# โปรแกรมประยุกต์บนระบบแอนดรอยด์ สำหรับเฝ้าติดตามการเคลื่อนไหวของผู้สูงอายุ

# ANDROID APPLICATION FOR MONITORING ELDERLY MOVEMENT


นลพรรณ ประสงค์แก้ว

NOLAPAN PRASONGKAEW

นิยตา แก้วบางกะพ้อม

NIYATA KAEWBANGKAPOM


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2557

# โปรแกรมประยุกต์บนระบบแอนดรอยด์ สำหรับเฝ้าติดตามการเคลื่อนไหวของผู้สูงอายุ

# ANDROID APPLICATION FOR MONITORING ELDERLY MOVEMENT


โดย

นลพรรณ ประสงค์แก้ว

นิยตา แก้วบางกะพ้อม


อาจารย์ที่ปรึกษา

ดร. เทอดศักดิ์ ลิ่วหาทอง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2557

**ปริญญานิพนธ์** ปีการศึกษา 2557
**สาขาวิชา** วิศวกรรมอิเล็กทรอนิกส์
**คณะ** วิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
**เรื่อง** โปรแกรมประยุกต์บนระบบแอนดรอยด์สำหรับเฝ้าติดตามการเคลื่อนไหวของผู้สูงอายุ
ANDROID APPLICATION FOR MONITORING ELDERLY MOVEMENT

**ผู้จัดทำ** นางสาวนลพรรณ ประสงค์แก้ว รหัสประจำตัว 54010683
นางสาวนิยตา แก้วบางกะพ้อม รหัสประจำตัว 54010717

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

(ดร.เทอดศักดิ์ ลิ่วหาทอง)
อาจารย์ที่ปรึกษา

| | |
|---|---|
| **หัวปริญญานิพนธ์** | โปรแกรมประยุกต์บนระบบแอนดรอยด์สำหรับเฝ้าติดตามการเคลื่อนไหวของผู้สูงอายุ |
| **นักศึกษา** | นางสาวนลพรรณ ประสงค์แก้ว รหัสประจำตัว 54010683 |
| | นางสาวนิยตา แก้วบางกะพ้อม รหัสประจำตัว 54010717 |
| **ปริญญา** | วิศวกรรมศาสตรบัณฑิต |
| **สาขาวิชา** | วิศวกรรมอิเล็กทรอนิกส์ |
| **ปีการศึกษา** | 2557 |
| **อาจารย์ที่ปรึกษาปริญญานิพนธ์** | ดร.เทอดศักดิ์ ลิ่วหาทอง |

## บทคัดย่อ

ปริญญานิพนธ์นี้นำเสนอการพัฒนาแอพพลิเคชั่นบนระบบปฏิบัติการแอนดรอยด์ สำหรับเป็นเครื่องมือที่ช่วยในการเก็บค่าข้อมูลสัญญาณที่วัดได้จากเซนเซอร์วัดความเร่ง 3 แกน ของระบบตรวจจับการล้ม (พัฒนาขึ้นโดยนักศึกษาปริญญาโท) เพื่อใช้เฝ้าติดตามสัญญาณการเคลื่อนไหวของผู้สูงอายุ แล้วแสดงผลเป็นกราฟแบบค่าเวลาจริง พร้อมการบันทึกวิดีโอการทดลองใช้ระบบตรวจจับการล้ม ทั้งนี้เพื่อนำผลลัพธ์ที่ได้จากแอพพลิเคชั่นนี้ไปวิเคราะห์ และพัฒนาอัลกอริทึมการตรวจจับการล้มต่อไป แอพพลิเคชั่นที่ถูกพัฒนาขึ้นนี้สามารถใช้ได้ทั้งบนอุปกรณ์สมาร์ทโฟน หรือ แท็บเล็ต ซึ่งเหมาะสมกับการนำไปใช้เก็บข้อมูลในสถานการณ์จริง

| | |
|---|---|
| **Thesis Title** | Android Application for Monitoring Elderly Movement |
| **Student** | Miss Nolapan Prasongkaew Student ID 54010683 |
| | Miss Niyata Kaewbangkapom Student ID 54010717 |
| **Degree** | Bachelor of Engineering |
| **Program** | Electronics Engineering |
| **Year** | 2014 |
| **Thesis Advisor** | Dr.Thurdsak Leauhathong |


## ABSTRACT

This thesis presents an application developing on Android operating system for receiving signal data that have measured from acceleration sensor of Falling Detector System (designed by graduate student), monitoring elderly movement and showing results as a real-time graph with recording experimental video of Falling Detector System. In order to develop an algorithms of Falling Detector System by analyzing results from an application. An application is developed on a Smartphone and Tablet which is suitable for usage environment.

# กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่องโปรแกรมประยุกต์บนระบบแอนดรอยด์สำหรับเฝ้าติดตามการเคลื่อนไหวของผู้สูงอายุ (Android Application for Monitoring Elderly Movement) นี้ สำเร็จลุล่วงไปได้ด้วยดี โดยได้รับความอนุเคราะห์และความช่วยเหลืออย่างดียิ่งของ ดร.เทอดศักดิ์ ลิ่วหาทอง อาจารย์ที่ปรึกษาโครงงาน และจากบุคคลหลายท่านด้วยกัน ทางผู้จัดทำจึงขอขอบคุณอาจารย์เทอดศักดิ์ ลิ่วหาทอง และ อาจารย์ทุกท่านที่คอยให้ความรู้ คำปรึกษา และคำแนะนำที่เป็นประโยชน์ต่อโครงงานนี้ ทำให้โครงงานนี้สำเร็จไปได้ด้วยดี ขอขอบคุณเพื่อนๆ ที่คอยแนะแนวทางในการดำเนินงาน ให้ความช่วยเหลือในการแก้ไขปัญหาที่เกิดขึ้น อีกทั้งยังคอยให้กำลังใจและติดตามเอาใจใส่เป็นอย่างดีมาโดยตลอด ตั้งแต่เริ่มต้นโครงงานจนกระทั่งโครงงานนี้สำเร็จไปได้ด้วยดี

นลพรรณ ประสงค์แก้ว
นิยตา แก้วบางกะพ้อม

# สารบัญ


หน้า

บทคัดย่อภาษาไทย ..... I
บทคัดย่อภาษาอังกฤษ ..... II
กิตติกรรมประกาศ ..... III
สารบัญ ..... IV
สารบัญรูป ..... V
บทที่ 1 บทนำ ..... 1
1.1 ความเป็นมาและความสำคัญ ..... 1
1.2 วัตถุประสงค์ ..... 2
1.3 การออกแบบโปรแกรมประยุกต์ ..... 2
1.4 ประโยชน์ที่คาดว่าจะได้รับ ..... 2

บทที่ 2 การพัฒนาโปรแกรมบนระบบแอนดรอยด์ ..... 3
2.1 การพัฒนาโปรแกรม ..... 3
2.1.1 โปรแกรมที่ใช้ในการเขียนแอพพลิเคชั่น ..... 3
2.1.2 พื้นฐานการเขียน Android กับโปรแกรมภาษา Java Syntax ..... 3
2.1.3 การเริ่มต้นพัฒนาแอพพลิเคชั่น ..... 6
2.2 การทำงานกับฐานข้อมูล SQLite ใน Application ..... 7
2.3 การใช้งาน Bluetooth ใน Application ..... 9
2.4 การบันทึกวิดีโอใน Application ..... 10
2.5 การสร้างกราฟใน Application ..... 11

บทที่ 3 การออกแบบระบบโปรแกรมประยุกต์ ..... 12
3.1 การส่งข้อมูลจากเครื่องตรวจจับการล้ม ..... 13
3.2 แผนผังการทำงานของโปรแกรมประยุกต์ ..... 14

บทที่ 4 ผลการทำงานของโปรแกรมประยุกต์ ..... 17
4.1 การทดลองเชื่อมต่อบลูทูธระหว่างแอพพลิเคชั่นกับเครื่องตรวจจับการล้ม ..... 17
4.2 การทดลองรับข้อมูลจากเครื่องตรวจจับการล้ม พร้อมกับการแสดงกราฟ ..... 18
4.3 การทดลองเก็บค่าความเร่งต่อเวลาระหว่างการทำกิจกรรม ..... 19
4.4 Android Activity ..... 26

บทที่ 5 สรุปผลการทดลอง ..... 29
5.1 สรุปผลการทดลอง ..... 29
5.2 ปัญหาและอุปสรรคในการทดลอง ..... 30

เอกสารอ้างอิง ..... 31

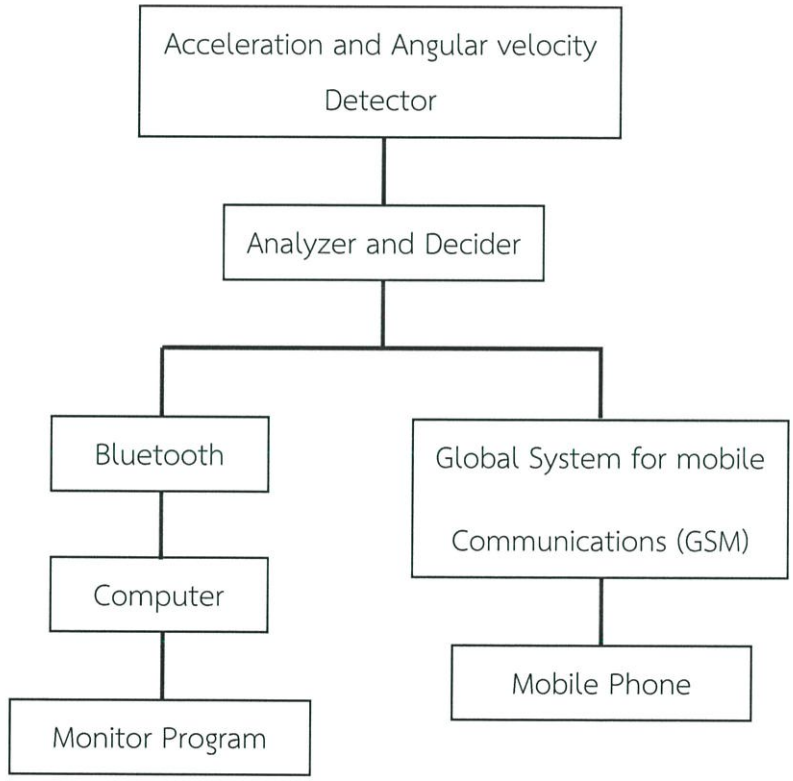
# สารบัญรูป


| รูปที่ | หน้า |
|---|---|
| 1.1 โครงสร้างของระบบตรวจจับการล้ม | 1 |
| 1.2 แผนผังการออกแบบโปรแกรมประยุกต์ | 2 |
| 2.1 หน้าต่างเริ่มต้นของโปรแกรมเมื่อสร้างโปรเจคแล้ว | 6 |
| 2.2 ฐานข้อมูลที่บันทึกไว้ในโปรแกรมประยุกต์ | 8 |
| 2.3 การเชื่อมต่อผ่านทางบลูทูธระหว่างโปรแกรมประยุกต์กับไมโครคอนโทรลเลอร์ | 9 |
| 2.4 การบันทึกวิดีโอจากแอพพลิเคชั่น | 10 |
| 2.5 การสร้างกราฟแบบต่างๆ โดยใช้ AndroidPlot | 11 |
| 3.1 โครงสร้างการทำงานของโปรแกรม | 12 |
| 3.2 แผนผังการส่งข้อมูลจากเครื่องตรวจจับการล้ม | 13 |
| 3.3 การส่งข้อมูลจากเครื่องตรวจจับการล้ม | 13 |
| 3.4 แผนผังการทำงานส่วนบันทึกข้อมูลลงทะเบียนลงในฐานข้อมูล | 14 |
| 3.5 แผนผังการทำงานส่วนรับข้อมูล แสดงกราฟ และบันทึกข้อมูล | 15 |
| 3.6 แผนผังการทำงานส่วนบันทึกวิดีโอ | 16 |
| 4.1 การเชื่อมต่อผ่านบลูทูธ | 17 |
| 4.2 การรับข้อมูลจากเครื่องตรวจจับการล้มพร้อมกับการพล็อตกราฟ | 18 |
| 4.3 การทดลองในท่าเดิน | 19 |
| 4.4 กราฟความเร่งในการเดินต่อเวลา | 19 |
| 4.5 การทดลองในท่าเดินขึ้นลงบันได | 20 |
| 4.6 กราฟความเร่งในการเดินขึ้นลงบันไดต่อเวลา | 20 |
| 4.7 การทดลองในท่าวิ่งเหยาะๆ | 21 |
| 4.8 กราฟความเร่งในการวิ่งเหยาะๆต่อเวลา | 21 |
| 4.9 การทดลองในท่านั่งแล้วล้ม | 22 |
| 4.10 กราฟความเร่งในการนั่งแล้วล้มต่อเวลา | 22 |
| 4.11 การทดลองในท่าล้มไปด้านหลัง | 23 |
| 4.12 กราฟความเร่งในการล้มไปด้านหลังต่อเวลา | 23 |
| 4.13 การทดลองในท่าล้มไปด้านหน้า | 24 |
| 4.14 กราฟความเร่งในการล้มไปด้านหน้าต่อเวลา | 24 |
| 4.15 การทดลองในท่าลุกนั่งเก้าอี้ | 25 |
| 4.16 กราฟความเร่งในการลุกนั่งเก้าอี้ต่อเวลา | 25 |
| 4.17 หน้าหลักของ Application | 26 |
| 4.18 หน้าลงทะเบียนรับข้อมูลประวัติผู้ใช้งาน | 27 |
| 4.19 หน้าข้อมูลประวัติผู้ใช้งาน | 27 |
| 4.20 หน้าการเชื่อมต่อบลูทูธ | 28 |
| 4.21 หน้าส่วนรับข้อมูล แสดงกราฟแบบเวลาจริง และบันทึกวิดีโอ | 28 |

