคล โดยสร้างความเชื่อถือมากขึ้นด้วย ใบรับรองดิจิตอล ซึ่งออกโดยองค์กรกลางที่เป็นที่เชื่อถือ เรียกว่า องค์กรรับรองความถูกต้องจะถูกนำมาใช้สำหรับยืนยันในการทำธุรกรรมว่า เป็นบุคคลนั้นๆจริงตามที่ได้อ้างไว้ ใบรับรองดิจิตอลที่ออกตามมาตรฐาน X.509 Version 3 ซึ่งเป็นมาตรฐานที่ได้รับความนิยมอย่าง แพร่หลายที่สุด จะประกอบด้วยข้อมูลดังต่อไปนี้

1) หมายเลขของใบรับรอง
2) วิธีการที่ใช้ในการเข้ารหัสข้อมูล
3) หน่วยงานที่ออกใบรับรอง
4) เวลาเริ่มใช้ใบรับรอง
5) เวลาที่ใบรับรองหมดอายุ
6) ผู้ได้รับการรับรอง
7) กุญแจสาธารณะของผู้ได้รับการรับรอง
8) ลายมือชื่อดิจิตอลของหน่วยงานที่ออกใบรับรอง

องค์กรออกใบรับรอง เป็นองค์กรที่น่าเชื่อถือ ที่ทำหน้าที่เป็นบุคคลที่ดำเนินการออกใบรับรองดิจิตอล ให้กับผู้ทำธุรกรรมอิเล็กทรอนิกส์ ที่ขอใช้บริการโดยบริการต่างๆขององค์กรออกใบรับรองนี้ได้แก่

- บริการเทคโนโลยีเข้ารหัส ซึ่งประกอบด้วยการผลิตกุญแจส่วนตัว การส่งมอบกุญแจส่วนตัว การผลิตกุญแจสาธารณะและกุญแจส่วนตัว การผลิตลายมือชื่อดิจิตอล และการรับรองลายมือชื่อดิจิตอล

- บริการที่เกี่ยวข้องกับการออกใบรับรอง มีหลายบริการ ซึ่งประกอบไปด้วย การออกใบรับรอง การตีพิมพ์ใบรับรองเพื่อเผยแพร่แก่บุคคลทั่วไป การเก็บต้นฉบับใบรับรอง และการกำหนดนโยบายการออกและอนุมัติใบรับรอง
- บริการเสริมต่าง ๆ ได้แก่ การลงทะเบียนการตรวจสอบสัญญาต่าง ๆ การกู้กุญแจ เป็นต้น

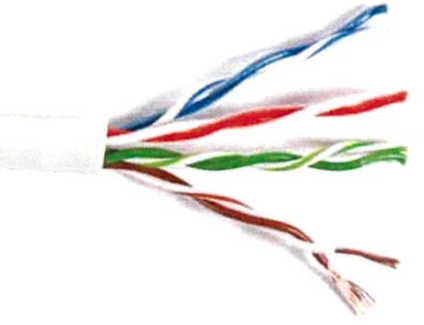
2.7 การรับส่งข้อมูล

2.7.1 การรับส่งข้อมูลแบบใช้สาย

เทคโนโลยีการรับส่งข้อมูลแบบใช้สาย แบ่งออกเป็น 3 ชนิด ดังนี้

2.7.1.1 สายตีเกลียวคู่

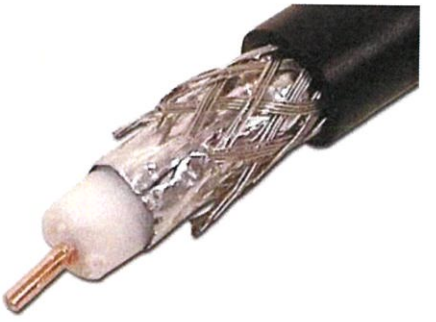
ประกอบด้วยเส้นทองแดง 2 เส้นที่หุ้มด้วยฉนวนพลาสติก พันบิดกันเป็นเกลียว เพื่อลดการบกวนจากคลื่นแม่เหล็กไฟฟ้า จากสายข้างเคียงภายในเคเบิลเดียวกัน หรือจากภายนอก เนื่องจากสายตีเกลียวคู่นี้ยอมให้สัญญาณไฟฟ้าความถี่สูงผ่านได้ สำหรับอัตราการส่งข้อมูลผ่านสายตีเกลียวคู่จะขึ้นอยู่กับความหนาของสาย คือ สายทองแดงที่มีเส้นผ่านศูนย์กลางกว้าง จะสามารถส่งสัญญาณไฟฟ้ากำลังแรงได้ ทำให้ส่งข้อมูลได้อัตราเร็วสูง โดยทั่วไปใช้สำหรับการส่งข้อมูลแบบดิจิทัล สามารถส่งได้ถึง 100 เมกะบิตต่อวินาที ในระยะทางไม่เกินร้อยเมตร



รูปที่ 2.21 ตัวอย่างสายตีเกลียวคู่

2.7.1.2. สายโคแอกซ์

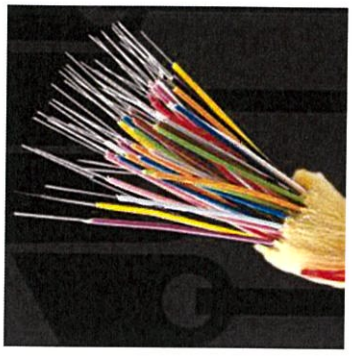
มีลักษณะเช่นเดียวกับสายที่ต่อมาจากเสาอากาศประกอบด้วยลวดทองแดงที่เป็นแกนหลักหุ้มด้วยฉนวนชั้นหนึ่ง เพื่อป้องกันกระแสไฟฟ้ารั่ว จากนั้นจะหุ้มด้วยตัวนำซึ่งทำจากลวดทองแดงถักเป็นเปียเพื่อป้องกันการรบกวนของคลื่นแม่เหล็กไฟฟ้า และสัญญาณรบกวนอื่นๆ ก่อนจะหุ้มด้วยฉนวนพลาสติกสัญญาณไฟฟ้าสามารถผ่านได้สูงมาก นิยมใช้เป็นช่องสื่อสารสัญญาณเชื่อมโยงผ่านใต้ทะเลและใต้ดิน



รูปที่ 2.22 ตัวอย่างสายโคแอกซ์

2.7.1.3 สายใยแก้วนำแสง หรือเส้นใยแก้วนำแสง

แกนกลางของสายประกอบด้วยเส้นใยแก้วหรือเส้นพลาสติกขนาดเล็กภายในกลวงหลายๆ เส้น อยู่รวมกัน เส้นใยแต่ละเส้นมีขนาดเล็กประมาณเส้นผมของมนุษย์ เส้นใยแต่ละเส้นห่อหุ้มด้วยเส้นใยอีกชนิดหนึ่งก่อนจะหุ้มชั้นนอกสุดด้วยฉนวน การส่งข้อมูลผ่านทางสื่อกลางชนิดนี้จะแตกต่างจากชนิดอื่นๆ ซึ่งจะใช้เลเซอร์วิ่งผ่านกลวงของเส้นใยแต่ละเส้น และอาศัยหลักการหักเหของแสง โดยใช้เส้นใยชั้นนอกเป็นกระจกสะท้อนแสง สามารถส่งข้อมูลด้วยอัตราความหนาแน่นของสัญญาณข้อมูลที่สูงมาก และไม่มีการก่อกวนของคลื่นแม่เหล็กไฟฟ้า



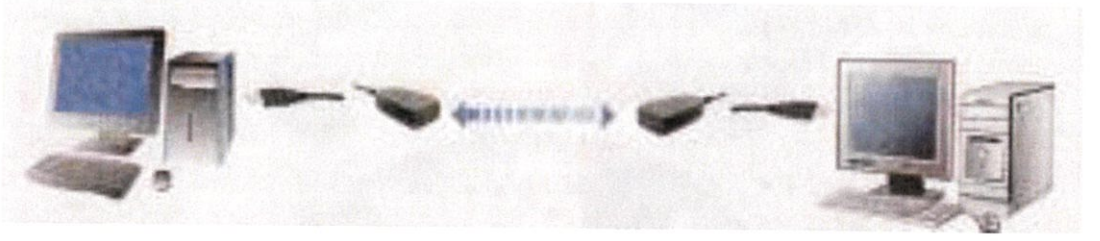
รูปที่ 2.23 ตัวอย่างสายใยแก้วนำแสง

2.7.2 การรับส่งข้อมูลแบบไร้สาย

เทคโนโลยีการรับส่งข้อมูลแบบไร้สาย แบ่งออกเป็น 4 ชนิด ได้แก่

2.7.2.1 อินฟราเรด (Infrared)

เป็นลักษณะของคลื่นที่ใช้ในการส่งข้อมูลระยะใกล้ๆ ในช่วงความถี่ที่แคบมาก ใช้ช่องทางสื่อสารน้อย มักใช้กับการสื่อสารข้อมูลที่ไม่มีสิ่งกีดขวางระหว่างตัวส่งกับตัวรับสัญญาณ โดยต้องใช้วิธีการสื่อสารตามแนวเส้นตรง ระยะทางไม่เกิน 1 – 2 เมตร ความเร็วประมาณ 4 -5 เมกกะบิตต่อนาที เช่น การส่งสัญญาณจากรีโมตคอนโทรลไปยังโทรทัศน์ การเชื่อมต่อคอมพิวเตอร์สองเครื่องโดยผ่านพอร์ตไออาร์ดีเอ เป็นต้น



รูปที่ 2.24 การเชื่อมต่อคอมพิวเตอร์สองเครื่องโดยผ่านพอร์ตไออาร์ดีเอ

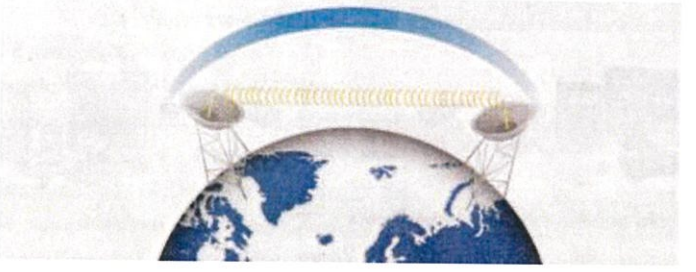
2.7.2.2 คลื่นวิทยุ (radio frequency)

ใช้ส่งสัญญาณไปในอากาศ โดยมีตัวกระจายสัญญาณส่งไปยังตัวรับสัญญาณ และใช้คลื่นวิทยุในช่วงความถี่ต่างๆ กัน มีความเร็วต่ำประมาณ 2 เมกกะบิตต่อนาที เช่น การสื่อสารในระบบวิทยุเอฟเอ็ม เอเอ็ม การสื่อสารโดยใช้ระบบไร้สาย และ บลูทูธ

รูปที่ 2.25 การส่งสัญญาณผ่านคลื่นวิทยุไปในอากาศ

2.7.2.3 ไมโครเวฟ (microwave)

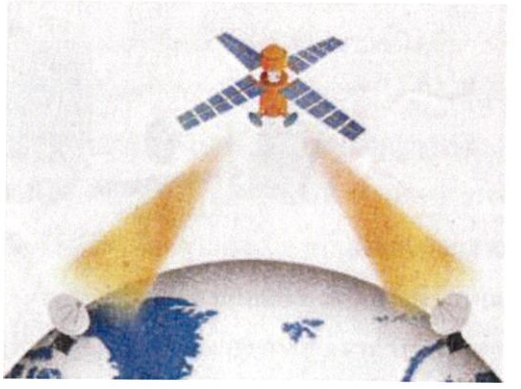
จะใช้การส่งสัญญาณคลื่นแม่เหล็กไฟฟ้าไปในอากาศพร้อมกับข้อมูลที่ต้องการส่ง และต้องมีสถานีที่ทำหน้าที่ส่งและรับข้อมูล และเนื่องจากสัญญาณไมโครเวฟจะเดินทางเป็นเส้นตรงไม่สามารถเลี้ยวหรือโค้งตามขอบโลกได้ จึงต้องมีการตั้งสถานีรับ – ส่งข้อมูลเป็นระยะๆ และส่งข้อมูลต่อกันเป็นทอดๆ ระหว่างสถานีต่อสถานี จนกว่าจะถึงสถานีปลายทาง และแต่ละสถานีจะตั้งอยู่ในที่สูง เช่น ดาดฟ้าของตึกสูง ยอดเขา เป็นต้น เพื่อหลีกเลี่ยงการชนสิ่งกีดขวางในแนวการเดินทางของสัญญาณ เหมาะกับการส่งข้อมูลในพื้นที่ห่างไกล และทุรกันดาร



รูปที่ 2.26 การส่งข้อมูลแบบไร้สายโดยใช้คลื่นไมโครเวฟ

2.7.2.4 ดาวเทียม (satellite)

เป็นสถานีรับส่งสัญญาณไมโครเวฟบนดาดฟ้า ซึ่งได้รับการพัฒนาขึ้นมาเพื่อหลีกเลียงข้อจำกัดของสถานีรับ – ส่งไมโครเวฟบนผิวโลก เพื่อใช้เป็นสถานีรับ – ส่ง สัญญาณไมโครเวฟบนอวกาศ และทวนสัญญาณในแนวโคจรของโลก ซึ่งจะต้องมีสถานีภาคพื้นดิน ทำหน้าที่รับและส่งสัญญาณขึ้นไปบนดาวเทียมที่โคจรอยู่สูงจากพื้นโลกประมาณ 35,600 ไมล์ โดยดาวเทียมเหล่านั้นจะเคลื่อนที่ด้วยความเร็วที่เท่ากับการหมุนของโลก จึงเสมือนกับดาวเทียมนั้นอยู่นิ่งกับที่ขณะที่โลกหมุนรอบตัวเอง ทำให้การส่งสัญญาณไมโครเวฟจากสถานีหนึ่งขึ้นไปบนดาวเทียมและการกระจายสัญญาณจากดาวเทียมลงมายังสถานีตามจุดต่างๆ บนผิวโลกเป็นไปอย่างแม่นยำ



รูปที่ 2.27 การส่งข้อมูลแบบไร้สายโดยใช้ดาวเทียมเป็นการรับส่งสัญญาณไมโครเวฟบนอวกาศ

2.8 ระบบจัดการข้อมูล

2.8.1 ฐานข้อมูล (Data base)

2.8.1.1 สาระสำคัญ

ฐานข้อมูลเป็นการจัดเก็บข้อมูลอย่างเป็นระบบ ทำให้ผู้ใช้สามารถใช้ข้อมูลที่เกี่ยวข้องในระบบงานต่าง ๆ ร่วมกันได้ โดยที่จะไม่เกิดความซ้ำซ้อนของข้อมูล และยังสามารถหลีกเลี่ยงความขัดแย้งของข้อมูลด้วย อีกทั้งข้อมูลในระบบก็จะถูกต้องเชื่อถือได้ และเป็นมาตรฐานเดียวกัน โดยจะมีการกำหนดระบบความปลอดภัยของข้อมูลขึ้น

นับได้ว่าปัจจุบันเป็นยุคของสารสนเทศ เป็นที่ยอมรับกันว่า สารสนเทศเป็นข้อมูลที่ผ่านการกลั่นกรองอย่างเหมาะสม สามารถนำมาใช้ประโยชน์อย่างมากมาย ไม่ว่าจะเป็นการนำมาใช้งานด้านธุรกิจ การบริหาร และกิจการอื่น ๆ องค์กรที่มีข้อมูลปริมาณมาก ๆ จะพบความยุ่งยากลำบากในการจัดเก็บข้อมูล ตลอดจนการนำข้อมูลที่ต้องการออกมาใช้ให้ทันต่อเหตุการณ์ ดังนั้นคอมพิวเตอร์จึงถูกนำมาใช้เป็นเครื่องมือช่วยในการจัดเก็บข้อมูล การประมวลผลข้อมูล ซึ่งทำให้ระบบการจัดเก็บข้อมูลเป็นไปได้สะดวก ทั้งนี้โปรแกรมแต่ละโปรแกรม จะต้องสร้างวิธีควบคุมและจัดการกับข้อมูลขึ้นเอง ฐานข้อมูลจึงเข้ามามีบทบาทสำคัญอย่างมาก โดยเฉพาะระบบงานต่าง ๆ ที่ใช้คอมพิวเตอร์ การออกแบบและพัฒนาระบบฐานข้อมูล จึงต้องคำนึงถึงการควบคุมและการจัดการความถูกต้องตลอดจนประสิทธิภาพในการ เรียกใช้ข้อมูลด้วย

2.8.1.2 ความรู้พื้นฐานเกี่ยวกับระบบฐานข้อมูล

ระบบฐานข้อมูล (Database System) หมายถึง โครงสร้างสารสนเทศที่ประกอบด้วยรายละเอียดของข้อมูลที่เกี่ยวข้องกันที่จะนำมาใช้ในระบบต่าง ๆ ร่วมกัน

ระบบฐานข้อมูล จึงนับว่าเป็นการจัดเก็บข้อมูลอย่างเป็นระบบ ซึ่งผู้ใช้สามารถจัดการกับข้อมูลได้ในลักษณะต่าง ๆ ทั้งการเพิ่ม การแก้ไข การลบ ตลอดจนการเรียกดูข้อมูล ซึ่งส่วนใหญ่จะเป็นการประยุกต์นำเอาระบบคอมพิวเตอร์เข้ามาช่วยในการจัดการ ฐานข้อมูล

2.8.1.3 นิยามและคำศัพท์พื้นฐานเกี่ยวกับระบบฐานข้อมูล

บิท (Bit) หมายถึง หน่วยของข้อมูลที่มีขนาดเล็กที่สุด

ไบท์ (Byte) หมายถึง หน่วยของข้อมูลที่เกิดจากการนำบิทมารวมกันเป็นตัวอักขระ (Character)

เขตข้อมูล (Field) หมายถึง หน่วยของข้อมูลที่ประกอบขึ้นจากตัวอักขระตั้งแต่หนึ่งตัวขึ้นไปมารวมกันแล้ว ได้ความหมายของสิ่งใดสิ่งหนึ่ง เช่น ชื่อ ที่อยู่ เป็นต้น

ระเบียน (Record) หมายถึง หน่วยของข้อมูลที่เกิดจากการนเอาเขตข้อมูลหลาย ๆ เขตข้อมูลมารวมกัน เพื่อเกิดเป็นข้อมูลเรื่องใดเรื่องหนึ่ง เช่น ข้อมูลของนักศึกษา 1 ระเบียน (1 คน) จะประกอบด้วย รหัสประจำตัวนักศึกษา 1 เขตข้อมูล, ชื่อนักศึกษา 1 เขตข้อมูล, ที่อยู่ 1 เขตข้อมูล

แฟ้มข้อมูล (File) หมายถึงหน่วยของข้อมูลที่เกิดจากการนำข้อมูลหลาย ๆ ระเบียนที่เป็นเรื่องเดียวกันมารวมกัน เช่น แฟ้มข้อมูลนักศึกษา แฟ้มข้อมูลลูกค้า แฟ้มข้อมูลพนักงาน

ส่วนในระบบฐานข้อมูล มีคำศัพท์ต่าง ๆ ที่เกี่ยวข้องดังนี้

เอนทิตี้ (Entity) หมายถึง ชื่อของสิ่งใดสิ่งหนึ่ง ได้แก่ คน สถานที่ สิ่งของ การกระทำ ซึ่งต้องการจัดก็บข้อมูลไว้ เช่น เอนทิตี้ลูกค้า เอนทิตี้พนักงาน

1) เอนทิตี้ชนิดอ่อนแอ (Weak Entity) เป็นเอนทิตี้ที่ไม่มีความหมาย หากขาดเอนทิตี้อื่นในฐานข้อมูล

2) แอททริบิวต์(Attribute) หมายถึง รายละเอียดข้อมูลที่แสดงลักษณะและคุณสมบัติของเอนทิตี้หนึ่ง ๆ เช่น เอนทิตี้นักศึกษา ประกอบด้วย - แอทริบิวต์รหัสนักศึกษา,- แอททริบิวต์ชื่อนักศึกษา แอททริบิวต์ที่อยู่นักศึกษา

3) ความสัมพันธ์ (Relationships) หมายถึง ความสัมพันธ์ระหว่างเอนทิตี้ เช่น ความสัมพันธ์ระหว่างเอนทิตี้นักศึกษาและเอนทิตี้คณะวิชาเป็นลักษณะว่า นักศึกษาแต่ละคนเรียนอยู่คณะวิชาใดคณะวิชาหนึ่ง

ความสัมพันธ์ระหว่างเอนทิตี้ แบ่งออกเป็น 3 ประเภท คือ

a) ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One-to-one Relationships) เป็นความสัมพันธ์ของข้อมูลในเอนทิตี้หนึ่งที่มีความสัมพันธ์กับข้อมูลในอีกเอนทิตี้หนึ่ง ในลักษณะหนึ่งต่อหนึ่ง (1 : 1)

b) ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One-to-many Relationships) เป็นความสัมพันธ์ของข้อมูลในเอนทิตี้หนึ่ง ที่มีความสัมพันธ์กับข้อมูลหลาย ๆ ข้อมูลในอีกเอนทิตี้หนึ่ง ในลักษณะ (1:m)

c) ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many-to-many Relationships) เป็นความสัมพันธ์ของข้อมูลสองเอนทิตี้ในลักษณะกลุ่มต่อกลุ่ม (m:n)

เอนทิตี้ใบสั่งซื้อแต่ละใบจะสามารถสั่งสินค้าได้มากกว่าหนึ่งชนิด ความสัมพันธ์ของข้อมูลจากเอนทิตี้ใบสั่งซื้อไปยังเอนทิตี้สินค้า จึงเป็นแบบหนึ่งต่อกลุ่ม (1:m) ในขณะที่สินค้าแต่ละชนิด

จะถูกสั่งอยู่ในใบสั่งซื้อหลายใบ ความสัมพันธ์ของข้อมูลจากเอนทิตี้สินค้าไปยังอินทิตี้ใบสั่งซื้อ จึงเป็นแบบหนึ่งต่อกลุ่ม (1:n) ดังนั้นความสัมพันธ์ของเอนทิตี้ทั้งสอง จึงเป็นแบบกลุ่มต่อกลุ่ม (m:n)

จากคำศัพท์ต่าง ๆ ที่เกี่ยวข้องกับระบบฐานข้อมูลที่ได้กล่าวมาแล้วข้างต้น จึงอาจให้นิยามของฐานข้อมูลในอีกลักษณะได้ว่า "ฐานข้อมูล" อาจหมายถึง โครงสร้างสารสนเทศ ที่ประกอบด้วยหลายๆ เอนทิตี้ที่มีความสัมพันธ์กัน

2.8.1.4 ความสำคัญของการประมวลผลแบบระบบฐานข้อมูล

จากการจัดเก็บข้อมูลรวมเป็นฐานข้อมูลจะก่อให้เกิดประโยชน์ดังนี้

1) สามารถลดความซ้ำซ้อนของข้อมูลได้ ซึ่งการเก็บข้อมูลชนิดเดียวกันไว้หลาย ๆ ที่ ทำให้เกิดความซ้ำซ้อน (Redundancy) ดังนั้นการนำข้อมูลมารวมเก็บไว้ในฐานข้อมูล จะช่วยลดปัญหาการเกิดความซ้ำซ้อนของข้อมูลได้ โดยระบบจัดการฐานข้อมูล (Database Management System : DBMS) จะช่วยควบคุมความซ้ำซ้อนได้ เนื่องจากระบบจัดการฐานข้อมูลจะทราบได้ตลอดเวลาว่ามีข้อมูลซ้ำซ้อนกันอยู่ ที่ใดบ้าง

2) หลีกเลี่ยงความขัดแย้งของข้อมูลได้ หาก มีการเก็บข้อมูลชนิดเดียวกันไว้หลาย ๆ ที่และมีการปรับปรุงข้อมูลเดียวกันนี้ แต่ปรับปรุงไม่ครบทุกที่ที่มีข้อมูลเก็บอยู่ก็จะทำให้เกิดปัญหาข้อมูลชนิดเดียวกัน อาจมีค่าไม่เหมือนกันในแต่ละที่ที่เก็บข้อมูลอยู่ จึงก่อให้เกิดความขัดแย้งของข้อมูลขึ้น (Inconsistency)

3) สามารถใช้ข้อมูลร่วมกันได้ ฐานข้อมูล จะเป็นการจัดเก็บข้อมูลรวมไว้ด้วยกัน ดังนั้นหากผู้ใช้ต้องการใช้ข้อมูลในฐานข้อมูลที่มาจากแฟ้มข้อมูลต่างๆ ก็จะทำได้โดยง่าย

4) สามารถรักษาความถูกต้องเชื่อถือได้ของข้อมูล บางครั้งพบว่าการจัดเก็บข้อมูลในฐานข้อมูลอาจมีข้อผิดพลาดเกิดขึ้น เช่น จากการที่ผู้ป้อนข้อมูลป้อนข้อมูลผิดพลาดคือป้อนจากตัวเลขหนึ่งไปเป็นอีกตัวเลขหนึ่ง โดยเฉพาะกรณีมีผู้ใช้หลายคนต้องใช้ข้อมูลจากฐานข้อมูลร่วมกัน หากผู้ใช้คนใดคนหนึ่งแก้ไขข้อมูลผิดพลาดก็ทำให้ผู้อื่นได้รับผลกระทบตามไป ด้วย ในระบบจัดการฐานข้อมูล (DBMS) จะสามารถใส่กฎเกณฑ์เพื่อควบคุมความผิดพลาดที่เกิดขึ้น

5) สามารถกำหนดความเป็นมาตรฐานเดียวกันของข้อมูลได้ การเก็บข้อมูลร่วมกัน ไว้ในฐานข้อมูล จะทำให้สามารถกำหนดมาตรฐานของข้อมูลได้รวมทั้ง มาตรฐานต่าง ๆ ในการจัดเก็บข้อมูลให้เป็นไปในลักษณะเดียวกันได้ เช่นการกำหนดรูปแบบการเขียนวันที่ ในลักษณะ วัน/เดือน/ปี หรือ ปี/เดือน/วัน ทั้งนี้จะมีผู้ที่คอยบริหารฐานข้อมูลที่เราเรียกว่า ผู้บริหารฐานข้อมูล (Database Administrator : DBA) เป็นผู้กำหนดมาตรฐานต่างๆสามารถกำหนดระบบความปลอดภัยของข้อมูลได้ ระบบความ ปลอดภัยในที่นี้ เป็นการป้องกันไม่ให้ผู้ใช้ที่ไม่มีสิทธิมาใช้ หรือมาเห็นข้อมูลบางอย่างในระบบ ผู้บริหารฐานข้อมูลจะสามารถกำหนดระดับการเรียกใช้ข้อมูลของผู้ใช้แต่ละคนได้ ตามความเหมาะสม

6) เกิดความเป็นอิสระของข้อมูล ในระบบฐาน ข้อมูลจะมีตัวจัดการฐานข้อมูลที่ทำหน้าที่เป็นตัวเชื่อมโยงกับฐานข้อมูล โปรแกรมต่าง ๆ อาจไม่จำเป็นต้องมีโครงสร้างข้อมูลทุกครั้ง ดังนั้นการแก้ไขข้อมูลบางครั้ง จึงอาจกระทำเฉพาะกับโปรแกรมที่เรียกใช้ข้อมูลที่เปลี่ยนแปลงเท่านั้น ส่วนโปรแกรมที่ไม่ได้เรียกใช้ข้อมูลดังกล่าว ก็จะเป็นอิสระจากการเปลี่ยนแปลง

2.8.1.5 รูปแบบของระบบฐานข้อมูล

รูปแบบของระบบฐานข้อมูล มีอยู่ด้วยกัน 3 ประเภท คือ

1) ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) เป็นการเก็บข้อมูลในรูปแบบที่เป็นตาราง (Table) หรือเรียกว่า รีเลชั่น (Relation) มีลักษณะเป็น 2 มิติ คือเป็นแถว (row) และเป็นคอลัมน์ (column) การเชื่อมโยงข้อมูลระหว่างตาราง จะเชื่อมโยงโดยใช้แอททริบิวต์ (attribute) หรือคอลัมน์ที่เหมือนกันทั้งสองตารางเป็นตัวเชื่อมโยงข้อมูล ฐานข้อมูลเชิงสัมพันธ์นี้จะเป็นรูปแบบของฐานข้อมูลที่นิยมใช้ในปัจจุบัน

2) ฐานข้อมูลแบบเครือข่าย (Network Database) ฐานข้อมูลแบบเครือข่ายจะเป็นการรวมระเบียนต่าง ๆ และความสัมพันธ์ระหว่างระเบียนแต่จะต่างกับฐานข้อมูลเชิงสัมพันธ์ คือ ในฐานข้อมูลเชิงสัมพันธ์จะแฝงความสัมพันธ์เอาไว้ โดยระเบียนที่มีความสัมพันธ์กัน จะต้องมีค่าของข้อมูลใน แอททริบิวต์ไดแอ ททริบิวต์หนึ่งเหมือนกัน แต่ฐานข้อมูลแบบเครือข่ายจะแสดงความสัมพันธ์อย่างชัดเจน

3) ฐานข้อมูลแบบลำดับชั้น (Hierarchical Database) ฐานข้อมูลแบบลำดับชั้น เป็นโครงโครงสร้างที่จัดเก็บข้อมูลในลักษณะความสัมพันธ์แบบพ่อ-ลูก (Parent-Child Relationship Type : PCR Type) หรือเป็นโครงสร้างรูปแบบต้นไม้ (Tree) ข้อมูลที่จัดเก็บในที่นี้ คือ ระเบียน (Record) ซึ่งประกอบด้วยค่าของเขตข้อมูล (Field) ของเอนทิตี้หนึ่ง ๆ

2.8.1.6 โปรแกรมฐานข้อมูลที่นิยมใช้

โปรแกรมฐานข้อมูลเป็นโปรแกรมหรือซอฟแวร์ที่ช่วยจัดการข้อมูลหรือรายการต่าง ๆ ที่อยู่ในฐานข้อมูล ไม่ว่าจะเป็นการจัดเก็บ การเรียกใช้ การปรับปรุงข้อมูล

โปรแกรมฐานข้อมูล จะช่วยให้ผู้ใช้สามารถค้นหาข้อมูลได้อย่างรวดเร็ว ซึ่งโปรแกรมฐานข้อมูลที่นิยมใช้มีอยู่ด้วยกันหลายตัว เช่น Access, FoxPro, Clipper, dBase, FoxBase, Oracle, SQL เป็นต้น โดยแต่ละโปรแกรมจะมีความสามารถต่างกัน บางโปรแกรมใช้ง่ายแต่จะจำกัดขอบเขตการใช้งาน บ่งโปรแกรมใช้งานยากกว่า แต่จะมีความสามารถในการทำงานมากกว่า

1) โปรแกรม Access นับเป็นโปรแกรมที่นิยมใช้กันมากในขณะนี้ โดยเฉพาะในระบบฐานข้อมูลขนาดใหญ่ สามารถสร้างแบบฟอร์มที่ต้องการจะเรียกดูข้อมูลในฐานข้อมูล หลังจากบันทึกข้อมูลในฐานข้อมูลเรียบร้อยแล้ว จะสามารถค้นหาหรือเรียกดูข้อมูลจากเขตข้อมูลใดก็ได้ นอกจากนี้ Access ยังมีระบบรักษาความปลอดภัยของข้อมูล โดยการกำหนดรหัสผ่านเพื่อป้องกันความปลอดภัยของข้อมูลในระบบได้ด้วย
2) โปรแกรม FoxPro เป็น โปรแกรมฐานข้อมูลที่มีผู้ใช้งานมากที่สุด เนื่องจากใช้ง่ายทั้งวิธีการเรียกจากเมนูของ FoxPro และประยุกต์โปรแกรมขึ้นใช้งาน โปรแกรมที่เขียนด้วย FoxPro จะสามารถใช้กลับ dBase คำสั่งและฟังก์ชั่นต่าง ๆ ใน dBase จะสามารถใช้งานบน FoxPro ได้ นอกจากนี้ใน FoxPro ยังมีเครื่องมือช่วยในการเขียนโปรแกรม เช่น การสร้างรายงาน
3) โปรแกรม dBase เป็นโปรแกรมฐานข้อมูลชนิดหนึ่ง การใช้งานจะคล้ายกับโปรแกรม FoxPro ข้อมูลรายงานที่อยู่ในไฟล์บน dBase จะสามารถส่งไปประมวลผลในโปรแกรม Word Processor ได้ และแม้แต่ Excel ก็สามารถอ่านไฟล์ .DBF ที่สร้างขึ้นโดยโปรแกรม dBase ได้ด้วย

4) โปรแกรม SQL เป็น โปรแกรมฐานข้อมูลที่มีโครงสร้างของภาษาที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพการทำงานสูง สามารถทำงานที่ซับซ้อนได้โดยใช้คำสั่งเพียงไม่กี่คำสั่ง โปรแกรม SQL จึงเหมาะที่จะใช้กับระบบฐานข้อมูลเชิงสัมพันธ์ และเป็นภาษาหนึ่งที่มีผู้นิยมใช้กันมาก โดยทั่วไปโปรแกรมฐานข้อมูลของบริษัทต่าง ๆ ที่มีใช้อยู่ในปัจจุบัน เช่น Oracle, DB2 ก็มักจะมีคำสั่ง SQL ที่ต่างจากมาตรฐานไปบ้างเพื่อให้เป็นจุดเด่นของแต่ละโปรแกรมไป

2.8.2 Server

2.8.2.1 Server คืออะไร

การเชื่อมต่ออินเตอร์เน็ตแต่ละครั้งจะต้องมีการเชื่อมต่อไปยัง Server ซึ่ง Server เป็นคอมพิวเตอร์ที่มีความสามารถสูง และมีโปรแกรมในที่คอยให้บริการกับลูกข่ายที่เข้ามาเชื่อมต่อกับ Server

รูปที่ 2.28 Server

2.8.2.2 ประเภทของเครื่อง Server

โดยประเภทของเครื่อง Server จะแบ่งเป็น 2 แบบคือ

1) แบบ Rack จะมีลักษณะเป็นแท่งสีเหลี่ยมยาว ๆ ถ้าใช้ Server แบบ Rack ค่าบริการที่จะนำServer ไปวางไว้ที่ Data Center จะถูกกว่าแบบ Tower
2) แบบ Tower จะเหมือนกับ PC ทั่ว ๆ ไปที่ใช้กันในบ้าน และค่าบริการการวางที่ Data Center จะแพงกว่าแบบ Rack เกือบเท่าตัว

2.8.2.3 ระบบปฏิบัติการที่ใช้ในเครื่อง Server

ระบบปฏิบัติการที่ใช้ในเครื่อง Server จะเป็น 3 ระบบปฏิบัติการนี้คือ

1) Linux สำหรับ Linux Distribution ที่ได้รับความนิยมได้แก่ Debian, Ubuntu, Redhat และ Fedora เป็นต้น Linux เป็นระบบปฏิบัติการที่ใช้งานโดยไม่เสียค่าใช้จ่าย พร้อมทั้งมีนักพัฒนาอยู่ทั่วโลกร่วมกันพัฒนาด้วย

2) Windows สำหรับ Windows ที่นิยมใช้เป็น server ได้แก่ Windows Server 2003 และ Windows Server 2008 ซึ่งเป็นระบบปฏิบัติการจากไมโครซอฟท์ที่มีความเสถียรและเป็นที่ยอมรับโดยทั่วไป

3) Unix สำหรับ Unix สำหรับระบบปฏิบัติการนี้เป็นระบบปฏิบัติการณ์ที่เก่าแก่ระบบหนึ่ง ที่ยังใช้งานอยู่จนถึงทุกวันนี้ ได้แก่ BSD

2.8.2.4 หน้าที่ของ Server

Server ทำหน้าที่เป็นเหมือนผู้ให้บริการต่าง ๆ ในโครงข่ายอินเตอร์เน็ต หรือโครงข่ายที่มีลูกข่าย เมื่อมีผู้ใช้งานมาขอใช้บริการ Server เครื่อง Server จะจัดสรรทรัพยากรที่มีอยู่ในเครื่องเพื่อให้บริการในทันที

ซึ่งบริการของ Server นั้นมีหลากหลายอย่างด้วยกัน โดยสามารถแบ่งได้เป็น 4 หน้าที่หลัก ๆดังต่อไปนี้

1) Web server คือโปรแกรมที่มีหน้าที่ให้บริการด้านการจัดการเว็บไซต์ โดยส่วนมากโปรแกรมที่นิยมใช้เป็น Web server จะเป็น Apache web server

2) Mail server คือโปรแกรมที่มีหน้าที่ให้บริการด้าน E-mail โปรแกรมที่ใช้ในด้าน Mail server มีอยู่หลายโปรแกรมด้วยกันแต่ที่นิยมกันจะมีอยู่ 3 โปรแกรมคือPostfix, qmail, courier

3) DNS server คือโปรแกรมที่มีหน้าที่ให้บริการด้านโดเมนเนมที่จะค่อยเปลี่ยนชื่อเว็บไซต์ที่เราต้องการให้เป็น IP Address โปรแกรมที่นิยมใช้คือ bind9

4) Database server คือโปรแกรมที่ทำหน้าที่ให้บริการด้านการจัดการดูแลข้อมูลต่างๆ ภายในเว็บไซต์ โปรแกรมที่มีการใช้งานส่วนใหญ่จะเป็น mysql, postgresql, DB2

โดยการทำงานของ Server จะสามารถทำงานพร้อมกันหลายๆอย่างได้ในเวลาเดียวกัน เนื่องจากความสามารถของเครื่อง Server ส่วนใหญ่จะมีความสามารถที่สูง โดยการทำงานแต่ละอย่างของ Server จะทำงานใน Port ที่ต่างกันไป

2.8.2.5 ประโยชน์ Server

Server เป็นเครื่องคอมพิวเตอร์ที่มีความสามารถในการให้บริการที่สูงมาก โดยประโยชน์หลัก ๆ ของ Server นั้นเป็นเครื่องคอมพิวเตอร์ที่คอยให้บริการกับผู้ใช้งานอินเตอร์เน็ตที่เข้ามาขอใช้บริการ

