



การพัฒนาแอปพลิเคชันสำหรับรวบรวมข้อมูลจากอุปกรณ์สวมใส่อัจฉริยะ

Development of an Application for Collecting Data from Smart  
Wearable Devices

นาย วิทวัส จันทิมฮะ 664230029

หมู่เรียน 66/46

โครงการนี้เป็นส่วนหนึ่งของการศึกษารายวิชา 7204903

โครงการด้านเทคโนโลยีสารสนเทศ 2

สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยราชภัฏนครปฐม

ภาคเรียนที่ 1 ปีการศึกษา 2567

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญของปัญหา

การเก็บข้อมูลสุขภาพผู้สูงอายุถือว่าจำเป็นในการวินิจฉัยโรคและอาการป่วยทางอายุรกรรม ตัวอย่างเช่น อัตราการเต้นหัวใจ การบาดเจ็บจากการเคลื่อนไหว รวมไปถึงการประสบอุบัติเหตุระหว่างการออกกำลังกาย เป็นต้น ทำให้การตรวจจับสุขภาพและการเคลื่อนไหวเป็นปัจจัยหนึ่งเพื่อช่วยป้องกันอาการเจ็บป่วยของผู้สูงอายุอีกแนวทางหนึ่ง เมื่อพิจารณาถึงปัจจัยการตรวจจับสุขภาพและการเคลื่อนไหว จากการศึกษาในเบื้องต้นพบว่ามีอุปกรณ์ที่ถูกสร้างมาทำการแก้ปัญหา เช่น การใช้กล้องตรวจจับการเคลื่อนไหว ตรวจจับการล้ม ด้วยการใช้เทคนิคการประมวลผลภาพ ซึ่งมีความสะดวก แต่ปัญหาอีกประการหนึ่งคือ ความเป็นส่วนตัวของผู้สูงอายุการติดตั้งกล้องในพื้นที่ส่วนตัวเป็นการรุกรานซึ่งอาจไม่ได้รับการยอมรับจากผู้สูงอายุ และยังมีความคลาดเคลื่อนสูงจากปัจจัยด้านแสงหรือการตีความท่าทางที่ผิดพลาด (เช่น การก้มเก็บของ) ซึ่งเป็นอุปสรรคสำคัญในการนำไปใช้งานจริง อีกตัวเลือกหนึ่งที่น่าสนใจคือ การใช้อุปกรณ์ประเภทสมาร์ทวอตช์ ที่มีความสามารถใกล้เคียงกันคือสามารถตรวจจับการเต้นของหัวใจ ตรวจจับการล้ม รวมไปถึงประวัติการออกกำลังกายได้ แต่ปัญหาที่พบเห็นจากการใช้อุปกรณ์นี้คือ อุปกรณ์จะอาศัยการเชื่อมต่อแบบบลูทูธ และมีการเชื่อมต่อกับแอปพลิเคชันของเจ้าของผลิตภัณฑ์ โดยตรง จึงจะสามารถทำการอ่านข้อมูลจากอุปกรณ์ดังกล่าวได้ อีกทั้งมีระบบการรักษาความปลอดภัยของอุปกรณ์เพื่อป้องกันการเข้าถึงโดยไม่รับอนุญาต ทำให้การนำอุปกรณ์สมาร์ทวอตช์มาใช้งาน จำเป็นต้องมีการเชื่อมต่อกับโทรศัพท์เคลื่อนที่ตลอดเวลา ทำให้การนำข้อมูลมาใช้ประโยชน์ทำได้ยากถึงแม้ว่าจะได้รับอนุญาตแล้วก็ตาม ดังนั้นมีพิจารณาจากการได้รับอนุญาตให้ทำการข้อมูลจากอุปกรณ์สมาร์ทวอตช์แล้ว ปัญหาต่อมาคือ การเชื่อมต่ออุปกรณ์ที่ได้รับอนุญาต ก็ยังมีปัญหาอีกประการหนึ่ง รุ่นและยี่ห้อของอุปกรณ์สมาร์ทวอตช์ที่มีความหลากหลายทั้งรุ่นและยี่ห้อ รวมไปถึงรูปแบบการแสดงผลที่แตกต่างกันในแต่ละแอปพลิเคชัน