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาและความสำคัญ

ปัจจุบันผู้สูงอายุในประเทศไทยมีจำนวนสูงขึ้นมาก และการล้มเป็นปัญหาหลักของผู้สูงอายุเหล่านี้ ดังนั้นการพัฒนาเครื่องมือเพื่อใช้วิเคราะห์และเฝ้าระวังการเคลื่อนที่ของผู้สูงอายุ เพื่อนำไปสู่การตรวจจับการล้มแบบอัตโนมัติจึงได้ถูกพัฒนาขึ้น ในโครงงานนี้ได้สนใจระบบอุปกรณ์ตรวจจับการล้มของผู้สูงอายุ (Falling Detector Device) ซึ่งกำลังพัฒนาอยู่โดยนักศึกษาปริญญาโท ซึ่งระบบตรวจจับการล้มของผู้สูงอายุ มีโครงสร้างดังนี้




**รูปที่ 1.1 โครงสร้างของระบบตรวจจับการล้ม**

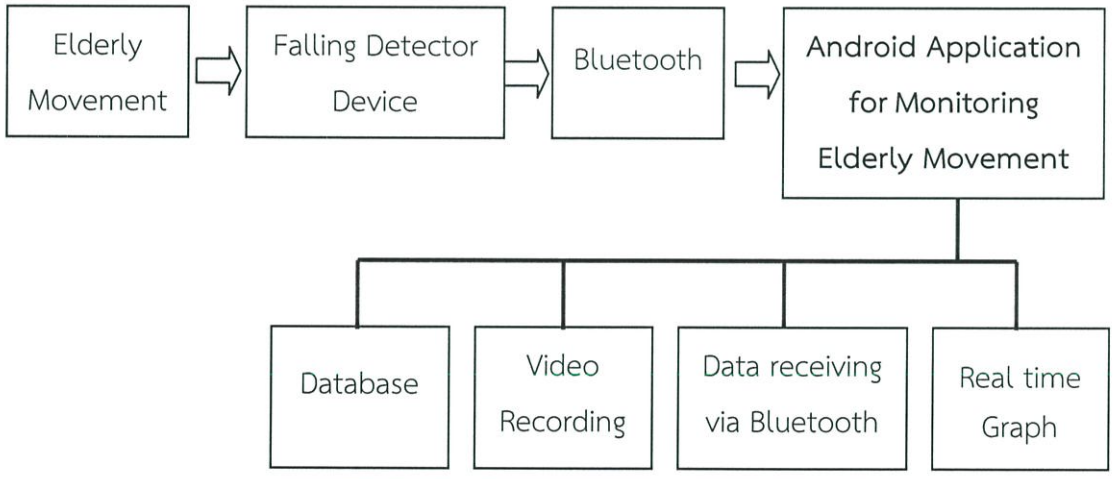
ปัญหาสำคัญของการพัฒนาระบบตรวจจับการล้มนี้ คือการเก็บข้อมูลการเคลื่อนไหวของผู้สูงอายุ โดยแต่เดิมนั้นการเก็บค่าข้อมูลการเคลื่อนไหวซึ่งได้จากเซนเซอร์วัดความเร่งที่อยู่ในอุปกรณ์นั้น จะใช้คอมพิวเตอร์ หรือคอมพิวเตอร์แบบพกพา ซึ่งจะมีโปรแกรมเฉพาะที่ถูกพัฒนาขึ้นเพื่อรับค่าสัญญาณจากเครื่องตรวจจับการล้ม เดิมนั้นการทดลองเก็บค่าข้อมูลสัญญาณจะทำในห้องทดลอง แต่เมื่อต้องมีการทดลองจริงกับผู้สูงอายุนอกห้องทดลอง การจะใช้คอมพิวเตอร์เก็บข้อมูลนั้นอาจจะไม่สะดวก เราจึงต้องการเครื่องมือในการเก็บข้อมูลที่มีขนาดเล็กลง และสามารถพกพาได้สะดวกมากขึ้น

ดังนั้นโปรแกรมประยุกต์บนระบบแอนดรอยด์สำหรับเฝ้าติดตามการเคลื่อนไหวของผู้สูงอายุ จึงถูกพัฒนาขึ้นเพื่อเป็นเครื่องมือที่ช่วยในการเก็บข้อมูลการเคลื่อนไหวของผู้สูงอายุ โดยใช้กับโทรศัพท์สมาร์ทโฟน หรืออุปกรณ์แท็บเล็ต ที่ปัจจุบันนี้มีใช้งานกันเป็นจำนวนมากแล้ว จึงเป็นอีกทางเลือกที่เราสามารถพัฒนาโปรแกรมหรือแอพพลิเคชั่นเพื่อรับค่าข้อมูลจากเครื่องตรวจจับการล้ม การเก็บข้อมูลนอกสถานที่จึงจะมีความสะดวกมากขึ้น

## 1.2 วัตถุประสงค์

ออกแบบแอพพลิเคชั่นบนระบบแอนดรอยด์สำหรับเฝ้าติดตามการเคลื่อนไหวของผู้สูงอายุ โดยจะนำไปใช้งานร่วมกับเครื่องตรวจจับการล้มของผู้สูงอายุ โดยโปรแกรมจะสามารถเก็บข้อมูลผู้ใช้ บันทึกวีดีโอ และการเก็บข้อมูลค่าสัญญาณการเคลื่อนไหวผ่านทางบลูทูธ แล้วนำค่าที่ได้มาสร้างเป็นกราฟ ซึ่งจะสามารถช่วยในการวิเคราะห์การล้มในรูปแบบต่างๆ ได้

## 1.3 การออกแบบโปรแกรมประยุกต์




**รูปที่ 1.2 แผนผังการออกแบบโปรแกรมประยุกต์**

## 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ความรู้เกี่ยวกับการเขียนโปรแกรมบนระบบแอนดรอยด์โดยใช้ภาษา JAVA
2. ได้ความรู้เรื่องการออกแบบโปรแกรม และการเก็บข้อมูลของผู้ใช้งาน
3. ได้สร้างแอพพลิเคชั่นสำหรับเฝ้าติดตามการเคลื่อนไหวของผู้สูงอายุ และพัฒนาฟังก์ชันต่างๆ ต่อไป

# บทที่ 2
# การพัฒนาโปรแกรมบนระบบแอนดรอยด์

## 2.1 การพัฒนาโปรแกรม

พื้นฐานและองค์ประกอบที่จะนำมาเขียนโปรแกรมบน Android จะใช้โครงสร้างของภาษา JAVA ในการพัฒนาเป็นหลัก และใน Android นั้นการเขียนโปรแกรมจะมี API Library ที่ถูกพัฒนาสำหรับ Android ให้เลือกใช้มากมายเช่น API Library ที่ช่วยจัดการเกี่ยวกับพวก Graphic การออกแบบ Multimedia หรือ API Library ที่เกี่ยวข้องกับ GPS, Bluetooth, EDGE, 3G, WIFI หรือ SQLite ที่จะเข้ามาจัดการเกี่ยวกับฐานข้อมูล Database

### 2.1.1 โปรแกรมที่ใช้ในการเขียนแอพพลิเคชั่น

โปรแกรมที่เราต้องทำการติดตั้งก่อน สำหรับใช้ในการพัฒนาแอพพลิเคชั่น มีดังต่อไปนี้

1) Eclipse Development Tools and Java Development Kit (JDK)
2) ADT (Android Development Tools Plug in for eclipse)
3) Android SDK
4) Android Virtual Device Manager (Emulator)

### 2.1.2 พื้นฐานการเขียน Android กับโปรแกรมภาษา Java Syntax

การเขียนโปรแกรมบน Android นั้นจะใช้ภาษา Java Platformในการพัฒนาและเขียนคำสั่งให้โปรแกรมทำงาน รูปในการเขียนเป็น OOP ทั้งหมด และ API Library ต่างๆ ที่อยู่ใน Android Framework ที่เราสามารถเรียกใช้งานได้ ก็ถูกพัฒนาด้วยภาษา Java เช่นเดียวกัน และจะมีให้เลือกเรียกใช้งานหลายตัวมาก แบ่งแยกตามวัตถุประสงค์ต่าง ๆ (เราสามารถพัฒนา Library ขึ้นมาใช้เองก็ได้ หรือจะดาวน์โหลด Library จากแหล่งต่างๆ ที่มีทั้งฟรีและเสียเงิน) โดยจะมีการแยก Package หรือ API Class Library ต่างๆ ถูกแยกจัดเก็บไว้ในแต่ล่ะหมวดหมู่ เพื่อความสะดวกและง่ายต่อการนำมาใช้ เช่น TextView จัดการเกี่ยวกับข้อความหรือ Text , Button จัดการเกี่ยวกับปุ่ม Button หรืออื่นๆ ที่ถูกจัดแยกไว้ตาม Class ที่อยู่ภายใต้ Widgets Class และถ้าจะใช้งานตัวไหนก็ค่อยทำการ Import เข้ามาใน Class ของเรา

ในโครงสร้างการเขียนโปรแกรม Android พื้นฐานทั่วไปจะเป็นการทำงานของ XML Layout ซึ่งจะผสานการทำงานร่วมกับ XML และJava โดย XML จะถูกออกแบบให้เป็นส่วนที่เป็น GUI และใช้ XML Syntax ในการวาง Layout ต่างๆ ของ Widgets หรือ Element ต่างๆ ส่วนใน

ภาษา Java จะเป็นชุดคำสั่งที่ควบควบคุมการทำงานของโปรแกรม และหน้าจอที่แสดงผลที่อยู่ในรูปแบบของ XML Layout

**2.1.2.1 การใช้ภาษา JAVA เบื้องต้น**

2.1.2.1.1 โครงสร้างคุมโปรแกรม

- การเขียนโปรแกรมเรียกว่า coding

- โปรแกรมประกอบด้วยหลายๆ statements (ประโยคคำสั่ง) แต่ละ statement สิ้นสุดด้วยเครื่องหมาย ; (semicolon) และจะทำงานตามลำดับเดียวกันกับลำดับของ statements ในโปรแกรมนั้น

- บางกรณีที่ไม่ต้องการให้โปรแกรมทำงานตามลำดับเดียวกันกับลำดับของ statements ต้องใช้ if, if...else, switch...case, และ loops เช่น for, while, do...while เพื่อเบี่ยงหรือข้ามบาง statements ไป

- การเขียน loops ถ้าเขียนไม่ถูกต้อง อาจวนไม่หยุด (เรียกว่า infinite loop) กด Ctrl+C ให้หยุด แล้วแก้ไข

2.1.2.1.2 JAVA OOP (Object oriented programming)

OOP คือการมองทุกอย่างให้เป็น object ทุก object จะประกอบด้วย 2 ส่วนคือ Attribute (object data) เป็นส่วนที่ใช้ในการเก็บข้อมูลต่างๆ ของ objectและ Method (object behavior) เป็นสิ่งที่ object นั้นสามารถทำได้