นอกจากที่เครื่อง Server ยังสามารถนำมาใช้ในสำนักงานได้อีกด้วย โดยประโยชน์ในการใช้เครื่อง Server ในสำนักงาน คือ ช่วยให้ประหยัดทรัพยากรต่าง ๆได้ เพราะว่าคอมพิวเตอร์ทุกตัวสามารถใช้งานทรัพยากรนั้น ๆ ได้เช่น เครื่องพิมพ์ ฮาร์ดดิสก์ เป็นต้น

2.8.2.6 ประเภทของ Server

Server สามารถแบ่งออกได้เป็น 4 ประเภทด้วยกัน โดยแบ่งตามลักษณะการทำงาน

1) File Server มีหน้าที่ในการจัดเก็บไฟล์เหมือนกับฮาร์ดดิสก์ ซึ่งผู้ใช้งานสามารถนำไฟล์ฝากไว้ใน File Server ได้

2) Print Server มีหน้าที่ในการเชื่อมต่อเครื่องปริ้นท์ให้สามารถใช้งานกับคอมพิวเตอร์ลูกข่ายเพื่อเป็นการประหยัดทรัพยากรนั่นเอง ซึ่งส่วนมากจะมีใช้ในองค์กรขนาดใหญ่

3) Database Server มีหน้าที่ในการรันระบบที่เป็นฐานข้อมูล DBMS (Database Management System) ซึ่งเป็นโปรแกรมฐานข้อมูลและตัวจัดการฐานข้อมูล เช่น SQL, Informix

4) Application Server มีหน้าที่ในการรันโปรแกรมประยุกต์ โดยมีการทำงานที่สอดคล้องกับผู้ใช้งาน

Server เป็นอุปกรณ์ที่มีส่วนสำคัญมากในระบบอินเตอร์เน็ตและในระบบเครือข่าย ซึ่งความสามารถของ Server นั้นเราสามารถประยุกต์ใช้ได้ตามหน้าที่และลักษณะงานให้เข้ากับ Server ประเภทต่าง ๆเพื่อประสิทธิภาพในการทำงานที่ดีที่สุดนั่นเอง

2.9 ฐานข้อมูลออนไลน์ไฟร์เบส (Firebase)

รูปที่ 2.29 Firebase Realtime Database

Firebase (ไฟร์เบส) คือ บริการ backend และ แพลตฟอร์ม ครบวงจรสำหรับนักพัฒนาแอพ และโปรแกรมประยุกต์บนเว็บแพลตฟอร์มที่มีเครื่องมือและโครงสร้างพื้นฐานที่ได้รับการออกแบบมาเพื่อช่วยให้นักพัฒนาสามารถสร้างแอพพลิเคที่มีคุณภาพสูง ไฟร์เบส ถูกสร้างขึ้นจากคุณสมบัติเสริมว่านักพัฒนาสามารถผสมและจับคู่เพื่อให้พอดีกับความต้องการของตน บริษัท ก่อตั้งขึ้นในปี 2011 โดยแอนดรูลีและเจมส์ เทมปลิน สินค้าเริ่มต้นไฟร์เบสเป็นฐานข้อมูลเรียลไทม์ซึ่งมี API ที่ช่วยให้นักพัฒนาในการจัดเก็บและซิงค์ข้อมูล Google Firebase 2.0 กูลเกิลได้ซื้อกิจการไฟร์เบส และมีการพัฒนาให้สามารถ จากบริการ backend เก็บข้อมูลอย่างเดียวมาเป็นแพลตฟอร์มครบวงจรสำหรับนักพัฒนาแอพ (รองรับ iOS, Android, Web) รองรับบริการแทบทุกอย่างที่นักพัฒนาแอพต้องใช้งาน

ในงานด้านแอพพลิเคชั่น ตัว Firebase ถือเป็นบริการฐานข้อมูลออนไลน์ตัวหนึ่ง ซึ่งแอพพลิเคชั่นส่วนใหญ่ต้องใช้งานฐานข้อมูลตรงส่วนนี้ แต่หากมองในมุมของ IoT ตัว Firebase ถือว่าเป็นตัวกลางการเชื่อมต่อทุกอุปกรณ์เข้าด้วยกันได้ โดยมีจุดเด่นคือ เรียลไทม์ และสามารถบันทึกข้อมูลไว้ได้

ในด้านของ API ตัว Firebase ไม่ได้อิงการใช้งานไปกับภาษาใดภาษาหนึ่ง กรณีที่ภาษาใด ๆ ไม่มีไลบารี่ให้ใช้งาน สามารถใช้ REST API (โปรโตคอล HTTP, HTTPs) ในการร้องขอข้อมูล (GET) หรือส่งข้อมูล (PUT) เข้าไปได้เลย

Firebase คือฐานข้อมูลประเภท NoSQL ฐานข้อมูล MySQL MSSQL และฐานข้อมูลชนิด RDBMS ต่าง ๆ จะมีลักษณะเป็นตารางข้อมูล มีคอลัมน์ มีการกำหนดชนิดของข้อมูลไว้อย่างชัดเจน และใช้ภาษา SQL ในการติดต่อเพื่อขอใช้ข้อมูล (SELECT) เพิ่มข้อมูล (INSERT) และลบข้อมูล (DELETE) สามารถกรองเอาเฉพาะข้อมูลที่ต้องการได้ด้วยการใช้ WHERE และบางครั้งมีปัญหาเรื่องช่องโหว่ (SQL Injection ถือเป็นวิธีพื้นฐานที่นิยมใช้และได้ผลมากที่สุดในขณะนี้)

ฐานข้อมูลชนิด NoSQL จะไม่ใช้ภาษา SQL ในการจัดการข้อมูล และออกแบบให้มีความยืดหยุ่นและเน้นความเร็วในการใช้งานมากที่สุด ฐานข้อมูล NoSQL ที่นิยมใช้งานในปัจจุบันคือ MongoDB ซึ่งมีการเก็บข้อมูลเป็นชนิด JSON (เจสัน) มีตารางเหมือนเดิม แต่ไม่มีคอลัมน์ข้อมูลที่ตายตัว ใน 1 แถวสามารถเก็บข้อมูลได้ทั้งข้อความ (String) ตัวเลข (Number) และอื่น ๆ รวมไปถึงอาเรย์ และออปเจ็ค

Firebase มีการทำงานคล้าย ๆ กับ MongoDB คือมีฐานข้อมูล แต่ไม่มีตาราง มีการเก็บข้อมูลในรูป JSON สามารถเพิ่มข้อมูลไปในออปเจ็คใด ๆ ก็ได้ แต่เก็บเป็นอาร์เรย์ไม่ได้ ถ้าต้องการเพิ่มข้อมูลแบบอาร์เรย์ จะต้องใช้การ PUT ข้อมูลเข้าไปต่อท้ายเรื่อย ๆ ซึ่งจะมี Key ที่ Firebase สร้างให้เป็นตัวอ้างอิง

Firebase กับ API เพื่ออุปกรณ์ IoT Firebase มี API ของหลายภาษาให้เลือกใช้งาน ทั้งภาษา Python (นิยมใช้ใน Embedded OS) JavaScript (ในบนหน้าเว็บไซต์) และรวมไปถึงใน ESP8266 ที่ใช้ Arduino IDE ด้วย และสิ่งที่ Google นำไฟร์เบสมาพัฒนาเพิ่ม ได้แก่

1) **Firebase Analytics** บริการวิเคราะห์ข้อมูล ดึงเทคโนโลยีมาจาก Google Analytics แถมยังเปิดให้ใช้ฟรีแบบไม่จำกัดปริมาณข้อมูลใดๆ

2) ระบบส่งข้อความแจ้งเตือน **Google Cloud Messaging (GCM)** เปลี่ยนชื่อมาเป็น **Firebase Cloud Messaging (FCM)** ใช้งานฟรีไม่จำกัดปริมาณข้อความ กูเกิลบอกว่าตอนนี้ FCM ให้บริการข้อความแจ้งเตือน 1.7 แสนล้านข้อความต่อวัน
3) **Firebase Storage** บริการพื้นที่เก็บข้อมูล เอาไว้เก็บภาพ วิดีโอ หรือไฟล์ขนาดใหญ่จากแอพของผู้ใช้ สร้างอยู่บน Google Cloud Storage
4) **Firebase Remote Config** ตัวช่วยอัพเดตคอนฟิกของแอพ สำหรับปรับแต่งค่าต่างๆ ในแอปพลิเคชันจากระยะไกล (เช่น เกมที่อยากปรับสมดุลของเกมตลอดเวลา) สามารถใช้ร่วมกับ Firebase Analytics เพื่อกำหนดผู้ใช้งานแยกเป็นกลุ่มๆ ได้
5) **Firebase Crash Reporting** ตัวรายงานการแครชของแอพ รองรับทั้ง iOS และ Android
6) **บริการทดสอบแอพบนฮาร์ดแวร์จริง Cloud Test Lab** ที่เปิดตัวเมื่อปีที่แล้ว เปลี่ยนชื่อมาเป็น Firebase Test Lab for Android
7) **Firebase Notifications** เป็นคอนโซลสำหรับนักพัฒนา เพื่อยิงข้อความผ่าน FCM ไปยังผู้ใช้ สำหรับโปรโมทหรือกระตุ้นให้ผู้ใช้กลับมาเปิดแอพของเรา (เช่น แจกของในเกม)
8) **Firebase Dynamic Links** บริการ URL กลางที่สามารถชี้ทางไปยังเพจต่างๆ แปรผันตามอุปกรณ์หรือคุณสมบัติของผู้ใช้ (เช่น แต่ละประเทศกดลิงก์เดียวกัน เข้าคนละเพจกัน)
9) **Firebase Invites** ระบบเชิญเพื่อนมาใช้แอพ มีฟีเจอร์ referral คนชวนได้สิทธิประโยชน์
10) **Firebase App Indexing** เปลี่ยนชื่อมาจาก Google App Indexing ที่ช่วยให้ Google Search ค้นเจอเนื้อหาภายในแอพ

2.9.1 Firebase Realtime Database

Firebase Realtime Database เป็น NoSQL cloud database ที่เก็บข้อมูลในรูปแบบของ JSON และมีการ sync ข้อมูลแบบ realtime กับทุกอุปกรณ์ที่เชื่อมต่อแบบอัตโนมัติในเวลาไม่ถึงวินาที รองรับการทำงานเมื่อออฟไลน์ (ข้อมูลจะถูกเก็บไว้ใน local จนกระทั่งกลับมา online ก็จะทำการ sync ข้อมูลให้อัตโนมัติ) รวมถึงมี Security Rules ให้เราสามารถออกแบบเงื่อนไขการเข้าถึงข้อมูลทั้งการ read และ write ได้ตามต้องการ ทั้ง Android, iOS และ Web

2.9.2 ไฟร์เบสและแอนดรอยด์สตูดิโอ

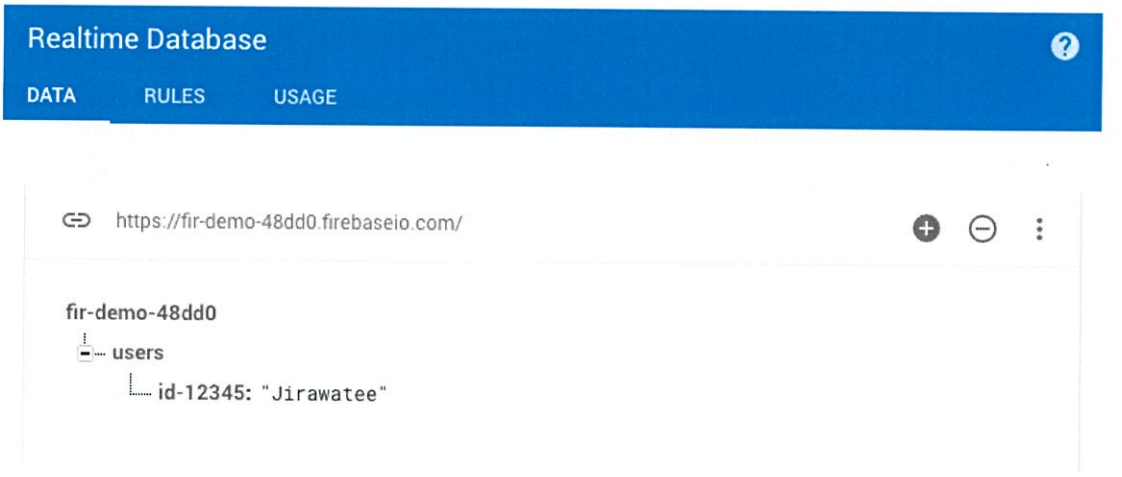
2.9.2.1 การเขียนข้อมูล

การ write, update หรือ delete ข้อมูลใน Firebase Realtime Database จะรองรับค่าหลายประเภททั้ง String, Long, Double, Boolean, Map<String, Object> และ List<Object> โดยการ write จะมีด้วยกัน 4 รูปแบบดังนี้

1) setValue() เป็นการ write หรือ update ข้อมูล ไปยัง path ที่เราอ้างถึงได้ เช่น users/<user-id>/<username>

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

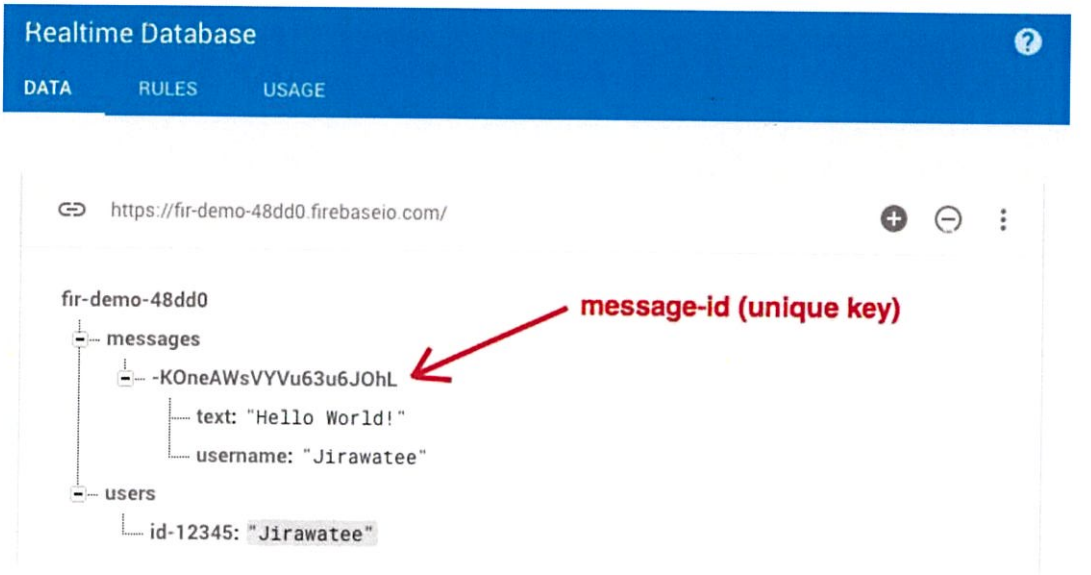
สามารถดูผลลัพธ์แบบ realtime ได้ที่ Firebase Console โดยไปที่เมนู Database แล้วเลือก tab แรก คือ Data ก็จะเห็นข้อมูลทั้งหมด



รูปที่ 2.30 แสดงการเขียนข้อมูลไปบนฐานข้อมูลออนไลน์

2) push() เป็นการเพิ่มชุดของข้อมูล ในที่นี้จะสร้าง model object ชื่อ FriendlyMessage ซึ่งจะบรรจุ text และ username ไว้ โดยการ push นั้น Firebase จะสร้าง unique key ของชุดข้อมูลนั้นๆ เพื่อใช้อ้างอิงต่อไปได้ เช่น messages/<message-id>/<data-model>

```
FriendlyMessage friendlyMessage = new FriendlyMessage("Hello World!",
"Jirawatee");

mMessageRef.push().setValue(friendlyMessage);
```



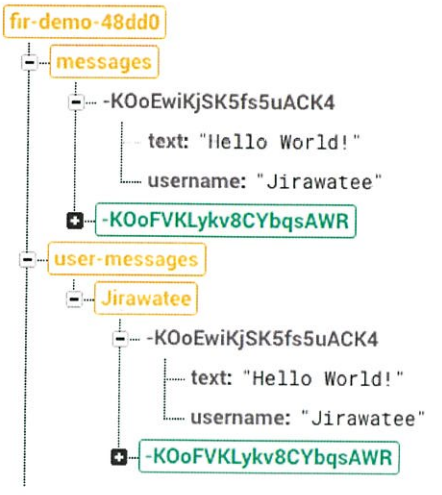
รูปที่ 2.31 แต่ละชุดข้อมูลถูกสร้าง unique key

3) updateChildren() เป็นการ write หรือ update ข้อมูลบางส่วน(บาง key) ตาม path ที่อ้างถึง โดยไม่ต้องทำการ replace ข้อมูลทั้งชุด และสามารถทำพร้อม ๆ กันได้หลาย object

ตัวอย่างจะเป็นการสร้าง post ใหม่ขึ้นมา โดยจะ write ข้อมูลไป 2 ที่คือ

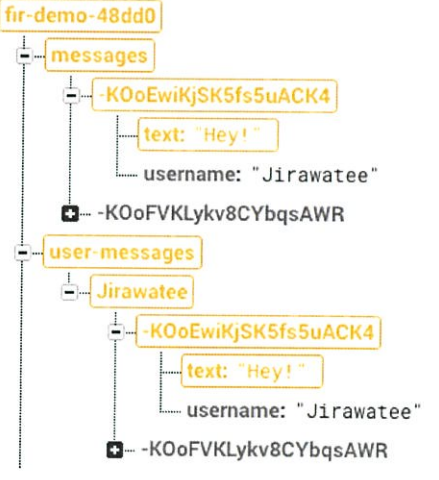
```
/user-messages/Jirawatee/$postid และ /messages/$postid

// push เป็นการ generate $postid ของ object ชื่อ posts ออกมาก่อนเพื่อใช้ใน // /user-
posts/$userid/$postid

String key = mMessagesRef.push().getKey();

HashMap<String, Object> postValues = new HashMap<>();

postValues.put("username", "Jirawatee");

postValues.put("text", "Hello World!");


Map<String, Object> childUpdates = new HashMap<>();

childUpdates.put("/messages/" + key, postValues);

childUpdates.put("/user-messages/Jirawatee/" + key, postValues);

rootRef.updateChildren(childUpdates);
```

สีที่แสดงใน console จะบอกสถานะแตกต่างกัน สีเหลืองแสดงการอัพเดทข้อมูล สีเขียวแสดงการเพิ่มข้อมูล สีแดงแสดงการลบข้อมูลออก และสีน้ำเงินแสดงการเคลื่อนย้ายข้อมูล



รูปที่ 2.32 แสดงแต่ละสีที่บอกความหมายที่ต่างกัน

ในกรณีที่ต้องการอัพเดทข้อมูลบางส่วน ก็สามารถทำได้พร้อม ๆ กันได้ โดยจะต้องรู้ username และ message-id เป็นตัวระบุในแต่ละ object



รูปที่ 2.33 แสดงการอัพเดตข้อมูลพร้อมกัน

4) runTransaction() เป็นการอัพเดทข้อมูล ที่มี concurrent เยอะๆ ที่อาจเกิดชนกัน เกิดข้อผิดพลาดได้ ตัวอย่างเช่น การกด like และกด unlike ที่โพสเดียวกัน เวลาเดียวกัน จะต้องมีการนับยอด like ตลอดเวลา ว่าช่วงเวลานั้นเป็นเท่าไร

```
postRef.runTransaction(new Transaction.Handler() {
    @Override
    public Transaction.Result doTransaction(MutableData mutable) {
        Post p = mutable.getValue(Post.class);
        if (p == null) {
            return Transaction.success(mutable);
        }

        if (p.stars.containsKey(getUid())) {
            // Unlike the post and remove self from likes
            p.starCount = p.starCount - 1;
            p.stars.remove(getUid());
        } else {
            // Like the post and add self to likes
            p.starCount = p.starCount + 1;
            p.stars.put(getUid(), true);
        }

        // Set value and report transaction success
        mutable.setValue(p);
        return Transaction.success(mutable);
    }

    @Override
    public void onComplete(DatabaseError databaseError, boolean b,
DataSnapshot dataSnapshot) {
        if (databaseError != null) {
```

```
                databaseError.getMessage()
                Log.w(TAG, databaseError.getMessage());
            } else {
                Log.d(TAG, "Transaction successful");
            }
        }
});
```

2.9.2.2 การลบข้อมูล (Delete)

การลบข้อมูลนั้น ให้ระบุ path ที่ต้องการจะลบ จากนั้นก็เรียกคำสั่ง removeValue() ตัวอย่างเช่น ต้องการลบข้อความทั้งหมดใน object ชื่อ messages

```
mMessageRef.removeValue();
```

นอกจากนั้นเรายังสามารถลบข้อมูล ได้ด้วยการส่งค่า null ไปที่ setValue(null) และสามารถใช้ค่า null กับเทคนิค updateChildren() เพื่อลบข้อมูลหลายๆ object ได้ด้วย

```
// ลบแบบ setValue()
mMessageRef.setValue(null);
// ลบแบบ updateChildren();
childUpdates.put("/messages/", null);
```

2.9.2.3 การอ่านข้อมูล

การอ่านข้อมูล จะมี 2 ประเภทแยกตาม Listener ดังนี้

1) ValueEventListener

จะอ่านข้อมูลตั้งแต่เริ่ม และจะอ่านข้อมูลทุกครั้งที่มีการเปลี่ยนแปลงของข้อมูลทั้งหมดภายใต้ path ที่เราอ้างถึง วิธีการคือใช้ออบเจ็คที่อ้างถึงมา addValueEventListener โดยจะมี callback 2 แบบ

onDataChange จะถูกเรียกตอนเริ่ม และถูกเรียกทุกครั้งที่ข้อมูลภายใต้ path ที่อ้างถึงมีการเปลี่ยนแปลง

onCancelled จะถูกเรียกเมื่อไม่สามารถอ่านข้อมูลจาก database ได้

```
mRootRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {

        String value = dataSnapshot.getValue(String.class);

        mTextView.setText(value);

    }


    @Override

    public void onCancelled(DatabaseError error) {

        mTextView.setText("Failed: " + databaseError.getMessage());

    }

});
```

ข้อควรระวังของ ValueEventListener คือไม่ควรอ้างถึง root เนื่องจากจะคอยดูข้อมูลทั้งหมดของ database ซึ่งอาจมีขนาดใหญ่ และ เปลือง bandwidth โดยใช่เหตุ จึงควรใช้วิธีนี้แบบเจาะจงเฉพาะ หรืออ่านข้อมูลทั้งหมดครั้งแรก แล้ว removeEventListener() ออกไป

2) ChildEventListener

จะคอยรับข้อมูลจาก การเพิ่ม, การเปลี่ยนแปลง, การลบ และ การย้าย เฉพาะของ child ที่อ้างถึง วิธีการคือใช้ object ที่อ้างถึงมา addChildEventListener โดยจะมี callback 5 แบบ

onChildAdded() จะถูกเรียกเมื่อมีการเพิ่มชุดข้อมูลเข้ามาใน child

onChildChanged() จะถูกเรียกเมื่อข้อมูลใน child มีการเปลี่ยนแปลง

onChildRemoved() จะถูกเรียกเมื่อข้อมูลใน child ถูกลบ

onChildMoved() จะถูกเรียกเมื่อมีการเรียงลำดับของข้อมูลใน child เกิดขึ้น

onCancelled() จะถูกเรียกเมื่อโหลดข้อมูลจาก child ไม่สำเร็จ

ตัวอย่างการ addChildEventListener ไปกับการ comment ซึ่งเหมาะกับ RecyclerView

```
ChildEventListener childEventListener = new ChildEventListener() {
    @Override
    public void onChildAdded(DataSnapshot dataSnapshot, String
previousChildName) {
        // A new comment has been added, add it to the displayed list
        Comment comment = dataSnapshot.getValue(Comment.class);
        // ...
    }

    @Override
    public void onChildChanged(DataSnapshot dataSnapshot, String
previousChildName) {
        // A comment has changed, use the key
        // to determine if we are displaying this
        // comment and if so displayed the changed comment.
        Comment newComment = dataSnapshot.getValue(Comment.class);
        String commentKey = dataSnapshot.getKey();
        // ...
    }

    @Override
    public void onChildRemoved(DataSnapshot dataSnapshot) {
        // A comment has changed, use the key
        // to determine if we are displaying this
        // comment and if so remove it.
        String commentKey = dataSnapshot.getKey();
        // ...
```

```
    }

    @Override

    public void onChildMoved(DataSnapshot dataSnapshot, String
previousChildName) {

        // A comment has changed position,

        // use the key to determine if we are

        // displaying this comment and if so move it.

        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);
```

การเพิ่ม listener เข้าไปหลายตัว นั่นก็แปลว่าจะต้องมีการ call เกิดขึ้นมากมายตามแต่ event ซึ่งหากเราไม่ได้ใช้ หรือออกจากหน้าดังกล่าว ก็ควรจะถอด listener เหล่านั้นออกไปด้วย เพื่อการใช้ bandwidth แบบคุ้มค่ามากที่สุดโดยสามารถถอดออกได้ด้วยคำสั่ง removeEventListener()

```
@Override

protected void onStop() {

   super.onStop();

   if (mValueEventListener != null) {

      mRootRef.removeEventListener(mValueEventListener);

   } }
```

บางครั้งอาจต้องการอ่านข้อมูลแค่ครั้งเดียวและไม่สนใจอีก Firebase ได้เตรียม addListenerForSingleValueEvent() สำหรับการอ่านข้อมูลเพียงครั้งเดียว ที่เมื่อได้ callback แล้วก็จะ remove listener ทิ้งอัตโนมัติ ตัวอย่างเช่น ดึงข้อมูลผู้ใช้ก่อนทำการโพส ว่าเขาผ่านการ sign-in มาหรือยัง หรือยังขาดข้อมูลอะไรที่ต้องการ หากครบถ้วนก็โพสได้ แต่หากไม่ครบอาจพาไปหน้า sign-in หรือพาไปหน้ากรอกข้อมูลให้ครบถ้วน

```
mDatabase.child("users").child(userId).addListenerForSingleValueEvent(new
ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        User user = dataSnapshot.getValue(User.class);
        if (user == null) {
            Toast.makeText(NewPostActivity.this, "Error: could not fetch user.",
Toast.LENGTH_LONG).show();
        } else {
            writeNewPost(userId, user.username, title, body);
        }
        finish();
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {
        Log.e(TAG, databaseError.getMessage());
    }
});
```

การเรียงลำดับและการกรองข้อมูล (Sorting and Filtering) การ query ข้อมูล ของ Firebase Realtime Database นั้น รองรับการ sort และ filter ได้ อย่างก็ดีการ sort และ filter สามารถทำให้การ query นั้นช้าได้ ทางที่ดีควรศึกษาเรื่องการทำ index (.indexOn) ไปด้วยจะทำให้การ query นั้นมีประสิทธิภาพมากขึ้น

การ Sort ข้อมูล มี 3 รูปแบบ ดังนี้ (Ordering Function)

orderByChild() เป็นการเรียงลำดับ value ของ child key ที่ถูกเลือก (การใช้งานคล้าย WHERE ใน SQL)

orderByKey() เป็นการเรียงลำดับ child key (ใช้เรียง PK เหมาะกับแสดงข้อมูลแบบมี limit หรือแสดง pagination)

orderByValue() เป็นการเรียงลำดับ child value (เหมาะกับการเรียงค่าตัวเลข)

// ตัวอย่างการเรียงลำดับโพสของฉันที่ได้รับคะแนนโหวตมากที่สุด

```
dbReference.child("user-posts").child(getUid()).orderByChild("starCount");
```

การ Filter ข้อมูล มี 5 รูปแบบ ดังนี้ (Querying Function)

limitToFirst() การระบุจำนวน item ซึ่งจะเรียงลำดับจากแถวแรก

limitToLast() การระบุจำนวน item ซึ่งจะเรียงลำดับจากแถวสุดท้าย

startAt() จะดึงจำนวน item ที่มากกว่าหรือเท่ากับที่ระบุ key หรือ value โดยขึ้นอยู่กับการ order-by

endAt() จะดึงจำนวน item ที่น้อยกว่าหรือเท่ากับที่ระบุใน key หรือ value โดยขึ้นอยู่กับการ order-by

equalTo() จะดึงจำนวน item ที่เท่ากับที่ระบุใน key หรือ value โดยขึ้นอยู่กับการ order-by

// ตัวอย่างการดึงข้อมูลโพส 100 อันล่าสุด

```
databaseReference.child("posts").limitToFirst(100);
```

2.9.2.4 การเปิดใช้งานโหมดออฟไลน์

ลักษณะการทำงานออฟไลน์ (Persistence Behavior) คือ Firebase Realtime Database มีการจัดการเรื่องการเชื่อมต่อให้แล้ว เมื่อไม่สามารถเชื่อมต่ออินเตอร์เน็ต หรืออยู่ในโหมดออฟไลน์ ยังสามารถใช้งานแอพได้ โดยตัว Firebase จะทำการ cache ข้อมูลไว้ทุกการกระทำ และจะทำการ sync ข้อมูลให้อัตโนมัติเมื่อเรากลับเข้าสู่โหมด online แม้ว่าเราจะปิดแล้วเปิดแอพใหม่ก็ตาม แค่ประกาศใช้งานเพียงบรรทัดเดียว

```
FirebaseDatabase.getInstance().setPersistenceEnabled(true);
```

Firebase Realtime Database จะ sync และเก็บข้อมูลใน local ของ client เฉพาะข้อมูลที่ตัว listener ทำงานอยู่ แต่สามารถจะ sync ข้อมูลส่วนอื่นที่ยังไม่ได้ active ได้ โดยให้อ้างถึง path ที่ต้องการ sync ตัวอย่างเช่น

```
DatabaseReference scoresRef =
FirebaseDatabase.getInstance().getReference("scores");

scoresRef.keepSynced(true);
```

และโดยทั่วไปจะเก็บข้อมูลลง cache ได้ไม่เกิน 10MB แต่หากแอพมีข้อมูลที่ต้องการ cache มากกว่านั้น Firebase Realtime Database จะ purge cache ออก แต่ข้อมูลที่เก็บแบบ keepSynced จะไม่ถูก purge

เมื่อแอพ sync ข้อมูลตอน online มาไว้ในเครื่องแล้ว และเมื่อแอพเข้าสู่โหมดออฟไลน์ ก็ยังสามารถ query ข้อมูลที่ sync มาแล้วได้ ตัวอย่างเช่น

```
// ตอน online ได้ผลลัพธ์ 4 แถว จากการ query
scoresRef.orderByValue().limitToLast(4)
// เมื่อ offline หากต้องการ query ผลลัพธ์ 2 แถว ก็ทำได้
scoresRef.orderByValue().limitToLast(2)
```

การ run ตัว Transactions ขณะออฟไลน์ เหมือนกับกรณีออฟไลน์ ปกติ คือเมื่อ run ตัว transaction ในขณะออฟไลน์ ข้อมูลต่าง ๆ ก็จะถูกเข้าคิวไว้ แต่หากมีการปิดแล้วเปิดแอพใหม่คิวจะไม่ถูกเก็บไว้ โดยควรแจ้งผู้ใช้ให้ทราบว่า transaction ไม่สำเร็จในกรณีที่ปิดแล้วเปิดแอพอีกครั้ง

2.9.3 Security & Rules

Firebase Realtime Database Rules เป็นการกำหนดการเข้าถึง database ทั้งการ read, write และการทำ index โดย rules ทั้งหมดจะอยู่บน Firebase server แล้วเราสามารถปรับเปลี่ยนและมีผลทันทีได้ตลอดเวลา ซึ่งมี syntax เป็น Javascript-like โดยจะแบ่ง rules ออกเป็น 4 ประเภทดังนี้

.read คือ การอนุญาตให้อ่าน

.write คือ การอนุญาตให้เขียน

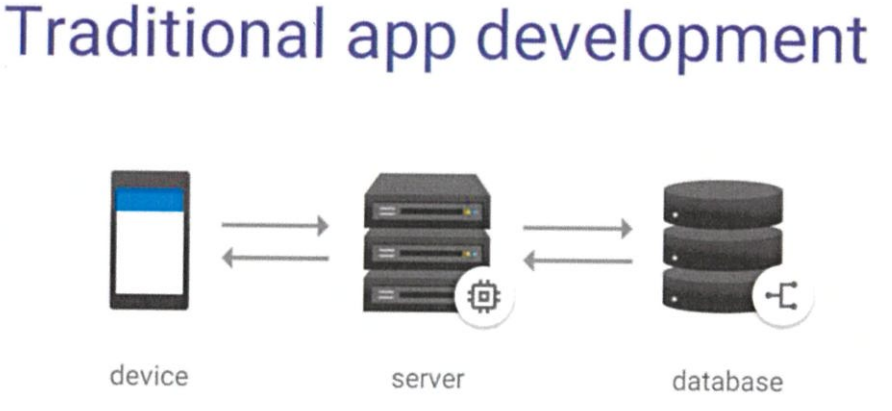
.validate คือ การตรวจสอบข้อมูลที่เข้ามา

.indexOn คือ การทำ index เพื่อความรวดเร็วในการเรียงลำดับ และการ query

โดยการกำหนด rules ต่างๆให้เข้าไปที่ Firebase Console เลือกโปรเจค จากนั้นเลือกเมนู Database แล้วเลือก tab ชื่อ RULES

2.9.4 Firebase Authentication

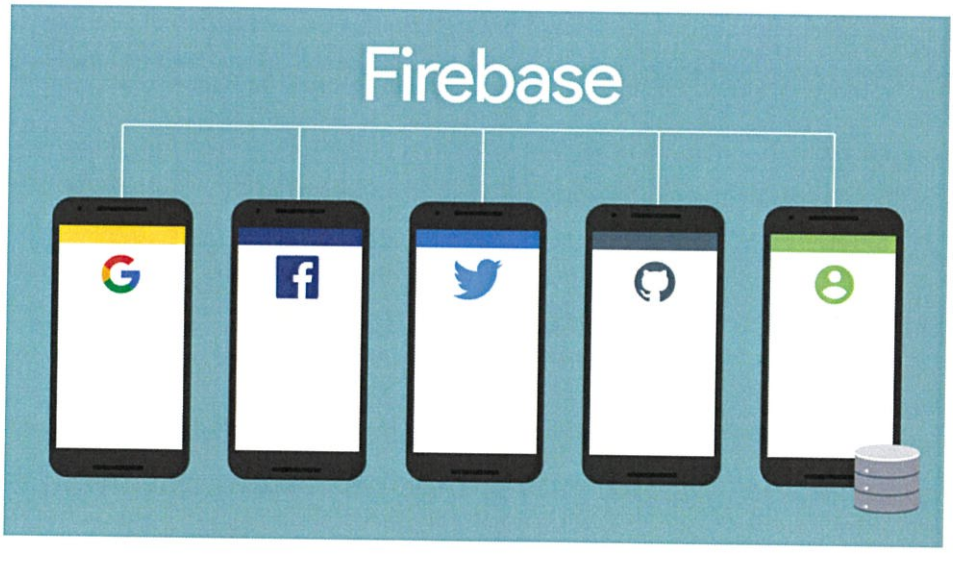
การพัฒนาระบบ Log-in ขึ้นมาโดยให้สามารถใช้งานข้าม platform ได้ทั้ง Android, iOS และ Web จะต้องหา server หรือ cloud ที่ performance ดี ๆ และ scale ได้, ต้องออกแบบ database เพื่อรองรับ third party ต่าง ๆ , ต้องทำความเข้าใจเรื่อง OAuth 2.0 และติดตั้ง SSL ลงไปเพื่อความปลอดภัย, ต้องพัฒนา backend และสุดท้ายก็พัฒนา frontend ทั้ง Android, iOS และ Web ซึ่ง Firebase Authentication เป็นบริการที่เตรียมทุกอย่างมาให้หมดแล้ว และไม่มีค่าใช้จ่าย



รูปที่ 2.34 แสดงขั้นตอนที่หลากหลายของการพัฒนาระบบ log-in

รูปที่ 2.35 แสดงการลดขั้นตอนจากเดิมด้วยบริการ Firebase Authentication

Firebase Authentication จะเป็นบริการที่เข้ามาจัดการ backend ให้ทั้งหมด ทั้ง การ register, การ sign-in, การ reset password, โดยจะมี SDK ให้ทั้ง Android, iOS และ Web นำไปติดตั้งและใช้งาน ซึ่งรองรับการ sign-in หลากหลายรูปแบบทั้งจาก social network ยอดนิยม, จาก Email และ Password ของผู้ใช้งาน หรือแบบไม่ระบุตัวตน(Anonymous) ก็ได้



รูปที่ 2.36 แสดงบริการ Firebase Authentication ที่รองรับหลายรูปแบบ

2.10 โปรแกรมช่วยออกแบบ Autodesk Inventor Professional

รูปที่ 2.37 Autodesk Inventor Professional

Autodesk Inventor Professional เป็นโปรแกรมเขียนแบบและออกแบบที่ถูกพัฒนาขึ้นมาเพื่อใช้ในงานออกแบบผลิตภัณฑ์ออกแบบเฟอร์นิเจอร์และออกแบบชิ้นส่วนเครื่องกล 3 มิติซึ่งมีฟังก์ชั่นการใช้งานดังต่อไปนี้

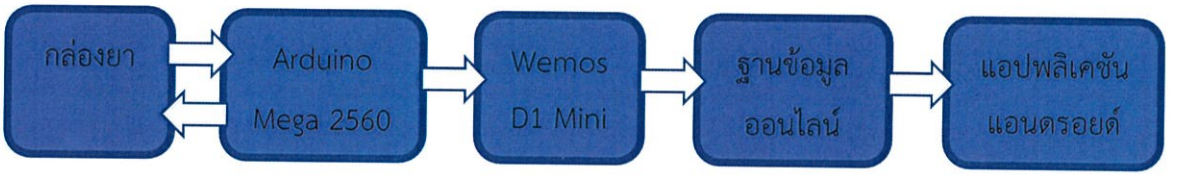
- การสร้าง Part Solid ใช้วิธีการและเทคโนโลยีของ Surface Modeling (NURBS)
- iPart Catalog (ชิ้นส่วนมาตรฐาน) สามารถ Download ชิ้นส่วนมาตรฐานจากเว็บไซต์ WWW. CBLISS.COM มาใช้งานได้ทันที
- Assembly Modeling สามารถประกอบชิ้นส่วน 3 มิติได้เร็วขึ้น โดยมีขนาดของไฟล์เล็กลงและใช้หน่วยความจำน้อย
- Drawing สามารถเปิดไฟล์ dwg 2 มิตินำมาสร้าง Part 3 มิติ, สร้าง Drawing 2 มิติจาก 3มิติ โดยอัตโนมัติและ บันทึกไฟล์เป็น *dwg ได
- Adaptive ใช้หาขนาดที่เหมาะสมของ Mechanical Link และชิ้นส่วนที่ประกอบกัน
- Motion ใช้ทดสอบการเคลื่อนที่และตรวจสอบหาชิ้นส่วนที่ขัดกัน