จากปัญหาเรื่อง การเชื่อมต่ออุปกรณ์สมาร์ทวอตช์ที่ได้รับอนุญาตจากผู้ป่วยหรือญาติของผู้ป่วย รุ่นและยี่ห้อของอุปกรณ์ที่มีความหลากหลาย รวมไปถึงการแสดงผลที่แตกต่างกันในแต่ละแอปพลิเคชัน ผู้วิจัยจึงมีแนวคิดในการแก้ปัญหาข้างต้นด้วยการเริ่มศึกษาและพัฒนาแอปพลิเคชันที่เป็นเครื่องมือกลาง โดยอาศัยซอฟต์แวร์โอเพนซอร์ส Gadgetbridge ซึ่งมีความสามารถในการเชื่อมต่อกับอุปกรณ์สมาร์ทวอตช์ที่หลากหลายถึง 406 รุ่นเป็นแกนหลัก เพื่อสร้างสะพานเชื่อมระหว่างข้อมูลสุขภาพที่ถูกจำกัดไว้กับผู้ใช้งานจริง

## 1.2 แนวคิดในการแก้ไขปัญหา

จากข้อจำกัดของการตรวจจับการล้มด้วยภาพ ซึ่งมีความผิดพลาดและความคลาดเคลื่อนสูง ผู้จัดทำจึงได้ปรับเปลี่ยนแนวทางการดำเนินโครงการมาสู่การเอาข้อมูลสุขภาพ ผ่านข้อมูลที่สามารถวัดได้จากอุปกรณ์ สมาร์ทวอตช์โดยมุ่งเน้นการวิเคราะห์ แนวโน้ม ของข้อมูลสุขภาพแทนการตรวจจับเหตุการณ์แบบเรียลไทม์ เพื่อแก้ไขปัญหาความหลากหลายของอุปกรณ์ในตลาดและหลีกเลี่ยงการผูกติดกับแอปพลิเคชันของผู้ผลิตรายใดรายหนึ่ง โครงการนี้จึงมีแนวคิดในการใช้ซอฟต์แวร์โอเพนซอร์ส Gadgetbridge ซึ่งมีความสามารถในการเชื่อมต่อกับอุปกรณ์สวมใส่ได้หลากหลายรุ่น มาทำหน้าที่เป็นตัวกลางในการรวบรวมข้อมูล

ดังนั้น แนวทางการแก้ปัญหาคือ การพัฒนาแอปพลิเคชันที่ทำหน้าที่เป็นเครื่องมือกลางซึ่งจะเข้าไป เข้าถึงฐานข้อมูลที่จัดเก็บโดยแอปพลิเคชัน Gadgetbridge โดยตรง แอปพลิเคชันที่พัฒนาขึ้นนี้จะมุ่งเน้นการดึงข้อมูลอัตราการเต้นของหัวใจออกมาประมวลผล และแสดงในรูปแบบของกราฟที่เข้าใจง่าย ซึ่งจะช่วยให้ผู้ใช้งานหรือผู้ดูแลสามารถติดตามแนวโน้มสุขภาพของตนเองได้อย่างสะดวกและมีประสิทธิภาพยิ่งขึ้น

1.3.1 เพื่อพัฒนาโมบายแอปพลิเคชันที่สามารถเข้าถึงและดึงข้อมูลกิจกรรมสุขภาพที่ได้จากฐานข้อมูล SQLite ของแอปพลิเคชัน Gadgetbridge โดยตรง โดยมีอุปกรณ์ที่เข้ากันได้ 406 ตัวรองรับ

1.3.2 เพื่อพัฒนากลไกการส่งออก (Export) ข้อมูลที่ดึงออกมา ให้อยู่ในรูปแบบไฟล์มาตรฐานที่หลากหลาย เพื่อตอบสนองต่อกลุ่มผู้ใช้งานที่แตกต่างกัน ได้แก่