หลักการสำคัญของ OOP คือ

1. Data hiding คือ ปกปิด source code ส่วนหนึ่งไว้ไม่ให้ผู้ที่ไม่เกี่ยวข้องสามารถเข้าถึง หรือแก้ไขข้อมูลได้โดยตรง ซึ่งจะต้องบอกคำสงวนในการควมคุมการเข้าถึงข้อมูล

– Public สามารถเข้าถึงได้จากทุกๆที่ของโปรแกรม

– Private ไม่สามารถเข้าถึงได้จากภายนอก class ส่วนมากจะใช้กับ attribute ของ class

2. encapsulation นั่นคือ attribute ต่างๆภายใน class นั้นจะไม่สามารถถูกเปลี่ยนแปลงค่าได้จากภายนอก class การเปลี่ยนแปลงค่าของ attribute จะสามารถทำได้จากการเรียก method ภายใน class เท่านั้น ซึ่งคือการกำหนดด้วยคำสงวน private นั่นเอง

3. inheritance & reusable คือการ นำ source code ส่วนที่มีการใช้ซ้ำๆ นำกลับมาใช้ใหม่การนำกลับมาใช้ใหม่ แบ่งได้เป็น 2 กรณี

- inheritance (การสืบทอดจากคลาสหลัก) มี superclass (base class/classหลัก) และ subclass (derived class/classสืบทอด) ซึ่งจะมี

attribute/method เพิ่มเติมจาก class พ่อ สามารถเป็นได้ทั้ง single inheritance คลาสลูกมีคลาสพ่อเพียงคลาสเดียว และ multiple inheritance คลาสลูกมีคลาสพ่อหลายคลาส

- composition (has-a relationship) เป็นการนำ class ที่มีอยู่เดิมมาเป็น attribute ของ class ใหม่

4. polymorphism ใช้ได้กับการสืบทอดแบบ is-a เท่านั้น คือการที่คลาสพ่อมี method ที่ภายในไม่มีการ implement ใดๆ เมื่อคลาสลูกสืบทอดคุณสมบัติไป คลาสลูกได้มีการเพิ่มส่วนของการ implement ใน method นั้น

- Method คือ บล็อกหรือกลุ่มของโค้ด สั่งให้คอมพิวเตอร์ กระทำอะไรสักอย่าง คำว่า Method ในภาษา Java ตรงกับคำ Function หรือ Procedure ในภาษาอื่นๆ

- Class เป็นแม่แบบของ object ก่อนที่จะสร้าง object ได้ก็ต้องมี class เสียก่อน object เป็น object ของ class ใด ก็จะมีคุณสมบัติ (attribute) และสามารถทำพฤติกรรม (method) ได้เหมือนที่ class มี

**2.1.2.2 คำสั่งพื้นฐานที่ใช้ในการเขียนโปรแกรมบนแอนดรอยด์**

โดยปกติแล้วแอพพลิเคชั่นจะทำงานแยกกันในแต่ละโพรเซส และในแต่ละโพเซสอาจจะมี Activity/Service ที่ทำงานอยู่มากกว่า 1 Activity/Service ดังนั้น ในแต่ละแอพพลิเคชั่นอาจจะมีมากกว่า 1 Activity ซึ่งในการเริ่มทำงานของ Activity จะเริ่มด้วย startAcivity() สำหรับแบบซิงโครนัส (Synchronous) และจะเริ่มด้วย startSubActivity() สำหรับแบบอะซิงโครนัส (Asynchronous) โดยในแต่ละ Activity จะมีสถานะการทำงานหลักที่แยกจากกันชัดเจนดังนี้

- Create (Bundle savedInstanceState) ส่วนนี้จะถูกเรียกใช้งานเมื่อเริ่มทำงาน ในกรณีที่มีการเรียกใช้งานเมธอด (Method) นี้ Android Framework จะนำ Bundle object ไปบันทึกไว้ใน Activity ก่อนที่ Activity จะทำงาน ซึ่งจากนั้นจะตามด้วยฟังก์ชั่น onStart()

- onStart() ส่วนนี้เป็นการระบุว่า Activity นั้นๆ จะถูกแสดงขึ้นมา จากนั้นสถานะจะถูกย้ายไปเป็นสถานะ onResume แต่ถ้า Activity นั้นไม่สามารถทำงานได้ด้วยเหตุผลบางอย่าง สถานะจะถูกย้ายไปเป็นสถานะ onStop

- onRestart() ส่วนนี้จะเป็นการระบุว่า Activity นั้นจะถูกแสดงขึ้นมาอีกครั้งหนึ่ง ซึ่งจะตามด้วยสถานะ onStart()

- onResume() ส่วนนี้จะถูกเรียกเมื่อ Activity นั้นๆ มีการติดต่อกับผู้ใช้งาน เช่น นักพัฒนาต้องการเรียก Activity นั้นขึ้นมาทำงานอีกรอบหนึ่ง หลังจากที่ Activity นั้นอยู่ในสถานะ onPause

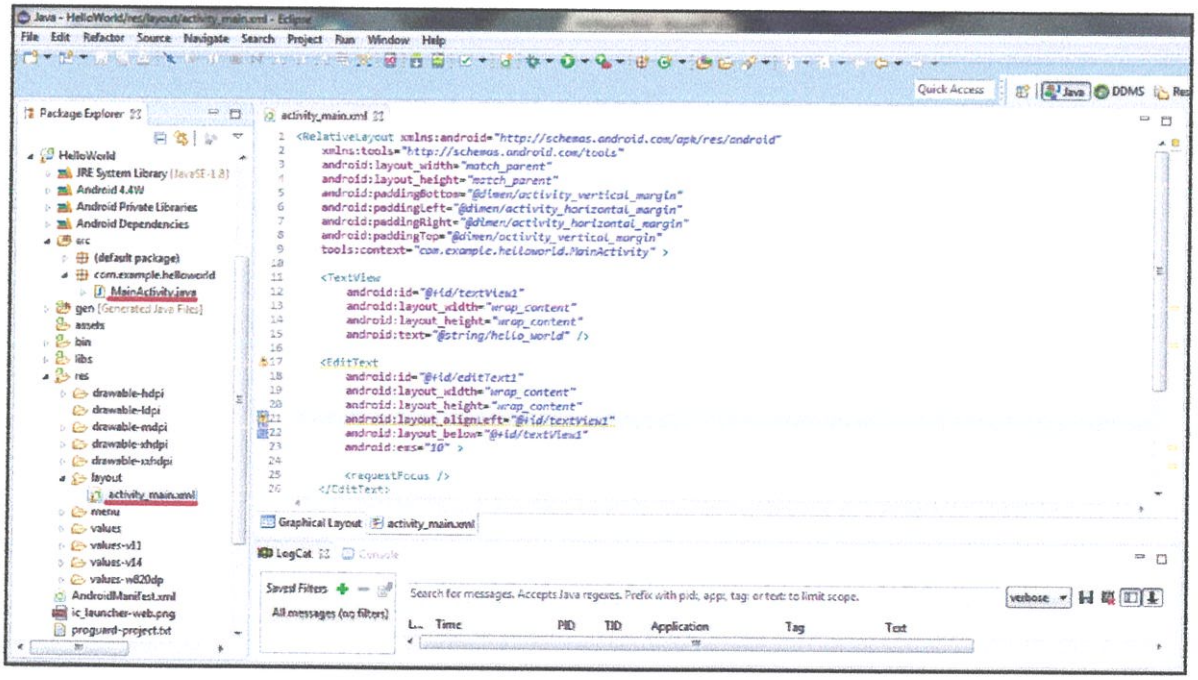
- onPause() ส่วนนี้จะถูกเรียกใช้เมื่อ Activity นั้นจะถูกเปลี่ยนไปเป็นการทำงานทางเบื้องหลัง (Background)

- onStop() ส่วนนี้จะถูกเรียกใช้งานเมื่อผู้ใช้ไม่ต้องการใช้งาน Activity นั้นๆ ในช่วงระยะเวลาหนึ่งๆ ซึ่งจะตามด้วยสถานะ onRestart() เมื่อต้องการกลับมาทำงานที่ Activity นั้นอีกครั้ง หรือตามด้วย สถานะ onDestroy() เมื่อต้องการปิด Activity นั้นๆ

- onDestroy() ส่วนนี้จะถูกเรียกเมื่อมีการปิดการทำงานของแต่ละ Activity

### 2.1.3 การเริ่มต้นพัฒนาแอพพลิเคชั่น

1. เริ่มต้นสร้างโปรเจ็กต์
2. ออกแบบหน้าจอแอพพลิเคชั่น การออกแบบหน้าจอแอพพลิเคชั่นเป็นการแก้ไขโค้ดใน XML (Extensible Markup Language)
3. เขียนโค้ดกำหนดการทำงานของแอพพลิเคชั่น เป็นการเขียนโค้ดในไฟล์ .java เพื่อกำหนดให้แอพพลิเคชั่นแสดงผลการทำงานตามต้องการ ออกทางหน้าจอ
4. ทดสอบการทำงานของแอพพลิเคชั่นผ่านโทรศัพท์จำลอง (Emulator) หรือโทรศัพท์จริงว่าสามารถทำงานได้ตามต้องการหรือไม่ หลังจากการพัฒนาโปรแกรมเสร็จสิ้น




**รูปที่ 2.1 หน้าต่างเริ่มต้นของโปรแกรมเมื่อสร้างโปรเจคแล้ว**

## 2.2 การทำงานกับฐานข้อมูล SQLite ใน Application

ฐานข้อมูล (Database) เหมาะสำหรับการเก็บข้อมูลที่มีโครงสร้างแบบเดียวกัน เช่น ข้อมูลรายชื่อผู้ติดต่อ ซึ่งแต่ละชื่อจะประกอบด้วยชื่อ เบอร์โทรศัพท์ และอีเมล์แอดเดรส เป็นต้น

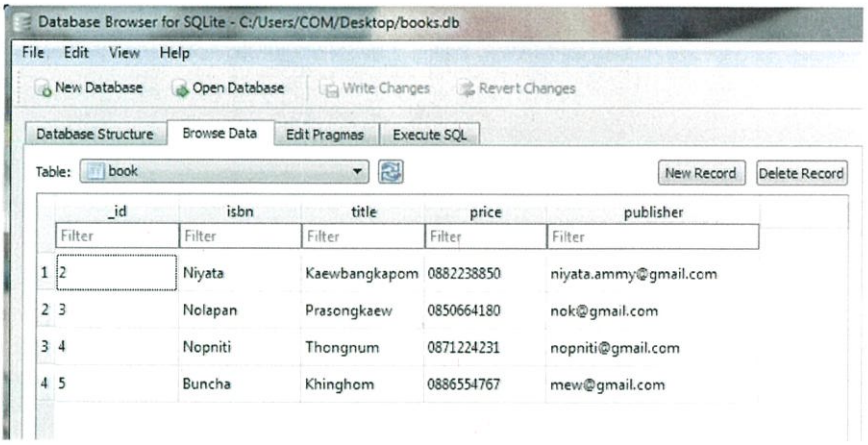
แอนดรอยด์มีระบบฐานข้อมูล SQLite รวมอยู่ในตัว และจัดเตรียม API ไว้ให้เราสร้างฐานข้อมูล และทำงานกับฐานข้อมูลได้อย่างง่ายดาย

ฐานข้อมูลที่แอพพลิเคชั่นของเราสร้างขึ้นจะถูกเก็บไว้ใน Internal Directory (ไดเร็คทอรีส่วนตัวของแอพบน Internal Storage) ซึ่งโดยดีฟอลต์แล้วแอพอื่นๆ จะไม่สามารถเข้าถึงได้

การใช้งานฐานข้อมูล SQLite นั้น โดยปกติเราจะไม่สร้างฐานข้อมูลเตรียมไว้ล่วงหน้าตั้งแต่ช่วงเขียนแอพ แต่จะสร้างขึ้นตอนที่แอพทำงานเลย (runtime) ซึ่งแอนดรอยด์มี คลาสตัวช่วย หรือ Helper Class ไว้ช่วยอำนวยความสะดวกในการสร้างฐานข้อมูล อัพเกรดฐานข้อมูล และเข้าถึงฐานข้อมูล เพื่ออ่าน/เขียนข้อมูล