- Presentation สร้างภาพเคลื่อนไหวแสดงการทำงานของชิ้นส่วนหรือเครื่องจักรกลและสามารถบันทึกไฟล์เป็น *AVI (ไฟล์วีดิโอ)ได้
- Sheet Metal สามารถสร้างงานพับแบบต่างๆและทำแผ่นคลี่งานโลหะแผ่นได้
- Weldment สามารถสร้างรอยเชื่อมทั้ง 2 มิติและ 3 มิติได้

บทที่ 3

ระเบียบวิธีวิจัย

การสร้างระบบการติดตามการรับประทานยาของผู้ป่วยนั้น เป็นการนำความรู้ทั้งทางด้านวงจรอิเล็กทรอนิกส์ และทางด้านเทคโนโลยี ได้แก่ การจัดเก็บข้อมูลบนฐานข้อมูลออนไลน์ และการออกแบบแอปพลิเคชันแอนดรอยด์ มาผสมผสานกันเพื่อส่งเสริมให้การรับประทานยาของผู้ป่วยมีประสิทธิภาพมากขึ้น รวมทั้งยังอยู่ในรูปแบบที่ใช้งานได้สะดวกและง่าย ซึ่งองค์ประกอบหลักของระบบนี้สามารถแบ่งออกได้เป็น 5 ส่วน ดังรูปที่ 3.1



รูปที่ 3.1 แผนภาพรวมในการดำเนินการวิจัย

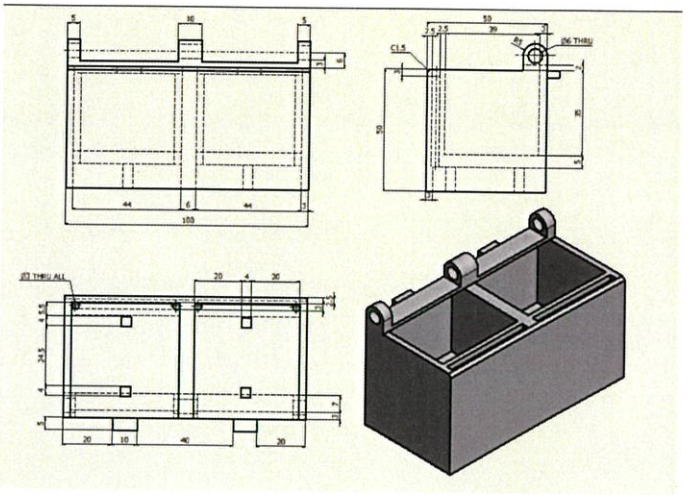
3.1 กล่องยา

ส่วนของกล่องยานั้น เป็นส่วนที่ผู้ป่วยนั้นจะใช้งานและสัมผัสโดยตรงเพื่อทำการรับประทานยาชนิดต่าง ๆ ตามแต่ละบุคคล ดังนั้น การออกแบบฮาร์ดแวร์ส่วนนี้จึงมีวัตถุประสงค์เพื่อที่จะรับรู้การรับประทานยาของผู้ป่วยจากกล่องยา ในที่นี้ ทางผู้วิจัยเลือกตรวจจับสถานะการเปิด-ปิดของแต่ละช่องของกล่องยา เนื่องจากแนวคิดว่าการรับประทานยาของผู้ป่วยทุกคนนั้นต้องผ่านกระบวนการนี้ทั้งสิ้น โดยได้ออกแบบไว้ดังนี้

3.1.1 การออกแบบกล่องยา

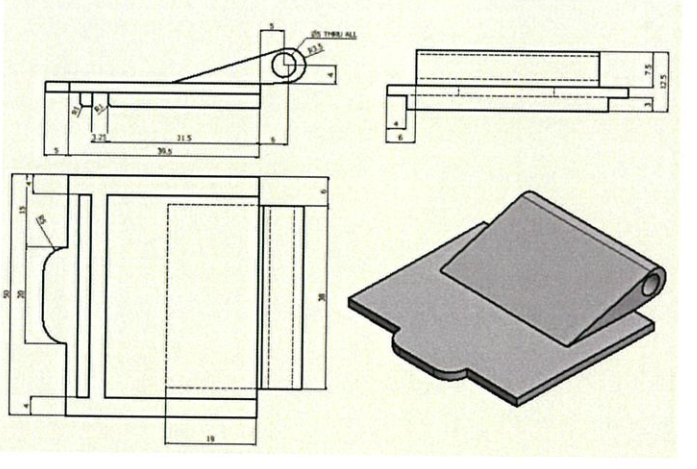
ผู้วิจัยได้ออกแบบกล่องยาโดยใช้โปรแกรม Autodesk Inventor Professional 2018 ในการออกแบบ ในเบื้องต้นผู้วิจัยจะทำการพัฒนาโดยใช้กล่องยา 4 ช่อง ซึ่งผู้วิจัยได้แบ่งงานออกแบบเป็น 6 ส่วน (ขนาดในภาพฉายมีหน่วยเป็นมิลลิเมตร) ดังนี้

ส่วนที่ 1 กล่องยา จะประกอบด้วยกล่องยาจำนวนสองช่อง ด้านจะมีรูสำหรับใส่แกนเหล็ก ซึ่งจะใส่แกนเดียวกับฝาของกล่องยา ด้านล่างจะมีส่วนที่ใช้ยึดกล่องยากับฐานของอุปกรณ์ และกล่องยาจะมีช่องสำหรับการติดคอปเปอร์เทปเพื่อทำการตรวจจับการเปิดปิดของฝากล่องยา ซึ่งภาพฉายของกล่องยาสามารถดูได้จากรูปที่ 3.2



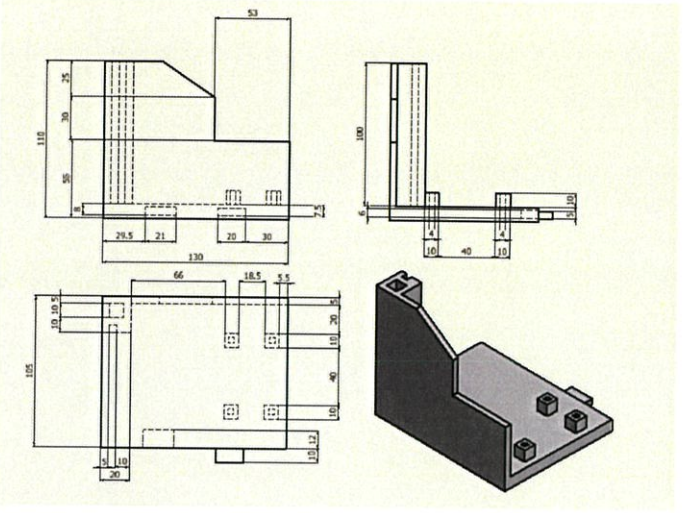
รูปที่ 3.2 แสดงภาพฉายของกล่องยา

ส่วนที่ 2 ฝากล่องยา จะมีรูสำหรับใส่แกนเหล็กซึ่งจะใส่แกนเดียวกับกล่องยา ด้านใต้ฝาจะมีส่วนที่ใช้สำหรับปิดช่องเก็บยา และส่วนที่ใช้สำหรับติดคอปเปอร์เทปเพื่อทำการตรวจจับการเปิดปิดฝากล่องยา ซึ่งภาพฉายของฝากล่องยาสามารถดูได้จากรูปที่ 3.3



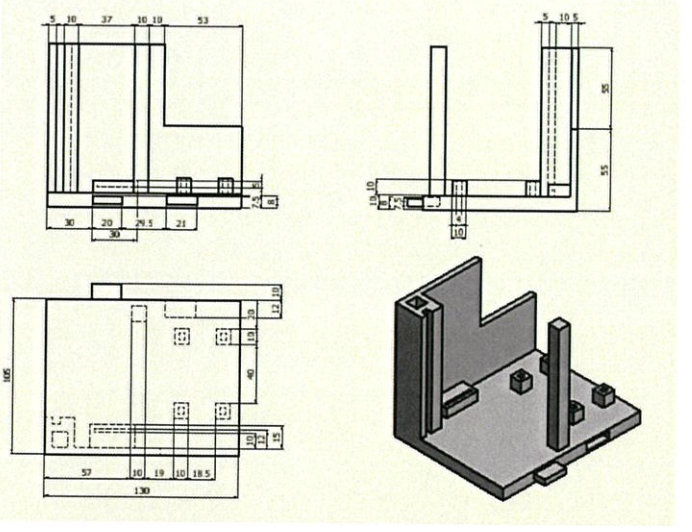
รูปที่ 3.3 แสดงภาพฉายของฝากล่องยา

ส่วนที่ 3 ฐานและฝาปิดด้านข้างฝั่งซ้ายของอุปกรณ์ จะประกอบด้วยช่องบริเวณฐานสำหรับยึดกล่องยากับอุปกรณ์ ช่องสำหรับยึดฝาปิดด้านบน และช่องสำหรับยึดฝาปิดด้านหลังของอุปกรณ์ ซึ่งภาพฉายของฐานและฝาปิดด้านข้างฝั่งซ้ายของอุปกรณ์สามารถดูได้จากรูปที่ 3.4



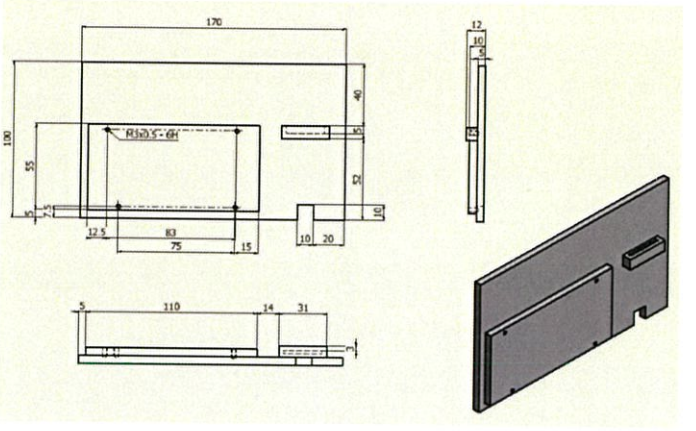
รูปที่ 3.4 แสดงภาพฉายของฝาผิดด้านข้างฝั่งซ้ายของอุปกรณ์

ส่วนที่ 4 ฐานและฝาปิดด้านข้างฝั่งขวาของอุปกรณ์ จะประกอบด้วยช่องบริเวณฐานสำหรับยึดกล่องยากับอุปกรณ์ ช่องและเสาสำหรับยึดฝาปิดด้านบน ช่องสำหรับยึดฝาปิดด้านหลังของอุปกรณ์ และช่องสำหรับใส่ RTC DS3231 ซึ่งภาพฉายของฐานและฝาปิดด้านข้างฝั่งขวาของอุปกรณ์สามารถดูได้จากรูปที่ 3.5



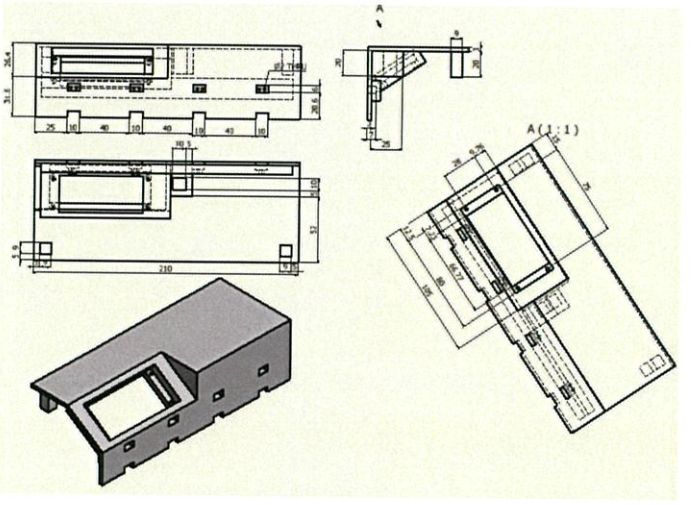
รูปที่ 3.5 แสดงภาพฉายของฝาผิดด้านข้างฝั่งขวาของอุปกรณ์

ส่วนที่ 5 ฝาปิดด้านหลังของอุปกรณ์ จะประกอบด้วย ช่องสำหรับใส่บอร์ด Wemos D1 Mini และบริเวณที่ใช้สำหรับยึดบอร์ด Arduino Mega 2560 ซึ่งภาพฉายของฝาปิดด้านหลังของอุปกรณ์สามารถดูได้จากรูปที่ 3.6



รูปที่ 3.6 แสดงภาพฉายของฝาผิดด้านหลังของอุปกรณ์

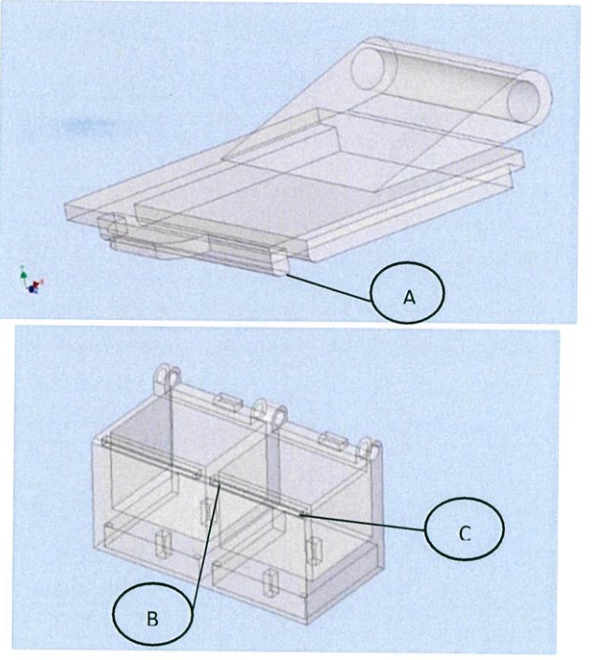
ส่วนที่ 6 ฝาปิดด้านบนของอุปกรณ์ จะประกอบด้วยช่องสำหรับใส่ LCD I2C ช่องสำหรับใส่ LED จำนวน 4 ช่องสำหรับ 4 ช่องของกล่องยา และด้านใต้จะเป็นส่วนที่ใช้ยึดฝาปิดด้านบนกับอุปกรณ์ซึ่งมีช่องสำหรับยึดเสาที่อยู่กับฐานอุปกรณ์และส่วนที่เป็นเสาสำหรับยึดกับฝาปิดด้านข้างของอุปกรณ์ ซึ่งภาพฉายของฝาปิดด้านหลังของอุปกรณ์สามารถดูได้จากรูปที่ 3.7



รูปที่ 3.7 แสดงภาพฉายของฝาผิดด้านบนของอุปกรณ์

3.1.2 การสร้างสวิตซ์ที่ฝากล่องยา

ผู้วิจัยได้นำคอปเปอร์เทป (Copper Tape) ติดกับกล่องยา โดยที่ในแต่ละช่องจะติดคอปเปอร์เทป 3 ตำแหน่ง คือ ที่ฝากล่อง 1 ตำแหน่ง บริเวณด้านใต้ฝากล่องยา(ตำแหน่ง A) และที่ตัวกล่องอีก 2 ตำแหน่ง บริเวณร่องที่เจาะไว้ใกล้กับรู(ตำแหน่ง B และ C) ดังรูปที่ 3.8 โดยที่ทั้งสองตำแหน่งนี้ต้องแปะคอปเปอร์เทปให้แยกจากกัน(ห้ามแปะเป็นแผ่นเดียวกัน) โดยการติดคอปเปอร์เทปลักษณะนี้เพื่อต้องการให้กล่องยาทำหน้าที่เป็นสวิตซ์ เมื่อปิดฝากล่องยาคอปเปอร์เทปที่ตำแหน่ง A จะทำหน้าที่ส่งต่อสัญญาณจากตำแหน่ง B ไปยังตำแหน่ง C



รูปที่ 3.8 แสดงตำแหน่งที่ติดคอปเปอร์เทปบนฝากล่องยา

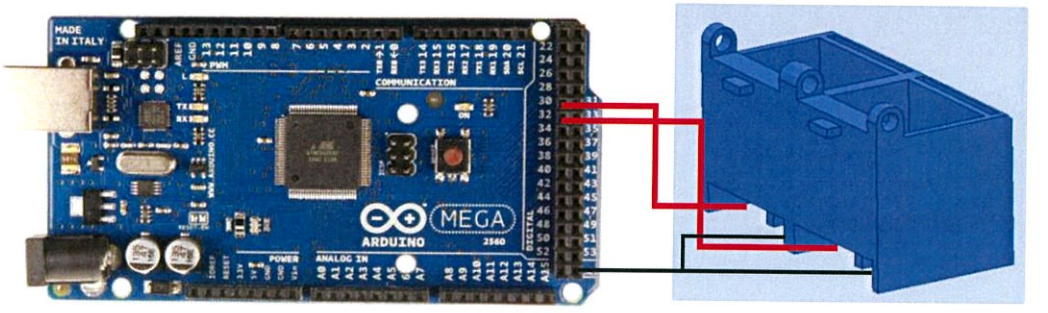
3.2 Arduino Mega 2560 R3

Arduino Mega 2560 R3 เป็นไมโครคอนโทรลเลอร์ที่เหมาะสมกับการออกแบบวงจรของกล่องยานี้ เนื่องจากเป็นอุปกรณ์มีขาที่เพียงพอในการต่อกับอุปกรณ์เสริมต่าง ๆ โดยทางผู้วิจัยได้ออกแบบให้สามารถทำงานได้หลากหลายเพื่อเป็นประโยชน์ต่อผู้ป่วย ดังนี้

3.2.1 การอ่านค่าสถานะ การเปิด-ปิดฝาของกล่องยา

หลังจากที่ผู้วิจัยได้ติดคอปเปอร์เทปดังข้อที่ 3.1.2 แล้วนั้น ผู้วิจัยได้นำ Arduino Mega 2560 R3 มาต่อเข้ากับกล่องยา โดยให้สายไฟด้านหนึ่งของแต่ละช่องของกล่องยาต่อเข้ากับกราวด์ของ

Arduino Mega 2560 R3 ส่วนขาอีกข้างหนึ่งต่อเข้ากับขาดิจิตอลของ Arduino Mega 2560 R3 โดยผู้วิจัยได้เลือกใช้ ขาดิจิตอลที่ 30,32,34,36 รูปที่ 3.9 แสดงตัวอย่างการต่อกล่องยาสำหรับสองช่อง

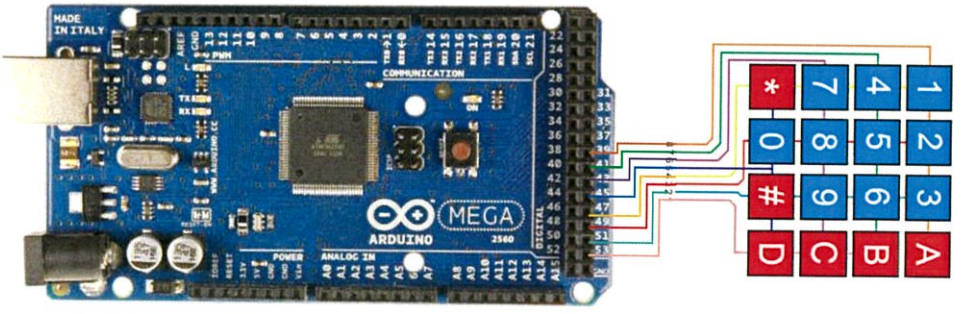


รูปที่ 3.9 แสดงการต่อกล่องยาเข้ากับ Arduino Mega 2560 R3

จากนั้นพิมพ์ชุดคำสั่งอ่านค่าดิจิตอลบนโปรแกรม Arduino IDE โดยที่ถ้าหากฝาปิด Arduino Mega จะรับค่า HIGH หากฝาเปิด Arduino Mega จะรับค่า Low จากนั่นให้พิมพ์ชุดคำสั่งให้แสดงสถานะการเปิด-ปิดฝากล่องยา (HIGH หรือ LOW) บน Serial Monitor

3.2.2 เชื่อมต่อ Arduino Mega 2560 R3 กับ Keypad

ผู้วิจัยเลือก Keypad เป็นอุปกรณ์เสริมในการกดเลือกการตั้งค่าเวลาบนกล่องยา โดยขาของ Keypad มีจำนวน 8 ขา โดยต่อกับขาดิจิตอลของ Arduino Mega 2560 R3 ขาที่ 53, 51, 49, 47, 45, 43, 41, 39 ตามลำดับ โดยต่อกับขาของ Keypad จากขาที่ 1 ไปยังขาที่ 8 ตามลำดับ ดังรูปที่ 3.10



รูปที่ 3.10 แสดงการต่อ Arduino Mega กับ Keypad

จากนั้นตั้งค่าปุ่มต่างๆของ Keypad โดยใช้ชุดคำสั่งดังนี้

```
const byte ROWS = 4; //four rows
```

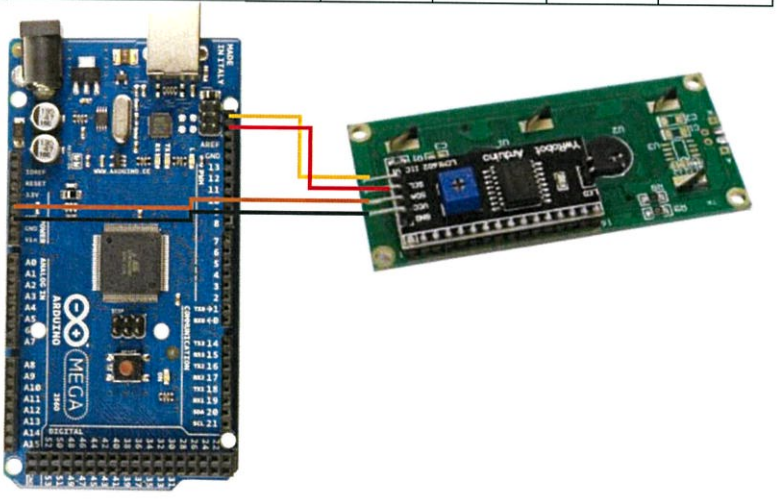
```
const byte COLS = 4; //four columns

char keys[ROWS][COLS] = {    {'1','2','3','A'},
                             {'4','5','6','B'},
                             {'7','8','9','C'},

                             {'*','0','#','D'}          };

byte rowPins[ROWS] = {7, 6, 5, 4}; //connect to the row pinouts of the keypad

byte colPins[COLS] = {11, 10, 9, 8};

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
```

3.2.3 หน้าจอแสดงผลบนกล่องยา

การเชื่อมต่อ Arduino 2560 R3 กับ LCD I2C 16X2 เริ่มจาก LCD I2C จะต้องต่อกับ LCD Controller จากนั้นจึงต้อง LCD controller กับ Arduino Mega โดยดูจากตารางที่ 3.1 กับรูปที่ 3.11

ตารางที่ 3.1 แสดงการเชื่อมต่อ LCD I2C กับ Arduino Mega

ขาของ LCD I2C	SCL	SDA	VCC	GND
ขาของ Arduino Mega	SCL1	SDA1	5V	GND



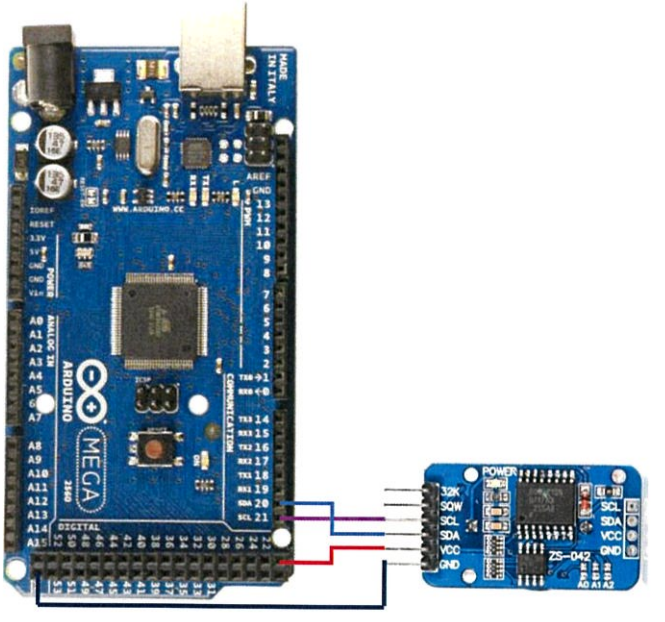
รูปที่ 3.11 แสดงการต่อ Arduino Mega กับ LCD I2C 16X2

จากนั้นทดสอบการทำงานของ LCD I2C ด้วยชุดคำสั่งดังนี้

```
#include <LiquidCrystal_I2C.h>        //ประกาศ Library ของจอ I2C
LiquidCrystal_I2C lcd(0x27,16,2);
// Set the LCD address to 0x27 for a 16 chars and 2 line display
void setup()  {
lcd.begin();                  // initialize the LCD
lcd.print("Hello !!!");       //ฟังก์ชั่นในการกำหนดข้อความที่ต้องการแสดงผล
lcd.setCursor(0, 1);          //ฟังก์ชั่นในการกำหนดตำแหน่ง Cursor
lcd.print("ThaiEasyElec");        }
```

3.2.4 เชื่อมต่อ Arduino 2560 R3 กับ RTC DS3231

เนื่องจากทางผู้วิจัยต้องการเวลาปัจจุบันมาอ้างอิงในการตั้งค่าเวลาการรับประทานยา จึงเลือกใช้ RTC DS3231 โดยการเชื่อมต่อ RTC DS3231 ต่อได้ดังรูปที่ 3.12 โดยเราจะต่อกับ RTC DS3231 เพียง 4 ขาเท่านั้น คือ ขา VCC GND SDA และ SCL



รูปที่ 3.12 แสดงการต่อ Arduino Mega กับ RTC DS3231

ทำการตั้งค่าเวลาเริ่มต้นให้ RTC DS3231 ดังนี้

```
#include <DS3231.h>
DS3231  rtc(SDA, SCL);
void setup()    {
        Serial.begin(9600);
        rtc.begin();
        rtc.setDOW(MONDAY);     // Set Day-of-Week to SUNDAY
        rtc.setTime(12, 0, 0);     // Set the time to 12:00:00 (24hr format)
        rtc.setDate(01, 01, 2018);   // Set the date to January 1st, 2014
}
```

หลังจากตั้งค่าให้กับ RTC DS3231 แล้วเราสามารถนำเวลามาใช้ได้โดยใช้คำสั่งดังต่อไปนี้

rtc.getDOWStr(); สำหรับรับค่าวันทั้ง 7 วันในหนึ่งสัปดาห์มาจาก RTC

rtc.getDateStr(); สำหรับรับค่าวันที่มาจาก RTC

rtc.getTimeStr(); สำหรับรับค่าเวลามาจาก RTC

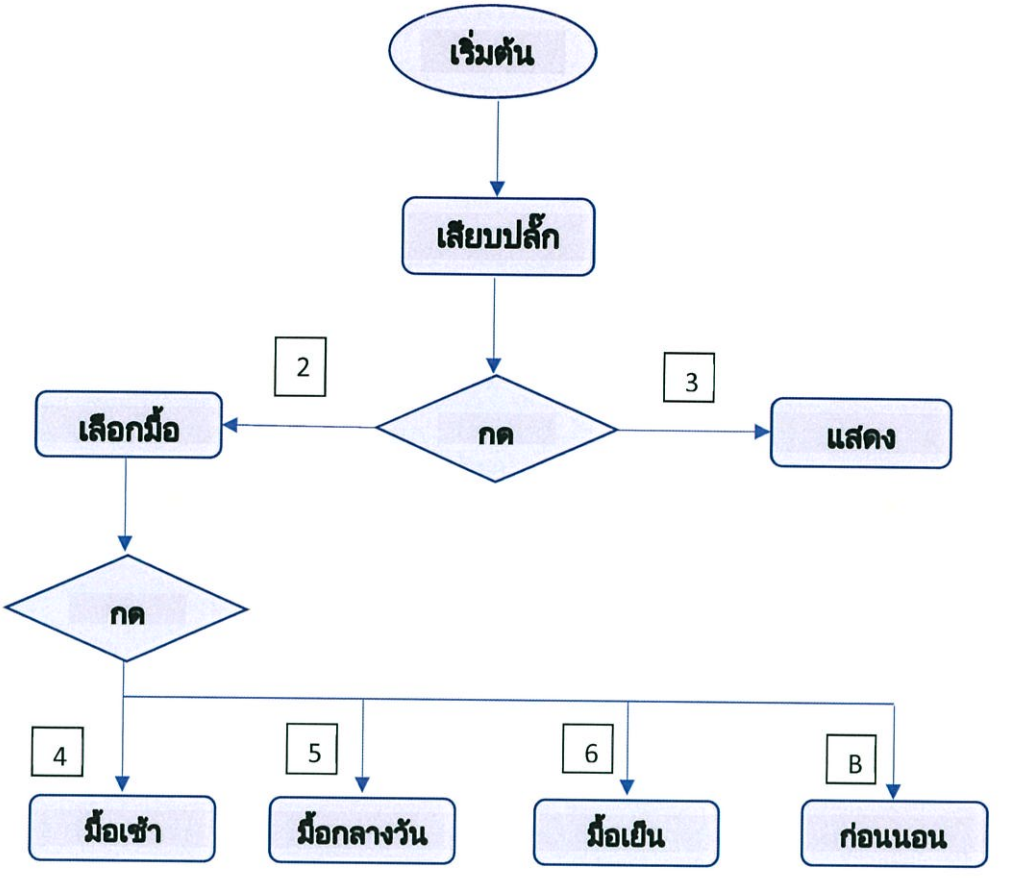
3.2.5 อุปกรณ์อื่นๆ กับ Arduino Mega 2560

ผู้วิจัยได้ต่อ LED จำนวน 4 ตัวเพื่อแจ้งเตือนเวลาการรับประทานยาซึ่งทั้ง 4 ตัวนี้จะแสดงถึงช่องทั้ง 4 ช่องของกล่องโดยผู้วิจัยได้ต่อ LED กับขาดิจิตอลที่ 46, 48, 50,52 ของ Arduino Mega ตามลำดับ

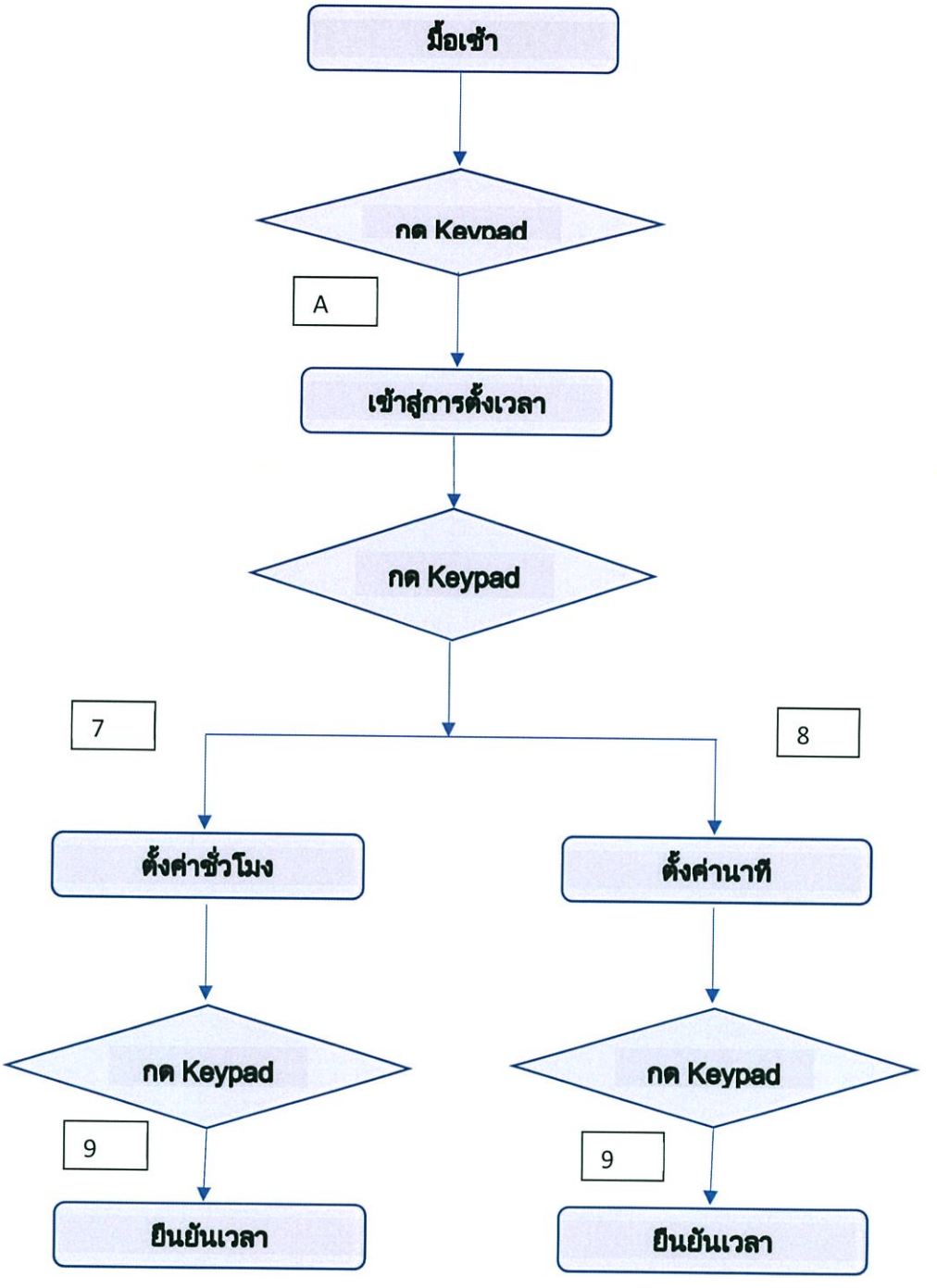
ผู้วิจัยได้ต่อ buzzer alarm สำหรับการแจ้งเตือนด้วยเสียงโดยต่อ buzzer ที่ขาดิจิตอลที่ 38 ของ Arduino Mega

3.2.6 การทำงานของ Arduino Mega 2560 R3 ภายในงานวิจัยนี้

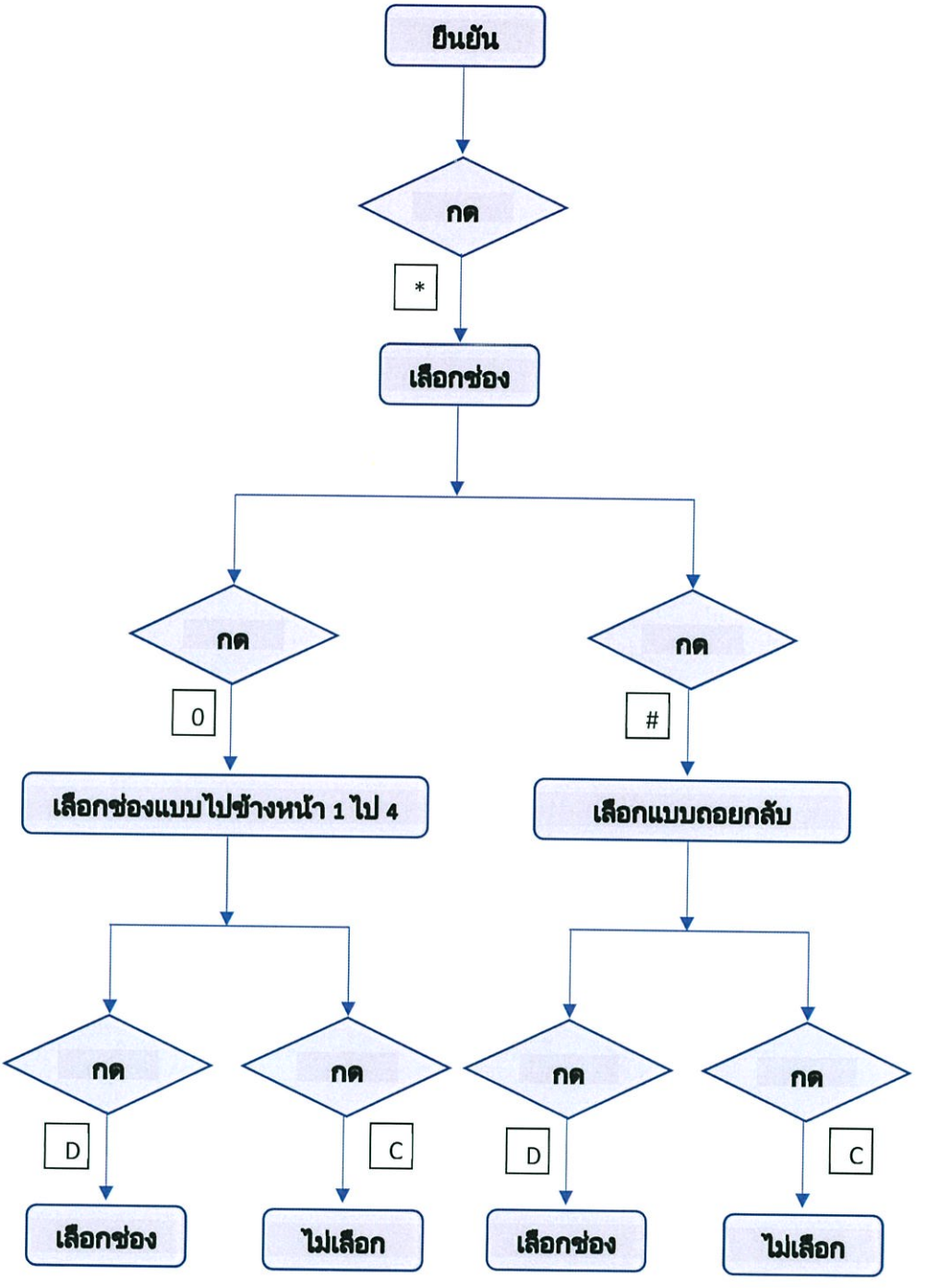
หลังจากการทดสอบอุปกรณ์ต่าง ๆ ในข้างต้นแล้ว ผู้วิจัยได้นำอุปกรณ์มาต่อเข้ากับArduino Mega 2560 R3 โดยให้สามารถทำงานได้ตามผังการทำงานดังรูปที่ 3.13 ถึงรูปที่ 3.16 ดังนี้