1.3.2.1 .db (SQLite Database) สำหรับนักพัฒนาที่ต้องการนำฐานข้อมูลไปใช้งานต่อทั้งชุด

1.3.2.2 .csv (Comma-Separated Values) สำหรับผู้ใช้ทั่วไปหรือนักวิเคราะห์ที่ต้องการนำข้อมูลไปใช้ในโปรแกรมสเปรดชีต

1.3.2.3 .xls (Microsoft Excel Spreadsheet) เพื่ออำนวยความสะดวกสูงสุดแก่ผู้ใช้ทั่วไปในการเปิดและวิเคราะห์ข้อมูลเบื้องต้น

1.3.3 เพื่อศึกษาและออกแบบโครงสร้างข้อมูลที่ส่งออกเข้าใจง่าย และง่ายต่อการนำไปพัฒนาต่อยอดในแอปพลิเคชันอื่นๆ

## 1.4 ขอบเขตการศึกษา

### 1.4.1 ขอบเขตของระบบ

1.4.1.1 ระบบสามารถส่งออก ข้อมูลที่ดึงออกมาให้อยู่ในรูปแบบไฟล์ได้ 3 รูปแบบ คือ .db, .csv, และ .xls

1.4.1.2 ระบบสามารถแสดงผลข้อมูลอัตราการเต้นของหัวใจในรูปแบบของกราฟที่เข้าใจง่าย

#### 1.4.2 ฮาร์ดแวร์ที่ใช้ในการพัฒนา

##### 1.4.2.1 โน้ตบุ๊ก เอ็มเอสไอ (MSI) รุ่น สิบสาม เจน อินเทล® คอร์™ ไอ

ห้า-หนึ่งสามสี่สองศูนย์เฮซ (13th Gen Intel® Core™ i5-13420H) เป็นเครื่องหลักพัฒนาแอปพลิเคชัน

##### 1.4.2.2 โทรศัพท์เคลื่อนที่ยี่ห้อซัมซุง รุ่นกาแลคซีเอห้าศูนย์ เป็นเครื่องทดสอบระบบ

##### 1.4.2.3 เสียมี่สมาร์ทรุ่น แบนด์เจ็ด จำนวน 1 เครื่อง เป็นอุปกรณ์ที่เก็บข้อมูล

#### 1.4.3 ซอฟต์แวร์ที่ใช้ในการพัฒนา

##### 1.4.3.1 ระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 11 โฮม

##### 1.4.3.2 โปรแกรมแอนดรอยด์สตูดิโอ เวอร์ชัน 2025.1.3.7

##### 1.4.3.3 ระบบจัดการฐานข้อมูลเอสคิวไลต์

##### 1.4.3.4 โปรแกรมวิซวลสตูดิโอ โค้ด

##### 1.4.3.5 เครื่องมือเจมินี ซีแอลไอ

##### 1.4.3.6 แก็ตเจ็ตบริดจ์ เวอร์ชัน 0.87.1

### 1.5 ประโยชน์ที่ได้คาดว่าจะได้รับ

1.5.1 ได้แอปพลิเคชันที่เป็นเครื่องมือช่วยอำนวยความสะดวกในการเข้าถึงข้อมูลสุขภาพส่วนบุคคลที่จัดเก็บโดย Gadgetbridge โดยไม่จำเป็นต้องมีความรู้ทางเทคนิคที่ซับซ้อน ทำให้สามารถนำข้อมูลไปใช้เฝ้าระวังแนวโน้มสุขภาพของตนเองหรือนำไปให้แพทย์ประกอบการวินิจฉัยได้ง่ายขึ้น

1.5.2 ได้เครื่องมือที่ช่วยส่งออกข้อมูลสุขภาพจาก Gadgetbridge ในรูปแบบไฟล์ต่างๆ ซึ่งสามารถนำไปใช้เป็นข้อมูลตั้งต้น ในการศึกษา หรือพัฒนาต่อยอดเป็นแอปพลิเคชันด้านสุขภาพอื่นๆ ในอนาคตได้