โดยมีคลาส MyDbHelper ทำหน้าที่เป็น Helper Class ที่ช่วยในการสร้างและทำงานกับฐานข้อมูล เนื่องจากเรากำหนดมันเป็นซับคลาสของ SQLiteOpenHelper ภายในคอนสตรัคเตอร์ของคลาส เราระบุชื่อและเวอร์ชั่นของฐานข้อมูลให้แอนดรอยด์รู้โดยเรียกไปยังคอนสตรัคเตอร์ของ SQLiteOpenHelper หลังจากนั้นเมื่อมีการเข้าถึงฐานข้อมูลเพื่ออ่าน/เขียนข้อมูล โดยใช้เมธอด getReadableDatabase, getWriteableDatabase แอนดรอยด์จะตรวจสอบว่ามีฐานข้อมูลหรือยัง ถ้ายังไม่มีก็จะสร้างฐานข้อมูลว่างๆให้ แล้วเรียกมายังเมธอด onCreate ใน Helper Class ของเรา พร้อมทั้งส่ง reference ของฐานข้อมูลมาเป็นพารามิเตอร์ เพื่อให้เราจัดเตรียมโครงสร้างของฐานข้อมูลตามที่ต้องการ เมื่อเราพัฒนาแอพเวอร์ชั่นใหม่ เราอาจออกแบบให้แอพทำงานกับฐานข้อมูลที่มีโครงสร้างเปลี่ยนไปจากเดิม เช่น มีการเพิ่มฟิลด์เข้ามา เป็นต้น ซึ่ง SQLiteOpenHelper มีเมธอด onUpgrade ไว้รองรับการอัพเกรดฐานข้อมูล โดยหากแอนดรอยด์พบว่าเลขเวอร์ชั่นของฐานข้อมูลที่ระบุในคอนสตรัคเตอร์มากกว่าเลขเวอร์ชั่นของฐานข้อมูลที่มีอยู่แล้ว มันจะเรียกมายังเมธอด onUpgrade ใน Helper Class ของเรา พร้อมทั้งส่งตัวฐานข้อมูล, เลขเวอร์ชั่นเดิม และเลขเวอร์ชั่นใหม่มาเป็นพารามิเตอร์ เพื่อให้เราแก้ไขโครงสร้างฐานข้อมูลหรือดำเนินการใดๆ ตามความเหมาะสม

เราสามารถดูข้อมูลทั้งหมดที่ถูกบันทึกไว้ในฐานข้อมูล (Database) ได้โดยการใช้โปรแกรม Database Browser for SQLite




รูปที่ 2.2 ฐานข้อมูลที่บันทึกไว้ในโปรแกรมประยุกต์

## 2.3 การใช้งาน Bluetooth ใน Application

บลูทูธจะประกอบไปด้วยการทำงานหลายแบบที่เรียกว่า Profile ในที่นี้ก็จะเป็นการใช้งาน Profile ของบลูทูธที่เรียกว่า Serial Port Profile (SPP) เป็นการส่งข้อมูลผ่านบลูทูธแบบอนุกรม ถ้านำไปใช้กับไมโครคอนโทรลเลอร์ก็สามารถใช้กับโมดูลบลูทูธที่รองรับการรับส่งข้อมูลแบบอนุกรมได้ทันที (Serial Bluetooth) โดยโมดูลดังกล่าวจะรับข้อมูลอนุกรมผ่านบลูทูธแล้วส่งออกมาเป็น UART ทันที จึงทำให้ง่ายต่อการใช้งานกับไมโครคอนโทรลเลอร์

สำหรับคำสั่งในการใช้งานบลูทูธก็จะมีคลาสที่ทำหน้าที่จัดการในส่วนของเซอร์วิสสำหรับเชื่อมต่อและรับส่งข้อมูลให้เรียบร้อยแล้ว โดยสิ่งที่ต้องจัดการก็คือการเข้าใช้งานบลูทูธ ตัวอย่างเช่น การเปิดบลูทูธ การค้นหาอุปกรณ์ เลือกอุปกรณ์ที่จะเชื่อมต่อ และการรับ-ส่งข้อมูลผ่านบลูทูธ โดยการเรียกใช้คำสั่งจากคลาสที่มี

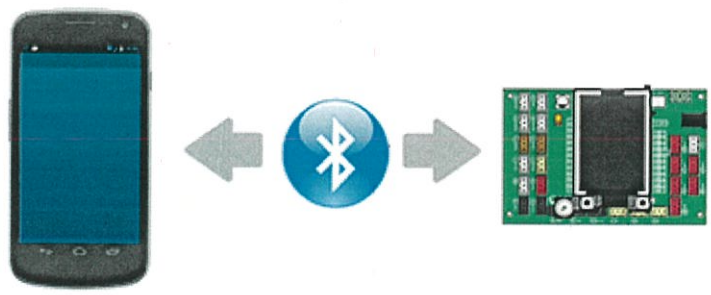
ขั้นตอนแรกสุดที่จะทำให้แอพพลิเคชันของเราทำงานร่วมกับบลูทูธได้นั้น ให้ทำการขอใช้ Permission ของบลูทูธก่อน เนื่องจากบลูทูธเป็นการใช้งานอุปกรณ์ภายนอกแอพพลิเคชัน ที่ AndroidManifest.xml ด้วยคำสั่งดังนี้

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

จากนั้นในการเรียกใช้งานบลูทูธเริ่มแรก คือเรียกใช้คลาสที่ทำหน้าที่จัดการการทำงานของบลูทูธ ซึ่งก็คือคลาส Bluetooth Adapter โดยจะประกาศไว้ใน onCreate เพื่อเรียกใช้งานบลูทูธ

```
BluetoothAdapter mybluetooth = BluetoothAdapter.getDefaultAdapter();
```

และอีกส่วน คือคลาส Bluetooth Service เป็นคลาสภายนอกที่เพิ่มเข้ามาทีหลังที่เอาไว้จัดการกับการรับส่งข้อมูล การเชื่อมต่อกับอุปกรณ์อื่น แล้วจากนั้นจึงทำการเขียนโปรแกรมเพิ่มเติมในส่วนอื่นตามที่เราได้ใช้งาน



**รูปที่ 2.3 การเชื่อมต่อผ่านทางบลูทูธระหว่างโปรแกรมประยุกต์กับไมโครคอนโทรลเลอร์**

## 2.4 การบันทึกวิดีโอใน Application

การทำให้ในแอพพลิเคชั่น สามารถใช้กล้องวิดีโอได้เลยโดยไม่ต้องใช้แอพพลิเคชั่นกล้องวิดีโอของเครื่องนั้น จะใช้คลาส MediaRecorder ร่วมกับ SurfaceView ที่ใช้สำหรับแสดงภาพรีวิวจากกล้อง เริ่มต้นจากการสร้างออบเจ็คของคลาส MediaRecorder และกำหนดให้แสดงภาพพรีวิวจากกล้องที่ SurfaceViewจากนั้นกำหนดแหล่งข้อมูลเสียงและภาพ โดยรับเสียงจากไมโครโฟนและรับภาพจากกล้อง

```
MediaRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
MediaRecorder.setVideoSource(MediaRecorder.VideoSource.CAMERA);
```

ต่อมากำหนดฟอร์แมตของไฟล์ .mp4 และกำหนดชื่อพาธของไฟล์ด้วยเมธอด setOutputFile เรียกเมธอด prepare เพื่อเตรียมการบันทึก และเรียกเมธอด start เพื่อเริ่มบันทึก stop เพื่อหยุดบันทึก หลังจากใช้งาน MediaRecorder เสร็จให้เรียกเมธอด release เพื่อปล่อยการใช้คืนสู่ระบบ เพิ่มการขอสิทธิ์ต่างๆ ที่จำเป็น ที่ AndroidManifest.xml ด้วยคำสั่งดังนี้

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.RECORD_VIDEO" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

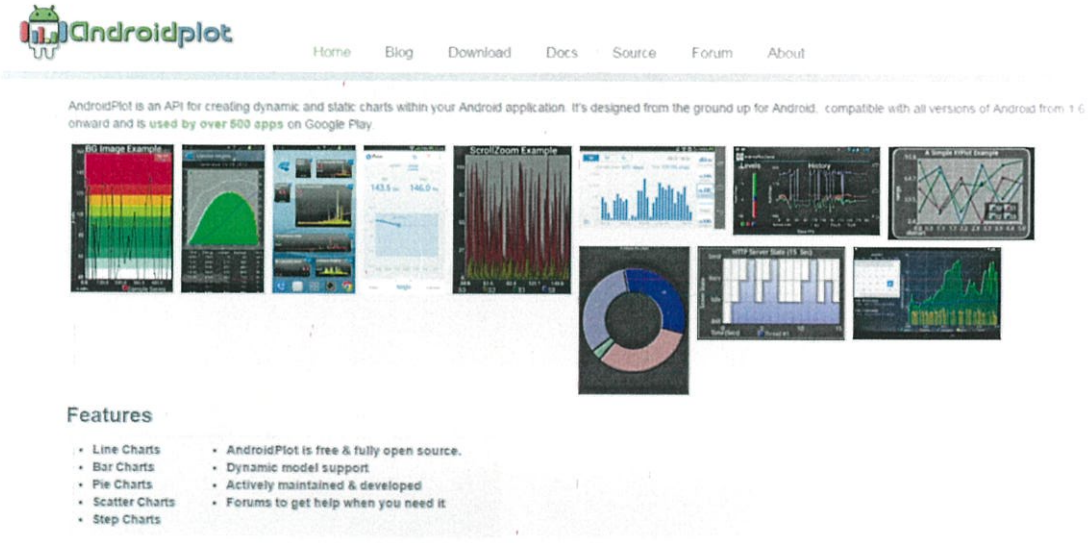
**รูปที่ 2.4 การบันทึกวิดีโอจากแอพพลิเคชั่น**

## 2.5 การสร้างกราฟใน Application

การสร้างกราฟในระบบแอนดรอยด์นี้สามารถทำได้หลายวิธี แต่ในที่นี้จะใช้งานร่วมกับ library ที่ชื่อว่า AndroidPlot

หลังจากสร้างโปรเจคเรียบร้อย เราจะเพิ่ม library AndroidPlot จากนั้นขั้นตอนการสร้างทำได้โดย

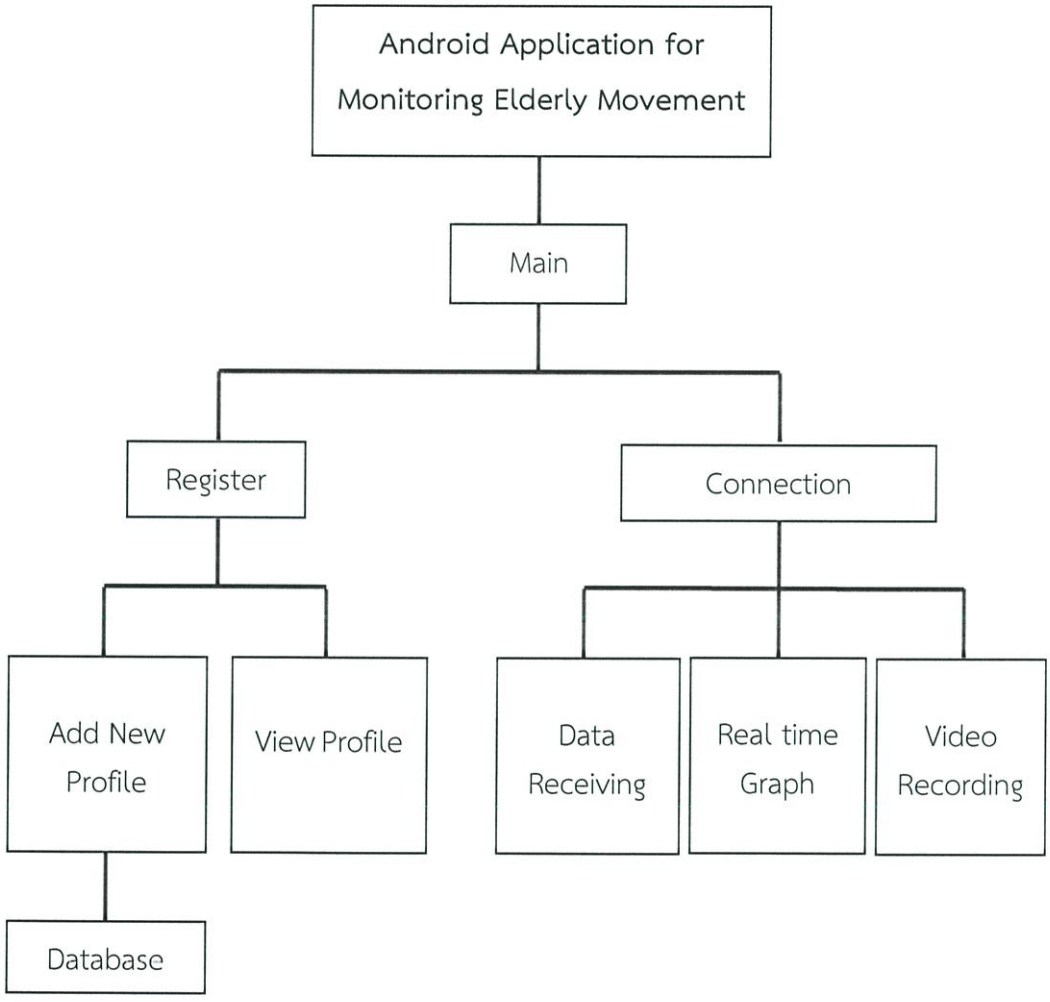
1. เลือกรูปแบบของกราฟที่ต้องการ ซึ่งเราจะทำในลักษณะ XY plot
2. เลือกข้อมูลที่ต้องการนำมา plot มาใส่ลงในเมธอดที่สร้างขึ้นเพื่อรับข้อมูลมา plot
3. ปรับแก้ไขต่างๆ เพื่อให้กราฟที่ได้แสดงผลได้ตามที่เราต้องการ โดยเราสามารถศึกษาเมธอดต่างๆ ได้จากในเว็บไซต์ของ AndroidPlot