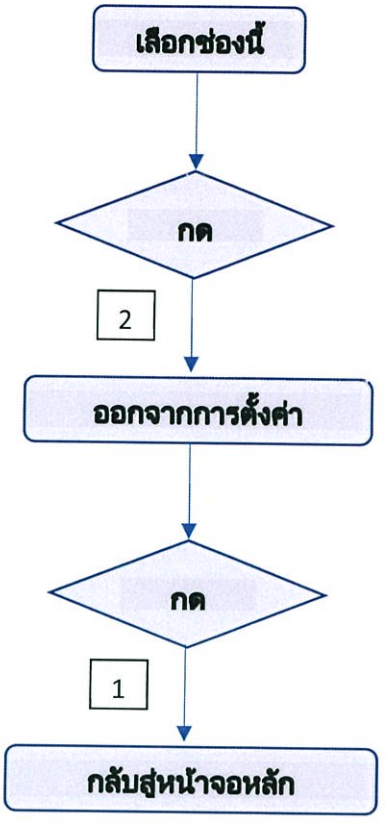
รูปที่ 3.13 แสดงแผนผังตั้งแต่เริ่มต้นไปยังเลือกมื้ออาหาร



รูปที่ 3.14 แสดงแผนผังหลังจากเลือกมื้ออาหารไปยังยืนยันการตั้งเวลา



รูปที่ 3.15 แสดงแผนผังหลังจากยืนยันการตั้งเวลาไปยังเลือกช่องที่กล่องยา



รูปที่ 3.16 แสดงแผนผังหลังจากเลือกช่องที่กล่องยาไปยังการกลับมาที่หน้าจอหลัก

3.3 Wemos D1 Mini

3.3.1 การเชื่อมต่อ Wemos D1 Mini กับสัญญาณอินเทอร์เน็ต

ทำการอัพโหลดชุดคำสั่งต่อไปนี้ลง Wemos D1 Mini

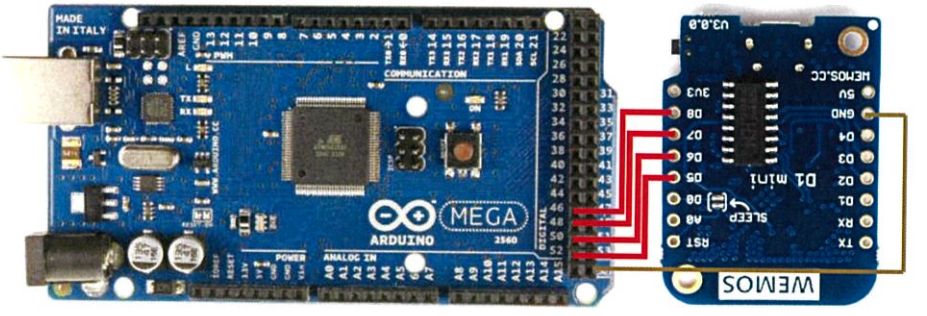
```
#include <ESP8266WiFi.h>
const char* ssid = "BMEKMITL";                  //ค่าของ SSID
const char* password = "KMITLBME";              //ค่าของ PASSWORD
void setup()    {
        Serial.begin(9600);
        WiFi.begin(ssid, password);                     //เชื่อมต่อกับ AP
        while (WiFi.status() != WL_CONNECTED)           //รอการเชื่อมต่อ
        {       delay(500);
                Serial.print(".");          }
```

```
        Serial.println("");
        Serial.println("WiFi connected");
        Serial.println("IP address: ");
        Serial.println(WiFi.localIP()); //แสดงหมายเลข IP
}
```

3.3.2 การรับค่าจาก Arduino Mega 2560 R3

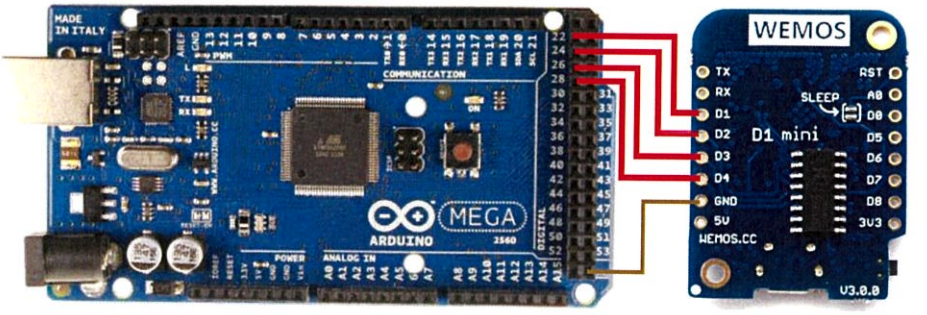
จากนั้นผู้วิจัยได้เชื่อมต่อ Arduino Mega กับ Wemos D1 Mini โดยการส่งข้อมูลแบบใช้สายโดยใช้รหัสแบบไบนารี่ ซึ่งจะต้องเชื่อมต่อกราวด์ของบอร์ดทั้งสองเข้าด้วยกัน โดยผู้วิจัยจะส่งข้อมูลสองส่วน ดังนี้

ส่วนที่ 1 เป็นการส่งข้อมูลการแจ้งเตือนจาก Arduino Mega ไปยัง Wemos D1 Mini ซึ่งจะนำข้อมูลเดียวกับที่ส่งไปให้ LED ส่งมายัง Wemos D1 Mini กล่าวคือ จะเชื่อมต่อสายไฟที่ขาดิจิตอลที่ 46, 48, 50,52 ของ Arduino Mega ต่อกับขาดิจิตอลที่ 5, 6, 7, 8 ของ Wemos D1 Mini ตามลำดับ ดังรูปที่ 3.17 หลังจากนั้น Wemos D1 Mini จะประมวลผลแล้วส่งค่าไปยังฐานข้อมูลออนไลน์ต่อไป



รูปที่ 3.17 แสดงการเชื่อมต่อเพื่อส่งข้อมูลการแจ้งเตือนจาก Arduino Mega ไปยัง Wemos D1 Mini

ส่วนที่ 2 เป็นการส่งข้อมูลสถานะการเปิดปิดฝาของกล่องยาจาก Arduino Mega ไปยัง Wemos D1 Mini โดยจะเชื่อมต่อสายไฟที่ขาดิจิตอลที่ 22, 24, 26, 28 ของ Arduino Mega ต่อกับขาดิจิตอลที่ 1, 2, 3, 4 ของ Wemos D1 Mini ตามลำดับ ดังรูปที่ 3.18 หลังจากนั้น Wemos D1 Mini จะประมวลผลแล้วส่งค่าไปยังฐานข้อมูลออนไลน์ต่อไป



รูปที่ 3.18 แสดงการเชื่อมต่อเพื่อส่งข้อมูลสถานะการเปิดปิดฝาของกล่องยาจาก Arduino Mega ไปยัง Wemos D1 Mini

3.3.3 การดึงเวลาสากลจากอินเทอร์เน็ตมาใช้

ผู้วิจัยต้องการนำเวลาสากลมาใช้เพื่อนำไปแสดงผลบนหน้าต่างประวัติการเปิดปิดฝากล่องยา หลังจากติดตั้ง ESP8266 บน Arduino IDE แล้ว ให้อัพโหลดชุดคำสั่งลงไปบนบอร์ดไมโครคอนโทรลเลอร์

```
#include <ESP8266WiFi.h>

#include <time.h>

const char* ssid = "ssid";                    //ใส่ชื่อ SSID Wifi

const char* password = "password";            //ใส่รหัสผ่าน

int timezone = 7 * 3600;                      //ตั้งค่า TimeZone ตามเวลาประเทศไทย

int dst = 0;                                  //กำหนดค่า Date Swing Time

void setup()    {

        Serial.begin(115200);

        Serial.setDebugOutput(true);

        WiFi.mode(WIFI_STA);                                    //เชื่อมต่อ Wifi

        WiFi.begin(ssid, password);
```

```
        Serial.println("\nConnecting to WiFi");
        while (WiFi.status() != WL_CONNECTED) {
                Serial.print(",");
                delay(1000);    }
        configTime(timezone, dst, "pool.ntp.org", "time.nist.gov");     //ดึงเวลาจาก Server
        Serial.println("\nWaiting for time");
        while (!time(nullptr)) {
                Serial.print(".");
                delay(1000);    }
        Serial.println("");          }
void loop()     {
        configTime(timezone, dst, "pool.ntp.org", "time.nist.gov"); //ดึงเวลาปัจจุบันจาก
Server
        time_t now = time(nullptr);
        struct tm* p_tm = localtime(&now);
        delay(1000);    }
```

จากนั้นนำค่า Parameter ต่างๆ มาใช้งานโดยใช้ชุดคำสั่ง

```
String daytime = String(p_tm->tm_mday)+"/"+String(p_tm-
>tm_mon+1)+"/"+String(p_tm->tm_year+1900)+"  "+String(p_tm-
>tm_hour)+":"+String(p_tm->tm_min)+":"+String(p_tm->tm_sec);
```

3.3.4 ชุดคำสั่งที่ใช้สำหรับส่งค่าไปยังฐานข้อมูลออนไลน์ไฟร์เบส

ใช้คำสั่ง Firebase.setString("A/B", "MESSAGE"); สำหรับการส่งข้อมูล String ไปยังไฟร์เบส จากชุดคำสั่งนี้จะได้รูปแบบบนไฟร์เบสจะเป็นดังนี้

A
↳ B: MESSAGE

ในที่นี้ผู้วิจัยจะให้ A เปรียบเสมือนเป็นบัญชีผู้ใช้ ส่วน B จะเป็นข้อมูลที่ผู้วิจัยต้องการจะแสดง Message ที่ข้อมูลใดซึ่งสามารถเพิ่มเข้าไปได้ ดังตัวอย่าง

```
Firebase.setString("A/B", "MESSAGE1");
Firebase.setString("A/C", "MESSAGE2");
```

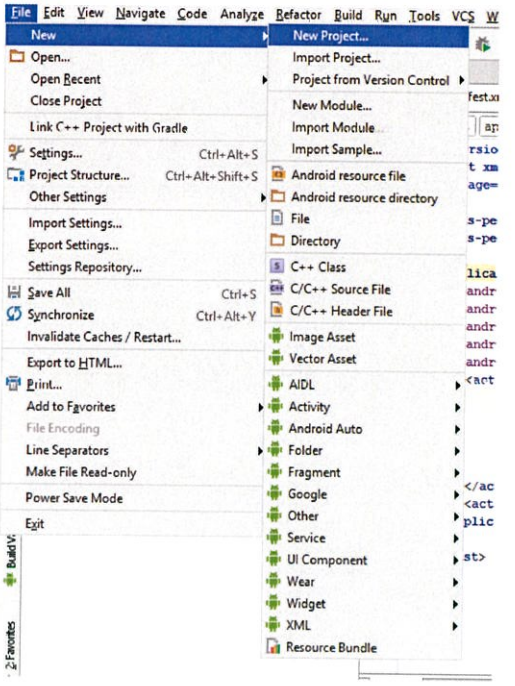
A
↳ B: MESSAGE1
↳ C: MESSAGE2

3.4 แอปพลิเคชันแอนดรอยด์

ทางผู้วิจัยได้ทำการออกแบบและสร้างแอปพลิเคชันแอนดรอยด์ที่ทำงานร่วมกับการใช้กล่องยา โดยมีการเชื่อมต่อแอปพลิเคชันกับกล่องยา เพื่อแจ้งเตือนเวลาการรับประทานยาได้ และนอกจากนั้นยังแสดงข้อมูลทั่วไปให้ผู้ป่วยสามารถเข้ามาดูได้อีกด้วย

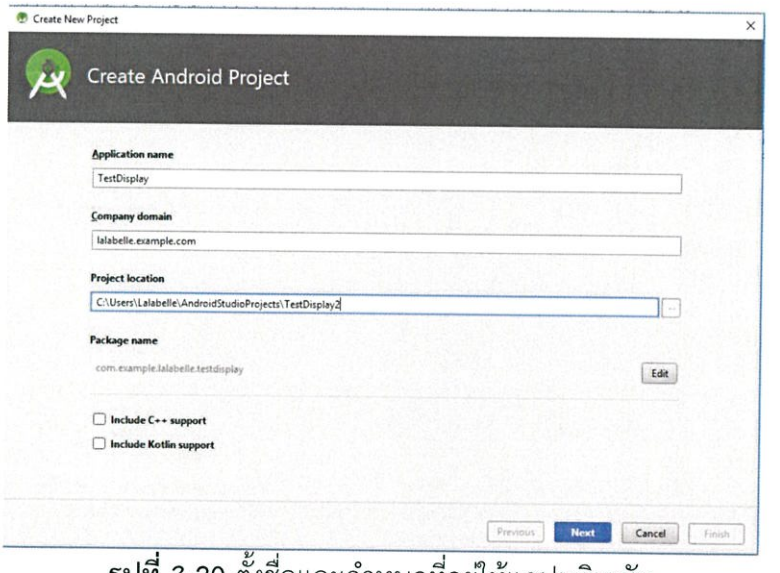
3.4.1 การสร้างโปรเจ็กต์ใหม่

การสร้างแอปพลิเคชัน ต้องเริ่มที่การสร้างโปรเจ็กต์แล้วพัฒนาต่อไปให้สมบูรณ์ จนกลายเป็นแอปพลิเคชันที่ใช้งานจริงได้ เริ่มต้นจากเปิดโปรแกรม Android Studio แล้วเริ่มสร้างชิ้นงานใหม่โดย File > New > New Project ดังรูปที่ 3.19



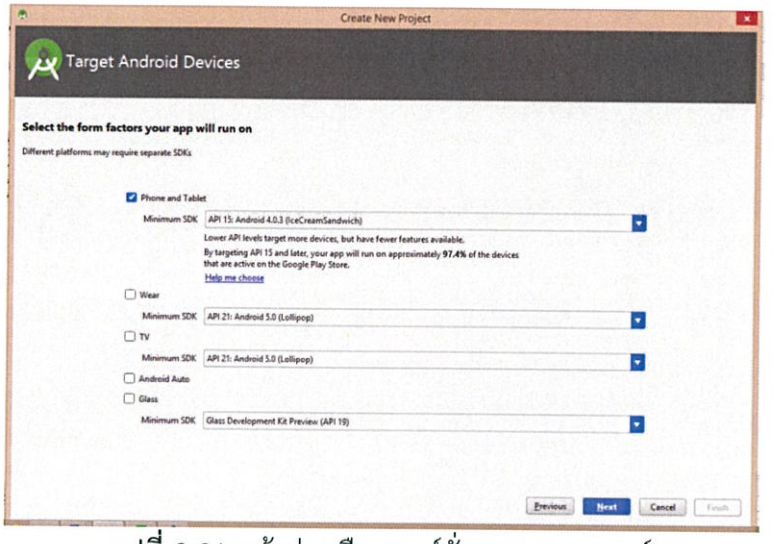
รูปที่ 3.19 การสร้างโปรเจ็กต์ใหม่บนโปรแกรม Android studio

จากนั้นทำการตั้งชื่อให้แอปพลิเคชัน ในที่นี้ใช้ชื่อ Multipledata การตั้งชื่อแอปพลิเคชัน Application name คือ ชื่อแอปพลิเคชันที่เราต้องการสร้าง ส่วน Company Domain คือ ชื่อเฉพาะที่เราต้องการระบุ เช่น ชื่อของบริษัท หรือชื่อผู้สร้างแอปพลิเคชัน Package name คือ ชื่อแพคเกจที่ใช้ในการเขียนแอปพลิเคชัน โดยจะเปลี่ยนตาม Company Domain และ Project location คือ path ของแอปพลิเคชันที่เราสร้าง เราต้องการเก็บไว้ที่ใด ดังรูปที่ 3.20 จากนั้นกด next



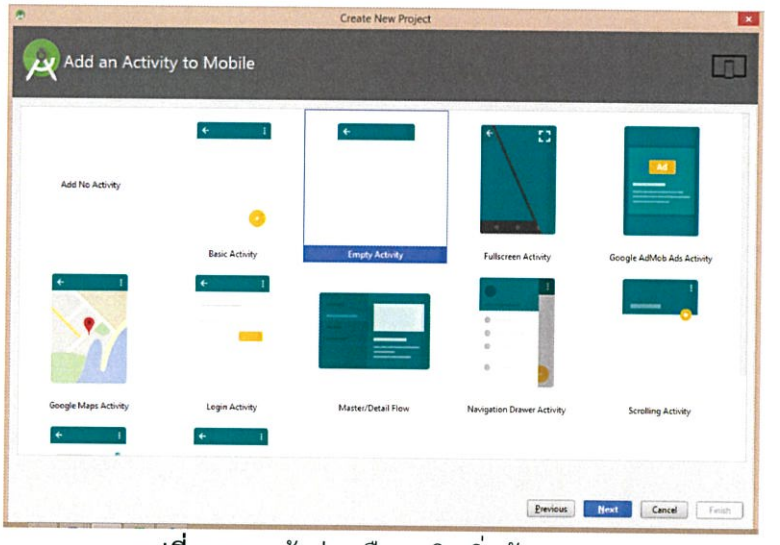
รูปที่ 3.20 ตั้งชื่อและกำหนดที่อยู่ให้แอปพลิเคชัน

จากนั้นเลือกเวอร์ชั่นของแอนดรอยด์ตามที่ต้องการให้ชิ้นงานนี้มีความสามารถประมวลผลได้ในขั้นต่ำสุด โดยในที่นี้เลือกใช้ API 15: Android 4.0.3 (IceCreamSandwich) ดังรูปที่ 3.21 แล้วคลิก Next



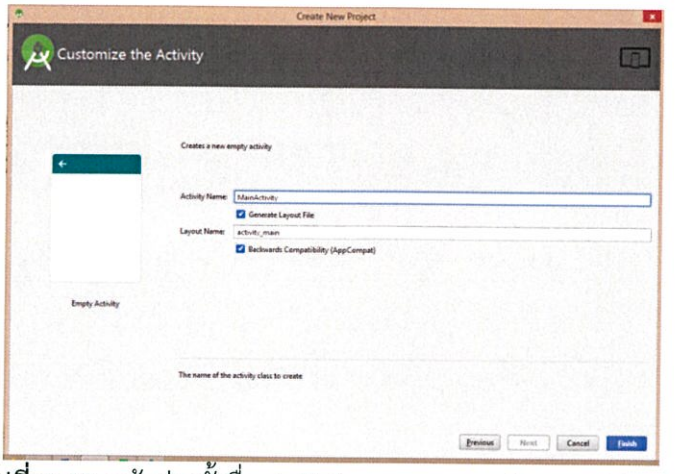
รูปที่ 3.21 หน้าต่างเลือกเวอร์ชั่นของแอนดรอยด์

ต่อมาเลือกชนิดเริ่มต้นของงาน ในที่นี้เลือก Empty Activity แล้วคลิก Next จากนั้นทำการตั้ง Activity name และ Layout Name ดังรูปที่ 3.22 แล้วคลิก Finish เป็นอันเสร็จสิ้นในขั้นตอนการสร้างชิ้นงานใหม่ในโปรแกรม Android Studio



รูปที่ 3.22 หน้าต่างเลือกชนิดเริ่มต้นของงาน

กำหนดชื่อไฟล์ Java (.class) และไฟล์ Layout (.xml) เริ่มต้น ดังรูปที่ 3.23



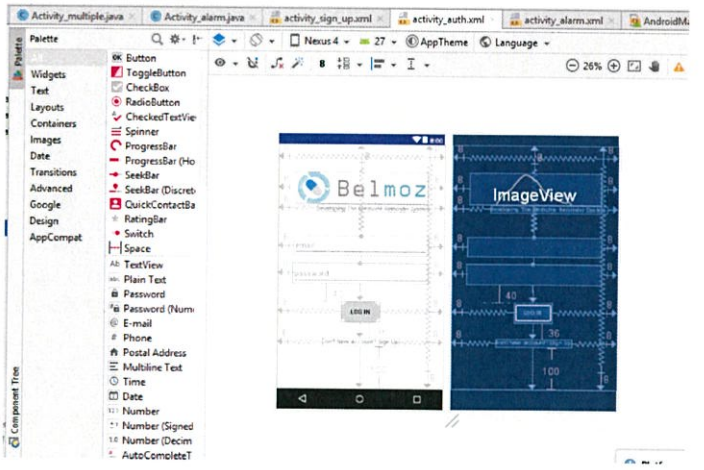
รูปที่ 3.23 หน้าต่างตั้งชื่อ Activity name และ Layout Name

หลังจากสร้างชิ้นงานใหม่ในโปรแกรม Android Studio เสร็จเรียบร้อย ทำการเขียนชุดคำสั่ง(Code) เพื่อออกแบบแอปพลิเคชันที่สามารถแสดงผลข้อมูลยาของผู้ป่วยได้บนสมาร์ทโฟน โดยข้อมูลที่แสดงนั้น มาจากฐานข้อมูลออนไลน์ไฟร์เบส นั่นเอง

3.4.2 การออกแบบแอปพลิเคชัน

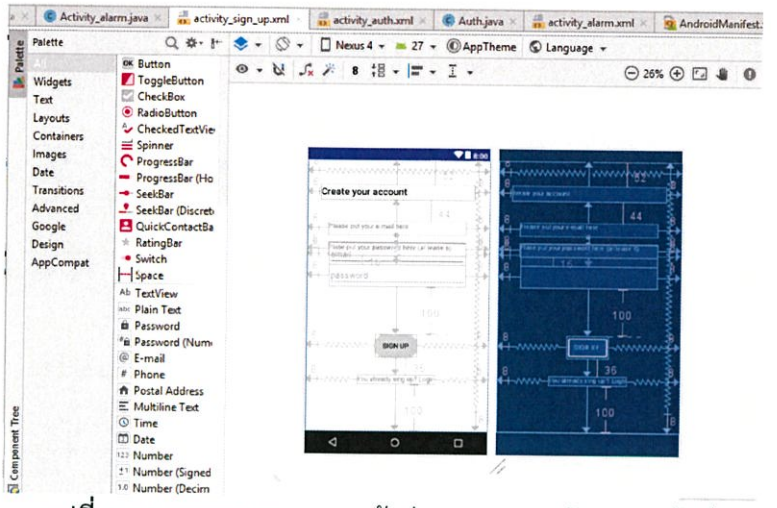
การออกแบบหน้าตา layout ที่ต้องการในแอปพลิเคชัน มีการเขียนชุดคำสั่งที่หน้าต่างจาวา โดยในโปรเจคนี้ได้ทั้งหมด 6 activities หลัก ดังนี้

1) activity_auth: ออกแบบหน้าต่างเข้าสู่ระบบ (log-in) ดังรูปที่ 3.24



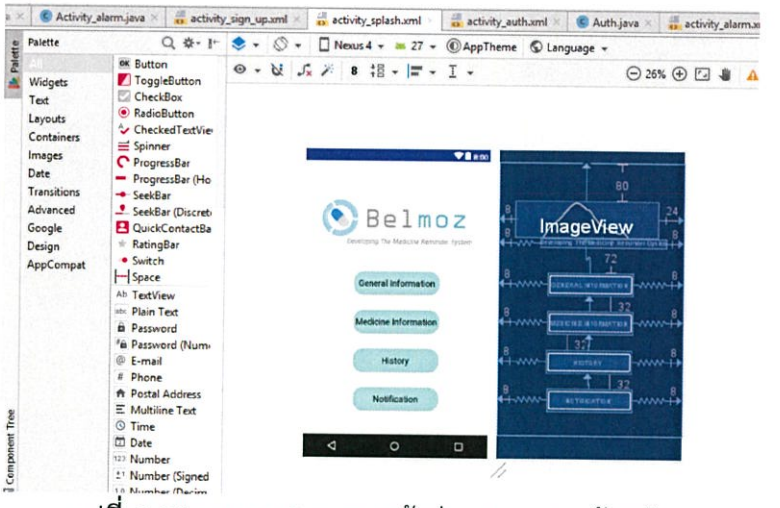
รูปที่ 3.24 มุมมอง Design หน้าต่าง layout เข้าสู่ระบบ

2) activity_sign_up: ออกแบบหน้าต่างสมัคร (Register) ดังรูปที่ 3.25



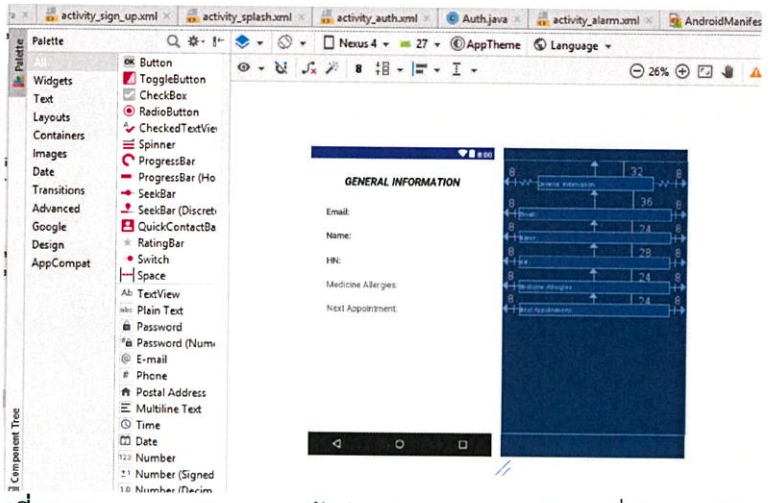
รูปที่ 3.25 มุมมอง Design หน้าต่าง layout สมัครแอคเค้าท์

3) activity_splash (Main Page): ออกแบบหน้าหลักของแอปพลิเคชัน จะมีปุ่มกดหลายปุ่มด้วยกัน ได้แก่ General Information, Medicine Information, History และ Notification ดังรูปที่ 3.26



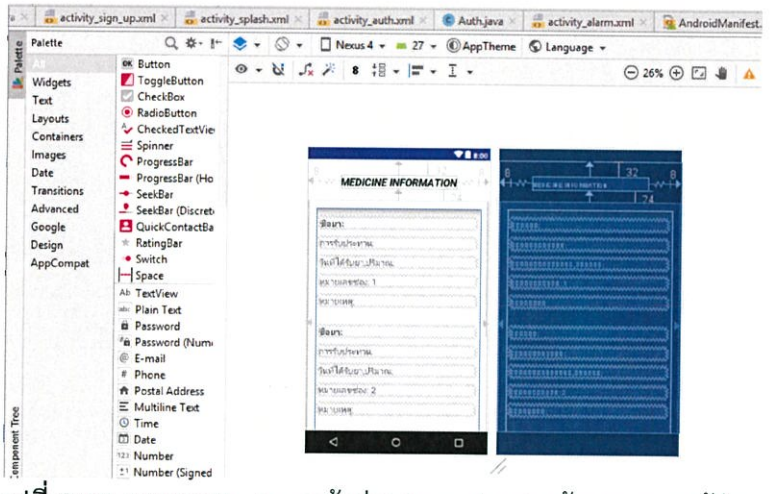
รูปที่ 3.26 มุมมอง Design หน้าต่าง layout หน้าหลัก

4) activity_general (General Information): ออกแบบหน้าต่างแสดงข้อมูลทั่วไปของผู้ป่วย ดังรูปที่ 3.27



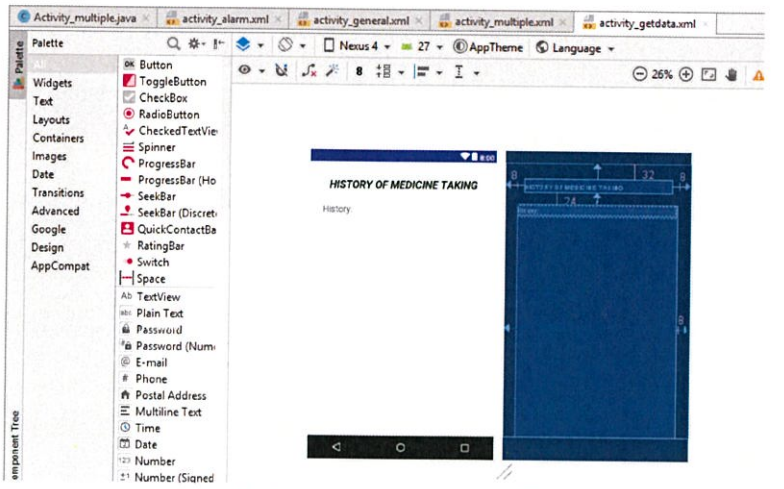
รูปที่ 3.27 มุมมอง Design หน้าต่าง layout แสดงข้อมูลทั่วไปของผู้ป่วย

5) activity_multiple (Medicine Information): ออกแบบหน้าต่างแสดงข้อมูลยาของผู้ป่วย ดังรูปที่ 3.28



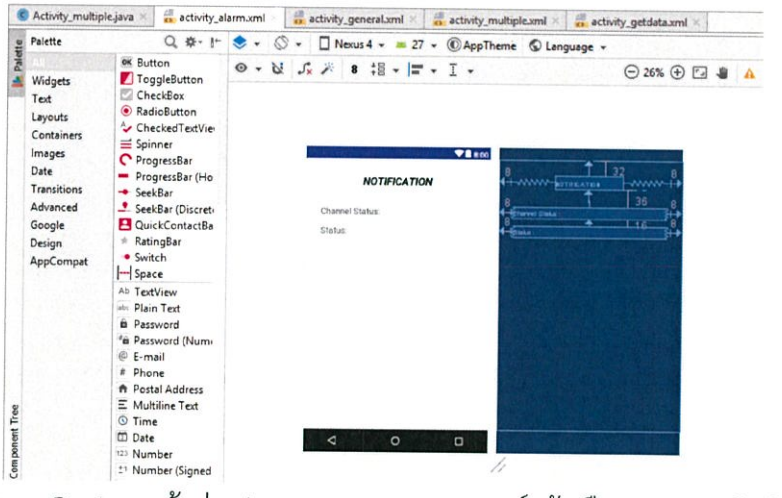
รูปที่ 3.28 มุมมอง Design หน้าต่าง layout แสดงข้อมูลยาของผู้ป่วย

6) activity_getdata (History): ออกแบบหน้าต่างแสดงประวัติการรับประทานยาของผู้ป่วย โดยเก็บค่าจากการเปิดปิดฝาของกล่องยา ดังรูปที่ 3.29



รูปที่ 3.29 มุมมอง Design หน้าต่าง layout แสดงประวัติการรับประทานยาของผู้ป่วย

7) activity_alarm (Notification): ออกแบบหน้าต่างแสดงสถานะของการแจ้งเตือนการรับประทานยา ดังรูปที่ 3.30 โดยค่า 0,1 และบอกสถานะการเปิดปิดฝาของกล่องยาว่าฝาใดกำลังเปิดอยู่หรือไม่มีฝาใดเปิดเลย



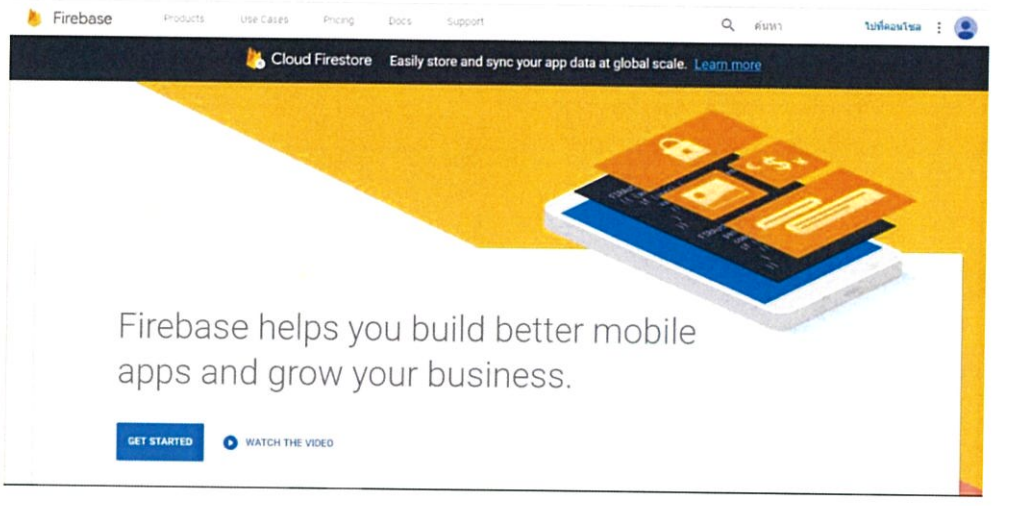
รูปที่ 3.30 มุมมอง Design หน้าต่าง layout แสดงสถานการณ์แจ้งเตือนและการเปิดปิดฝากล่องยา

3.5 ฐานข้อมูลออนไลน์

ในการศึกษาวิจัยระบบการติดตามการรับประทานยาของผู้ป่วยนี้ ผู้วิจัยได้เลือกใช้ฐานข้อมูลออนไลน์ไฟร์เบส (Firebase) โดยไฟร์เบสเป็นฐานข้อมูลออนไลน์ของกูเกิล (Google) ที่เปิดให้ใช้

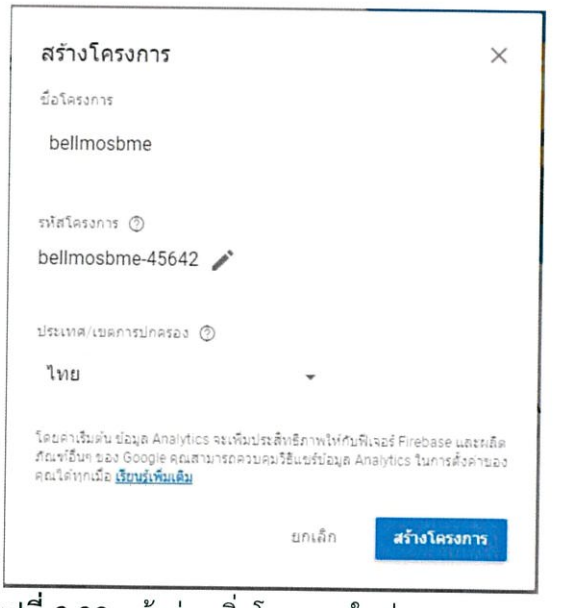
บริการโดยไม่เสียค่าใช้จ่าย โดยมีจุดเด่นคือ เรียลไทม์ และสามารถบันทึกข้อมูลไว้ได้ ซึ่งผู้วิจัยจะใช้ไฟร์เบสโดยการเข้าระบบด้วยจีเมล (Gmail)

3.5.1 การเริ่มต้นใช้งานฐานข้อมูลออนไลน์ไฟร์เบส



รูปที่ 3.31 หน้าเว็บฐานข้อมูลออนไลน์ Firebase (Database Realtime)

หลังจากนั้นผู้วิจัยจะสร้างโปรเจคบนไฟร์เบส โดยกดเพิ่มโครงการ แล้วกรอกชื่อ รหัสโครงการ ประเทศ ดังรูปที่ 3.32 แล้วกดสร้างโครงการ เมื่อสร้างเสร็จแล้วให้เก็บ Host Name ของโปรเจคนั้น ๆ ไว้ เพราะจะต้องใช้ในการเขียนชุดคำสั่ง ซึ่ง Host Name ไม่จำเป็นต้องมี https:// กับ / โดย Host Name สามารถดูได้ที่หน้าต่าง Database บนไฟร์เบส ผู้วิจัยได้ใช้รหัสโครงการว่า bellmosbme และ Host Name ของโปรเจคของผู้วิจัยคือ bellmosbme.firebaseio.com

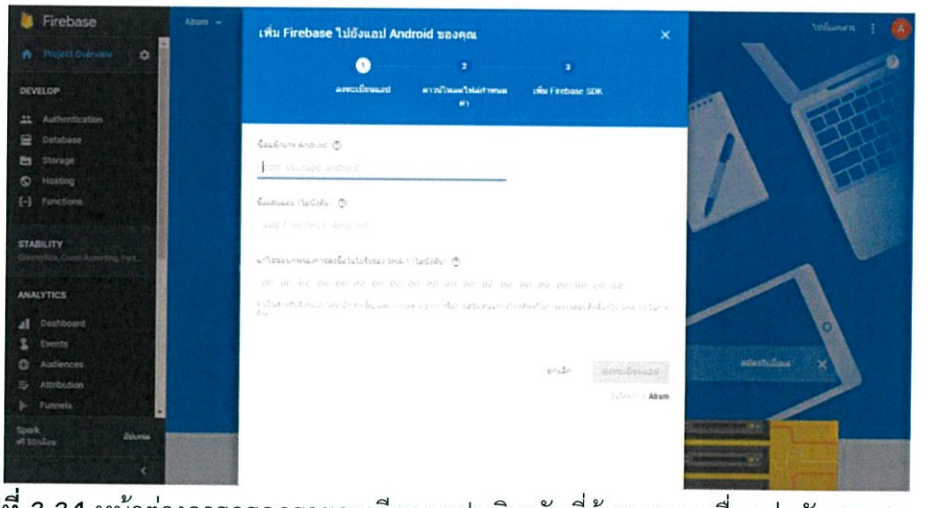


รูปที่ 3.32 หน้าต่างเพิ่มโครงการใหม่บน Firebase

จากนั้นทำการเลือกเพิ่มแอปพลิเคชันในการใช้งาน โดยจะเห็นได้ว่ามีตัวเลือกที่เป็นแอปพลิเคชันทั้งแบบ iOS, Android และเว็บ ดังรูปที่ 3.33 ในที่นี้เลือก “เพิ่ม Firebase ไปยังแอป Android ของคุณ”