### 1.6 นิยามศัพท์เฉพาะ

#### 1.6.1 แอปพลิเคชันแก็ตเจ็ตบริดจ์ (Gadgetbridge)

คือโมบายแอปพลิเคชันประเภทโอเพนซอร์สสำหรับระบบปฏิบัติการแอนดรอยด์ ที่ใช้เชื่อมต่อกับอุปกรณ์สวมใส่อัจฉริยะ มีจุดเด่นสำคัญคือการเน้นความเป็นส่วนตัวของผู้ใช้ โดยข้อมูลสุขภาพทั้งหมดจะถูกเก็บไว้ในฐานข้อมูลบนอุปกรณ์ของผู้ใช้เท่านั้น และไม่ส่งข้อมูลกลับไปยังเซิร์ฟเวอร์ภายนอก ในโครงการนี้ Gadgetbridge จึงทำหน้าที่เป็นแหล่งข้อมูลหลัก ที่แอปพลิเคชันที่พัฒนาขึ้นจะเข้าไปอ่านข้อมูลเพื่อนำมาประมวลผลต่อไป

#### 1.6.2 ฐานข้อมูลเอสคิวไลต์ (SQLite)

เป็นระบบจัดการฐานข้อมูลขนาดเล็กที่ไม่ต้องการการติดตั้งเซิร์ฟเวอร์ เนื่องจากมีขนาดเล็กและทำงานได้ในตัวเองจึงเป็นมาตรฐานที่นิยมใช้ในแอปพลิเคชันแอนดรอยด์ส่วนใหญ่ รวมถึง Gadgetbridge เพื่อจัดเก็บข้อมูลที่มีโครงสร้างภายในเครื่อง ดังนั้น บทบาทสำคัญในโครงการนี้คือการที่แอปพลิเคชันที่พัฒนาขึ้นต้องมีความสามารถในการเชื่อมต่อและสืบค้นข้อมูลจากไฟล์ฐานข้อมูล SQLite นี้โดยตรง

#### 1.6.5 เครื่องมือเจมินี ซีแอลไอ (Gemini CLI)

เป็นเครื่องมือสำหรับผู้พัฒนาที่ใช้สื่อสารกับ AI รุ่น Gemini ของ Google ผ่านหน้าต่างคำสั่ง ในโครงการนี้ Gemini CLI ทำหน้าที่เป็น ผู้ช่วยเขียนโปรแกรม เพื่อเร่งกระบวนการพัฒนาแก้ปัญหา และให้คำแนะนำทางเทคนิค

#### 1.6.6 โปรแกรมวิซวลสตูดิโอ โค้ด (Visual Studio Code)

เป็นโปรแกรมที่ใช้จัดการไฟล์เอกสารและไฟล์อื่นๆ ที่ไม่ใช่โค้ดหลัก ตรวจสอบไฟล์ข้อมูลที่ส่งออกมา (เช่น .csv, .txt) และใช้เขียนสคริปต์เสริมเล็กๆ เพื่อทำงานอัตโนมัติ

#### 1.6.7 โปรแกรมแอนดรอยด์สตูดิโอ เวอร์ชัน 2025.1.3.7 (Visual Studio Code)

เป็นสภาพแวดล้อมสำหรับการพัฒนาแบบเบ็ดเสร็จ และเป็นเครื่องมือหลักอย่างเป็นทางการสำหรับการสร้างแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ โดยประกอบด้วยเครื่องมือสำคัญที่ครบครันตั้งแต่โปรแกรมแก้ไขโค้ด เครื่องมือออกแบบหน้าจอ ระบบคอมไพล์โปรเจกต์ ไปจนถึงเครื่องมือสำหรับดีบั๊กและวิเคราะห์ประสิทธิภาพจึงถือเป็นโปรแกรมหลักที่ใช้ในการเขียนโค้ดออกแบบ คอมไพล์ และทดสอบแอปพลิเคชันในโครงการนี้ทั้งหมด