รูปที่ 2.5 การสร้างกราฟแบบต่างๆ โดยใช้ AndroidPlot

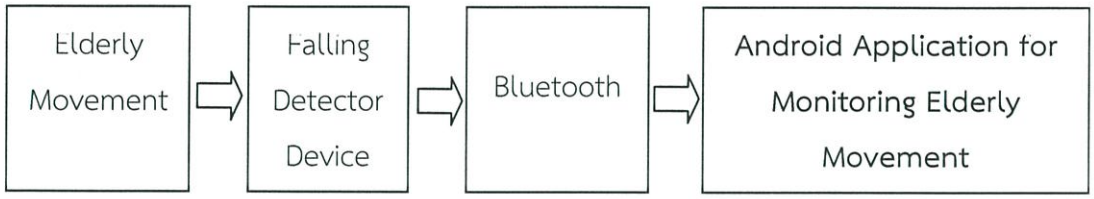
# บทที่ 3
# การออกแบบระบบโปรแกรมประยุกต์

ในการทำโครงงานโปรแกรมประยุกต์บนระบบแอนดรอยด์สำหรับเฝ้าติดตามการเคลื่อนไหวของผู้สูงอายุ ได้มีการออกแบบโครงสร้างการทำงานของโปรแกรม โดยส่วนแรกเป็นหน้าหลักของโปรแกรม ที่จะเลือกไปส่วนการทำงานอื่น ต่อมายังส่วนหน้าลงทะเบียนรับข้อมูลประวัติผู้ใช้งาน และส่วนการเชื่อมต่อบลูทูธ เพื่อรับข้อมูล แล้วพล็อตกราฟแบบเวลาจริง พร้อมกับการบันทึกวิดีโอ




รูปที่ 3.1 โครงสร้างการทำงานของโปรแกรม

## 3.1 การส่งข้อมูลจากเครื่องตรวจจับการล้ม




**รูปที่ 3.2 แผนผังการส่งข้อมูลจากเครื่องตรวจจับการล้ม**

เมื่อเซนเซอร์จากเครื่องตรวจจับคนล้มจับสัญญาณได้ จะส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ และทำการแปลงสัญญาณข้อมูลจากสัญญาณอนาลอกเป็นสัญญาณดิจิตอล จากนั้นส่งข้อมูลไปยังสมาร์ทโฟน ผ่านการเชื่อมต่อบลูทูธ เพื่อสังเกตการเปลี่ยนแปลงอิริยาบทต่างๆ

โดยที่การแปลงสัญญาณข้อมูลจากสัญญาณอนาลอกเป็นสัญญาณดิจิตอลนั้น เราเลือกใช้ไมโครคอนโทรลเลอร์ที่สามารถแปลงข้อมูลได้ 10 bit เพราะว่ามีความละเอียด และแม่นยำ แต่เนื่องจากการรับ-ส่งข้อมูลทางบลูทูธนั้น สามารถรับข้อมูลได้สูงสุด 8 bit จึงต้องมีการตัดส่งข้อมูล โดยแบ่งส่งเป็น 5 ชุด ชุดละ 7 bit อีก 1 bit เป็นตัวเช็คว่าขึ้นข้อมูลชุดใหม่แล้ว ซึ่งมีข้อมูลทั้ง 3 ระนาบ ได้แก่ แกน X, แกน Y และแกน Z ดังรูป

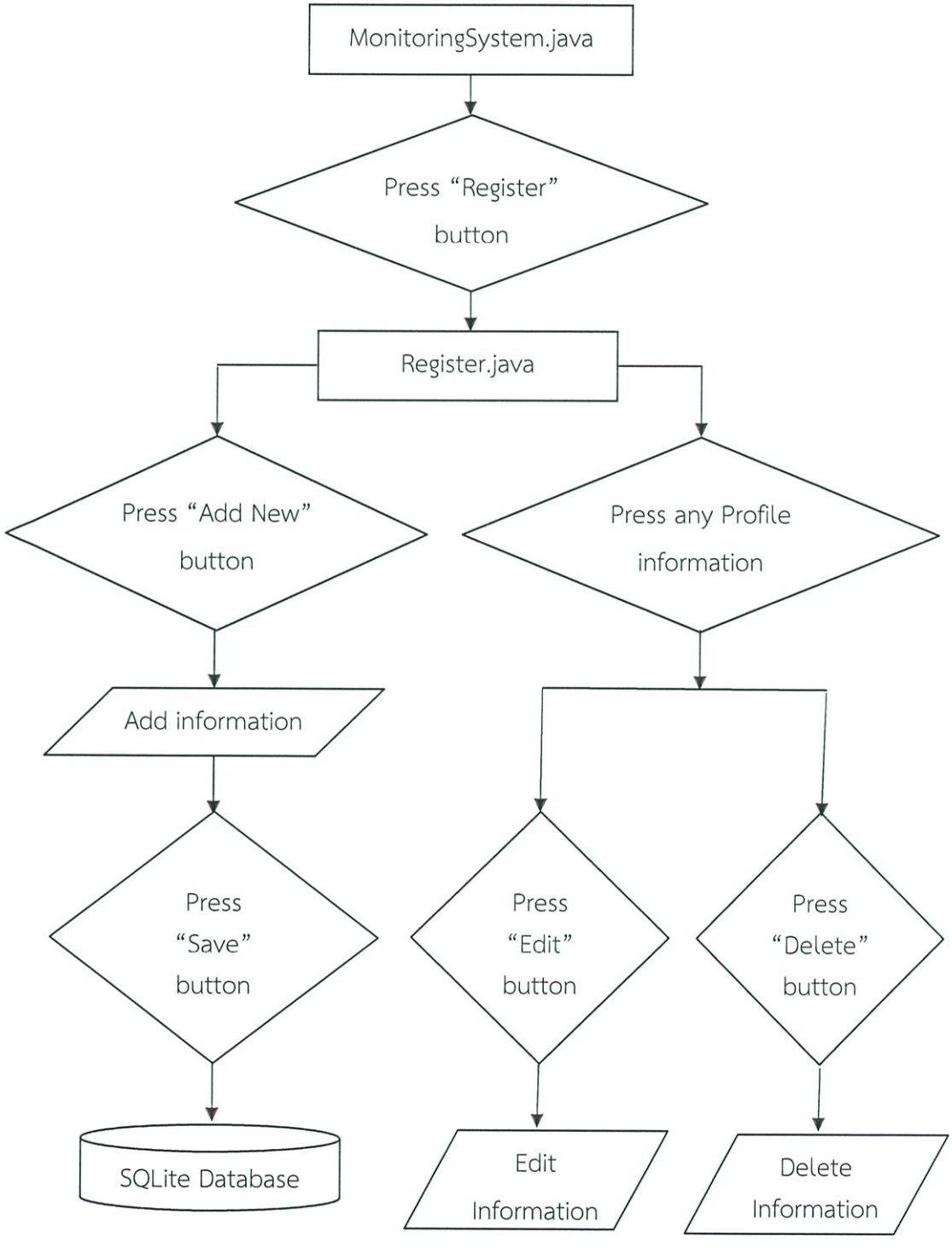
| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **X** | $X_9$ | $X_8$ | $X_7$ | $X_6$ | $X_5$ | $X_4$ | $X_3$ | $X_2$ | $X_1$ | $X_0$ |
| **Y** | $Y_9$ | $Y_8$ | $Y_7$ | $Y_6$ | $Y_5$ | $Y_4$ | $Y_3$ | $Y_2$ | $Y_1$ | $Y_0$ |
| **Z** | $Z_9$ | $Z_8$ | $Z_7$ | $Z_6$ | $Z_5$ | $Z_4$ | $Z_3$ | $Z_2$ | $Z_1$ | $Z_0$ |

| | **ตัวเช็ค** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **data1** | 1 | $X_6$ | $X_5$ | $X_4$ | $X_3$ | $X_2$ | $X_1$ | $X_0$ |
| **data2** | 0 | $Y_3$ | $Y_2$ | $Y_1$ | $Y_0$ | $X_9$ | $X_8$ | $X_7$ |
| **data3** | 0 | $Z_0$ | $Y_9$ | $Y_8$ | $Y_7$ | $Y_6$ | $Y_5$ | $Y_4$ |
| **data4** | 0 | $Z_7$ | $Z_6$ | $Z_5$ | $Z_4$ | $Z_3$ | $Z_2$ | $Z_1$ |
| **data5** | 0 | 0 | 0 | 0 | 0 | 0 | $Z_9$ | $Z_8$ |

**รูปที่ 3.3 การส่งข้อมูลจากเครื่องตรวจจับการล้ม**

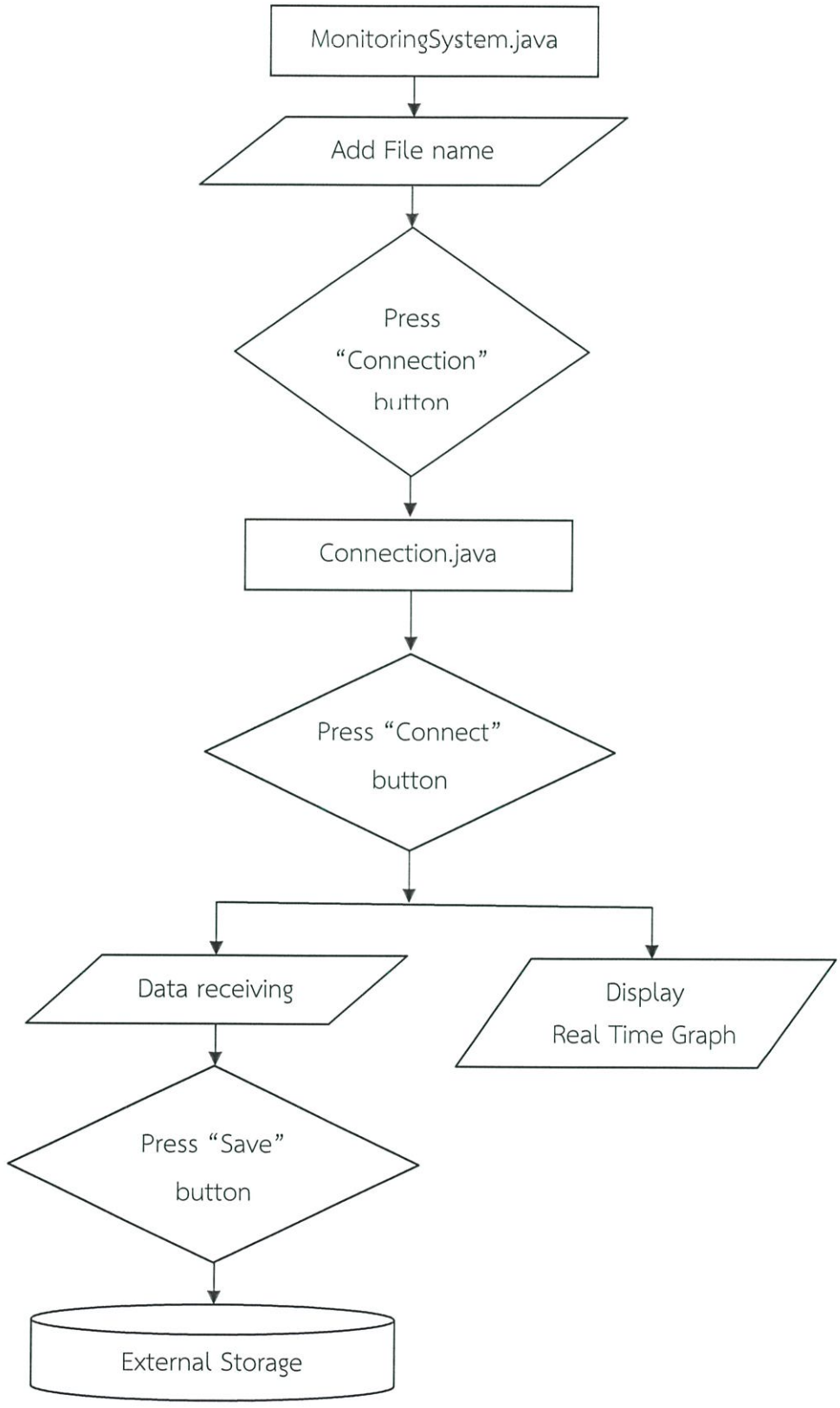
## 3.2 แผนผังการทำงานของโปรแกรมประยุกต์

3.2.1 แผนผังการทำงานส่วนหน้าลงทะเบียนรับข้อมูลประวัติผู้ใช้งาน ซึ่งจะถูกเก็บไว้ในฐานข้อมูล เพื่อสามารถเรียกดูภายหลัง แก้ไขข้อมูล และลบข้อมูลได้