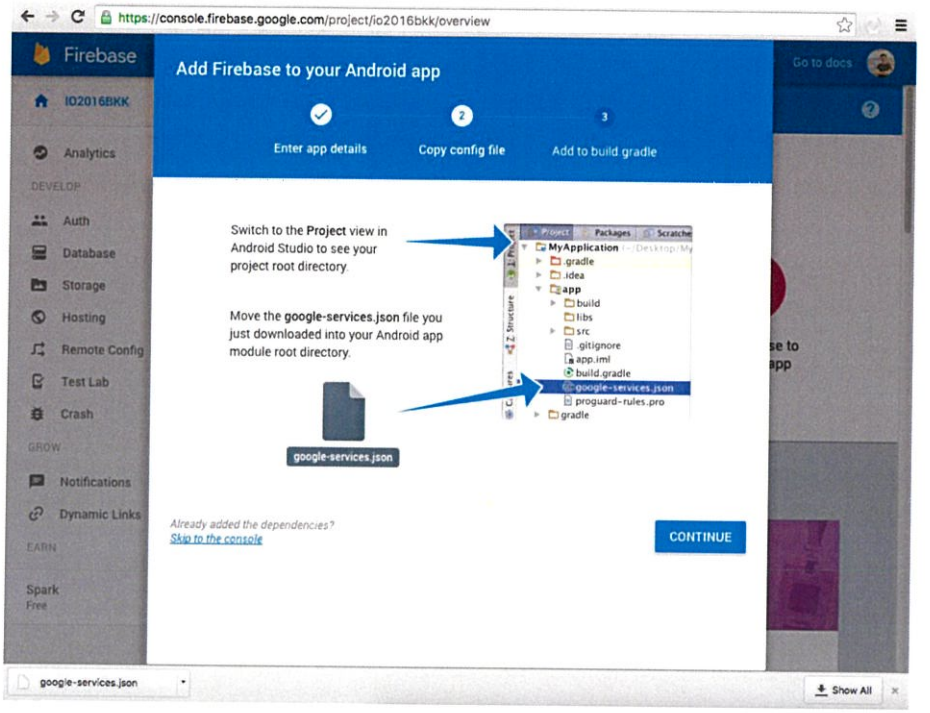
รูปที่ 3.33 หน้าเว็บเริ่มต้นการใช้งานทั้งแอปแบบ Android, iOS และเว็บ

การเชื่อมต่อโปรเจคแอนดรอยด์กับ Firebase นั้นทำได้โดยต้องกรอกรายละเอียดของแอปพลิเคชัน เริ่มจาก Package name ของแอปพลิเคชัน ส่วน Debug signing certificate SHA-1 จะใส่หรือไม่ก็ได้ (Optional) ดังรูปที่ 3.34



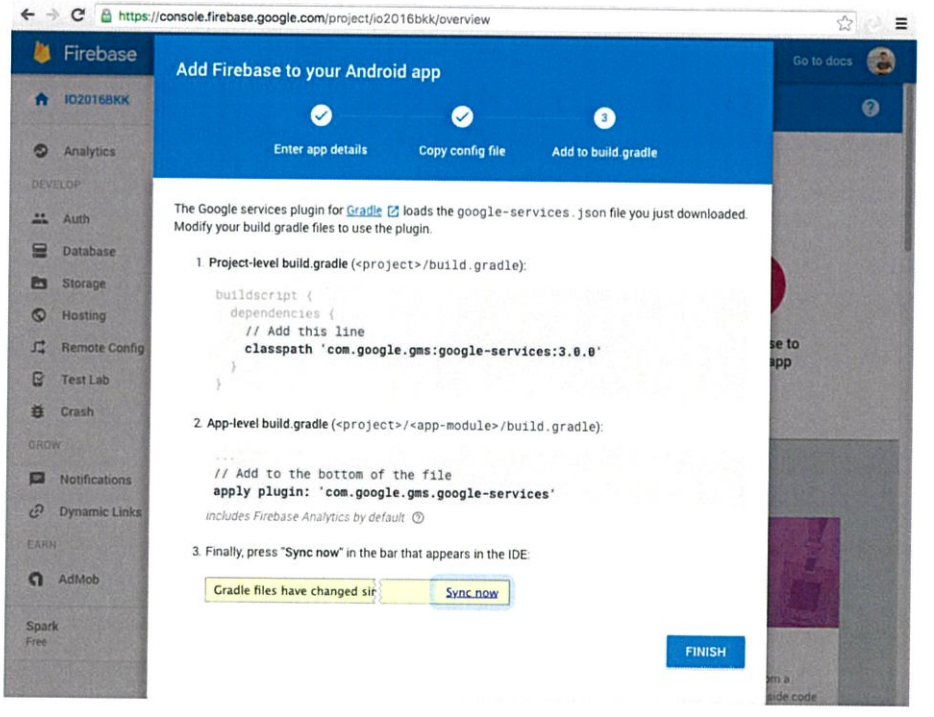
รูปที่ 3.34 หน้าต่างการกรอกรายละเอียดแอปพลิเคชันที่ต้องการจะเชื่อมต่อกับ Firebase

หลังจากกรอกชื่อ Package name แล้วกดปุ่ม ADD APP จะไปสู่ขั้นตอนที่ 2 ซึ่งในขั้นตอนนี้เราจะได้ไฟล์ google-services.json มาให้เรา copy ไฟล์ไปวางในโฟลเดอร์ app ของ Project (เปลี่ยนวิวเป็นแบบ Project เพื่อให้เห็น Directory ทั้งหมด) ใน Android Studio ดังรูปที่ 3.35



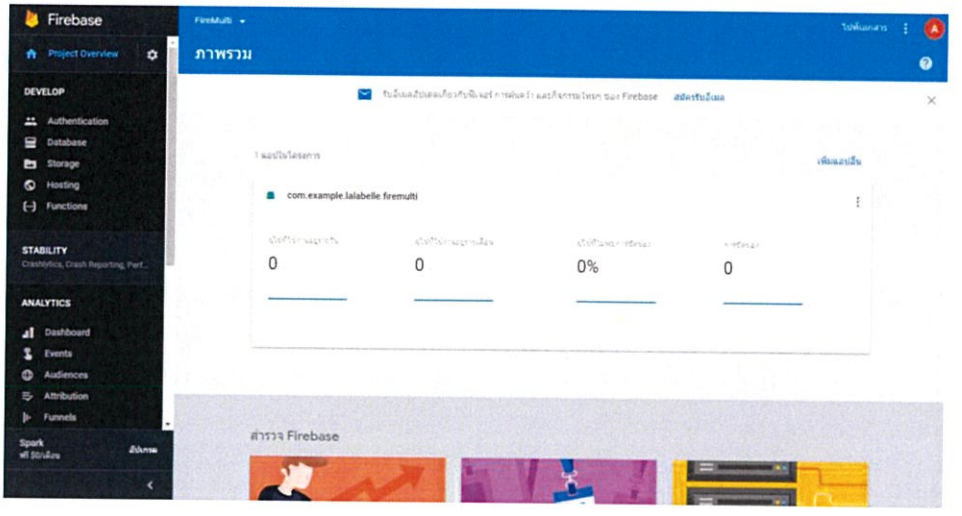
รูปที่ 3.35 ขั้นตอนการโหลดไฟล์ google-services.json

เสร็จแล้วกด CONTINUE ก็จะมาถึง ขั้นตอนที่ 3 ขั้นตอนนี้จะเพิ่ม code เพื่อใช้งาน Google services plugin ดังรูปที่ 3.36



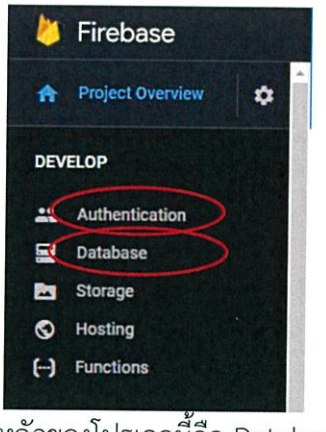
รูปที่ 3.36 หน้าต่างขั้นตอนที่ 3 ให้นำชุดคำสั่งไปใส่ใน Android Studio

สำหรับหน้าขั้นตอนที่ 3 ก็กดปุ่ม FINISH ไป แอปพลิเคชันของเราก็จะถูกเพิ่มใน Firebase เรียบร้อย ดังรูปที่ 3.37



รูปที่ 3.37 แอปพลิเคชันถูกเพิ่มลงใน Firebase เรียบร้อยแล้ว

โดยทางซ้ายมือจะมีแถบการทำงานให้เลือก โดย 2 ส่วนหลักของแอปพลิเคชันนี้คือ Database และ Authenication ดังรูปที่ 3.38

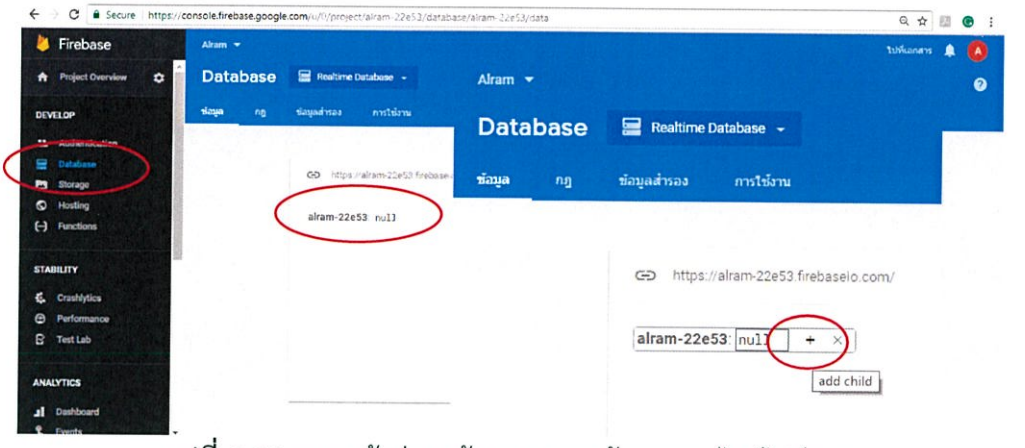


รูปที่ 3.38 แถบเครื่องมือหลักของโปรเจคนี้คือ Database และ Authenication

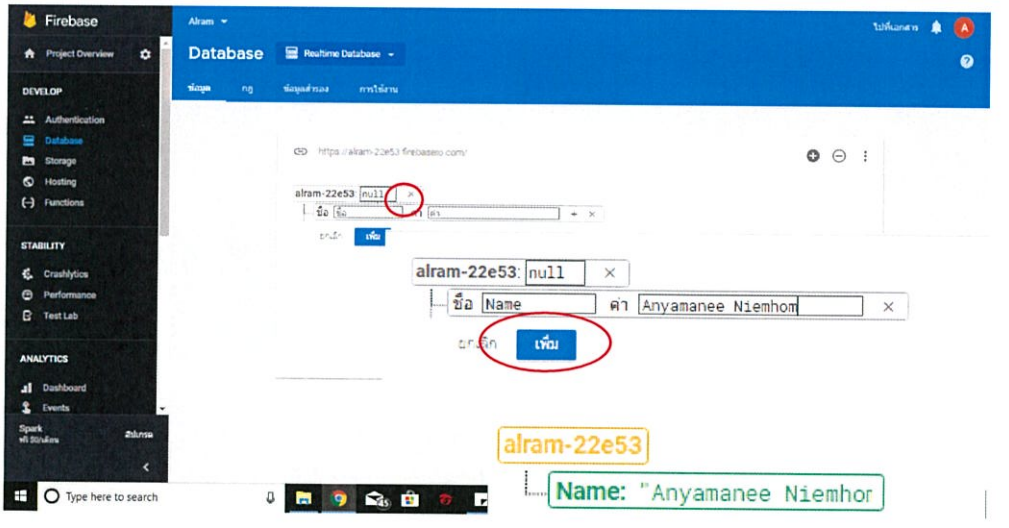
3.5.2 การสร้างฐานข้อมูลบนฐานข้อมูลออนไลน์ไฟร์เบส

เลือกที่แถบการทำงานด้านซ้ายมือไปที่ Database จะเริ่มจากไม่มีข้อมูล (null) ทำการเพิ่มข้อมูลโดยเลือกที่ + จะมีการเพิ่มชื่อ และค่าของข้อมูล ในที่นี้ใส่ชื่อ Name ค่า Anyamanee

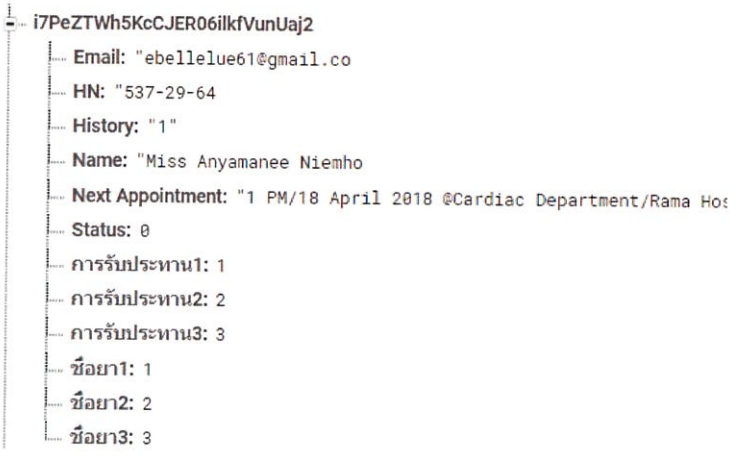
Niemhom ดังรูปที่ 3.39 จากนั้นกดเพิ่ม ดังรูปที่ 3.40 ข้อมูลจะถูกบันทึกและเก็บไว้ในหน้าต่างนี้ ทำการเพิ่มข้อมูลที่ต้องการจนครบ ดังรูปที่ 3.41



รูปที่ 3.39 แสดงเข้าสู่ฐานข้อมูลบนฐานข้อมูลออนไลน์ไฟร์เบส



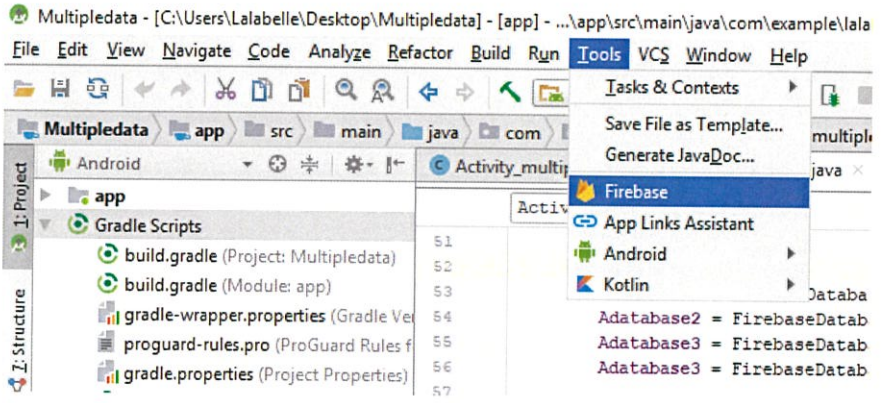
รูปที่ 3.40 แสดงการเพิ่มข้อมูลบนฐานข้อมูลออนไลน์ไฟร์เบส



รูปที่ 3.41 แสดงข้อมูลที่หลากหลายบนฐานข้อมูลออนไลน์ไฟร์เบส

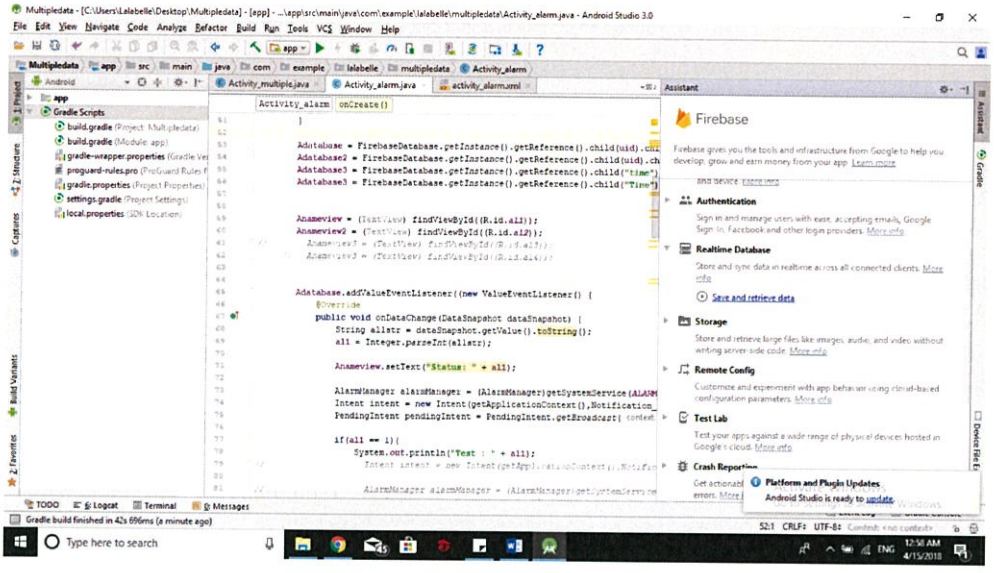
3.5.3 การเชื่อมต่อฐานข้อมูลออนไลน์ไฟร์เบสกับแอปพลิเคชัน

ในการอัพเดตใหม่ล่าสุด ทำให้ขั้นตอนการเชื่อมต่อไฟร์เบสกับโปรแกรม Android Studio ได้ง่ายขึ้น โดยสามารถเรียกเครื่องมือที่เป็นตัวช่วยในการเชื่อมต่อ (Assistant) ดังรูปที่ 3.42 แล้วเครื่องนี้จะทำการเพิ่มชุดคำสั่งเข้าไปในไฟล์ให้เองเพื่อทำการเชื่อมต่อ



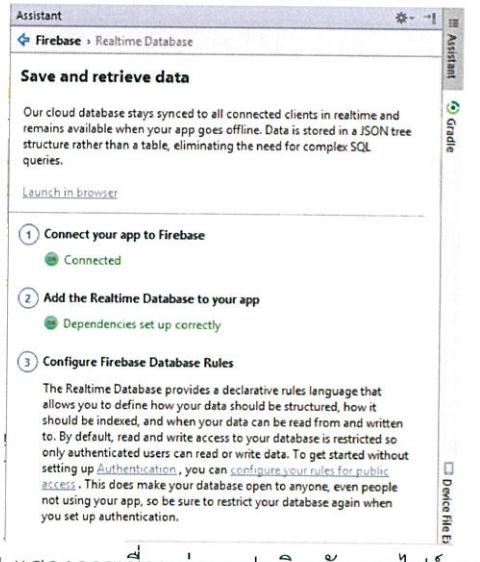
รูปที่ 3.42 แสดงแถบเครื่องมือช่วย Firebase

จะได้หน้าต่างตัวช่วย (Assistant) ขึ้นมา ดังรูปที่ 3.43



รูปที่ 3.43 แสดงหน้าต่างตัวช่วย Firebase

เลือก save and retrieve data แล้วกดเชื่อมต่อข้อมูล ทำตามลำดับที่กำหนดไว้ให้ ดังรูปที่ 3.44 ก็เป็นอันเสร็จสิ้น

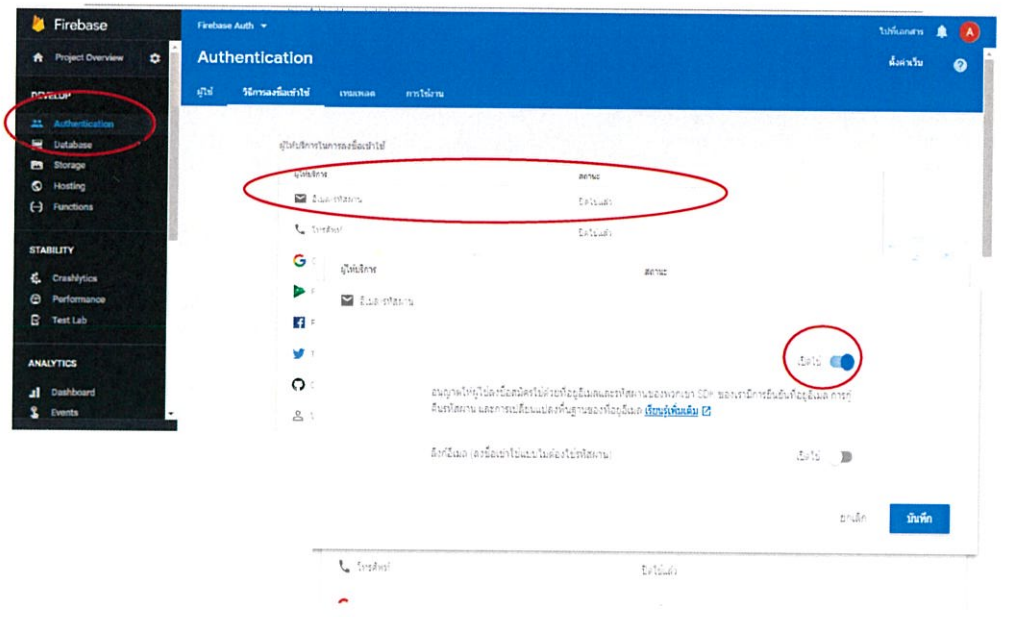


รูปที่ 3.44 แสดงการเชื่อมต่อแอปพลิเคชันและไฟร์เบสเสร็จสิ้น

3.5.4 การสร้างระบบการสร้างบัญชีผู้ใช้และการเข้าสู่ระบบ

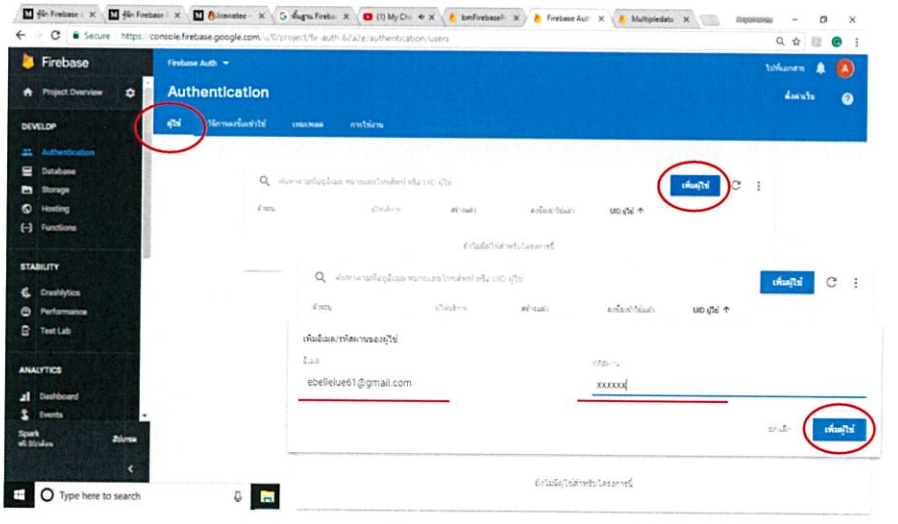
3.5.4.1 การสร้างบัญชีผู้ใช้

ใน Firebase Authentication นั้นสามารถจะสร้างจากการ sign-in ด้วย social network หรือสร้างบัญชีด้วย email และ password ของผู้ใช้เอง เริ่มโดยเข้าไปที่ Firebase Console เลือกเมนู Develop ทางด้านซ้าย แล้วเลือกแถบชื่อ Authentication จากนั้นให้กดเลือกที่แถวของ อีเมล/รหัสผ่าน เพื่อที่จะไปเปิดการใช้งาน ดังรูปที่ 3.45



รูปที่ 3.45 แสดงการเปิดใช้งานอีเมล/รหัสผ่าน

ต่อไปคือการสร้างบัญชีด้วย email และ password จะมีด้วยกัน 2 วิธี โดยวิธีแรก สร้างจาก Firebase Console โดยคลิกที่แถบแถวบนอันแรก ชื่อ ผู้ใช้ จากนั้นกดปุ่มเพิ่มผู้ใช้ แล้วกรอก email และ password (ขั้นต่ำ 6 characters) ดังรูปที่ 3.46



รูปที่ 3.47 แสดงการเพิ่มบัญชีอีเมล/รหัสผ่าน บนไฟร์เบส

วิธีที่สอง คือสร้างบนแอปพลิเคชัน โดยสร้าง layout ที่ android studio เริ่มด้วยการประกาศตัวแปร FirebaseAuth และ รับค่า Instance ก่อน ดังนี้

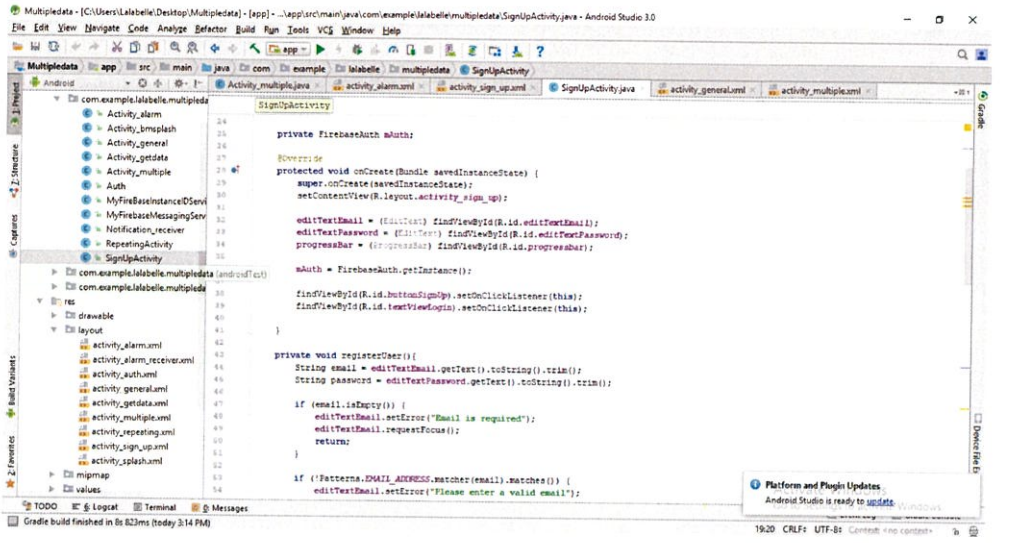
```
private FirebaseAuth mAuth;

// ...

mAuth = FirebaseAuth.getInstance();
```

ต่อไปสร้าง AuthStateListener เพื่อรอรับข้อมูล user ที่ได้จากการ sign-up และ sign-in โดยสามารถรับข้อมูล user ได้จาก getCurrentUser() หากค่าของ getCurrentUser เป็น null แปลว่า user ยังไม่ได้ sign-in นั่นเอง

```
private FirebaseAuth.AuthStateListener mAuthListener;


// ...

@Override

protected void onCreate(Bundle savedInstanceState) {

  // ...

  mAuthListener = new FirebaseAuth.AuthStateListener() {

    @Override

    public void onAuthStateChanged(FirebaseAuth firebaseAuth) {

      FirebaseUser user = firebaseAuth.getCurrentUser();

      if (user != null) {

        // User is signed in

      } else {

        // User is signed out

      }
```

```
            // ...
        }
    };
    // ...
}
@Override
public void onStart() {
    super.onStart();
    mAuth.addAuthStateListener(mAuthListener);
}

@Override
public void onStop() {
    super.onStop();
    if (mAuthListener != null) {
        mAuth.removeAuthStateListener(mAuthListener);
    }
}
```

ซึ่งสามารถดูหน้าต่าง SignUpActivity ที่เขียนชุดคำสั่งในการเพิ่มบัญชีผู้ใช้ได้ดังรูปที่ 3.47



รูปที่ 3.47 แสดงหน้าต่าง SignUpActivity ที่เขียนชุดคำสั่งในการเพิ่มบัญชีผู้ใช้

3.5.4.2 การเข้าสู่ระบบ

โดยขั้นตอนก็คล้ายกับการเพิ่มบัญชีผู้ใช้ จากนั้นก็ทำตาม code ด้านล่างด้วยการส่ง email และ password เข้าไป และผ่านการตรวจสอบโดยเงื่อนไข

```
mAuth.signInWithEmailAndPassword(email, password)
.addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete( Task<AuthResult> task) {
           if (!task.isSuccessful()) {
              Log.w(TAG, "signInWithEmail", task.getException().getMessage());
              Toast.makeText(EmailPasswordActivity.this, "Authentication failed.",
Toast.LENGTH_SHORT).show();
           }
        // ...
        }
    });
```

3.5.5 การนำข้อมูลบนฐานข้อมูลออนไลน์ไปแสดงบนแอปพลิเคชัน

ในส่วนของข้อมูลที่ส่งระหว่างฐานข้อมูลออนไลน์กับแอปพลิเคชัน มีจุดประสงค์ให้แอปพลิเคชันทำการแสดงข้อมูลที่มาจากฐานข้อมูลออนไลน์ไฟร์เบส ที่เป็นเรียลไทม์ โดยมีการออกแบบชุดคำสั่งบน Android Studio มากมายดังที่กล่าวข้างต้น ต้องเริ่มจากการเขียนข้อมูล (Write) เริ่มด้วยการประกาศตัวแปร DatabaseReference รับค่า Instance และอ้างถึง path ที่เราต้องการใน database

```
DatabaseReference mRootRef = FirebaseDatabase.getInstance().getReference();
```

จากนั้นก็อ้างอิงไปที่ path ที่เราต้องการจะจัดการข้อมูล ตัวอย่างคือ users และ messages

```
DatabaseReference mUsersRef = mRootRef.child("users");
DatabaseReference mMessagesRef = mRootRef.child("messages");
```

ทางผู้วิจัยเลือกรูปแบบในการเขียนข้อมูลแบบ setValue() ซึ่งเป็นการ write หรือ update ข้อมูล ไปยัง path ที่เราอ้างถึงได้ เช่น users/<user-id>/<username> จากนั้นเราสามารถดูผลลัพธ์แบบ realtime ได้ที่ Firebase Console โดยไปที่เมนู Database แล้วเลือก tab แรก คือ Data เราก็จะเห็นข้อมูลทั้งหมด

หลังจากเขียนข้อมูลตามข้างต้นแล้วต้องทำการอ่านข้อมูล (Read) เริ่มด้วยการประกาศตัวแปร DatabaseReference รับค่า Instance และอ้างถึง path ที่เราต้องการใน database

```
DatabaseReference mRootRef = FirebaseDatabase.getInstance().getReference();
```

ทางผู้วิจัยเลือกรูปแบบในการอ่านข้อมูลแบบ ValueEventListener จะอ่านข้อมูลตั้งแต่เริ่ม และ จะอ่านข้อมูลทุกครั้งที่มีการเปลี่ยนแปลงของข้อมูลทั้งหมด

```
mRootRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        String value = dataSnapshot.getValue(String.class);
        mTextView.setText(value);
    }
```

```
    @Override
    public void onCancelled(DatabaseError error) {
        mTextView.setText("Failed: " + databaseError.getMessage());
    }
});
```

และเนื่องจากเรื่องความปลอดภัยของการเข้าถึงข้อมูลของผู้ป่วย จึงทำการกรองการเข้าถึงข้อมูลจาก unique key ที่มาจากการเข้าสู่ระบบที่เฉพาะเจาะจง ทำให้แต่ละแอคเคาท์สามารถเข้าไปเห็นได้แค่ข้อมูลของตนเอง และแอคเคาท์อื่นไม่สามารถเข้ามาถึงข้อมูลตนได้ โดยมีการสร้างเงื่อนไขของ unique key ดังนี้

```
FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
if (user != null) {
    uid = user.getUid();
}
```

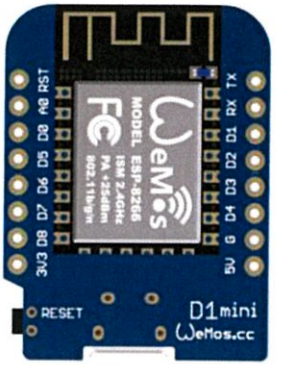
และเพิ่มการเลือกอ่านข้อมูลด้วยคำสั่ง

```
.child(uid)
```

3.5.6 การส่งข้อมูลจาก NodeMCU ไปยังฐานข้อมูลด้วยระบบการส่งข้อมูลแบบไร้สาย

ในส่วนของข้อมูลที่ส่งระหว่างฐานข้อมูลออนไลน์กับวงจรอิเล็กทรอนิกส์นั้น ผู้วิจัยต้องทำการเชื่อมต่อข้อมูลทั้งสองก่อน โดยเข้าไปที่ Project Overview -> การตั้งค่าบริการ -> บัญชีบริการ -> ข้อมูลลับฐานข้อมูล แล้วกดให้แสดงรหัสลับ (Secret Key) ซึ่ง Secret Key กับ Host Name นั้น ผู้วิจัยจะนำไปใช้ในชุดคำสั่ง เพื่อเชื่อมต่อข้อมูลระหว่างวงจรอิเล็กทรอนิกส์กับฐานข้อมูลออนไลน์ไฟร์เบส

NodeMCU เป็นอุปกรณ์ที่ผู้วิจัยเลือกใช้ในการส่งข้อมูลแบบไร้สายไปยังฐานข้อมูล และผู้วิจัยได้เลือกใช้ไฟร์เบส (Firebase) ซึ่งเป็นฐานข้อมูลที่เปิดให้ใช้ฟรีของกูเกิล (Google) เป็นฐานข้อมูล ส่วนของการจัดเก็บข้อมูล ซึ่ง NodeMCU ที่ใช้ผู้วิจัยเลือกใช้คือ Wemos D1 Mini



รูปที่ 3.48 แสดง NodeMCU Wemos D1 Mini

โดยที่ Wemos D1 Mini จะมีขา GPIO ทั้งหมด 8 ขา คือ ขา D1 ถึงขา D8 และมีขาสำหรับจ่ายไฟ 2 ขา คือ ขา 3V3 เป็นขาที่จ่ายไฟ 3.3 โวลต์ และ ขา 5V สำหรับจ่ายไฟ 5 โวลต์ ส่วนขา G คือ ขาที่ใช้ต่อกับกราวด์ของวงจร ดังรูปที่ 3.48

จากนั้นผู้วิจัยจะนำเอาท์พุททั้ง 4 เอาท์พุทของวงจรเข้ารหัสมาต่อเข้าที่ขา D5 D6 D7 และ D8 โดยต่อเข้ากับเอาท์พุท A B C และ D ตามลำดับ ส่วนไฟของวงจรทั้งหมดจะต่อเข้ากับขา 5V ของ Wemos D1 Mini และกราวด์ของวงจร จะต่อเข้ากับ ขา G ของ Wemos D1 Mini

หลังจากนั้นอัพโหลดคำสั่งเข้า Wemos D1 Mini โดยใช้โปรแกรมอาร์ดูโน (Arduino) โดยที่ก่อนการอัพโหลดจะต้องดาวน์โหลดและติดตั้งไลบราลี (Library) สำหรับไฟร์เบส (Firebase) บนโปรแกรมอาร์ดูโน ทำการตั้งค่าในชุดคำสั่งต่อไปนี้

1. ตั้งค่าการเชื่อมต่อกับฐานข้อมูล

ตั้งค่า Host name ของโปรเจค Firebase ที่ต้องการส่งข้อมูลไปจัดเก็บ

```
#define FIREBASE_HOST "bellmosbme.firebaseio.com"
```

ตั้งค่า Secret Key ของโปรเจค Firebase ที่ต้องการส่งข้อมูลไปจัดเก็บ

```
#define FIREBASE_AUTH "vunSDYepAnmTbSo50j3zE1goRRjNXoXw9zPiPplX"
```

2. ตั้งค่าการเชื่อมต่ออินเตอร์เน็ต

ตั้งค่าชื่อของสัญญาณไร้สายที่ต้องการเชื่อมต่อ

```
#define WIFI_SSID "BMEKMITL"
```

ตั้งค่ารหัสเพื่อเชื่อมต่อกับสัญญาณไร้สาย

```
#define WIFI_PASSWORD "KMITLBME"
```

โดยข้อมูลที่ส่งไปนั้น มาจากการตรวจจับจากตัวไมโครคอนโทรลเลอร์ส่งไปยังฐานข้อมูลออนไลน์และไปแสดงบนแอปพลิเคชัน สามารถแบ่งเป็น 2 ประเภท ได้แก่ สถานะแสดงการแจ้งเตือนและการตรวจจับการเปิดปิดฝาของกล่องยา ในสถานะแสดงการแจ้งเตือน จะทำการส่งค่า 0 และ 1 โดยค่า 0 แสดงว่ายังไม่มีการแจ้งเตือนการรับประทานยา และ 1 แสดงว่ามีการแจ้งเตือนการรับประทานยา และในส่วนของการตรวจจับการเปิดปิดฝา จะส่งค่าไปแสดง Channel Status ยกตัวอย่าง ถ้าฝาที่ช่อง 1 เปิดจะแสดง Channel 1 is opened และ ถ้าทุกฝาปิดหมดจะแสดง All channel is closed

3.5.7 การสร้างการแจ้งเตือนบนแอปพลิเคชัน

ทำการสร้างเงื่อนไขการรับค่าไร้สายจากไมโครคอนโทรลเลอร์ โดยค่า 1 ให้มีการ alarm และ 0 ไม่ต้อง alarm ดังคำสั่ง ต่อไปนี้

```
AlarmManager alarmManager = (AlarmManager)getSystemService(ALARM_SERVICE);
Intent intent = new
Intent(getApplicationContext(),Notification_receiver.class);
PendingIntent pendingIntent =
PendingIntent.getBroadcast(Activity_alarm.this, 0, intent,
PendingIntent.FLAG_CANCEL_CURRENT);

if(all == 1){
             System.out.println("Test : " + all);

alarmManager.setRepeating(AlarmManager.RTC_WAKEUP,System.currentTimeMillis()
, (5 * 1000), pendingIntent);
                }
   else {
             alarmManager.cancel(pendingIntent);
                }

                System.out.println("Test out: " + all);
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        }));
```