## บทที่ 2

### ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในการพัฒนาแอปพลิเคชันสำหรับรวบรวมและแสดงผลข้อมูลอัตราการเต้นของหัวใจจากอุปกรณ์สวมใส่อัจฉริยะ ผู้จัดทำได้ศึกษาทฤษฎี หลักการ และงานวิจัยที่เกี่ยวข้อง เพื่อใช้เป็นพื้นฐานในการออกแบบและพัฒนาระบบให้บรรลุตามวัตถุประสงค์ที่กำหนดไว้ โดยมีรายละเอียดดังต่อไปนี้

#### 2.1 ทฤษฎีที่เกี่ยวข้องกับปัญหา

##### 2.1.1 สภาวะสังคมผู้สูงอายุและความเสี่ยงด้านสุขภาพ

สังคมผู้สูงอายุ (Aging Society) คือสภาวะที่สังคมมีสัดส่วนของประชากรผู้ที่มีอายุ 60 ปีขึ้นไปเพิ่มสูงขึ้น ในขณะที่อัตราการเกิดและประชากรวัยทำงานลดลง สภาวะดังกล่าวได้นำมาซึ่งความท้าทายด้านสาธารณสุขโดยตรง เนื่องจากผู้สูงอายุเป็นกลุ่มวัยที่มีความเสื่อมถอยของร่างกายและมีความเสี่ยงต่อปัญหาสุขภาพสูง

หนึ่งในความเสี่ยงที่สำคัญและเป็นอันตรายที่สุดคือ การพลัดตกหกล้ม ซึ่งเป็นปัญหาที่โครงการนี้ให้ความสำคัญ จากข้อมูลพบว่าผู้ที่มีอายุ 65 ปีขึ้นไปมีความเสี่ยงต่อการหกล้มสูงถึง 28 – 35% และยังเป็นสาเหตุการเสียชีวิตอันดับสองในกลุ่มผู้สูงอายุของไทย นอกเหนือจากอุบัติเหตุที่เกิดขึ้นเฉียบพลันแล้ว การเฝ้าระวังความผิดปกติของข้อมูลสุขภาพพื้นฐาน เช่น อัตราการเต้นของหัวใจ ก็มีความสำคัญอย่างยิ่งในการประเมินความเสี่ยงและป้องกันเหตุฉุกเฉินที่อาจเกิดขึ้น ดังนั้นการนำเทคโนโลยีเข้ามาช่วยในการรวบรวมและติดตามข้อมูลสุขภาพของผู้สูงอายุจึงเป็นแนวทางที่จำเป็นและมีประโยชน์อย่างยิ่งในสภาวะสังคมปัจจุบัน

##### 2.1.2 บทบาทของอุปกรณ์สวมใส่อัจฉริยะในการเฝ้าระวังสุขภาพ

เทคโนโลยีอุปกรณ์สวมใส่อัจฉริยะ (Smart Wearable Devices) ได้เข้ามามีบทบาทสำคัญในการเฝ้าระวังและติดตามสุขภาพส่วนบุคคล เนื่องจากความสามารถในการเก็บรวบรวมข้อมูลทางสรีรวิทยาได้อย่างต่อเนื่องตลอด 24 ชั่วโมง อุปกรณ์เหล่านี้มักใช้เซ็นเซอร์ประเภท Photoplethysmography (PPG) ซึ่งใช้หลักการปล่อยแสงสีเขียวไปที่ผิวหนังเพื่อตรวจจับการเปลี่ยนแปลงของปริมาณเลือดที่ไหลผ่านหลอดเลือดฝอย ทำให้สามารถประมาณค่าอัตราการเต้นของหัวใจได้อย่างต่อเนื่องและไม่รบกวนการใช้ชีวิตประจำวัน (J. Allen, 2007) ข้อมูลที่ได้จากอุปกรณ์เหล่านี้จึงสามารถใช้เป็นดัชนีชี้วัดเบื้องต้นถึงความผิดปกติที่อาจเกิดขึ้นกับร่างกายได้