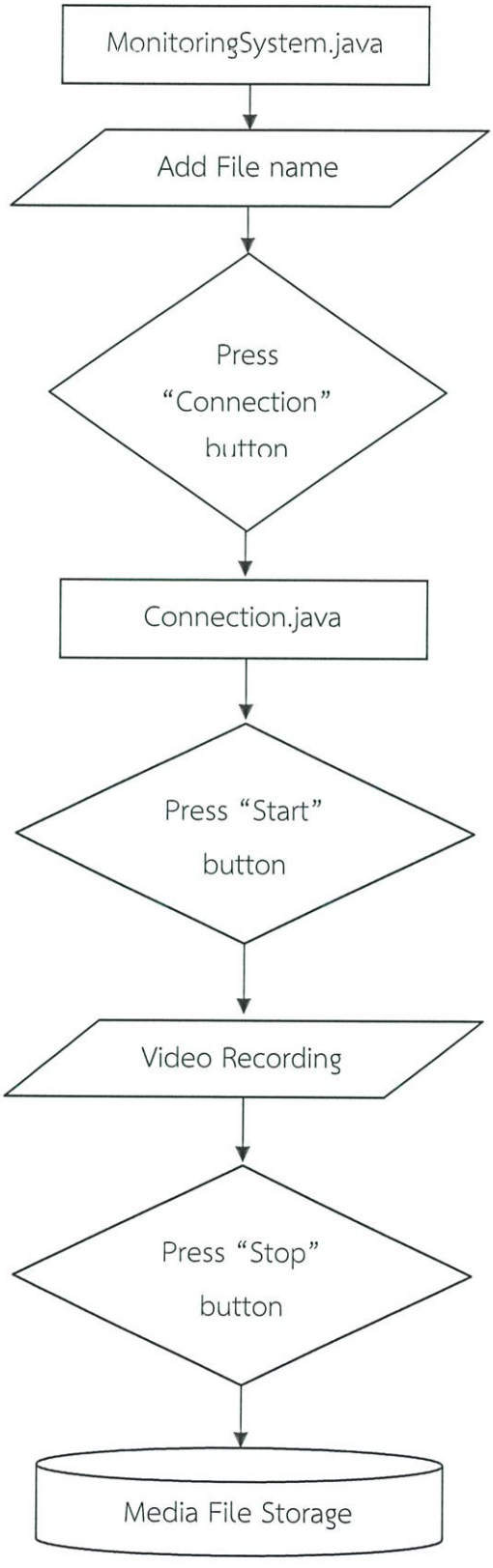

**รูปที่ 3.4 แผนผังการทำงานส่วนบันทึกข้อมูลลงทะเบียนลงในฐานข้อมูล**

3.2.2 แผนผังการทำงานส่วนรับข้อมูล แสดงกราฟ และบันทึกข้อมูล




**รูปที่ 3.5 แผนผังการทำงานส่วนรับข้อมูล แสดงกราฟ และบันทึกข้อมูล**

3.2.3 แผนผังการทำงานส่วนบันทึกวิดีโอ




**รูปที่ 3.6 แผนผังการทำงานส่วนบันทึกวิดีโอ**

# บทที่ 4
# ผลการทำงานของโปรแกรมประยุกต์

## 4.1 การทดลองเชื่อมต่อบลูทูธระหว่างแอพพลิเคชั่นกับเครื่องตรวจจับการล้ม

### 4.1.1 วัตถุประสงค์

เพื่อทดลองการเชื่อมต่อระหว่างแอพพลิเคชั่นกับเครื่องตรวจจับการล้มผ่านทางบลูทูธ

### 4.1.2 ผลการทดลอง

เมื่อทดลองเชื่อมต่อเครื่องตรวจจับคนล้มเข้ากับแอพพลิชั่นผ่านทางบลูทูธ สามารถทำการเชื่อมต่อได้

รูปที่ 4.1 การเชื่อมต่อผ่านบลูทูธ

## 4.2 การทดลองรับข้อมูลจากเครื่องตรวจจับการล้ม พร้อมกับการแสดงกราฟ

### 4.2.1 วัตถุประสงค์

เพื่อทดลองการรับข้อมูล และแสดงกราฟแบบค่าเวลาจริงในแอพพลิเคชั่นที่ส่งผ่านบลูทูธมาจากเครื่องตรวจจับการล้ม

### 4.2.2 ผลการทดลอง

เมื่อทดลองเชื่อมต่อเครื่องตรวจจับคนล้มเข้ากับแอพพลิชั่นผ่านทางบลูทูธ สามารถทำการรับข้อมูล และแสดงผลกราฟแบบค่าเวลาจริงได้

รูปที่ 4.2 การรับข้อมูลจากเครื่องตรวจจับการล้มพร้อมกับการพล็อตกราฟ

## 4.3 การทดลองเก็บค่าความเร่งต่อเวลาระหว่างการทำกิจกรรม

### 4.3.1 วัตถุประสงค์

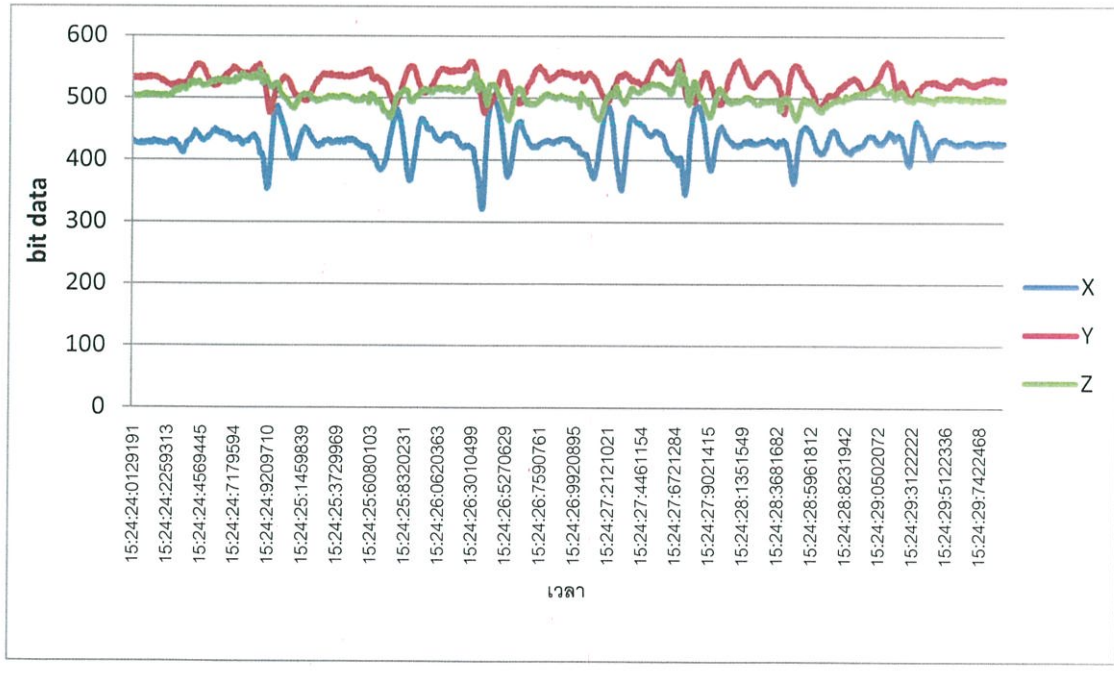
เพื่อทดลองเก็บข้อมูลค่าความเร่ง (bit data) ต่อเวลาที่ได้จากเครื่องตรวจจับการล้มระหว่างการทำกิจกรรมต่างๆ แล้วแสดงเป็นกราฟแบบค่าเวลาจริงในแอพพลิเคชั่น

### 4.3.2 ผลการทดลอง

เมื่อทดลองเก็บค่าข้อมูล สามารถแสดงผลกราฟแบบค่าเวลาจริงได้ และนำข้อมูลที่เก็บได้ในแต่ละกิจกรรมมาเขียนกราฟความสัมพันธ์ระหว่างค่าความเร่ง (bit data) กับเวลา จะพบการเปลี่ยนแปลงกราฟเป็นไปตามลักษณะของแต่ละกิจกรรม ดังนี้