และในคลาสของ Notification_receiver จะกำหนดรายละเอียดของการแจ้งเตือน เช่น ข้อความแจ้งเตือน เสียง โลโก้ และเวลา เป็นต้น

บทที่ 4

ผลการทดลองและอภิปรายผล

การพัฒนาระบบการแจ้งเตือนการรับประทานยาของผู้ป่วยสูงอายุด้วยการใช้กล่องยาควบคุมด้วยไมโครคอนโทรลเลอร์ โดยมีการส่งข้อมูลผ่านฐานข้อมูลออนไลน์เพื่อการแจ้งเตือนบนสมาร์ทโฟนแอนดรอยด์ ระบบมีการออกแบบให้เหมาะสมกับการใช้งาน มีการทำการทดสอบความสามารถของระบบหลายด้าน ได้แก่ ความสามารถในการตรวจจับการเปิดฝาของกล่องยาโดยไมโครคอนโทรลเลอร์และกระบวนการแจ้งเตือนของระบบทั้งในกล่องยาและแอพพลิเคชั่น ผลการทดลองจะแสดงระดับของประสิทธิภาพและความแม่นยำของระบบดังกล่าว

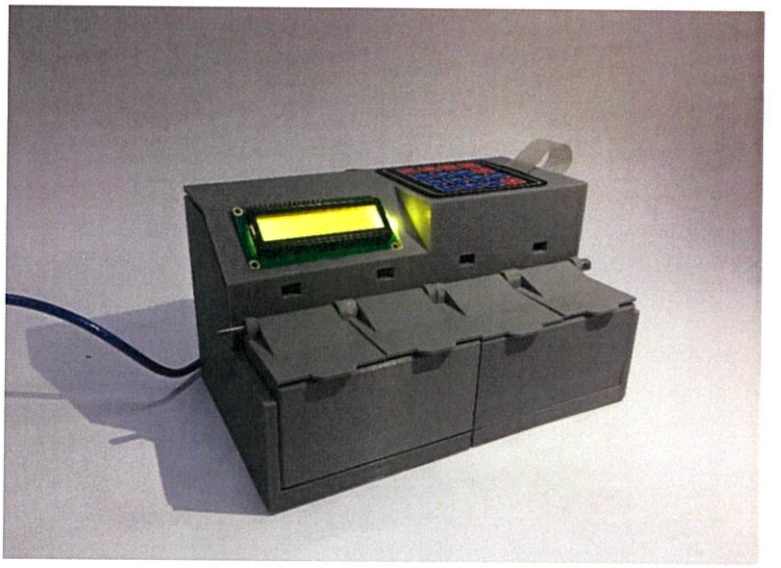
4.1 ภาพรวมของระบบ

ระบบการแจ้งเตือนการรับประทานยาของผู้ป่วยสูงอายุถูกแบ่งเป็น 3 ส่วนหลัก ได้แก่

1) กล่องยาพร้อมไมโครคอนโทรลเลอร์
2) แอพพลิเคชั่นแอนดรอยด์
3) ฐานข้อมูลออนไลน์ไฟร์เบส

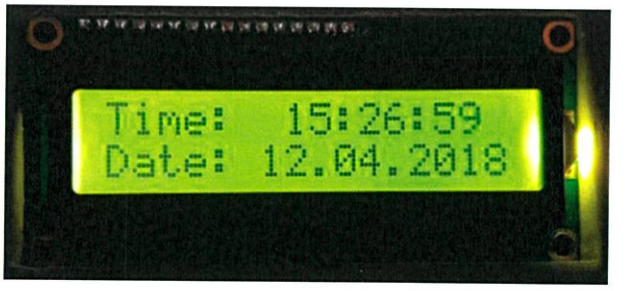
4.1.1 กล่องยาพร้อมไมโครคอนโทรลเลอร์

ส่วนของกล่องยาที่ปริ้นมาจากเครื่องปริ้นท์สามมิติจะมีช่องใส่ยา 4 ช่อง ส่วนช่องข้างหลังจะเก็บไมโครคอนโทรลเลอร์ไว้ โดยบนกล่องยามีหน้าจอแอลซีดีแสดงเมื่อทำการตั้งค่าเวลาการแจ้งเตือนด้วยแพดปุ่มกด ผลที่ได้แสดงดังรูปที่ 4.1 ถึงรูปที่ 4.7

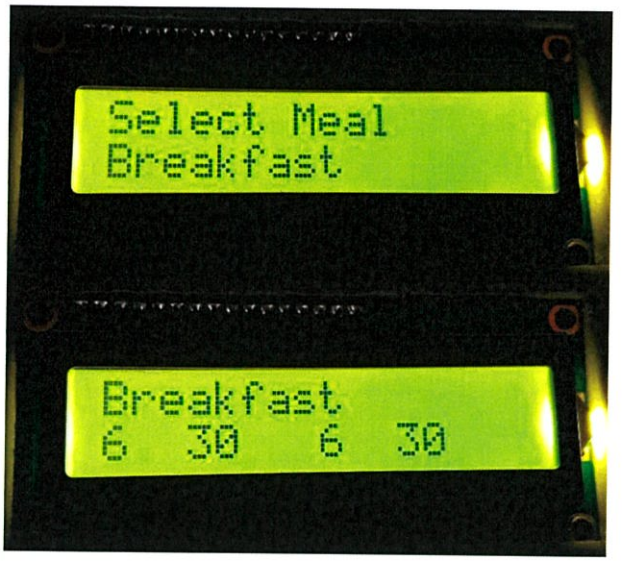


รูปที่ 4.1 กล่องยาพร้อมไมโครคอนโทรลเลอร์

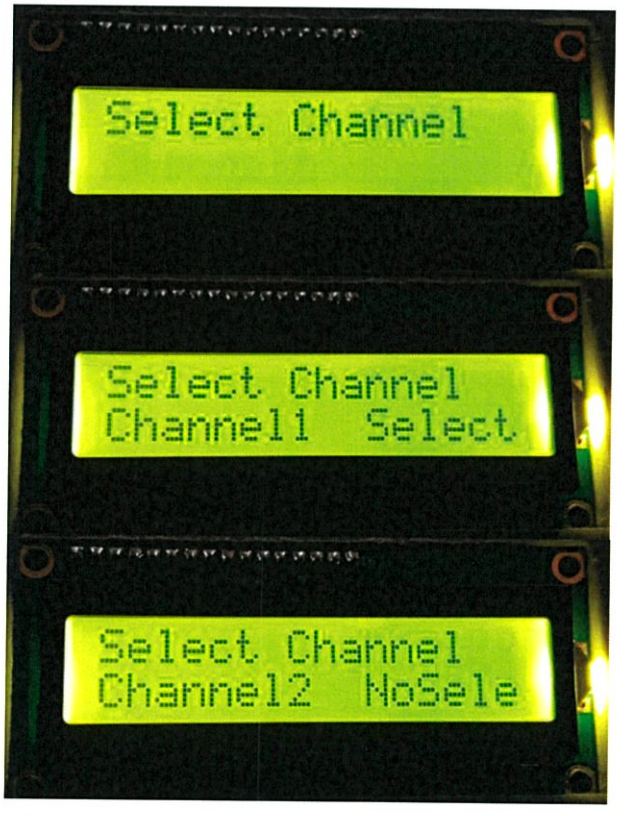
รูปที่ 4.2 แสดงหน้าต่างหน้าแรกของอุปกรณ์



รูปที่ 4.3 แสดงหน้าต่างแสดงเวลาของอุปกรณ์



รูปที่ 4.4 แสดงตัวอย่างการตั้งค่าเวลาสำหรับมื้ออาหาร



รูปที่ 4.5 แสดงตัวอย่างการตั้งค่าการเลือกช่องของกล่องยา

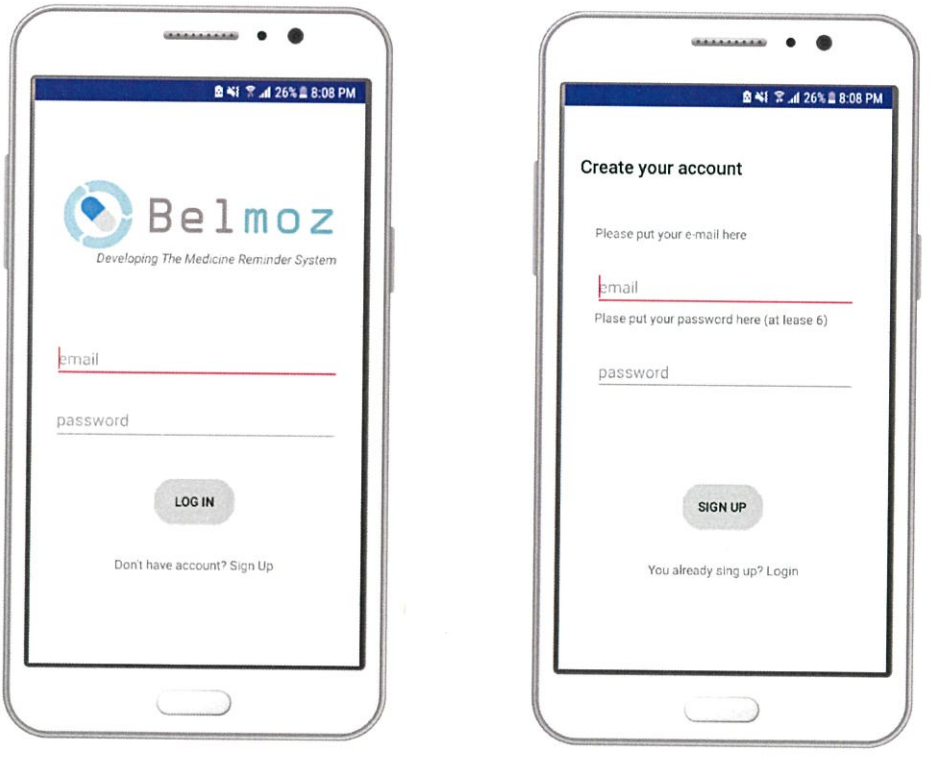
รูปที่ 4.6 อุปกรณ์ในขณะที่มีการแจ้งเตือน

รูปที่ 4.7 อุปกรณ์ในขณะที่มีการเปิดฝาล่องยาหลังจากการแจ้งเตือน

4.1.2 แอปพลิเคชันบนแอนดรอยด์สมาร์ทโฟน

ส่วนของแอปพลิเคชันบนแอนดรอยด์สมาร์ทโฟน รูปที่ 4.8 เมื่อคลิกเข้าไปในแอปพลิเคชัน “Belmoz” จะเจอหน้าตาแอปพลิเคชั่นที่ให้ทำการล็อคอิน หรือถ้าไม่มีแอคเคาท์ให้ทำการสมัครสมาชิก ดังรูปที่ 4.9

รูปที่ 4.8 ไอคอนแอปพลิเคชันชื่อ “Belmoz”

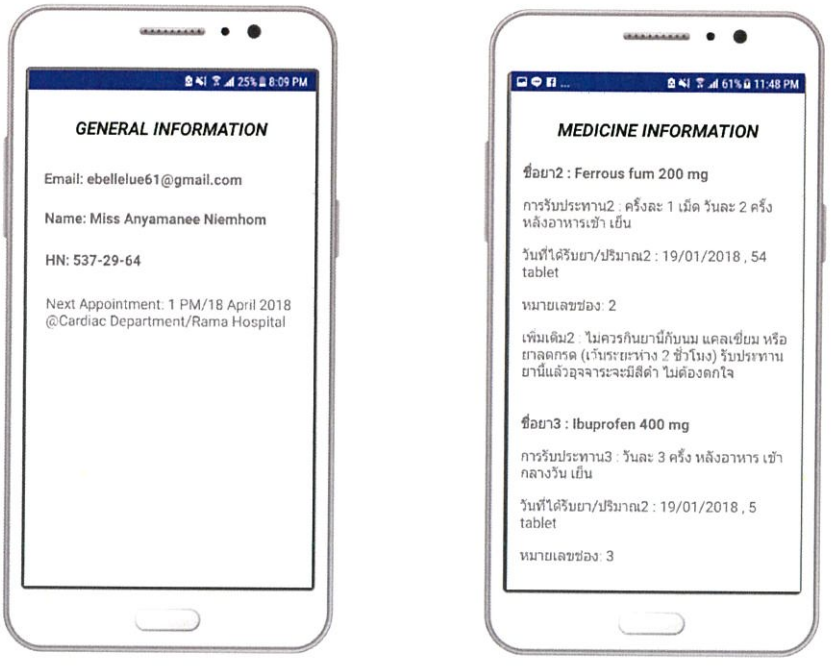


รูปที่ 4.9 แสดงหน้าจอเข้าสู่ระบบและสมัครหน้าจอสมัครสมาชิกบนแอนดรอยด์สมาร์ทโฟน

รูปที่ 4.10 แสดงหน้าจอหลังจากเข้าสู่ระบบจะเจอหน้าหลักที่มีปุ่มเข้าไปดูข้อมูลต่าง ๆ

หลังจากทำการเข้าสู่ระบบ(Log In) จะเข้าสู่หน้าจอหลักของแอปพลิเคชัน ดังรูปที่ 4.10 โดยจะมีปุ่มให้กดเพื่อเข้าถึงข้อมูลต่าง ๆ โดยมี 4 ปุ่ม ดังนี้

1) General Information: ข้อมูลทั่วไปของผู้ป่วย ดังรูปที่ 4.11
2) Medicine Information: ข้อมูลยาของผู้ป่วย ดังรูปที่ 4.11



รูปที่ 4.11 หน้าจอแสดงหน้าข้อมูลทั่วไปของผู้ป่วย และข้อมูลยา

3) History: ประวัติการรับประทานยาของผู้ป่วย ซึ่งเก็บมาจากประวัติการเปิดฝากล่องยา ดังรูปที่ 4.12

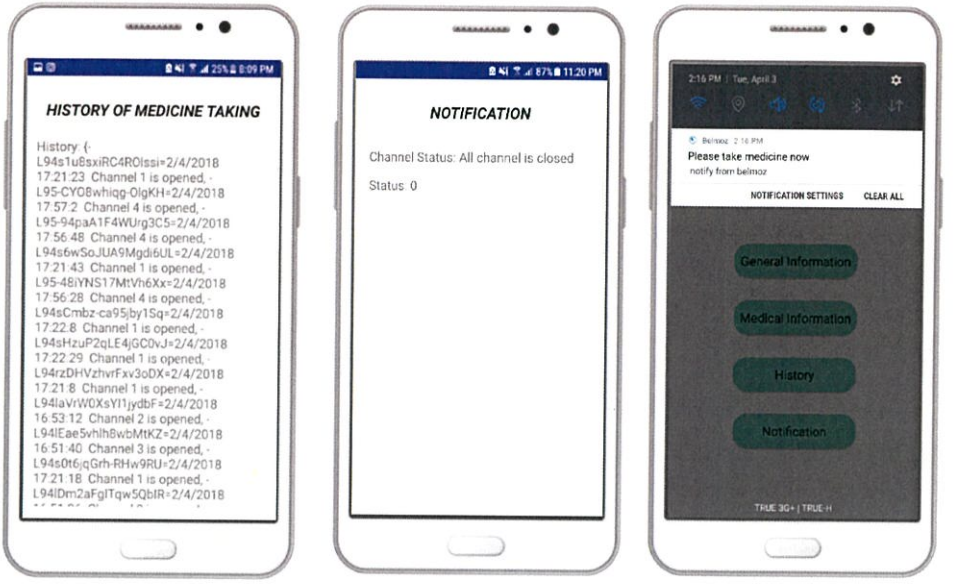
4) Notification: แสดงสถานะ(Status) การแจ้งเตือนการทานยา ดังรูปที่ 4.12 โดย

0 แสดงถึงเวลานั้นไม่ใช่เวลารับประทานยา

1 แสดงถึงเวลานั้นเป็นเวลาที่ต้องทำการรับประทานยา

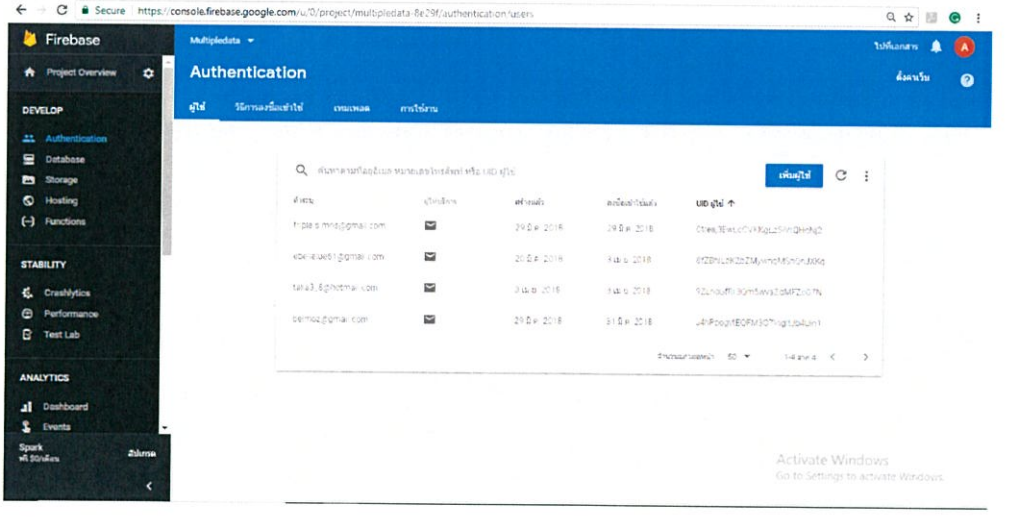
และนอกจากนี้ยังบอกสถานะของฝาบนกล่องยาว่าเปิดหรือปิดอยู่ เช่น เวลาปกติทุกฝาจะปิดหมดก็จะแสดง All channel is closed ถ้าฝาที่ช่อง 1 ถูกเปิดจะแสดง Channel 1 is opened

การแจ้งเตือนจะแสดงขึ้นที่แถบข้างบนของสมาร์ทโฟนเมื่อถึงเวลา ดังรูปที่ 4.12

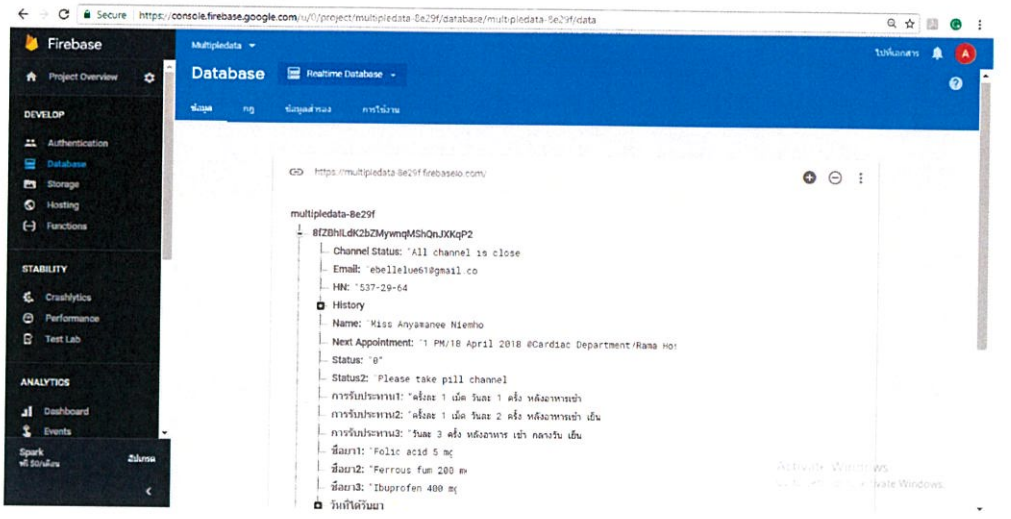


รูปที่ 4.12 หน้าจอแสดงหน้าประวัติการรับประทานยาของผู้ป่วย และข้อมูลสถานการณ์แจ้งเตือน

นอกจาก 2 ส่วนหลักในระบบแล้ว อีกส่วนสำคัญที่จัดเก็บข้อมูลทั้งหมดก็คือ ฐานข้อมูลออนไลน์ไฟร์เบส โดยข้อมูลต่างๆ จะถูกแยกตามรหัสของแต่ละแอคเคาท์ ที่ต้องทำการเข้าสู่ระบบ เนื่องจากเป็นเรื่องของความปลอดภัยของข้อมูล ดังรูปที่ 4.13 และ 4.14



รูปที่ 4.13 แสดงรายชื่ออีเมล์ที่ทำการสมัครสมาชิกเพื่อใช้แอคคเคาท์เข้าสู่ระบบบนฐานข้อมูลออนไลน์ไฟร์เบส



รูปที่ 4.14 แสดงฐานข้อมูลที่เก็บไว้บนฐานข้อมูลออนไลน์ไฟร์เบส ข้อมูลจะอยู่ในรหัสของแต่ละแอคเคาท์

4.2 การทดสอบ

หลังจากการออกแบบอุปกรณ์เสร็จสิ้น มีการทดสอบความแม่นยำในการทำงานของระบบแบบต่าง ๆ ได้แก่ ความสามารถในการตรวจจับการเปิดและปิดฝา การแจ้งเตือนที่กล่องยา การแจ้งเตือนที่แอปพลิเคชัน การตั้งเวลาที่กล่องยา

4.2.1 การทดสอบการตรวจจับการเปิดปิดฝากล่องยา

การทดสอบการตรวจจับการเปิดปิดฝากล่องยา จะทดสอบโดยการสังเกตผลบนฐานข้อมูลออนไลน์ Firebase บันทึกผลโดยให้สัญลักษณ์ 0 แทนฝากล่องยาถูกปิด สัญลักษณ์ 1 ฝากล่องยาถูกเปิด

การทดสอบการตรวจจับการเปิดฝา กล่องยาโดยการเปิดฝากล่องยาให้ครบทุกกรณีที่ระบุ ในขณะที่ฝากล่องยาของช่องอื่นๆปิดอยู่ ผลการทดสอบสามารถดูได้จากตารางที่ 4.1

ตารางที่ 4.1 แสดงผลบนฐานข้อมูลออนไลน์ สำหรับการทดสอบการตรวจจับการเปิดฝา กล่องยา

ครั้งที่	1				2				3				4				5			
กรณี	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
เปิดฝาช่องที่ 1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
เปิดฝาช่องที่ 2	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
เปิดฝาช่องที่ 3	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
เปิดฝาช่องที่ 4	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
เปิดฝาช่องที่ 1 และ 2	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
เปิดฝาช่องที่ 1 และ 3	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
เปิดฝาช่องที่ 1 และ 4	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
เปิดฝาช่องที่ 2 และ 3	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
เปิดฝาช่องที่ 2 และ 4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
เปิดฝาช่องที่ 3 และ 4	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
เปิดฝาช่องที่ 1 2 และ 3	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0
เปิดฝาช่องที่ 1 2 และ 4	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1
เปิดฝาช่องที่ 1 3 และ 4	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1
เปิดฝาช่องที่ 2 3 และ 4	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
เปิดฝาทุกช่อง	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

การทดสอบการตรวจจับการปิดฝา กล่องยาโดยการเปิดฝากล่องยาให้ครบทุกกรณีที่ระบุ ในขณะที่ฝากล่องยาของช่องอื่นๆเปิดอยู่ ผลการทดสอบสามารถดูได้จากตารางที่ 4.2

ตารางที่ 4.2 แสดงผลบนฐานข้อมูลออนไลน์ สำหรับการทดสอบการตรวจจับการปิดฝากล่องยา

ครั้งที่	1				2				3				4				5			
กรณี	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
ปิดฝาช่องที่ 1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1
ปิดฝาช่องที่ 2	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1
ปิดฝาช่องที่ 3	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1
ปิดฝาช่องที่ 4	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0
ปิดฝาช่องที่ 1 และ 2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
ปิดฝาช่องที่ 1 และ 3	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
ปิดฝาช่องที่ 1 และ 4	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	1	1	1	1
ปิดฝาช่องที่ 2 และ 3	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1
ปิดฝาช่องที่ 2 และ 4	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
ปิดฝาช่องที่ 3 และ 4	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0
ปิดฝาช่องที่ 1 2 และ 3	0	0	0	1	0	0	0	1	1	0	1	1	0	0	0	1	1	0	0	1
ปิดฝาช่องที่ 1 2 และ 4	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1
ปิดฝาช่องที่ 1 3 และ 4	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	0	0	1	0	0
ปิดฝาช่องที่ 2 3 และ 4	1	0	0	0	1	1	1	0	1	0	0	0	1	0	0	0	1	0	0	0
ปิดฝาทุกช่อง	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0

จากตารางที่ 4.1 ถึง 4.2 จะได้ผลการทดสอบดังนี้

การทดสอบการเปิดฝาของแต่ละช่อง

ช่องที่ 1 ให้ผลร้อยละ 100.0 จากการทดสอบการเปิดฝาช่องที่ 1 รวมกันทั้งหมด 40 ครั้ง

ช่องที่ 2 ให้ผลร้อยละ 100.0 จากการทดสอบการเปิดฝาช่องที่ 2 รวมกันทั้งหมด 40 ครั้ง

ช่องที่ 3 ให้ผลร้อยละ 100.0 จากการทดสอบการเปิดฝาช่องที่ 3 รวมกันทั้งหมด 40 ครั้ง

ช่องที่ 4 ให้ผลร้อยละ 100.0 จากการทดสอบการเปิดฝาช่องที่ 4 รวมกันทั้งหมด 40 ครั้ง

การทดสอบการปิดฝาของแต่ละช่อง

ช่องที่ 1 ให้ผลร้อยละ 85.00 จากการทดสอบการปิดฝาช่องที่ 1 รวมกันทั้งหมด 40 ครั้ง

ช่องที่ 2 ให้ผลร้อยละ 95.00 จากการทดสอบการปิดฝาช่องที่ 2 รวมกันทั้งหมด 40 ครั้ง

ช่องที่ 3 ให้ผลร้อยละ 87.50 จากการทดสอบการปิดฝาช่องที่ 3 รวมกันทั้งหมด 40 ครั้ง

ช่องที่ 4 ให้ผลร้อยละ 95.00 จากการทดสอบการปิดฝาช่องที่ 4 รวมกันทั้งหมด 40 ครั้ง

ผลการทดสอบรวมของการทดสอบการตรวจจับการเปิดปิดฝากล่องยาของอุปกรณ์

ช่องที่ 1 สามารถทำงานได้ร้อยละ 92.50

ช่องที่ 2 สามารถทำงานได้ร้อยละ 97.50

ช่องที่ 3 สามารถทำงานได้ร้อยละ 93.75

ช่องที่ 4 สามารถทำงานได้ร้อยละ 97.50

และอุปกรณ์สามาตรวจจับการเปิดปิดฝากล่องยาได้ร้อยละ 94.67

4.2.2 การทดสอบการแจ้งเตือนที่กล่องยา

จะทดสอบโดยการตั้งเวลาบนไมโครคอนโทรลเลอร์แล้วสังเกตผลการแจ้งเตือนจาก LED หรือเสียง ผลการทดสอบสามารถดูได้จากตารางที่ 4.3

ตารางที่ 4.3 แสดงผลการทดสอบการแจ้งเตือนที่กล่องยา

เวลาที่ตั้ง	เวลาที่มีการแจ้งเตือน	การแจ้งเตือนด้วย LED	การแจ้งเตือนด้วยเสียง
10.05	10.05	มี	มี
10.15	10.15	มี	มี
10.25	10.25	มี	มี
10.40	10.40	มี	มี
11.00	11.00	มี	มี
14.00	14.00	มี	มี
14.30	14.30	มี	มี
14.45	14.45	มี	มี
15.00	15.00	มี	มี
15.15	15.15	มี	มี

4.2.3 การทดสอบการแจ้งเตือนด้วยไฟที่แต่ละช่องกล่องยา

การทดสอบนี้จะทดสอบโดยการตั้งค่าช่องของกล่องยาให้แจ้งเตือนด้วยไฟทีละสองช่อง สัญลักษณ์ 1 แสดงถึงมีการแจ้งเตือนที่ช่องของกล่องยาช่องนั้น สัญลักษณ์ 0 แสดงถึงไม่มีการแจ้งเตือนที่ช่องของกล่องยาช่องนั้น ผลการทดสอบสามารถดูได้จากตารางที่ 4.4

ตารางที่ 4.4 แสดงผลการทดสอบการแจ้งเตือนด้วยไฟที่แต่ละช่องกล่องยา

ช่องที่ตั้งค่าให้มีการแจ้งเตือน	ผลครั้งที่ 1				ผลครั้งที่ 2				ผลครั้งที่ 3			
ช่องที่ 1 2 และ 3	1	1	1	0	0	0	1	0	1	1	1	0
ช่องที่ 1 2 และ 4	1	1	0	1	1	1	0	1	1	1	0	1
ช่องที่ 1 3 และ 4	1	0	1	1	1	0	1	1	1	0	1	1
ช่องที่ 2 3 และ 4	0	1	1	1	0	1	1	1	0	1	1	1

จากการทดสอบการแจ้งเตือนด้วยไฟที่แต่ละช่องกล่องยาพบว่ากล่องยาสามารถแจ้งเตือนได้ร้อยละ 91.67

4.2.4 การแจ้งเตือนที่แอปพลิเคชัน

การทดสอบตั้งค่าให้แอปพลิเคชันแจ้งเตือน ผลการทดสอบสามารถดูได้จากตารางที่ 4.5

ตารางที่ 4.5 แสดงผลการทดสอบการแจ้งเตือนที่แอปพลิเคชันที่ตั้งเวลาไว้ 1 นาที

จำนวนครั้ง	1	2	3	4	5	6	7	8	9	10	เฉลี่ย
การแจ้งเตือน (วินาที)	0.57	0.17	1.25	0.19	0.50	1.19	1.22	0.18	1.30	1.20	
จำนวนครั้ง	11	12	13	14	15	16	17	18	19	20	
การแจ้งเตือน (วินาที)	0.14	1.20	1.18	0.43	0.50	1.18	1.39	1.45	1.30	1.02	1.21

ได้ค่าเฉลี่ยเวลาในการแจ้งเตือนบนแอปพลิเคชันเป็น 1.21 นาที ความแม่นยำ 65 %

4.2.5 การแจ้งเตือนซ้ำที่แอปพลิเคชัน

การทดสอบการตั้งค่าให้แอปพลิเคชันแจ้งเตือนหลังจากเวลาที่กำหนดไว้บนกล่องยา 05.00 นาที ผลการทดสอบสามารถดูได้จากตารางที่ 4.6

ตารางที่ 4.6 แสดงผลการทดสอบการแจ้งเตือนซ้ำที่แอปพลิเคชัน

จำนวนครั้ง	1	2	3	4	5	6	7	8	9	10	เฉลี่ย
การแจ้งเตือน (วินาที)	04.56	04.59	05.01	05.00	04.57	04.58	05.00	04.58	04.59	05.00	
จำนวนครั้ง	11	12	13	14	15	16	17	18	19	20	
การแจ้งเตือน (วินาที)	05.00	05.01	04.59	04.55	05.01	04.57	04.59	04.59	05.00	04.57	04.59

ได้ค่าเฉลี่ยเวลาในการแจ้งเตือนซ้ำบนแอปพลิเคชันเป็น 04.59 นาที = ความแม่นยำ 98.33%

4.2.6 ตารางสรุปการทดสอบ

ตารางที่ 4.7 แสดงผลสรุปของการทดสอบ

การทดสอบ	% ความแม่นยำ
ความสามารถในการตรวจจับการเปิดฝากล่องยา	94.67%
การแจ้งเตือนที่กล่องยา	91.67%
การแจ้งเตือนบนแอปพลิเคชัน	65%
การแจ้งเตือนซ้ำที่แอปพลิเคชัน	98.33%

ความแม่นยำรวมของระบบ = 87.41 %

ปัจจัยของความคลาดเคลื่อนในระบบคาดว่ามาจากการส่งผ่านข้อมูลทางอินเตอร์เน็ต ส่งผลให้การแจ้งเตือนบนสมาร์ทโฟนมีความคลาดเคลื่อนเล็กน้อย

บทที่ 5

บทสรุป

5.1 สรุป

การพัฒนาระบบการแจ้งเตือนการรับประทานยานี้ มุ่งเน้นไปที่ผู้สูงอายุที่อาศัยอยู่ในบ้าน เนื่องจากการวิจัยพบว่าผู้สูงอายุในประเทศไทยที่อาศัยอยู่ในบ้านประสบปัญหาการรับประทานยาหลายด้าน เช่น การรับประทานยาไม่ตรงเวลาหรือรับประทานยาผิด โดยระบบนี้มีกล่องยาที่สามารถตั้งค่าเวลาการแจ้งเตือนการรับประทานยา และมีความแม่นยำรวมถึง 87.41% ซึ่งถ้าผู้ป่วยสูงอายุปฏิบัติตามแนวทางของระบบจะเพิ่มความถูกต้องและแม่นยำในการรับประทานยาของคนไข้ และส่งผลให้เพิ่มประสิทธิภาพต่อการรับการรักษาด้วยยาจากแพทย์ นอกจากนี้ระบบยังสามารถเก็บประวัติการรับประทานยาไว้ในฐานข้อมูลออนไลน์ให้แพทย์สามารถเข้าถึงได้สะดวกอีกด้วย มีการแยกข้อมูลออกเป็นแอคเคาท์เนื่องจากคำนึงถึงเรื่องความปลอดภัยของข้อมูล และจากในยุคปัจจุบันที่มีการใช้สมาร์ทโฟนเป็นจำนวนมาก จึงทำการส่งผ่านข้อมูลยาไปแสดงและแจ้งเตือนที่แอปพลิเคชันแอนดรอยด์ ทำให้ผู้ป่วยหรือญาติของผู้ป่วยสามารถดูรายละเอียดที่เกี่ยวข้องกับการรับประทานยาได้อย่างสะดวก รวดเร็วและถูกต้อง แต่ระบบนี้ยังมีข้อจำกัดในเรื่องของการตั้งเวลาแจ้งเตือนที่ยังไม่หลากหลาย ลำดับเวลาในการเลือกช่องของกล่องยา และการแสดงประวัติการรับประทานยาบนแอปพลิเคชันที่ยังจัดเรียงไม่เป็นระเบียบ

5.2 ปัญหาและอุปสรรคในการทำงาน

5.2.1 ข้อจำกัดอินพุทและเอาท์พุทของไมโครคอนโทรลเลอร์ ทำให้ต้องแก้ไขรูปแบบของอุปกรณ์

5.2.2 รูปแบบของอุปกรณ์มีความละเอียด ทำให้เกิดปัญหากับการพิมพ์ 3D

5.2.3 ปัญหาขนาดของแบบที่พิมพ์ออกมาไม่ตรงตามที่ต้องการ

5.2.4 ไฟร์เบสเป็นฐานข้อมูลออนไลน์ที่ยังใหม่ จึงมีการพัฒนาตลอดเวลา และเปลี่ยนรูปแบบการใช้บ่อยครั้ง

5.2.5 อุปกรณ์ข้อมูลไร้สาย บางครั้งไม่สามารถเชื่อมต่ออินเตอร์เน็ตได้

5.2.6 การแจ้งเตือนบนแอปพลิเคชันขึ้นอยู่กับอินเทอร์เน็ตในเวลานั้น

5.3 ข้อเสนอแนะ

5.2.1 พัฒนาอุปกรณ์ให้สามารถเปิดปิดฝาของกล่องยาได้ง่ายขึ้น

5.2.2 เพิ่มระบบความปลอดภัยให้อุปกรณ์ เช่น ระบบล็อค

5.2.3 พัฒนาอุปกรณ์ให้มีขนาดเล็กลง

5.2.4 เปลี่ยนการจ่ายไฟให้อุปกรณ์จากแบตเตอรี่

5.2.5 เพิ่มเติมข้อมูลยาให้มากขึ้น และจัดระเบียบให้ดีขึ้น

5.2.6 ทดลองใช้ระบบกับผู้สูงอายุจริง เพื่อประเมินผลได้ชัดเจนมากขึ้น

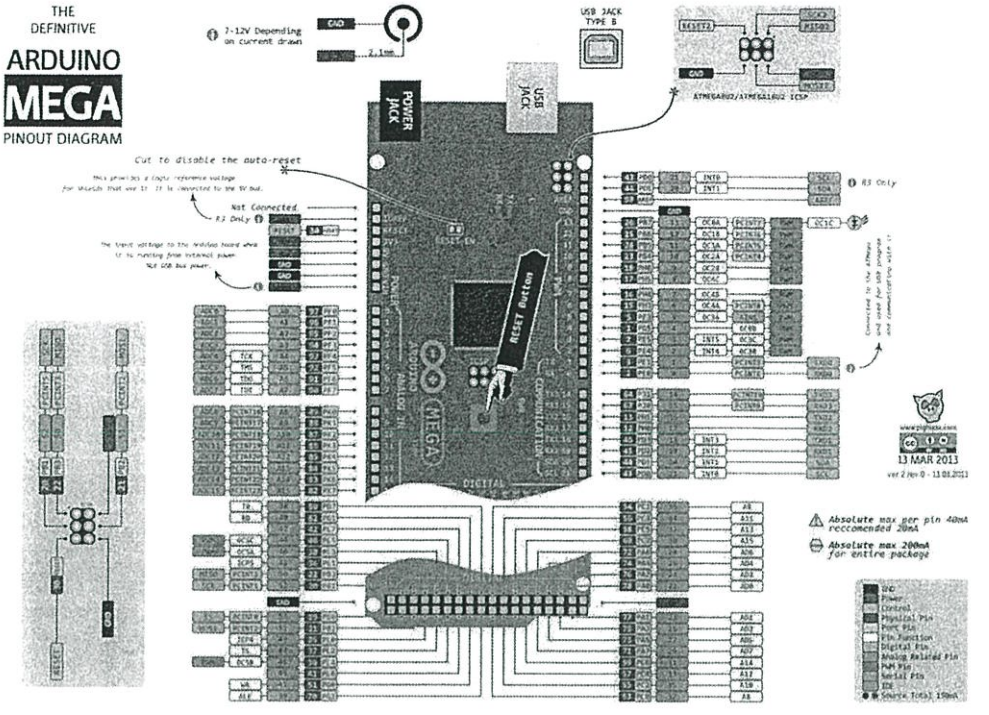
บรรณานุกรม

1. บัญชา ปะสีละเตสัง. (2559). การเขียนโปรแกรม Java และ Android. กรุงเทพมหานคร: ซีเอ็ดยูเคชั่น.
2. ไม่ปรากฏนามผู้แต่งและปีที่แต่ง. บทที่ 3 รหัสแทนข้อมูล. สืบค้นเมื่อ 14 พฤศจิกายน 2560 จาก เว็บไซต์: http://jumrus.crru.ac.th/cp1701_digital/lesson3.pdf
3. นางอุตสาห์ โนราช. (2557). การรับส่งข้อมูล. สืบค้นเมื่อ 15 พฤศจิกายน 2560 จาก เว็บไซต์: https://kruudsa2011.wordpress.com/การรับส่งข้อมูล/
4. ไม่ปรากฏนามผู้แต่งและปีที่แต่ง. Server คืออะไร ทำหน้าที่อะไร มีประโยชน์อย่างไร Server มีกี่ประเภท. สืบค้นเมื่อ 15 พฤศจิกายน 2560 จาก เว็บไซต์: http://www.เกร็ดความรู้.net/server/
5. น.อ.รุ่งธรรม บัวแตง. (2557). Database (ฐานข้อมูล) คืออะไร (เพิ่มเติม). สืบค้นเมื่อ 15 พฤศจิกายน 2560 จาก เว็บไซต์: http://www.dstd.mi.th/board/index.php?topic=370.0
6. Archai. (2560). ทำความรู้จักกับ Android Studio. สืบค้นเมื่อ 18 พฤศจิกายน 2560 จาก เว็บไซต์: http://arctech.in.th/articles/125
7. Jirawatee. (2559). รู้จัก Firebase และการเพิ่มเข้าไปในโปรเจค Android. สืบค้นเมื่อ 18 พฤศจิกายน 2560 จาก เว็บไซต์: https://developers.ascendcorp.com/รู้จัก-Firebase-และวิธีการเพิ่มมันเข้าไปใน-android-project-e51e38bccbfa
8. Siriwimon Sunthon. (2557). ศึกษาข้อมูลของบอร์ด Arduino Mega2560. สืบค้นเมื่อ 3 เมษายน 2561 จาก เว็บไซต์: http://mbeddedweekly.blogspot.com/2014/08/arduino-mega2560.html
9. หทัยรัตน์ สุขศรี, เพ็ญนภา ศรีหริ่ง, ทิพาพร กาญจนราช. (2554). ปัญหาการใช้ยาของผู้ป่วยในการรักษาตนเองที่บ้านในเขตชุมชนอำเภอเมือง จังหวัดขอนแก่น. สืบค้นเมื่อ 19 มีนาคม 2561 จาก เว็บไซต์: https://cscd.kku.ac.th/2016/uploads/proceeding/060911_153105.pdf

10. ภก. ธีรัตถ์ เหลืองมั่นคง. (2554). ยาก่อนอาหาร ยาหลังอาหาร ลืมกินยาตามเวลา อันตรายหรือไม่. สืบค้นเมื่อ 19 มีนาคม 2561 จาก เว็บไซต์: http://www.pharmacy.mahidol.ac.th/th/knowledge/article/83/ยาก่อนอาหาร-ยาหลังอาหาร-ลืมกินยาตามเวลา-อันตรายหรือไม่/
11. Embedded Systems Lab (ESL@KMUTNB). (2557). การอ่านและแสดงวันเวลาจากโมดูล DS3231 RTC. สืบค้นเมื่อ 19 มีนาคม 2561 จาก เว็บไซต์: http://cpre.kmutnb.ac.th/esl/learning/index.php?article=ds3231-i2c-rtc
12. Montien Ngamkaew. วิธีดึงเวลาสากลจาก Internet มาใช้งานกับESP8266. สืบค้นเมื่อ 19 มีนาคม 2561. จาก เว็บไซต์: https://montienfocus.blogspot.com/2016/06/internet-esp8266.html
13. ผศ.ดร. จตุรงค ลังกาพินธุ์. คูมือการใช้โปรแกรม Autodesk Inventor Professional ขั้นพื้นฐาน. สืบค้นเมื่อ 19 มีนาคม 2561 จากเว็บไซต์: http://www.satreephuket.ac.th/krukaew/NG23204/inventor1.pdf
14. Benznest. (2559). Android Code : ทำ Notification การแจ้งเตือนแบบพื้นฐาน. สืบค้นเมื่อ 19 มีนาคม 2561 จากเว็บไซต์: https://benzneststudios.com/blog/android/how-to-use-notification-basic-on-android/

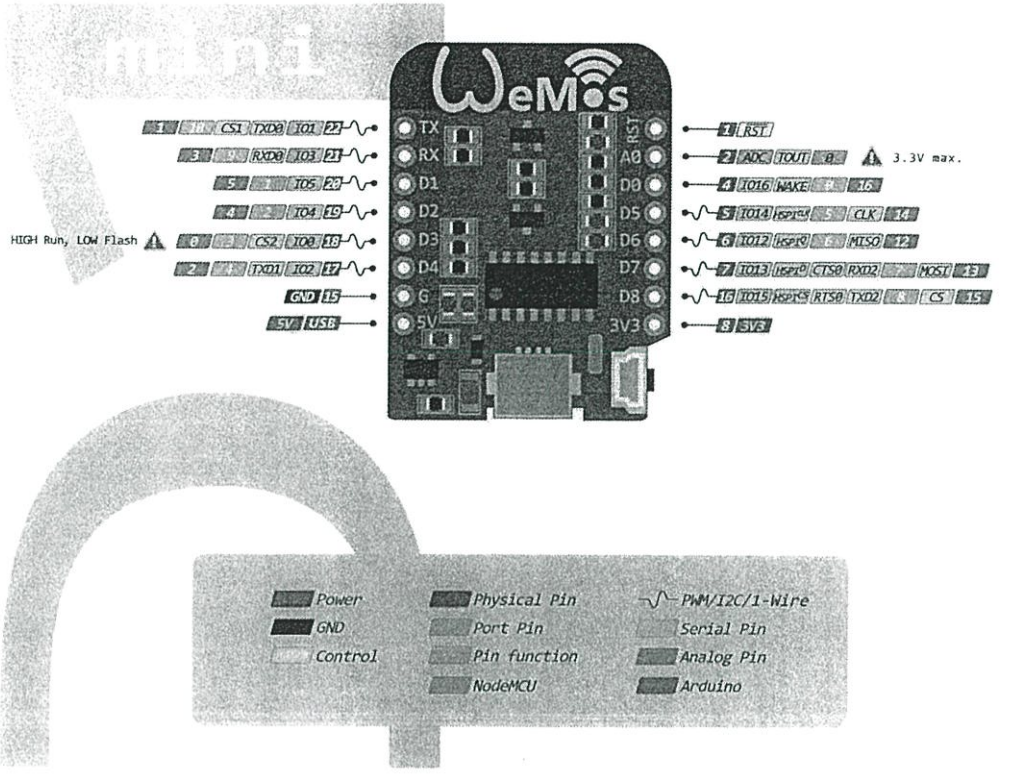
ภาคผนวก

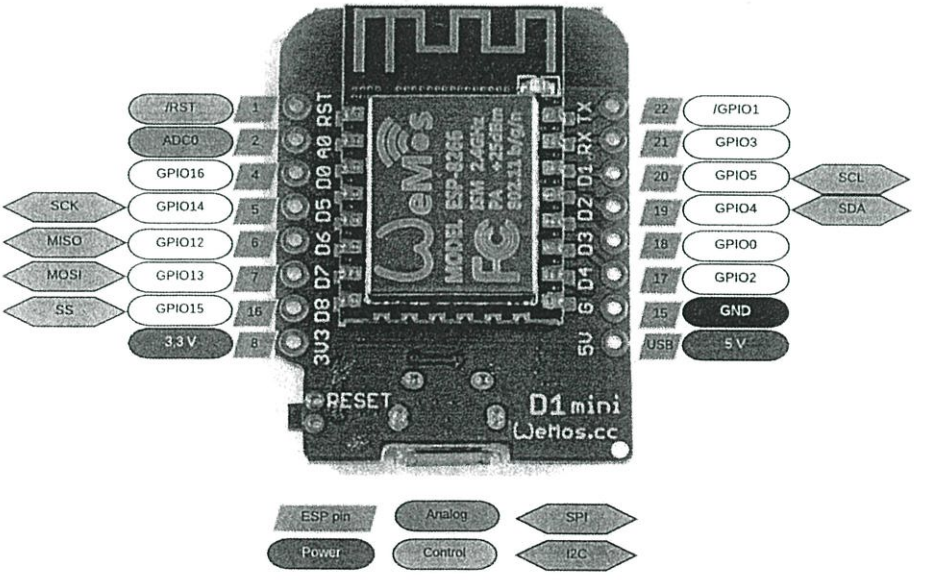
Arduino Mega 2560 R3



WeMos D1 mini pins and diagram

Diagram



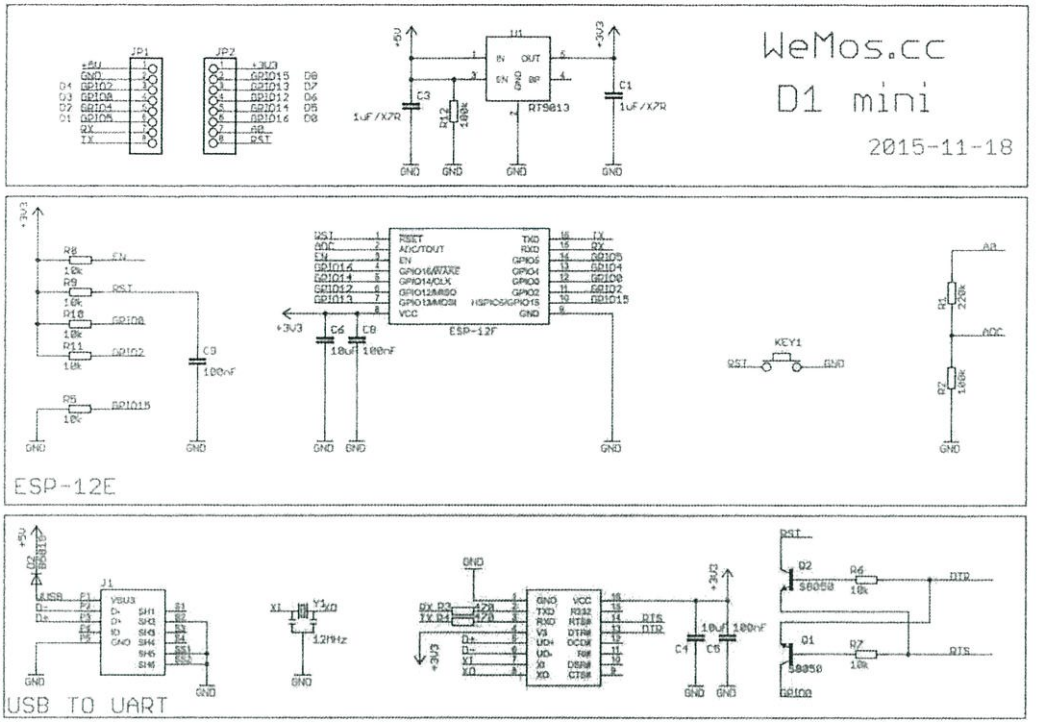


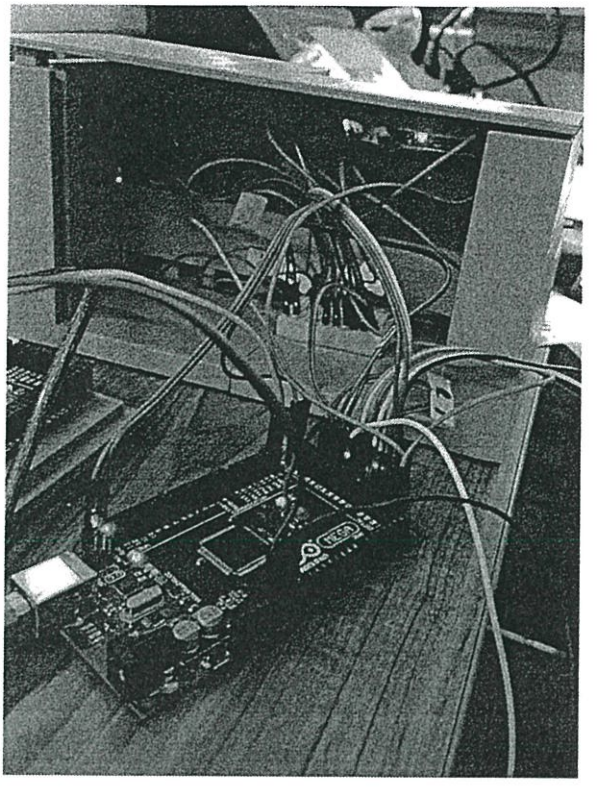
Pin

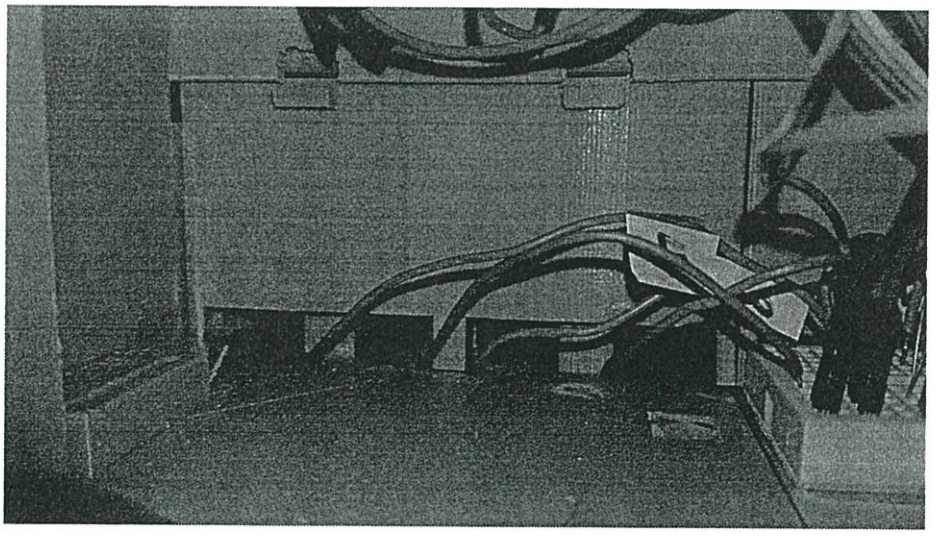
Pin	Function	ESP-8266 Pin
TX	TXD	TXD
RX	RXD	RXD
A0	Analog input, max 3.3V input	A0
D0	IO	GPIO16
D1	IO, SCL	GPIO5
D2	IO, SDA	GPIO4
D3	IO,10k Pull-up	GPIO0
D4	IO, 10k pull-up, BUILTIN_LED	GPIO2
D5	IO, SCK	GPIO14
D6	IO, MISO	GPIO12
D7	IO, MOSI	GPIO13
D8	IO,10k pull-down, SS	GPIO15
G	Ground	GND
5V	5V	–
3V3	3.3V	3.3V
RST	Reset	RST

**All IO have interrupt/pwm/I2C/one-wire supported(except D0)*

d1 mini shematics







ชุดคำสั่งที่ใช้บน Arduino Mega 2560 R3

```
#include <Keypad.h> //http://www.arduino.cc/playground/uploads/Code/Keypad.zip

#include <Wire.h>

#include <LiquidCrystal_I2C.h> //ประกาศ Library ของจอ I2C

#include <DS3231.h>


// Set the LCD address to 0x27 for a 16 chars and 2 line display

LiquidCrystal_I2C lcd(0x27,16,2);

DS3231  rtc(SDA, SCL);

const byte ROWS = 4; // Four rows

const byte COLS = 4; //  columns


// Define the Keymap

char keys[ROWS][COLS] = {

  {'D','#','0','*'},

  {'C','9','8','7'},

  {'B','6','5','4'},

  {'A','3','2','1'}                    };

byte rowPins[ROWS] = {47, 49, 51, 53}; //connect to the row pinouts of the keypad

byte colPins[COLS] = {39, 41, 43, 45}; //connect to the column pinouts of the keypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

int h=0;

int m=0;
```

```
int N=0;

int n;

int H1,M1;

int H2,M2;

int H3,M3;

int H4,M4;

char a;

int Hor;

int Min;

int Sec;

int i=10;

int I;

int A=0;

int P1[4];

int P2[4];

int P3[4];

int P4[4];

int P5[4];

int P6[4];

int P7[4];

int p11=1;

int p22=1;

int p33=1,

int p44=1;
```

```
int p1,p2,p3,p4,p5,p6,p7;
int k=1;
Time t;

void setup() {
  Serial.begin(9600);
  pinMode(22, OUTPUT);
  pinMode(24, OUTPUT);
  pinMode(26, OUTPUT);
  pinMode(28, OUTPUT);
  pinMode(30, INPUT);
  pinMode(32, INPUT);
  pinMode(34, INPUT);
  pinMode(36, INPUT);
  pinMode(38, OUTPUT);
  pinMode(46, OUTPUT);
  pinMode(48, OUTPUT);
  pinMode(50, OUTPUT);
  pinMode(52, OUTPUT);
  Wire.begin();
  lcd.begin();
  rtc.begin();
  lcd.setCursor(0, 0);lcd.print("Belmoz");
}
```

```
void loop() {
  if(i==0){
  a=keypad.getKey();
  // put your main code here, to run repeatedly:
switch(a){
  case '7' :h=h+1;lcd.clear();break;
  case '8' :m=m+5;lcd.clear();break;
  case '9' :H1=h;M1=m;lcd.clear();break;
  case '2' : i=10;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Meal");break;
  case '1' : i=5;lcd.clear();break;
  case '*' : i=6;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Channel");break;
  default : ; break;}

  lcd.setCursor(0, 0);lcd.print("Breakfast");

  if(h<24){lcd.setCursor(0, 1);lcd.print(h);}
  else{h=0;lcd.setCursor(0, 1);lcd.print(h);}

  if(m<60){lcd.setCursor(3, 1);lcd.print(m);}
  else{m=0;lcd.setCursor(3, 1);lcd.print(m);}

  lcd.setCursor(8, 1);lcd.print(H1);
  lcd.setCursor(11, 1);lcd.print(M1); }
```

```
  else if(i==1){
  a=keypad.getKey();

switch(a){
  case '7' :h=h+1;lcd.clear();break;
  case '8' :m=m+5;lcd.clear();break;
  case '9' :H2=h;M2=m;lcd.clear();break;
  case '2' : i=10;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Meal");break;
  case '1' : i=5;lcd.clear();break;
  case '*' : i=7;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Channel");break;
  default : ;break;}

lcd.setCursor(0, 0);lcd.print("Lunch");

 if(h<24){lcd.setCursor(0, 1);lcd.print(h);}
 else{h=0;lcd.setCursor(0, 1);lcd.print(h);}

 if(m<60){lcd.setCursor(3, 1);lcd.print(m);}
 else{m=0;lcd.setCursor(3, 1);lcd.print(m);}

 lcd.setCursor(8, 1);lcd.print(H2);
 lcd.setCursor(11, 1);lcd.print(M2); }
```

```
  else if(i==2){

  a=keypad.getKey();


switch(a){

  case '7' :h=h+1;lcd.clear();break;

  case '8' :m=m+5;lcd.clear();break;

  case '9' :H3=h;M3=m;lcd.clear();break;

  case '2' : i=10;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Meal");break;

  case '1' : i=5;lcd.clear();break;

  case '*' : i=8;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Channel");break;

  default : ;break;}


  lcd.setCursor(0, 0);lcd.print("Dinner");


  if(h<24){lcd.setCursor(0, 1);lcd.print(h);}

  else{h=0;lcd.setCursor(0, 1);lcd.print(h);}


  if(m<60){lcd.setCursor(3, 1);lcd.print(m);}

  else{m=0;lcd.setCursor(3, 1);lcd.print(m);}


  lcd.setCursor(8, 1);lcd.print(H3);

  lcd.setCursor(11, 1);lcd.print(M3); }


  else if(i==3){
```

```
  a=keypad.getKey();


switch(a){
  case '7' :h=h+1;lcd.clear();break;
  case '8' :m=m+5;lcd.clear();break;
  case '9' :H4=h;M4=m;lcd.clear();break;
  case '2' : i=10;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Meal");break;
  case '1' : i=5;lcd.clear();break;
  case '*' : i=9;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Channel");break;
  default : ;break;}


lcd.setCursor(0, 0);lcd.print("Bedtime");


 if(h<24){lcd.setCursor(0, 1);lcd.print(h);}
 else{h=0;lcd.setCursor(0, 1);lcd.print(h);}


 if(m<60){lcd.setCursor(3, 1);lcd.print(m);}
 else{m=0;lcd.setCursor(3, 1);lcd.print(m);}


 lcd.setCursor(8, 1);lcd.print(H4);
 lcd.setCursor(11, 1);lcd.print(M4);}


 else if(i==4){
   lcd.setCursor(0,0);
```

```
  lcd.print("Time:  ");
  lcd.print(rtc.getTimeStr());

  lcd.setCursor(0,1);
  lcd.print("Date: ");
  lcd.print(rtc.getDateStr());

  char ekey = keypad.getKey();
 switch (ekey){
 case '1' : i=5;lcd.clear();break;
 default : ;break;}
  }

   else if(i==5){
     lcd.setCursor(0,0);
  lcd.print("Welcome");

  lcd.setCursor(0,1);
  lcd.print("Belmoz");

  char ekey = keypad.getKey();
 switch (ekey){
 case '3' : i=4;lcd.clear();break;
 case '2' : i=10;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Meal");break;
```

```
case '1' : i=5;lcd.clear();break;

default : ;break;}

  }


  else if (i==6){

    String CH[]={"Please Select>>>","Channel1","Channel2","Channel3","Channel4","<<<Please
Select"};

    a=keypad.getKey();

switch(a){

  case '0' :A=A+1;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Channel");lcd.setCursor(0,
1);lcd.print(CH[A]);break;

  case '#' :A=A-1;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Channel");lcd.setCursor(0,
1);lcd.print(CH[A]);break;

  case 'D' :P1[p11]=1;lcd.setCursor(0, 0);lcd.print("Select Channel");lcd.setCursor(0,
1);lcd.print(CH[A]);lcd.setCursor(10,1);lcd.print("Selected");p11=p11+1;break;

  case 'C' :P1[p11]=0;lcd.setCursor(0, 0);lcd.print("Select Channel");lcd.setCursor(0,
1);lcd.print(CH[A]);lcd.setCursor(10,1);lcd.print("NoSelected");p11=p11+1;break;

  case '2' : i=10;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Meal");break;

  case '1' : i=5;lcd.clear();break;

  case '*' : i=6;p11=1;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Channel");p11=1;break;

  default : ;break;}

}


   else if (i==7){
```

```
   String CH[]={"Please Select>>>","Channel1","Channel2","Channel3","Channel4","<<<Please
Select"};
   a=keypad.getKey();
switch(a){
  case '0' :A=A+1;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Channel");lcd.setCursor(0,
1);lcd.print(CH[A]);break;
  case '#' :A=A-1;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Channel");lcd.setCursor(0,
1);lcd.print(CH[A]);break;
  case 'D' :P2[p22]=1;lcd.setCursor(0, 0);lcd.print("Select Channel");lcd.setCursor(0,
1);lcd.print(CH[A]);lcd.setCursor(10,1);lcd.print("Selected");p22=p22+1;break;
  case 'C' :P2[p22]=0;lcd.setCursor(0, 0);lcd.print("Select Channel");lcd.setCursor(0,
1);lcd.print(CH[A]);lcd.setCursor(10,1);lcd.print("NoSelected");p22=p22+1;break;
  case '2' : i=10;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Meal");break;
  case '1' : i=5;lcd.clear();break;
  case '*' : i=7;p22=1;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Channel");p22=1;break;
  default : ;break;}
 }

   else if (i==8){
   String CH[]={"Please Select>>>","Channel1","Channel2","Channel3","Channel4","<<<Please
Select"};
   a=keypad.getKey();
switch(a){
  case '0' :A=A+1;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Channel");lcd.setCursor(0,
1);lcd.print(CH[A]);break;
```

```
  case '#' :A=A-1;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Channel");lcd.setCursor(0,
1);lcd.print(CH[A]);break;

  case 'D' :P3[p33]=1;lcd.setCursor(0, 0);lcd.print("Select Channel");lcd.setCursor(0,
1);lcd.print(CH[A]);lcd.setCursor(10,1);lcd.print("Selected");p33=p33+1;break;

  case 'C' :P3[p33]=0;lcd.setCursor(0, 0);lcd.print("Select Channel");lcd.setCursor(0,
1);lcd.print(CH[A]);lcd.setCursor(10,1);lcd.print("NoSelected");p33=p33+1;break;

  case '2' : i=10;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Meal");break;

  case '1' : i=5;lcd.clear();break;

  case '*' : i=8;p33=1;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Channel");p33=1;break;

  default : ;break;}

 }


  else if (i==9){

   String CH[]={"Please Select>>>","Channel1","Channel2","Channel3","Channel4","<<<Please
Select"};

   a=keypad.getKey();

switch(a){

 case '0' :A=A+1;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Channel");lcd.setCursor(0,
1);lcd.print(CH[A]);break;

 case '#' :A=A-1;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Channel");lcd.setCursor(0,
1);lcd.print(CH[A]);break;

 case 'D' :P4[p44]=1;lcd.setCursor(0, 0);lcd.print("Select Channel");lcd.setCursor(0,
1);lcd.print(CH[A]);lcd.setCursor(10,1);lcd.print("Selected");p44=p44+1;break;

 case 'C' :P4[p44]=0;lcd.setCursor(0, 0);lcd.print("Select Channel");lcd.setCursor(0,
1);lcd.print(CH[A]);lcd.setCursor(10,1);lcd.print("NoSelected");p44=p44+1;break;

 case '2' : i=10;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Meal");break;
```

```
  case '1' : i=5;lcd.clear();break;
  case '*' : i=9;p44=1;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Channel");p44=1;break;
  default : ;break;}
 }

  else{
    char ekey = keypad.getKey();
switch (ekey){
case '4': I=0;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Meal");lcd.setCursor(0,
1);lcd.print("Breakfast");break;
case '5': I=1;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Meal");lcd.setCursor(0,
1);lcd.print("Lunch");break;
case '6' :I=2;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Meal");lcd.setCursor(0,
1);lcd.print("Dinner");break;
case 'B' :I=3;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Meal");lcd.setCursor(0,
1);lcd.print("Bedtime");break;
case '2' : i=10;lcd.clear();lcd.setCursor(0, 0);lcd.print("Select Meal");break;
case 'A' : i=I;lcd.clear();break;
case '1' : i=5;lcd.clear();break;
default: ; break;}}

int M11=M1-30;
int M12,H12;
if(M11>=0){
  M12=M11;
```

```
  H12=H1;}
 else if(M11<0){
  M12=60+M11;
  H12=H1-1;}
 else{;}

 int M21=M2-30;
 int M22,H22;
 if(M21>=0){
  M22=M21;
  H22=H2;}
 else if(M21<0){
  M22=60+M21;
  H22=H2-1;}
 else{;}

int M31=M3-30;
int M32,H32;
if(M31>=0){
  M32=M31;
  H32=H3;}
else if(M31<0){
  M32=60+M31;
  H32=H3-1;}
```

```
 else{;}


 if(k%5==0){
 Serial.print(P1[1]);
 Serial.print(P1[2]);
 Serial.print(P1[3]);
 Serial.println(P1[4]);
 Serial.print(P2[1]);
 Serial.print(P2[2]);
 Serial.print(P2[3]);
 Serial.println(P2[4]);
 Serial.print(P3[1]);
 Serial.print(P3[2]);
 Serial.print(P3[3]);
 Serial.println(P3[4]);
 Serial.print(P4[1]);
 Serial.print(P4[2]);
 Serial.print(P4[3]);
 Serial.println(P4[4]);
 }
 else{k=k+1;}


 t = rtc.getTime();
Hor = t.hour;
```

```
Min = t.min;

Sec = t.sec;


if(Hor==H1 && Min==M1 && Sec==0){

  p1=P1[1]+(2*P1[2])+(4*P1[3])+(8*P1[4]);

  switch(p1){

case 0 : digitalWrite(46,LOW);digitalWrite(48,LOW);digitalWrite(50,LOW);digitalWrite(52,LOW); break;

case 1 :
digitalWrite(46,HIGH);digitalWrite(48,LOW);digitalWrite(50,LOW);digitalWrite(52,LOW);digitalWrite(38,H
IGH);break;

case 2 :
digitalWrite(46,LOW);digitalWrite(48,HIGH);digitalWrite(50,LOW);digitalWrite(52,LOW);digitalWrite(38,H
IGH);break;

case 3 :
digitalWrite(46,HIGH);digitalWrite(48,HIGH);digitalWrite(50,LOW);digitalWrite(52,LOW);digitalWrite(38,
HIGH);break;

case 4 :
digitalWrite(46,LOW);digitalWrite(48,LOW);digitalWrite(50,HIGH);digitalWrite(52,LOW);digitalWrite(38,H
IGH);break;

case 5 :
digitalWrite(46,HIGH);digitalWrite(48,LOW);digitalWrite(50,HIGH);digitalWrite(52,LOW);digitalWrite(38,
HIGH);break;

case 6 :
digitalWrite(46,LOW);digitalWrite(48,HIGH);digitalWrite(50,HIGH);digitalWrite(52,LOW);digitalWrite(38,
HIGH);break;

case 7 :
digitalWrite(46,HIGH);digitalWrite(48,HIGH);digitalWrite(50,HIGH);digitalWrite(52,LOW);digitalWrite(38,
HIGH);break;
```

```
 case 8 :
 digitalWrite(46,LOW);digitalWrite(48,LOW);digitalWrite(50,LOW);digitalWrite(52,HIGH);digitalWrite(38,H
 IGH);break;

 case 9 :
 digitalWrite(46,HIGH);digitalWrite(48,LOW);digitalWrite(50,LOW);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 case 10 :
 digitalWrite(46,LOW);digitalWrite(48,HIGH);digitalWrite(50,LOW);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 case 11 :
 digitalWrite(46,HIGH);digitalWrite(48,HIGH);digitalWrite(50,LOW);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 case 12 :
 digitalWrite(46,LOW);digitalWrite(48,LOW);digitalWrite(50,HIGH);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 case 13 :
 digitalWrite(46,HIGH);digitalWrite(48,LOW);digitalWrite(50,HIGH);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 case 14 :
 digitalWrite(46,LOW);digitalWrite(48,HIGH);digitalWrite(50,HIGH);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 case 15 :
 digitalWrite(46,HIGH);digitalWrite(48,HIGH);digitalWrite(50,HIGH);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 default: ;  }

 }

 else{;}
```

```
if(Hor==H2 && Min==M2 && Sec==0){
  p2=P2[1]+(2*P2[2])+(4*P2[3])+(8*P2[4]);
  switch(p2){
case 0 : digitalWrite(46,LOW);digitalWrite(48,LOW);digitalWrite(50,LOW);digitalWrite(52,LOW); break;
case 1 :
digitalWrite(46,HIGH);digitalWrite(48,LOW);digitalWrite(50,LOW);digitalWrite(52,LOW);digitalWrite(38,H
IGH);break;
case 2 :
digitalWrite(46,LOW);digitalWrite(48,HIGH);digitalWrite(50,LOW);digitalWrite(52,LOW);digitalWrite(38,H
IGH);break;
case 3 :
digitalWrite(46,HIGH);digitalWrite(48,HIGH);digitalWrite(50,LOW);digitalWrite(52,LOW);digitalWrite(38,
HIGH);break;
case 4 :
digitalWrite(46,LOW);digitalWrite(48,LOW);digitalWrite(50,HIGH);digitalWrite(52,LOW);digitalWrite(38,H
IGH);break;
case 5 :
digitalWrite(46,HIGH);digitalWrite(48,LOW);digitalWrite(50,HIGH);digitalWrite(52,LOW);digitalWrite(38,
HIGH);break;
case 6 :
digitalWrite(46,LOW);digitalWrite(48,HIGH);digitalWrite(50,HIGH);digitalWrite(52,LOW);digitalWrite(38,
HIGH);break;
case 7 :
digitalWrite(46,HIGH);digitalWrite(48,HIGH);digitalWrite(50,HIGH);digitalWrite(52,LOW);digitalWrite(38,
HIGH);break;
case 8 :
digitalWrite(46,LOW);digitalWrite(48,LOW);digitalWrite(50,LOW);digitalWrite(52,HIGH);digitalWrite(38,H
IGH);break;
```

```
 case 9 :
 digitalWrite(46,HIGH);digitalWrite(48,LOW);digitalWrite(50,LOW);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 case 10 :
 digitalWrite(46,LOW);digitalWrite(48,HIGH);digitalWrite(50,LOW);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 case 11 :
 digitalWrite(46,HIGH);digitalWrite(48,HIGH);digitalWrite(50,LOW);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 case 12 :
 digitalWrite(46,LOW);digitalWrite(48,LOW);digitalWrite(50,HIGH);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 case 13 :
 digitalWrite(46,HIGH);digitalWrite(48,LOW);digitalWrite(50,HIGH);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 case 14 :
 digitalWrite(46,LOW);digitalWrite(48,HIGH);digitalWrite(50,HIGH);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 case 15 :
 digitalWrite(46,HIGH);digitalWrite(48,HIGH);digitalWrite(50,HIGH);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 default: ;  }

 }

 else{;}


 if(Hor==H3 && Min==M3 && Sec==0){

   p3=P3[1]+(2*P3[2])+(4*P3[3])+(8*P3[4]);

   switch(p3){
```

```
case 0 : digitalWrite(46,LOW);digitalWrite(48,LOW);digitalWrite(50,LOW);digitalWrite(52,LOW); break;

case 1 :
digitalWrite(46,HIGH);digitalWrite(48,LOW);digitalWrite(50,LOW);digitalWrite(52,LOW);digitalWrite(38,H
IGH);break;

case 2 :
digitalWrite(46,LOW);digitalWrite(48,HIGH);digitalWrite(50,LOW);digitalWrite(52,LOW);digitalWrite(38,H
IGH);break;

case 3 :
digitalWrite(46,HIGH);digitalWrite(48,HIGH);digitalWrite(50,LOW);digitalWrite(52,LOW);digitalWrite(38,
HIGH);break;

case 4 :
digitalWrite(46,LOW);digitalWrite(48,LOW);digitalWrite(50,HIGH);digitalWrite(52,LOW);digitalWrite(38,H
IGH);break;

case 5 :
digitalWrite(46,HIGH);digitalWrite(48,LOW);digitalWrite(50,HIGH);digitalWrite(52,LOW);digitalWrite(38,
HIGH);break;

case 6 :
digitalWrite(46,LOW);digitalWrite(48,HIGH);digitalWrite(50,HIGH);digitalWrite(52,LOW);digitalWrite(38,
HIGH);break;

case 7 :
digitalWrite(46,HIGH);digitalWrite(48,HIGH);digitalWrite(50,HIGH);digitalWrite(52,LOW);digitalWrite(38,
HIGH);break;

case 8 :
digitalWrite(46,LOW);digitalWrite(48,LOW);digitalWrite(50,LOW);digitalWrite(52,HIGH);digitalWrite(38,H
IGH);break;

case 9 :
digitalWrite(46,HIGH);digitalWrite(48,LOW);digitalWrite(50,LOW);digitalWrite(52,HIGH);digitalWrite(38,
HIGH);break;
```

```
 case 10 :
 digitalWrite(46,LOW);digitalWrite(48,HIGH);digitalWrite(50,LOW);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 case 11 :
 digitalWrite(46,HIGH);digitalWrite(48,HIGH);digitalWrite(50,LOW);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 case 12 :
 digitalWrite(46,LOW);digitalWrite(48,LOW);digitalWrite(50,HIGH);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 case 13 :
 digitalWrite(46,HIGH);digitalWrite(48,LOW);digitalWrite(50,HIGH);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 case 14 :
 digitalWrite(46,LOW);digitalWrite(48,HIGH);digitalWrite(50,HIGH);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 case 15 :
 digitalWrite(46,HIGH);digitalWrite(48,HIGH);digitalWrite(50,HIGH);digitalWrite(52,HIGH);digitalWrite(38,
 HIGH);break;

 default: ;  }

 }

 else{;}


 if(Hor==H4 && Min==M4 && Sec==0){

  p4=P4[1]+(2*P4[2])+(4*P4[3])+(8*P4[4]);

  switch(p4){

case 0 : digitalWrite(46,LOW);digitalWrite(48,LOW);digitalWrite(50,LOW);digitalWrite(52,LOW); break;
```

case 1 :

digitalWrite(46,HIGH);digitalWrite(48,LOW);digitalWrite(50,LOW);digitalWrite(52,LOW);digitalWrite(38,HIGH);break;

case 2 :

digitalWrite(46,LOW);digitalWrite(48,HIGH);digitalWrite(50,LOW);digitalWrite(52,LOW);digitalWrite(38,HIGH);break;

case 3 :

digitalWrite(46,HIGH);digitalWrite(48,HIGH);digitalWrite(50,LOW);digitalWrite(52,LOW);digitalWrite(38,HIGH);break;

case 4 :

digitalWrite(46,LOW);digitalWrite(48,LOW);digitalWrite(50,HIGH);digitalWrite(52,LOW);digitalWrite(38,HIGH);break;

case 5 :

digitalWrite(46,HIGH);digitalWrite(48,LOW);digitalWrite(50,HIGH);digitalWrite(52,LOW);digitalWrite(38,HIGH);break;

case 6 :

digitalWrite(46,LOW);digitalWrite(48,HIGH);digitalWrite(50,HIGH);digitalWrite(52,LOW);digitalWrite(38,HIGH);break;

case 7 :

digitalWrite(46,HIGH);digitalWrite(48,HIGH);digitalWrite(50,HIGH);digitalWrite(52,LOW);digitalWrite(38,HIGH);break;

case 8 :

digitalWrite(46,LOW);digitalWrite(48,LOW);digitalWrite(50,LOW);digitalWrite(52,HIGH);digitalWrite(38,HIGH);break;

case 9 :

digitalWrite(46,HIGH);digitalWrite(48,LOW);digitalWrite(50,LOW);digitalWrite(52,HIGH);digitalWrite(38,HIGH);break;

```
case 10 :
digitalWrite(46,LOW);digitalWrite(48,HIGH);digitalWrite(50,LOW);digitalWrite(52,HIGH);digitalWrite(38,
HIGH);break;

case 11 :
digitalWrite(46,HIGH);digitalWrite(48,HIGH);digitalWrite(50,LOW);digitalWrite(52,HIGH);digitalWrite(38,
HIGH);break;

case 12 :
digitalWrite(46,LOW);digitalWrite(48,LOW);digitalWrite(50,HIGH);digitalWrite(52,HIGH);digitalWrite(38,
HIGH);break;

case 13 :
digitalWrite(46,HIGH);digitalWrite(48,LOW);digitalWrite(50,HIGH);digitalWrite(52,HIGH);digitalWrite(38,
HIGH);break;

case 14 :
digitalWrite(46,LOW);digitalWrite(48,HIGH);digitalWrite(50,HIGH);digitalWrite(52,HIGH);digitalWrite(38,
HIGH);break;

case 15 :
digitalWrite(46,HIGH);digitalWrite(48,HIGH);digitalWrite(50,HIGH);digitalWrite(52,HIGH);digitalWrite(38,
HIGH);break;

default: ; }

}

else{;}


int a1=digitalRead(30);

int a2=digitalRead(32);

int a3=digitalRead(34);

int a4=digitalRead(36);
```

```
if (a1==0){digitalWrite(46,LOW);
digitalWrite(38,LOW);
digitalWrite(22,HIGH);
p1=0;
}
else{digitalWrite(22,LOW);}

if (a2==0){digitalWrite(48,LOW);
digitalWrite(38,LOW);
digitalWrite(24,HIGH);
p2=0;
}
else{digitalWrite(24,LOW);}

if (a3==0){digitalWrite(50,LOW);
digitalWrite(38,LOW);
digitalWrite(26,HIGH);
p3=0;
}
else{digitalWrite(26,LOW);}

if (a4==0){digitalWrite(52,LOW);
digitalWrite(38,LOW);
digitalWrite(28,HIGH);
```

```
p4=0;

}

else{digitalWrite(28,LOW);}


Serial.print(a);

 }
```