## 2.2 เทคโนโลยีที่เกี่ยวข้องกับการพัฒนาระบบ

### 2.2.1 แอปพลิเคชัน แก็ดเจ็ตบริดจ์ (Gadgetbridge)

แก็ดเจ็ตบริดจ์เป็นโมบายแอปพลิเคชันประเภทโอเพนซอร์สสำหรับระบบปฏิบัติการแอนดรอยด์ที่ใช้เชื่อมต่อกับอุปกรณ์สวมใส่ มีจุดเด่นสำคัญคือการเน้นความเป็นส่วนตัวของผู้ใช้ โดยข้อมูลสุขภาพทั้งหมดจะถูกเก็บไว้ในฐานข้อมูลบนอุปกรณ์ของผู้ใช้เท่านั้น และไม่ส่งข้อมูลกลับไปยังเซิร์ฟเวอร์ภายนอก ในโครงการนี้ Gadgetbridge ทำหน้าที่เป็นตัวกลางและเป็นแหล่งข้อมูลหลัก แอปพลิเคชันที่พัฒนาขึ้นจะเข้าไปอ่านข้อมูล

### 2.2.2 ฐานข้อมูลเอสคิวไลต์ (SQLite)

เอสคิวไลต์เป็นระบบจัดการฐานข้อมูลขนาดเล็กที่ไม่ต้องการการติดตั้งเซิร์ฟเวอร์ (Serverless) และถูกผนวกรวมเป็นส่วนหนึ่งของแอปพลิเคชันโดยตรง แอปพลิเคชันแอนดรอยด์ส่วนใหญ่นิยมใช้ SQLite ในการจัดเก็บข้อมูลภายในเครื่อง เช่นเดียวกับ Gadgetbridge ดังนั้น บทบาทสำคัญในโครงการนี้คือการที่แอปพลิเคชันที่พัฒนาขึ้นต้องมีความสามารถในการเชื่อมต่อและสืบค้นข้อมูลจากไฟล์ฐานข้อมูล SQLite นี้โดยตรง

### 2.2.3 รูปแบบไฟล์สำหรับการส่งออกข้อมูล (Data Export Formats)

เพื่อให้ข้อมูลที่ได้สามารถนำไปใช้งานต่อยอดได้หลากหลาย วัตถุประสงค์ของโครงการนี้จึงกำหนดให้มีการส่งออกไฟล์ในรูปแบบมาตรฐานต่างๆ ได้แก่:

2.3.3.1 db (SQLite Database) สำหรับนักพัฒนาที่ต้องการนำฐานข้อมูลไปใช้งานต่อทั้งชุด

2.3.3.2 .csv (Comma-Separated Values) สำหรับผู้ใช้ทั่วไปหรือนักวิเคราะห์ที่ต้องการนำข้อมูลไปใช้ในโปรแกรมสเปรดชีต

2.3.3.3 .xls (Microsoft Excel Spreadsheet): เพื่ออำนวยความสะดวกสูงสุดแก่ผู้ใช้ทั่วไปในการเปิดและวิเคราะห์ข้อมูลเบื้องต้น

### 2.2.4 เทคโนโลยีการสื่อสารไร้สายสำหรับอุปกรณ์สวมใส่

ในการที่อุปกรณ์สวมใส่อัจฉริยะ (Smartwatch) จะสามารถส่งข้อมูลสุขภาพมายังสามารถพอนได้นั้น จำเป็นต้องอาศัยโปรโตคอลการสื่อสารไร้สายระยะสั้น ซึ่งเทคโนโลยีที่นิยมใช้ในปัจจุบันมี 2 ประเภทหลัก คือ