#### 4.3.2.1 ขณะเดิน

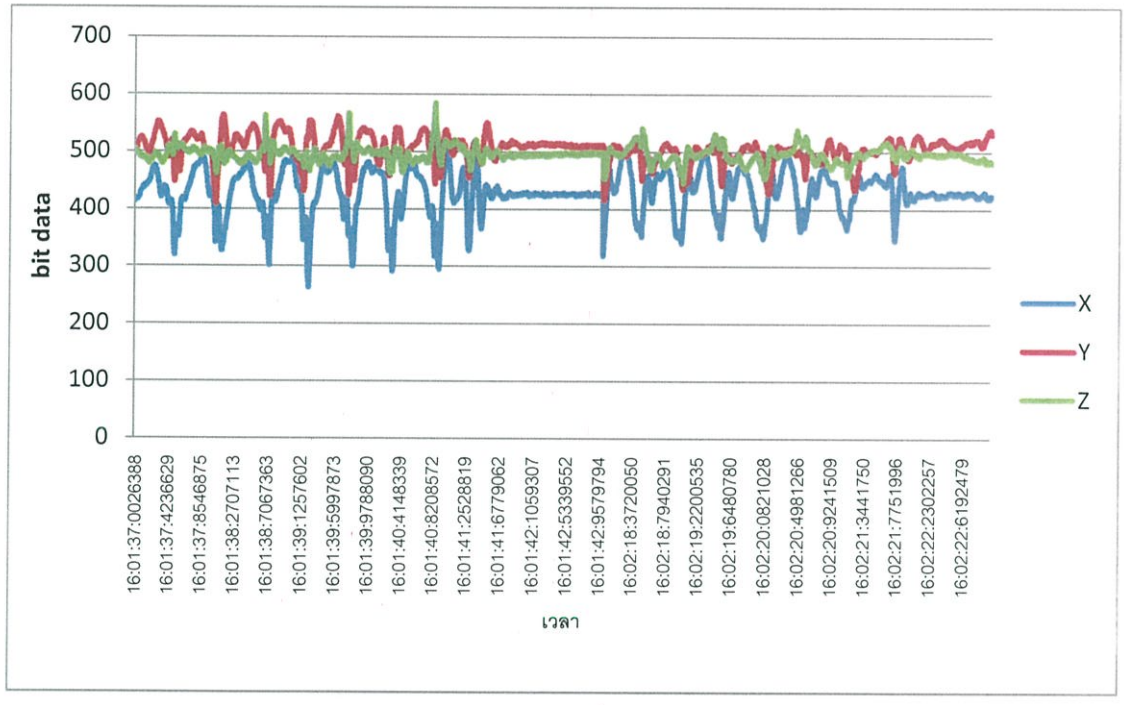
รูปที่ 4.3 การทดลองในท่าเดิน




รูปที่ 4.4 กราฟความเร่งในการเดินต่อเวลา

4.3.2.2 ขณะเดินขึ้นลงบันได

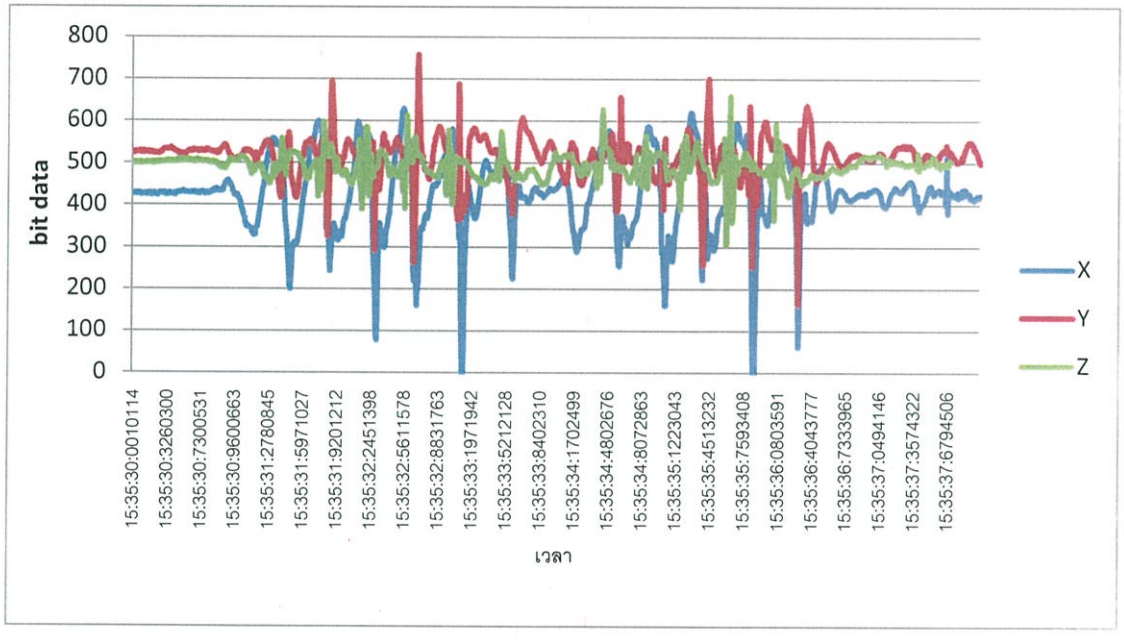
รูปที่ 4.5 ทดลองในท่าเดินขึ้นลงบันได




รูปที่ 4.6 กราฟความเร่งในการเดินขึ้นลงบันไดต่อเวลา

4.3.2.3 ขณะวิ่งเหยาะๆ

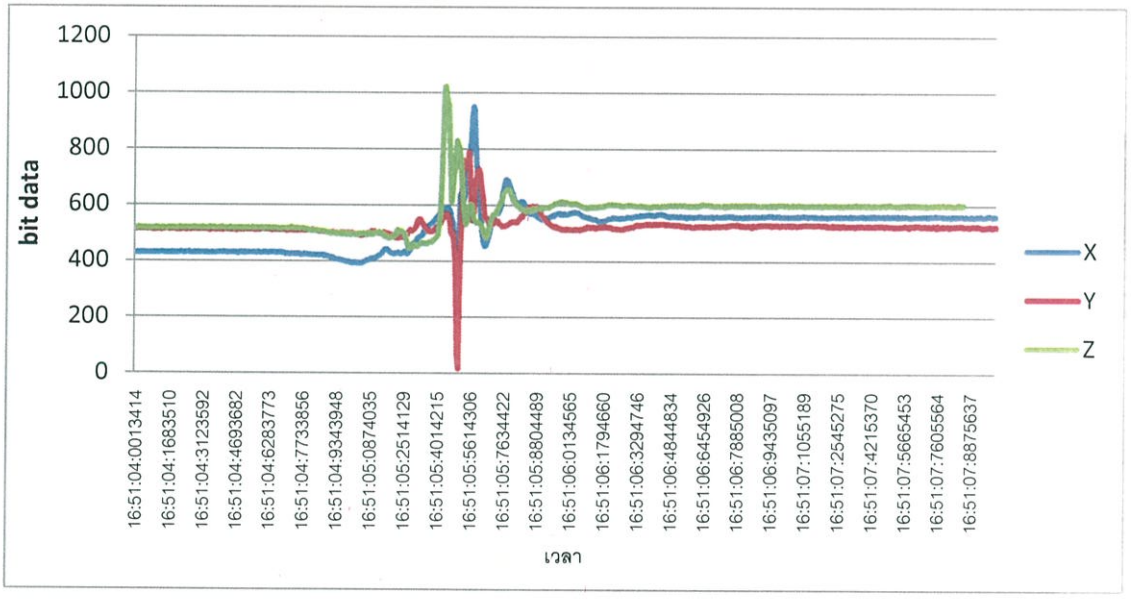
รูปที่ 4.7 การทดลองในท่าวิ่งเหยาะๆ




รูปที่ 4.8 กราฟความเร่งในการวิ่งเหยาะๆต่อเวลา

4.3.2.4 ขณะนั่งแล้วล้ม

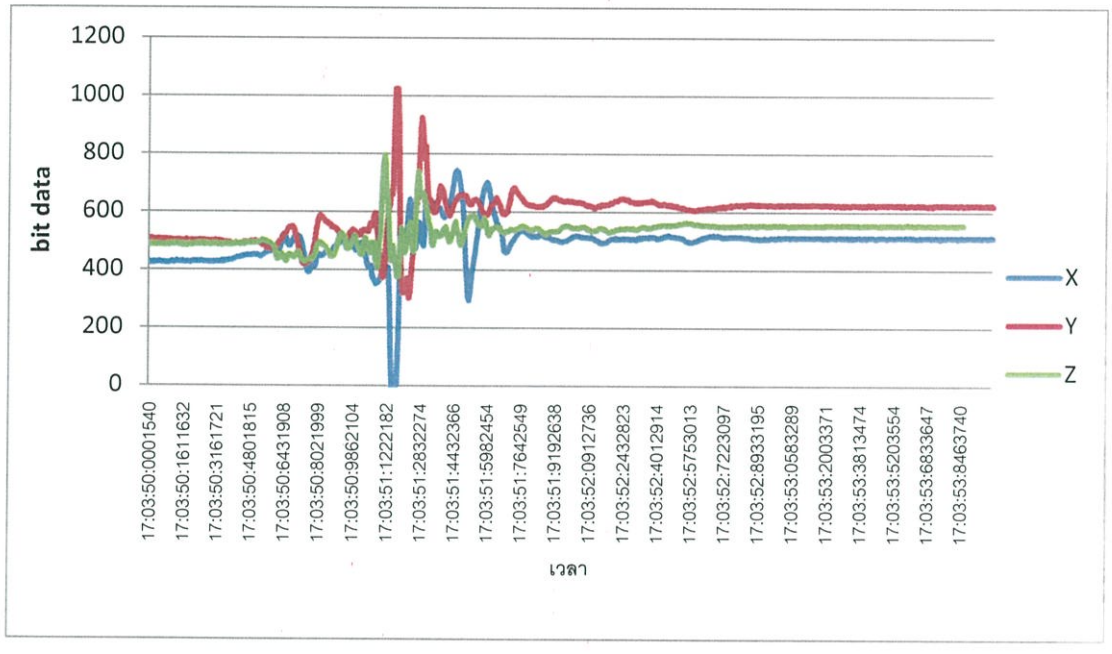
รูปที่ 4.9 การทดลองในท่านั่งแล้วล้ม




รูปที่ 4.10 กราฟความเร่งในการนั่งแล้วล้มต่อเวลา

4.3.2.5 ขณะล้มไปด้านหลัง

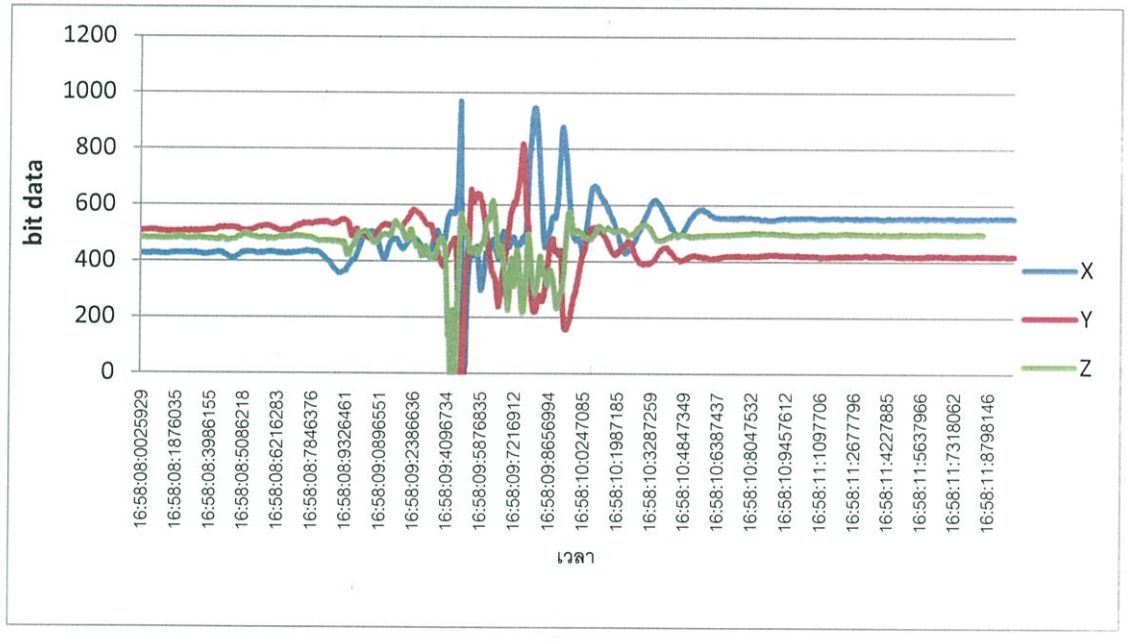
รูปที่ 4.11 การทดลองในท่าล้มไปด้านหลัง




รูปที่ 4.12 กราฟความเร่งในการล้มไปด้านหลังต่อเวลา

4.3.2.6 ขณะล้มไปด้านหน้า

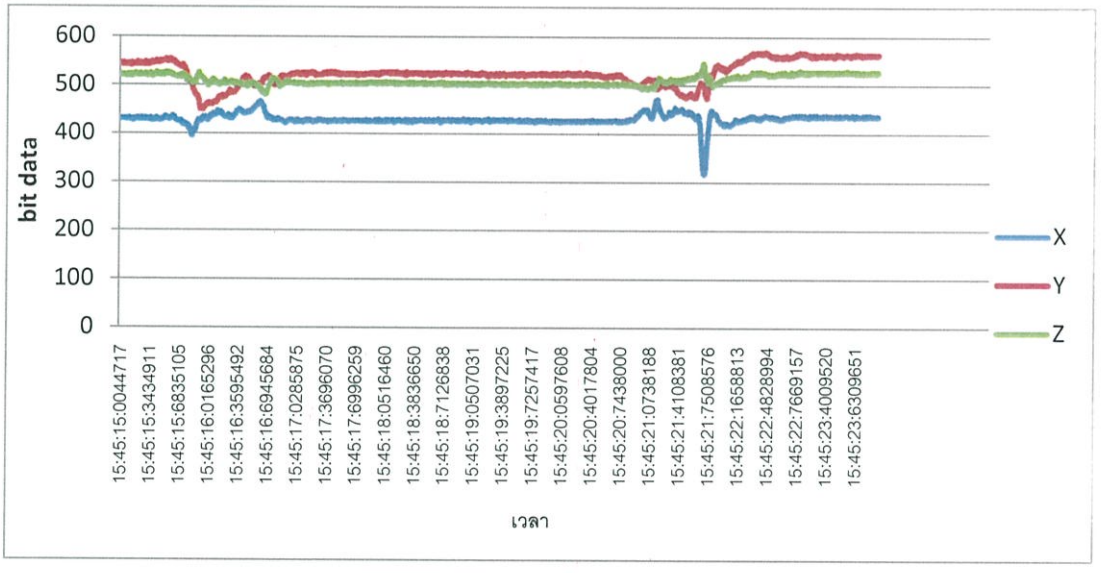
รูปที่ 4.13 การทดลองในท่าล้มไปด้านหน้า




รูปที่ 4.14 กราฟความเร่งในการล้มไปด้านหน้าต่อเวลา

4.3.2.7 ขณะลุกนั่งเก้าอี้

รูปที่ 4.15 การทดลองในท่าลุกนั่งเก้าอี้

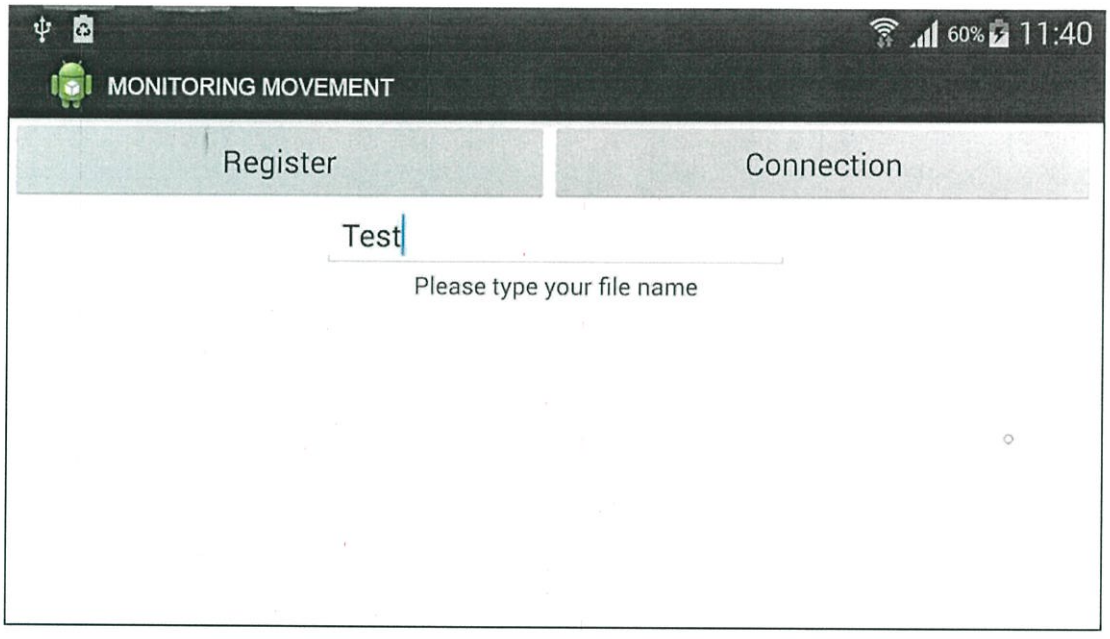



รูปที่ 4.16 กราฟความเร่งในการลุกนั่งเก้าอี้ต่อเวลา

## 4.4 Android Activity

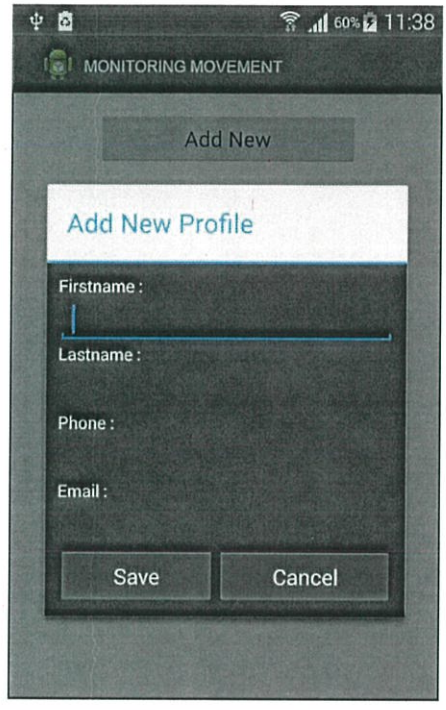
การแสดงผลในแต่ละหน้าของ Android Application for Monitoring Elderly Movement

### 4.4.1 หน้าหลักของ Application ที่จะเชื่อมต่อไปยังส่วนอื่นๆ




รูปที่ 4.17 หน้าหลักของ Application

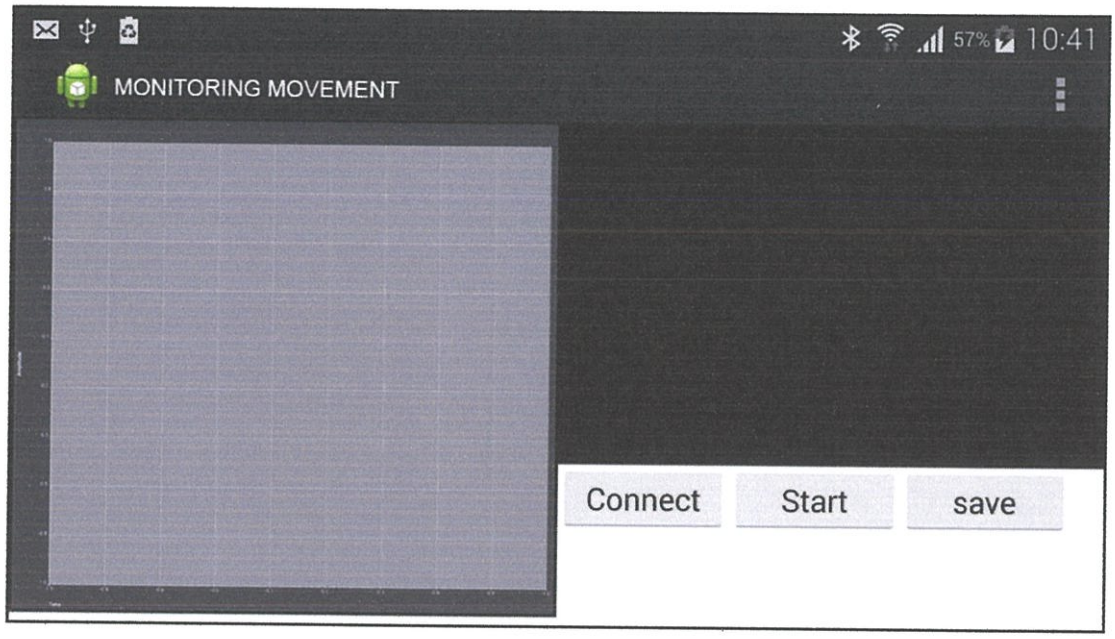
**4.4.2 หน้าลงทะเบียนรับข้อมูลประวัติผู้ใช้งาน** ซึ่งจะถูกเก็บไว้ในฐานข้อมูล เพื่อสามารถเรียกดูภายหลัง แก้ไขข้อมูล และลบข้อมูลได้




รูปที่ 4.18 หน้าลงทะเบียนรับข้อมูลประวัติผู้ใช้งาน

รูปที่ 4.19 หน้าข้อมูลประวัติผู้ใช้งาน

### 4.4.3 หน้าการเชื่อมต่อบลูทูธ และรับข้อมูล




รูปที่ 4.20 หน้าการเชื่อมต่อบลูทูธ

### 4.4.4 หน้าส่วนรับข้อมูล แสดงกราฟแบบเวลาจริง และบันทึกวิดีโอ

รูปที่ 4.21 หน้าส่วนรับข้อมูล แสดงกราฟแบบเวลาจริง และบันทึกวิดีโอ

# บทที่ 5
# สรุปและวิจารณ์ผลการทดลอง

## 5.1 สรุปผลการทดลอง

จากการทดลองเป็นการทดลองที่ศึกษาการเขียนโปรแกรมโดยใช้ภาษา JAVA แล้วนำมาประยุกต์ใช้ในการพัฒนาแอพพลิเคชั่นบนระบบปฏิบัติการแอนดรอยด์ เพื่อทำงานร่วมกับเครื่องตรวจจับการล้มของผู้สูงอายุ โดยจะประกอบด้วยส่วนที่เก็บข้อมูลประวัติของผู้ใช้ ส่วนเก็บค่าข้อมูลการเคลื่อนไหวจากเซนเซอร์วัดความเร่งของเครื่องตรวจจับการล้มโดยการเชื่อมต่อทางบลูทูธ ซึ่งมีการทดลองเก็บค่าความเร่งเพื่อมาเขียนกราฟผลการทดลองเก็บข้อมูลค่าความเร่ง (bit data) ต่อเวลาที่ได้จากเครื่องตรวจจับการล้มระหว่างการทำกิจกรรมต่างๆ พบว่าค่าที่ได้เมื่อนำกลับมาเขียนกราฟมีข้อมูลผิดพลาดเล็กน้อยในบางจุด เมื่อนำกราฟมาเปรียบเทียบกับวิดีโอที่บันทึกรูปแบบการเคลื่อนไหวในส่วนบันทึกวิดีโอ แสดงให้เห็นว่ากราฟจะมีความชันขึ้นเมื่อมีการเปลี่ยนแปลงลักษณะการเคลื่อนไหว ต่อมาจะเป็นส่วนแสดงผลกราฟสัญญาณแบบค่าเวลาจริงที่วัดได้จากเครื่องตรวจจับการล้ม กราฟที่ได้ค่อนข้างสัมพันธ์กับลักษณะการเคลื่อนไหว มีบางจุดที่มีการผิดพลาดเล็กน้อยซึ่งอาจเนื่องมาจากความเร็วในการรับส่งข้อมูลอาจมีการหน่วงเวลาไปบ้าง

การใช้โปรแกรมประยุกต์นี้ ทำให้การนำเครื่องตรวจจับการล้มไปทดลองกับผู้สูงอายุในสถานการณ์จริงสามารถทำได้สะดวกมากขึ้น เนื่องจากการเก็บค่าโดยใช้โปรแกรมประยุกต์บนสมาร์ทโฟนสามารถทำได้ง่าย และยังแสดงผลของกราฟได้เลย ทำให้ทราบว่าลักษณะการเคลื่อนไหวต่างๆ ส่งผลให้มีการแสดงค่าสัญญาณอย่างไร พร้อมกับมีการบันทึกวิดีโอการทดลองไปพร้อมกัน ซึ่งจะสามารถนำข้อมูลที่ได้จากแอพพลิเคชั่นกลับมาวิเคราะห์ต่อที่ห้องทดลองได้สะดวกมากยิ่งขึ้น

## 5.2 ปัญหาและอุปสรรคในการทดลอง

ในส่วนที่เป็นการเชื่อมต่อบลูทูธระหว่างแอนดรอยด์แอพพลิเคชั่นกับคอนโทรลเลอร์ของเครื่องตรวจจับการล้ม เป็นส่วนที่มีความยาก เนื่องจากการเชื่อมต่อบลูทูธในระบบแอนดรอยด์นั้นค่อยข้างมีความซับซ้อนต้องมีการเปิดใช้งานฟังก์ชันย่อยในหลายส่วน ทำให้มีความผิดพลาดระหว่างการทำค่อนข้างมาก

นอกจากนั้นในเรื่องของการสร้างกราฟ ต้องทำการศึกษาเพิ่มเติมค่อนข้างมากเนื่องจากการสร้างกราฟต้องใช้ฟังก์ชันในหลายส่วนมารวมเข้าด้วยกัน จึงเลือกใช้ไลบารี่ที่ต้องศึกษาการใช้งาน และการนำมาประยุกต์ใช้กับข้อมูลที่เรารับมาจากการเชื่อมต่อบลูทูธ

# เอกสารอ้างอิง


[1] ดรจักรชัย โสอินทร์. Android App Development ฉบับสมบูรณ์. นนทบุรี: ไอดีซีฯ; 2555.

[2] พร้อมเลิศ หล่อวิจิตร. คู่มือเขียนแอพ Android ฉบับรวมโค้ด ปรับปรุงใหม่. กรุงเทพฯ: โปรวิชั่น; 2557.

[3] ไพบูลย์ สวัสดิ์ปัญญาโชติ. The Android Developer's Cookbook: รวมโค้ด Android App. กรุงเทพฯ: ทรู ดิจิตอล คอนเท้นท์ แอนด์ มีเดีย; 2554.

[4] ศุภชัย สมพานิช. Professional Android Programming. นนทบุรี: ไอดีซีฯ; 2557.

[5] อนรรฆนงค์ คุณมณี. คู่มือเขียนโปรแกรมภาษา JAVA ฉบับผู้เริ่มต้น. นนทบุรี: ไอดีซีฯ; 2551.

[6] ดร.จักรชัย โสอินทร์. Basic Android App Development. นนทบุรี: ไอดีซีฯ; 2554.