ชุดคำสั่งบน Wemos D1 Mini

```
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include <stdio.h>
#include <time.h>

// Set these to run example.
#define FIREBASE_HOST "multipledata-8e29f.firebaseio.com"
#define FIREBASE_AUTH "KX8dRpFch7fhMoYfNzNfvNVay7XOXaccUtTBE4wk"
#define WIFI_SSID "BMEKMITL"
#define WIFI_PASSWORD "KMITLBME"

 int L1;
  int L4;
  int L;
  int p;
int timezone = 7 * 3600;                    //ตั้งค่า TimeZone ตามเวลาประเทศไทย
int dst = 0;

void setup() {
  pinMode(D1, INPUT);
  pinMode(D2, INPUT);
  pinMode(D3, INPUT);
```

```
 pinMode(D4, INPUT);
   pinMode(D5, INPUT),
   pinMode(D6, INPUT);
   pinMode(D7, INPUT);
   pinMode(D8, INPUT);
   pinMode(D0, OUTPUT);

 Serial.begin(115200);
Serial.setDebugOutput(true);

 // connect to wifi.
 WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
 Serial.print("connecting");
 while (WiFi.status() != WL_CONNECTED) {
   Serial.print(".");
   delay(500);
}
Serial.println();
//digitalWrite(D0,HIGH);
Serial.print("connected: ");
Serial.println(WiFi.localIP());

  configTime(timezone, dst, "pool.ntp.org", "time.nist.gov");     //ดึงเวลาจาก Server
  Serial.println("\nWaiting for time");
```

```
    while (!time(nullptr)) {
      Serial.print(".");
      delay(1000);              }
    Serial.println("");
    if(WiFi.status()==WL_CONNECTED){digitalWrite(D0,HIGH);}
    else{digitalWrite(D0,LOW);}
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);              }

void loop() {
configTime(timezone, dst, "pool.ntp.org", "time.nist.gov");    //ดึงเวลาปัจจุบันจาก Server อีกครั้ง
  time_t now = time(nullptr);
  struct tm* p_tm = localtime(&now);

  String daytime = String(p_tm->tm_mday)+"/"+String(p_tm->tm_mon+1)+"/"+String(p_tm-
>tm_year+1900)+"  "+String(p_tm->tm_hour)+":"+String(p_tm->tm_min)+":"+String(p_tm->tm_sec);
  String date1 = daytime+"  Channel 1 is opened";
  String date2 = daytime+"  Channel 2 is opened";
  String date3 = daytime+"  Channel 3 is opened";
  String date4 = daytime+"  Channel 4 is opened";

int a1 = digitalRead(D1);
int a2 = digitalRead(D2);
int a3 = digitalRead(D3);
int a4 = digitalRead(D4);
```

```
  if (Firebase.failed()) {
      Serial.print("setting /number failed:");
      Serial.println(Firebase.error());
      return;
  }

int a=a1+(2*a2)+(4*a3)+(8*a4);

int p1 = digitalRead(D5);
int p2 = digitalRead(D6);
int p3 = digitalRead(D7);
int p4 = digitalRead(D8);

p=p1+(2*p2)+(4*p3)+(8*p4);

switch(a){
case 0 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "All channel is closed"); break;
case 1 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 1 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date1);break;
case 2 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 2 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date2);break;
```

case 3 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 1 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date1);delay(1000);

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 2 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date2);delay(1000);break;

case 4 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 3 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date3);break;

case 5 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 1 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date1);delay(1000);

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 3 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date3);delay(1000);break;

case 6 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 2 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date2);delay(1000);

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 3 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date3);delay(1000);break;

case 7 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 1 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date1);delay(1000);

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 2 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date2);delay(1000);

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 3 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date3);delay(1000);break;

case 8 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 4 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date4);break;

case 9 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 1 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date1);delay(1000);

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 4 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date4);delay(1000);break;

case 10 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 1 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date2);delay(1000);

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 4 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date4);delay(1000);break;

case 11 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 1 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date1);delay(1000);

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 2 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date2);delay(1000);

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 4 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date4);delay(1000);break;

case 12 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 3 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date3);delay(1000);

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 4 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date4);delay(1000);break;

case 13 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 1 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date1);delay(1000);

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 3 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date3);delay(1000);

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 4 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date4);delay(1000);break;

case 14 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 4 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date4);delay(1000);

```
Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 2 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date2);delay(1000);

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 3 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date3);delay(1000);break;

case 15 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 1 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date1);delay(1000);

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 2 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date2);delay(1000);

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 3 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date3);delay(1000);

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", "Channel 4 is opened");Firebase.pushString("8fZBhILdK2bZMywnqMShQnJXKqP2/History", date4);delay(1000);break;

default: Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Channel Status", " "); break; }


 switch(p){

case 0 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", " ");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status", "0"); break;

case 1 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 1");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status", "1");break;

case 2 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 2");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status", "1");break;

case 3 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 1");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 2");

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status", "1");break;
```

case 4 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 3");break;

case 5 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 1");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 3");

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status", "1");break;

case 6 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 2");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 3");

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status", "1");break;

case 7 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 1");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 2");

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 3");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status", "1");break;

case 8 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 4");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status", "1");break;

case 9 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 1");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 4");

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status", "1");break;

case 10 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 2");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 4");

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status", "1");break;

case 11 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 1");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 2");

Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 4");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status", "1");break;

```
case 12 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill
channel 3");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill
channel 4");

        Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status", "1");break;

case 13 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill
channel 1");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill
channel 3");

        Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel
4");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status", "1");break;

case 14 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill
channel 2");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill
channel 3");

        Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel
4");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status", "1");break;

case 15 : Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill
channel 1");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill
channel 2");

        Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel
3");Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", "Please take pill channel 4");

        Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status", "1");break;

default: Firebase.setString("8fZBhILdK2bZMywnqMShQnJXKqP2/Status2", " "); break; }
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.lalabelle.multipledata">
    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="com.android.alarm.permission.SET_ALARM"
/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/druglogo"
        android:label="Belmoz"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".Auth">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <service android:name=".MyFireBaseInstanceIDService">
            <intent-filter>
                <action android:name="com.google.firebase.INSTANCE_ID_EVENT"
/>
            </intent-filter>
        </service>
        <service
            android:name=".MyFirebaseMessagingService"
            android:enabled="true"
            android:exported="true">
            <intent-filter>
                <action android:name="com.google.firebase.MESSAGING_EVENT"
/>
            </intent-filter>
        </service>

        <activity android:name=".SignUpActivity" />
        <activity android:name=".Activity_multiple" />
        <activity android:name=".Activity_getdata" />
        <activity android:name=".Activity_general" />
        <activity android:name=".Activity_bmsplash" />
        <activity android:name=".Activity_alarm" />
        <receiver android:name=".Notification_receiver" /></activity>
    </application>

</manifest>
```

build.gradle

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 26
    defaultConfig {
        applicationId "com.example.lalabelle.multipledata"
        minSdkVersion 15
        targetSdkVersion 26
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner
"android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'),
'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.google.firebase:firebase-database:11.0.4'
    implementation 'com.android.support.constraint:constraint-layout:1.0.2'
    implementation 'com.google.android.gms:play-services-basement:11.0.4'
    implementation 'com.google.firebase:firebase-messaging:11.0.4'
    implementation 'com.google.firebase:firebase-auth:11.0.4'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.1'
    androidTestImplementation 'com.android.support.test.espresso:espresso-
core:3.0.1'
}
apply plugin: 'com.google.gms.google-services'
```

Auth.java

```
package com.example.lalabelle.multipledata;

import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.util.Patterns;
import android.view.View;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

public class Auth extends AppCompatActivity implements View.OnClickListener
{

    FirebaseAuth mAuth;
    EditText editTextEmail , editTextPassword;
    ProgressBar progressBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_auth);

        mAuth = FirebaseAuth.getInstance();

        editTextEmail = (EditText) findViewById(R.id.editTextEmail);
        editTextPassword = (EditText) findViewById(R.id.editTextPassword);
        progressBar = (ProgressBar) findViewById(R.id.progressbar);

        findViewById(R.id.textViewSignUp).setOnClickListener(this);
        findViewById(R.id.buttonLogin).setOnClickListener(this);


    }

    @Override
    public void onClick(View view) {
        switch (view.getId()){
            case R.id.textViewSignUp:
                startActivity(new Intent(this,SignUpActivity.class));
                break;

            case R.id.buttonLogin:
                userLogin();
                break;
        }
    }

    private void userLogin() {

        String email = editTextEmail.getText().toString().trim();
        String password = editTextPassword.getText().toString().trim();

        if (email.isEmpty()) {
```

```
            editTextEmail.setError("Email is required");
            editTextEmail.requestFocus();
            return;
        }

        if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
            editTextEmail.setError("Please enter a valid email");
            editTextEmail.requestFocus();
            return;
        }

        if (password.isEmpty()) {
            editTextPassword.setError("Password is required");
            editTextPassword.requestFocus();
            return;
        }

        if (password.length() < 6) {
            editTextPassword.setError("Minimum lenght of password should be
6");
            editTextPassword.requestFocus();
            return;
        }

        progressBar.setVisibility(View.VISIBLE);


mAuth.signInWithEmailAndPassword(email,password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                progressBar.setVisibility(View.GONE);
                if(task.isSuccessful()) {
                    Intent intent = new
Intent(Auth.this,Activity_bmsplash.class);
                    intent .addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                    startActivity(intent);

                }else{

Toast.makeText(getApplicationContext(),task.getException().getMessage(),Toas
t.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```

activity_auth.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

    tools:context="com.example.lalabelle.multipledata.Auth">

    <EditText
        android:id="@+id/editTextEmail"
        android:layout_width="311dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:ems="10"

        android:hint="email"
        android:inputType="textEmailAddress"
        app:layout_constraintBottom_toTopOf="@+id/editTextPassword"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.953" />

    <EditText
        android:id="@+id/editTextPassword"
        android:layout_width="313dp"
        android:layout_height="48dp"
        android:layout_marginBottom="40dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:ems="10"
        android:hint="password"
        android:inputType="textPassword"
        app:layout_constraintBottom_toTopOf="@+id/buttonLogin"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.494"
        app:layout_constraintStart_toStartOf="parent" />

    <Button
        android:id="@+id/buttonLogin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="36dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:text="Log in"
        android:background="@drawable/buttonstyle2"
        app:layout_constraintBottom_toTopOf="@+id/textViewSignUp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent" />

    <TextView
        android:id="@+id/textViewSignUp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:layout_marginBottom="100dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:text="Don't have account? Sign Up"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
        app:layout_constraintStart_toStartOf="parent" />

    <ProgressBar
        android:id="@+id/progressbar"
        style="?android:attr/progressBarStyle"
        android:layout_width="37dp"
        android:layout_height="25dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:visibility="gone"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.908"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.049" />

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="304dp"
        android:layout_height="75dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:src="@drawable/belmozz"
        app:layout_constraintBottom_toTopOf="@+id/editTextEmail"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.465" />

    <TextView
        android:id="@+id/textView9"
        android:layout_width="271dp"
        android:layout_height="18dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:text="Developing The Medicine Reminder System"
        android:textAppearance="@style/TextAppearance.AppCompat.Small"
        android:textStyle="italic"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.829"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView2" />

</android.support.constraint.ConstraintLayout>
```

SignUpActivity .java

```
package com.example.lalabelle.multipledata;

import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.util.Patterns;
import android.view.View;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseAuthUserCollisionException;

public class SignUpActivity extends AppCompatActivity implements
View.OnClickListener {

    EditText editTextEmail , editTextPassword;

    ProgressBar progressBar;

    private FirebaseAuth mAuth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sign_up);

        editTextEmail = (EditText) findViewById(R.id.editTextEmail);
        editTextPassword = (EditText) findViewById(R.id.editTextPassword);
        progressBar = (ProgressBar) findViewById(R.id.progressbar);

        mAuth = FirebaseAuth.getInstance();

        findViewById(R.id.buttonSignUp).setOnClickListener(this);
        findViewById(R.id.textViewLogin).setOnClickListener(this);

    }

    private void registerUser(){
        String email = editTextEmail.getText().toString().trim();
        String password = editTextPassword.getText().toString().trim();

        if (email.isEmpty()) {
            editTextEmail.setError("Email is required");
            editTextEmail.requestFocus();
            return;
        }

        if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
            editTextEmail.setError("Please enter a valid email");
            editTextEmail.requestFocus();
            return;
        }

        if (password.isEmpty()) {
            editTextPassword.setError("Password is required");
```

```
            editTextPassword.requestFocus();
            return;
        }

        if (password.length() < 6) {
            editTextPassword.setError("Minimum lenght of password should be
6");
            editTextPassword.requestFocus();
            return;
        }

        progressBar.setVisibility(View.VISIBLE);


mAuth.createUserWithEmailAndPassword(email,password).addOnCompleteListener(n
ew OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                progressBar.setVisibility(View.GONE);

                if(task.isSuccessful()){
                    Intent intent = new
Intent(SignUpActivity.this,Activity_bmsplash.class);
                    intent .addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                    startActivity(intent);
                }
                else {

                    if(task.getException() instanceof
FirebaseAuthUserCollisionException) {

                        Toast.makeText(getApplicationContext(), "You are
already registered", Toast.LENGTH_SHORT).show();
                    }
                    else

Toast.makeText(getApplicationContext(),task.getException().getMessage(),Toas
t.LENGTH_SHORT).show();

                    }
                }
            }
        );
    }

    @Override
    public void onClick(View view) {
        switch (view.getId()){
            case R.id.buttonSignUp:
                registerUser();
                break;

            case R.id.textViewLogin:
                startActivity(new Intent(this,Auth.class));
                break;
        }
    }
}
```

activity_sign_up.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.lalabelle.multipledata.SignUpActivity">

    <EditText
        android:id="@+id/editTextEmail"
        android:layout_width="288dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:ems="10"
        android:hint="email"
        android:inputType="textEmailAddress"
        app:layout_constraintBottom_toTopOf="@+id/textView8"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.495"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView7"
        app:layout_constraintVertical_bias="0.0" />

    <EditText
        android:id="@+id/editTextPassword"
        android:layout_width="288dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="100dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:ems="10"
        android:hint="password"
        android:inputType="textPassword"
        app:layout_constraintBottom_toTopOf="@+id/buttonSignUp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <Button
        android:id="@+id/buttonSignUp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="36dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:background="@drawable/buttonstyle2"
        android:text="Sign Up"
        app:layout_constraintBottom_toTopOf="@+id/textViewLogin"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent" />

    <TextView
        android:id="@+id/textViewLogin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="100dp"
        android:layout_marginEnd="8dp"
```

```
        android:layout_marginStart="8dp"
        android:text="You already sing up? Login"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <ProgressBar
        android:id="@+id/progressbar"
        style="?android:attr/progressBarStyle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:visibility="gone"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.934"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.041" />

    <TextView
        android:id="@+id/textView5"
        android:layout_width="324dp"
        android:layout_height="33dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="52dp"
        android:text="Create your account"

android:textAppearance="@style/TextAppearance.AppCompat.Light.SearchResult.T
itle"
        android:textColorLink="@color/colorPrimaryDark"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:textColorLink="@color/colorPrimary" />

    <TextView
        android:id="@+id/textView7"
        android:layout_width="290dp"
        android:layout_height="30dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="44dp"
        android:text="Please put your e-mail here"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView5" />

    <TextView
        android:id="@+id/textView8"
        android:layout_width="290dp"
        android:layout_height="30dp"
        android:layout_marginBottom="16dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:text="Plase put your password here (at lease 6)"
        app:layout_constraintBottom_toTopOf="@+id/editTextPassword"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
</android.support.constraint.ConstraintLayout>
```

Activity_multiple.java

```
package com.example.lalabelle.multipledata;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class Activity_multiple extends AppCompatActivity {

    private DatabaseReference mDatabase,
mDatabase2,mDatabase3,mDatabase4,mDatabase5,mDatabase6;
    private DatabaseReference
mDatabase7,mDatabase8,mDatabase9,mDatabase10,mDatabase11,mDatabase12;
    private DatabaseReference
mDatabase13,mDatabase14,mDatabase15,mDatabase16,mDatabase17,mDatabase18;
    private TextView mNameView, mNameView2 ,mNameView3;
    private TextView mQuantity, mQuantity2, mQuantity3;
    private TextView mDqview1, mDqview2, mDqview3;
    private TextView mdetail1,mdetail2,mdetail3;
    String uid;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_multiple);

        FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
        if (user != null) {
            uid = user.getUid();
        }

        mDatabase =
FirebaseDatabase.getInstance().getReference().child(uid).child("ชื่อยา1");
        mDatabase2 =
FirebaseDatabase.getInstance().getReference().child(uid).child("การรับประทาน
1");
        mDatabase7 =
FirebaseDatabase.getInstance().getReference().child(uid).child("หมายเหตุ1");
        mDatabase3 =
FirebaseDatabase.getInstance().getReference().child(uid).child("ชื่อยา2");
        mDatabase4 =
FirebaseDatabase.getInstance().getReference().child(uid).child("การรับประทาน
2");
        mDatabase8 =
FirebaseDatabase.getInstance().getReference().child(uid).child("หมายเหตุ2");
        mDatabase5 =
FirebaseDatabase.getInstance().getReference().child(uid).child("ชื่อยา3");
        mDatabase6 =
FirebaseDatabase.getInstance().getReference().child(uid).child("การรับประทาน
3");
```

```
        mDatabase9 =
FirebaseDatabase.getInstance().getReference().child(uid).child("หมายเหตุ3");

        mDatabase10 =
FirebaseDatabase.getInstance().getReference().child(uid).child("วันที่ได้รับยา
,ปริมาณ1");
        mDatabase11 =
FirebaseDatabase.getInstance().getReference().child(uid).child("วันที่ได้รับยา
,ปริมาณ2");
        mDatabase12 =
FirebaseDatabase.getInstance().getReference().child(uid).child("วันที่ได้รับยา
,ปริมาณ3");



        mNameView = (TextView) findViewById(R.id.name_view);
        mQuantity = (TextView) findViewById(R.id.quan_view);
        mdetail1 = (TextView) findViewById(R.id.detail_view);
        mNameView2 = (TextView) findViewById(R.id.name2_view);
        mQuantity2 = (TextView) findViewById(R.id.quan2_view);
        mdetail2 = (TextView) findViewById(R.id.detail2_view);
        mNameView3 = (TextView) findViewById(R.id.name3_view);
        mQuantity3 = (TextView) findViewById(R.id.quan3_view);
        mdetail3 = (TextView) findViewById(R.id.detail3_view);

        mDqview1 = (TextView) findViewById(R.id.dq1_view);
        mDqview2 = (TextView) findViewById(R.id.dq2_view);
        mDqview3 = (TextView) findViewById(R.id.dq3_view);


        mDatabase.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {

                String name = dataSnapshot.getValue().toString();
                mNameView.setText("ชื่อยา1 : " + name);

            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });



        mDatabase2.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {

                String taking = dataSnapshot.getValue().toString();
                mQuantity.setText("การรับประทาน1 : " + taking);

            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
```

```
});

mDatabase7.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {

        String detail1 = dataSnapshot.getValue().toString();
        mdetail1.setText("เพิ่มเติม1 : " + detail1);

    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

    }
});


mDatabase3.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {

        String name2 = dataSnapshot.getValue().toString();
        mNameView2.setText("ชื่อยา2 : " + name2);

    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

    }
});


mDatabase4.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {

        String taking2 = dataSnapshot.getValue().toString();
        mQuantity2.setText("การรับประทาน2 : " + taking2);

    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

    }
});

mDatabase8.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {

        String detail2 = dataSnapshot.getValue().toString();
        mdetail2.setText("เพิ่มเติม2 : " + detail2);

    }

    @Override
    public void onCancelled(DatabaseError databaseError) {
```

```
    }
});

mDatabase5.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {

        String name3 = dataSnapshot.getValue().toString();
        mNameView3.setText("ชื่อยา3 : " + name3);

    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

    }
});

mDatabase6.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {

        String taking3 = dataSnapshot.getValue().toString();
        mQuantity3.setText("การรับประทาน3 : " + taking3);

    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

    }
});

mDatabase9.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {

        String detail3 = dataSnapshot.getValue().toString();
        mdetail3.setText("เพิ่มเติม3 : " + detail3);

    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

    }
});

mDatabase10.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {

        String name = dataSnapshot.getValue().toString();
        mDqview1.setText("วันที่ได้รับยา/ปริมาณ1 : " + name);

    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

    }
});
```

```
        mDatabase11.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {

                String name = dataSnapshot.getValue().toString();
                mDqview2.setText("วันที่ได้รับยา/ปริมาณ2 : " + name);

            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });

        mDatabase12.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {

                String name = dataSnapshot.getValue().toString();
                mDqview3.setText("วันที่ได้รับยา/ปริมาณ2 : " + name);

            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });


    }

}
```

activity_multiple.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.lalabelle.multipledata.Activity_multiple">


    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="32dp"
        android:selectAllOnFocus="false"
```

```
        android:text="MEDICINE INFORMATION"
        android:textAllCaps="false"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textColorLink="@color/green"
        android:textIsSelectable="false"
        android:textStyle="bold|italic"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ScrollView
        android:id="@+id/scrollView3"
        android:layout_width="379dp"
        android:layout_height="517dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="24dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="16dp"
            android:orientation="vertical"
            tools:layout_editor_absoluteX="24dp"
            tools:layout_editor_absoluteY="50dp">

            <TextView
                android:id="@+id/name_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="20dp"
                android:text="ชื่อยา:"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                android:textStyle="bold"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/textView" />

            <TextView
                android:id="@+id/quan_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="16dp"
                android:text="การรับประทาน:"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/name_view" />

            <TextView
                android:id="@+id/dql_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
```

```
                android:layout_marginStart="8dp"
                android:layout_marginTop="16dp"
                android:text="วันที่ได้รับยา,ปริมาณ:"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/quan_view_view" />

            <TextView
                android:id="@+id/ch_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="16dp"
                android:text="หมายเลขช่อง: 1"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/dq1_view" />

            <TextView
                android:id="@+id/detail_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="16dp"
                android:text="หมายเหตุ:"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/ch_view" />



            <TextView
                android:id="@+id/name2_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="40dp"
                android:text="ชื่อยา:"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                android:textStyle="bold"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/quan_view" />

            <TextView
                android:id="@+id/quan2_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="16dp"
```

```
                android:text="การรับประทาน:"

 android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/name2_view" />

            <TextView
                android:id="@+id/dq2_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="16dp"
                android:text="วันที่ได้รับยา,ปริมาณ:"

 android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/quan2_view" />

            <TextView
                android:id="@+id/ch2_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="16dp"
                android:text="หมายเลขช่อง: 2"

 android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/dq2_view" />

            <TextView
                android:id="@+id/detail2_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="16dp"
                android:text="หมายเหตุ:"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/ch2_view" />


            <TextView
                android:id="@+id/name3_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="40dp"
                android:text="ชื่อยา:"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                android:textStyle="bold"
                app:layout_constraintEnd_toEndOf="parent"
```

```
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/quan2_view" />

            <TextView
                android:id="@+id/quan3_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="16dp"
                android:text="การรับประทาน:"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/name3_view" />

            <TextView
                android:id="@+id/dq3_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="16dp"
                android:text="วันที่ได้รับยา,ปริมาณ:"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/quan3_view" />

            <TextView
                android:id="@+id/ch3_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="16dp"
                android:text="หมายเลขช่อง: 3"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/dq3_view" />

            <TextView
                android:id="@+id/detail3_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="16dp"
                android:text="หมายเหตุ:"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/ch3_view" />


            <TextView
                android:id="@+id/name4_view"
```

```
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="40dp"
                android:text="ชื่อยา:"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                android:textStyle="bold"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/quan3_view" />

            <TextView
                android:id="@+id/quan4_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="16dp"
                android:text="การรับประทาน:"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/name4_view" />

            <TextView
                android:id="@+id/dq4_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="16dp"
                android:text="วันที่ได้รับยา,ปริมาณ:"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/quan4_view" />

            <TextView
                android:id="@+id/ch4_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="16dp"
                android:text="หมายเลขช่อง:"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/dq4_view" />


            <TextView
                android:id="@+id/detail4_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
```

```
                android:layout_marginStart="8dp"
                android:layout_marginTop="16dp"
                android:text="หมวดหมู่:"

 android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/ch4_view" />

            <TextView
                android:id="@+id/free5_view"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="16dp"

 android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"

 app:layout_constraintTop_toBottomOf="@+id/detail4_view_view_view" />


        </LinearLayout>


    </ScrollView>

</android.support.constraint.ConstraintLayout>
```

Activity_getdata.java

```
package com.example.lalabelle.multipledata;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import java.util.ArrayList;
import java.util.List;

public class Activity_getdata extends AppCompatActivity {

    private DatabaseReference Hdatabase ;

    private TextView HnameView ;
    String uid;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_getdata);

        FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
        if (user != null) {
            uid = user.getUid();
        }

        Hdatabase =
FirebaseDatabase.getInstance().getReference().child(uid).child("History");
        HnameView = (TextView) findViewById((R.id.history1));

        Hdatabase.addValueEventListener((new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {

                        String history = dataSnapshot.getValue().toString();

                        HnameView.setText("History: "+ history);

                }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        }));
    }
}
```

activity_getdata.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.lalabelle.multipledata.Activity_getdata">


    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="32dp"
        android:text="HISTORY OF MEDICINE TAKING"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textStyle="bold|italic"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.496"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ScrollView
        android:id="@+id/scrollView2"
        android:layout_width="332dp"
        android:layout_height="514dp"
        android:layout_marginEnd="8dp"
```

```
        android:layout_marginStart="8dp"
        android:layout_marginTop="24dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView3">

        <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:orientation="vertical">


            <TextView
                android:id="@+id/history1"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:text="History: "
                android:inputType="textMultiLine"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                tools:layout_editor_absoluteY="135dp" />

        </LinearLayout>

    </ScrollView>

</android.support.constraint.ConstraintLayout>
```

Activity_general.java

```
package com.example.lalabelle.multipledata;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class Activity_general extends AppCompatActivity {

    private DatabaseReference GdatabaseUser,Gdatabase , Gdatabase2
,Gdatabase3,Gdatabase4;
    private TextView GnameviewUser, Gnameview , Gnameview2, Gnameview3,
Gnameview4;
    String uid;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_general);
```

```
        FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
        if (user != null) {
            uid = user.getUid();
        }

        GdatabaseUser =
FirebaseDatabase.getInstance().getReference().child(uid).child("Email");
        Gdatabase =
FirebaseDatabase.getInstance().getReference().child(uid).child("Name");
        Gdatabase2 =
FirebaseDatabase.getInstance().getReference().child(uid).child("HN");
        Gdatabase3 =
FirebaseDatabase.getInstance().getReference().child(uid).child("Next
Appointment");
        Gdatabase4 =
FirebaseDatabase.getInstance().getReference().child(uid).child("Medicine
Allergies");

        GnameviewUser = (TextView) findViewById((R.id.Pinfo));
        Gnameview = (TextView) findViewById((R.id.Pinfo1));
        Gnameview2 = (TextView) findViewById((R.id.Pinfo2));
        Gnameview3 = (TextView) findViewById((R.id.Pinfo3));
        Gnameview4 = (TextView) findViewById((R.id.pinmed));

        GdatabaseUser.addValueEventListener((new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                String Email = dataSnapshot.getValue().toString();
                GnameviewUser.setText("Email: " + Email);
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        }));

        Gdatabase.addValueEventListener((new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                String Name = dataSnapshot.getValue().toString();
                Gnameview.setText("Name: " + Name);
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        }));


        Gdatabase2.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                String hn = dataSnapshot.getValue().toString();
                Gnameview2.setText("HN: " + hn);
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });
```

```
        Gdatabase3.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                String Appointment = dataSnapshot.getValue().toString();
                Gnameview3.setText("Next Appointment: " + Appointment);
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });

        Gdatabase4.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                String medal = dataSnapshot.getValue().toString();
                Gnameview4.setText("Medicine Allergies: " + medal);
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });

    }
}
```

activity_general.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.lalabelle.multipledata.Activity_general">

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="32dp"
        android:text="General Information"
        android:textAllCaps="true"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textStyle="bold|italic"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
    android:id="@+id/Pinfo1"
    android:layout_width="317dp"
    android:layout_height="24dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="24dp"
    android:text="Name:  "
    android:textAppearance="@style/TextAppearance.AppCompat.Medium"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.509"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/Pinfo" />

<TextView
    android:id="@+id/Pinfo2"
    android:layout_width="314dp"
    android:layout_height="26dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="28dp"
    android:text="HN: "
    android:textAppearance="@style/TextAppearance.AppCompat.Medium"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/Pinfo1" />

<TextView
    android:id="@+id/Pinfo3"
    android:layout_width="312dp"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="24dp"
    android:text="Next Appointment: "
    android:textAppearance="@style/TextAppearance.AppCompat.Medium"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/pinmed" />

<TextView
    android:id="@+id/Pinfo"
    android:layout_width="319dp"
    android:layout_height="25dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="36dp"
    android:text="Email:"
    android:textAppearance="@style/TextAppearance.AppCompat.Medium"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.51"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2" />

<TextView
    android:id="@+id/pinmed"
    android:layout_width="314dp"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="24dp"
```

```
        android:text="Medicine Allergies:"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/Pinfo2" />

</android.support.constraint.ConstraintLayout>
```

Activity_bmsplash.java

```
package com.example.lalabelle.multipledata;
import android.annotation.SuppressLint;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Display;
import android.view.View;
import android.widget.Button;

public class Activity_bmsplash extends AppCompatActivity {

    public Button buttonsub;
    public Button button2;
    public Button button3;
    public Button button4;

        public void init(){
            buttonsub = (Button)findViewById(R.id.buttonsub);
            buttonsub.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {

                    Intent start_but = new
Intent(Activity_bmsplash.this,Activity_multiple.class);

                    startActivity(start_but);
                }
            });
        }


        public void init2(){
            button2 = (Button)findViewById(R.id.button2);
            button2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                    Intent start_but2 = new
Intent(Activity_bmsplash.this,Activity_general.class);

                    startActivity(start_but2);
            }
        });
    }


    public void init3(){
        button3 = (Button)findViewById(R.id.button3);
        button3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                Intent start_but3 = new
Intent(Activity_bmsplash.this,Activity_getdata.class);

                startActivity(start_but3);
            }
        });
    }

    public void init4(){
```

```
        button4 = (Button)findViewById(R.id.button4);
        button4.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                Intent start_but4 = new
Intent(Activity_bmsplash.this,Activity_alarm.class);

                startActivity(start_but4);
            }
        });
    }

    @SuppressLint("WrongViewCast")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);

        init();
        init2();
        init3();
        init4();
    }

}
```

activity_splash.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.lalabelle.multipledata.Activity_bmsplash">

    <ImageView
        android:id="@+id/t1"
        android:layout_width="298dp"
        android:layout_height="77dp"
        android:layout_marginEnd="24dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="80dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/belmozz" />

    <Button
        android:id="@+id/buttonsub"
        android:layout_width="177dp"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="32dp"
        android:background="@drawable/buttonstyle"
```

```
        android:text="Medicine Information"
        android:textAppearance="@android:style/TextAppearance.Medium"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button2" />

    <Button
        android:id="@+id/button2"
        android:layout_width="177dp"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="72dp"
        android:background="@drawable/buttonstyle"
        android:text="General Information"
        android:textAppearance="@android:style/TextAppearance.Medium"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/t1" />

    <Button
        android:id="@+id/button3"
        android:layout_width="177dp"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="32dp"
        android:background="@drawable/buttonstyle"
        android:text="History"
        android:textAppearance="@android:style/TextAppearance.Medium"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/buttonsub" />

    <Button
        android:id="@+id/button4"
        android:layout_width="177dp"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="32dp"
        android:background="@drawable/buttonstyle"
        android:text="Notification"
        android:textAppearance="@android:style/TextAppearance.Medium"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button3" />

    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:text="Developing The Medicine Reminder System"
        android:textStyle="italic"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.738"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/t1" />


</android.support.constraint.ConstraintLayout>
```

Activity_alarm.java

```
package com.example.lalabelle.multipledata;

import android.app.Notification;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.support.annotation.Nullable;
import android.support.v4.app.NotificationCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import android.app.AlarmManager;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;


public class Activity_alarm extends AppCompatActivity {
    public DatabaseReference Adatabase , Adatabase2 ,Adatabase3,Adatabase4;
    public TextView Anameview , Anameview2, Anameview3,Anameview4;
    Button button;
    public int al1,al2,al4,al3;

    public String string1;
    String uid;


    public  Activity_alarm (){
          string1="m";
    }
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_alarm);

        FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
        if (user != null) {
            uid = user.getUid();
        }

        Adatabase =
FirebaseDatabase.getInstance().getReference().child(uid).child("Status");
        Adatabase2 =
FirebaseDatabase.getInstance().getReference().child(uid).child("Channel
Status");
        Adatabase3 =
FirebaseDatabase.getInstance().getReference().child("time");
        Adatabase3 =
FirebaseDatabase.getInstance().getReference().child("Time");
```

```
        Anameview = (TextView) findViewById((R.id.al1));
        Anameview2 = (TextView) findViewById((R.id.al2));


        Adatabase.addValueEventListener((new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                String al1str = dataSnapshot.getValue().toString();
                al1 = Integer.parseInt(al1str);

                Anameview.setText("Status: " + al1);

                AlarmManager alarmManager =
(AlarmManager)getSystemService(ALARM_SERVICE);
                Intent intent = new
Intent(getApplicationContext(),Notification_receiver.class);
                PendingIntent pendingIntent =
PendingIntent.getBroadcast(Activity_alarm.this, 0, intent,
PendingIntent.FLAG_CANCEL_CURRENT);

                if(al1 == 1){
                    System.out.println("Test : " + al1);

                    alarmManager.setRepeating(AlarmManager.RTC_WAKEUP,
System.currentTimeMillis(), (5 * 1000), pendingIntent);
                }
                else {
                    alarmManager.cancel(pendingIntent);
                }

                System.out.println("Test out: " + al1);
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        }));


        Adatabase2.addValueEventListener((new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                 String al2 = dataSnapshot.getValue().toString();

                Anameview2.setText("Channel Status: " + al2);
             }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        }));
    }}
```

activity_alarm.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.lalabelle.multipledata.Activity_alarm">

    <TextView
        android:id="@+id/al"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="32dp"
        android:text="NOTIFICATION"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textStyle="bold|italic"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.496"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/al1"
        android:layout_width="321dp"
        android:layout_height="0dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:text="Status: "
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/al2" />

    <TextView
        android:id="@+id/al2"
        android:layout_width="321dp"
        android:layout_height="0dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="36dp"
        android:text="Channel Status:"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/al" />


</android.support.constraint.ConstraintLayout>
```

Notification_receiver.java

```
package com.example.lalabelle.multipledata;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.NotificationCompat;
import android.widget.TextView;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.example.lalabelle.multipledata.Activity_alarm;
/**
 * Created by apple on 3/10/18.
 */

public class Notification_receiver extends BroadcastReceiver{

    private String msg1 = " notify from belmoz " ;


        @Override
        public void onReceive(Context context, Intent intent) {

            NotificationManager notificationManager = (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);
            Intent intent1 = new Intent(context,Activity_multiple.class);
            intent1.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            //if we want ring on notifcation then uncomment below line.
        Uri alarmSound =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
            PendingIntent pendingIntent =
PendingIntent.getActivity(context,100,intent1,PendingIntent.FLAG_UPDATE_CURR
ENT);
            NotificationCompat.Builder builder = new
NotificationCompat.Builder(context).
                    setContentIntent(pendingIntent).
                    setSmallIcon(R.drawable.druglogo).
                    setContentText(msg1).
                    setContentTitle("Please take medicine now").
                    setSound(alarmSound).
                    setVibrate(new long[] { 500, 1000, 500 }).
                    setVisibility(NotificationCompat.VISIBILITY_PUBLIC).
        setAutoCancel(true);

            notificationManager.notify(100,builder.build());

    }
}
```