#### 2.2.4.1 บลูทูธพลังงานต่ำ (Bluetooth Low Energy - BLE)

เป็นมาตรฐานการสื่อสารไร้สายที่ถูกออกแบบมาเพื่ออุปกรณ์ขนาดเล็กที่ใช้พลังงานจากแบตเตอรี่โดยเฉพาะ เช่น อุปกรณ์สวมใส่, เซ็นเซอร์ทางการแพทย์, และอุปกรณ์ IoT (Internet of Things) จุดเด่นที่สุดของ BLE คือ อัตราการใช้พลังงานที่ต่ำมาก ทำให้สามารถทำงานได้ต่อเนื่องเป็นเวลานานโดยไม่ต้องชาร์จบ่อยครั้ง BLE เหมาะสำหรับการส่งข้อมูลเป็นช่วงๆ (Bursts of data) เช่น การส่งข้อมูลอัตราการเต้นของหัวใจทุกๆ 1-5 นาที ซึ่งสอดคล้องกับการทำงานของอุปกรณ์ในโครงการนี้เป็นอย่างดี

#### 2.2.4.2 เทคโนโลยี ANT+

เป็นโปรโตคอลการสื่อสารไร้สายอีกประเภทหนึ่งที่นิยมใช้ในกลุ่มอุปกรณ์กีฬาและฟิตเนสโดยเฉพาะ เช่น สายคาดวัดอัตราการเต้นของหัวใจ, เซ็นเซอร์วัดรอบขาจักรยาน, หรือ Foot Pod สำหรับวิ่ง จุดเด่นของ ANT+ คือความสามารถในการสร้างเครือข่ายที่ยืดหยุ่น ทำให้อุปกรณ์หนึ่งสามารถส่งข้อมูลไปยังอุปกรณ์รับหลายๆ ตัวได้พร้อมกัน (เช่น ส่งข้อมูลจากสายคาดหน้าอกไปยังนาฬิกาและคอมพิวเตอร์จักรยานได้ในเวลาเดียวกัน) แม้ว่าในโครงการนี้จะเน้นการเชื่อมต่อแบบหนึ่งต่อหนึ่ง แต่การทำความเข้าใจเทคโนโลยี ANT+ ก็เป็นสิ่งสำคัญเนื่องจากอุปกรณ์บางรุ่นที่ Gadgetbridge รองรับอาจใช้โปรโตคอลนี้ในการสื่อสาร

#### 2.2.5 ภาษาและเครื่องมือในการพัฒนา (Development Languages and Tools)

##### 2.2.5.1 ภาษาโคทลิน (Kotlin)

เป็นภาษาโปรแกรมเชิงวัตถุ ที่ทำงานบน Java Virtual Machine (JVM) ซึ่งถูกพัฒนาขึ้นโดยบริษัท JetBrains ในปี ค.ศ. 2016 Google ได้ประกาศให้ Kotlin เป็นภาษาโปรแกรมที่รองรับอย่างเป็นทางการสำหรับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ และต่อมาได้แนะนำให้เป็นภาษาหลัก (Preferred Language) สำหรับการพัฒนาแอปพลิเคชันใหม่ๆ (Google, n.d.)

จุดเด่นของ Kotlin คือไวยากรณ์ (Syntax) ที่มีความกระชับและทันสมัย ช่วยลดปริมาณโค้ดที่ต้องเขียนลงได้อย่างมากเมื่อเทียบกับภาษา Java นอกจากนี้ยังมีคุณสมบัติด้านความปลอดภัยที่สำคัญคือ Null Safety ซึ่งถูกออกแบบมาเพื่อช่วยขจัดข้อผิดพลาดประเภท NullPointerException หรือ "ข้อผิดพลาดพินล้นดอลลาร์" ซึ่งเป็นหนึ่งในข้อผิดพลาดที่พบบ่อยที่สุดในการพัฒนาซอฟต์แวร์ (T. Hoare, 2009) ในโครงการนี้ ภาษา Kotlin ถูกใช้เป็นภาษาหลักในการพัฒนาตรรกะการทำงาน (Business Logic) ทั้งหมดของแอปพลิเคชัน

##### 2.2.5.2 ภาษา XML (Extensible Markup Language)

เป็นภาษามาร์กอัป (Markup Language) ที่ใช้ในการกำหนดโครงสร้างและจัดเก็บข้อมูลในรูปแบบที่ทั้งมนุษย์และเครื่องคอมพิวเตอร์สามารถอ่านเข้าใจได้ ในบริบทของการพัฒนาแอปพลิเคชันแอนดรอยด์ XML ถูกใช้เป็นภาษาหลักในการออกแบบและกำหนดโครงสร้างของส่วนติดต่อผู้ใช้งาน (User Interface - UI)

นักพัฒนาจะใช้ XML เพื่อกำหนดองค์ประกอบต่างๆ บนหน้าจอ เช่น ปุ่ม (Button), กล่องข้อความ (TextView), หรือช่องรับข้อมูล (EditText) รวมถึงกำหนดคุณลักษณะต่างๆ เช่น ขนาด, สี, และการจัดวางองค์ประกอบเหล่านั้น การแยกโค้ดส่วน UI (ที่เขียนด้วย XML) ออกจากโค้ดส่วนตรรกะการทำงาน (ที่เขียนด้วย Kotlin) ช่วยให้การพัฒนาและการบำรุงรักษาแอปพลิเคชันทำได้ง่ายและเป็นระเบียบมากขึ้น (Google, n.d.)



## 2.3 งานวิจัยและแอปพลิเคชันที่เกี่ยวข้อง

จากการทบทวนวรรณกรรมและสำรวจแอปพลิเคชันที่มีอยู่ในปัจจุบัน พบว่าแอปพลิเคชันด้านสุขภาพส่วนใหญ่มักส่งข้อมูลของผู้ใช้กลับไปยังเซิร์ฟเวอร์ของผู้พัฒนา ซึ่งอาจก่อให้เกิดความกังวลด้านความเป็นส่วนตัวของข้อมูล (T. Vo, H. Nguyen, and V. Le, 2021) ในขณะที่แอปพลิเคชันที่เน้นความเป็นส่วนตัวสูงอย่าง Gadgetbridge กลับขาดคุณสมบัติที่จำเป็นในการส่งออกและแสดงผลข้อมูลเพื่อการวิเคราะห์ (Freeyourgadget, 2024)

จากการสำรวจเพิ่มเติม พบว่ามีความพยายามในการเข้าถึงข้อมูลของ Gadgetbridge ในกลุ่มนักพัฒนา โดยมีการเผยแพร่สคริปต์ (Script) สำหรับคอมพิวเตอร์บนแพลตฟอร์มอย่าง GitHub ซึ่งสามารถอ่านและแปลงข้อมูลจากไฟล์ฐานข้อมูล SQLite ได้ (M. Sh. et al., 2023) อย่างไรก็ตาม เครื่องมือเหล่านี้ต้องการความรู้ทางเทคนิคในการใช้งานผ่าน Command-line และไม่ได้อยู่ในรูปแบบของแอปพลิเคชันที่ผู้ใช้ทั่วไปสามารถใช้งานได้โดยง่าย นอกจากนี้ยังมีแอปพลิเคชันประเภทวิเคราะห์ข้อมูลทั่วไปที่สามารถเปิดไฟล์ CSV และสร้างกราฟได้ แต่ก็ไม่สามารถแก้ปัญหาหลักคือ ผู้ใช้ยังคงต้องผ่านขั้นตอนที่ซับซ้อนในการส่งออกไฟล์ฐานข้อมูลจาก Gadgetbridge และแปลงเป็นไฟล์ CSV ด้วยตนเองก่อน ซึ่งเป็นอุปสรรคสำคัญ

ปัจจุบันยังไม่พบแอปพลิเคชันที่ทำหน้าที่เป็นเครื่องมือกลาง (Middleware Tool) สำหรับผู้ใช้ Gadgetbridge โดยเฉพาะ ที่มุ่งเน้นการแก้ปัญหาการเข้าถึงและแสดงผลข้อมูลอัตราการเต้นของหัวใจในรูปแบบกราฟที่เข้าใจง่าย เพื่อนำไปใช้ประกอบการปรึกษาแพทย์โดยตรง โครงการนี้จึงถูกพัฒนาขึ้นเพื่อเติมเต็มช่องว่างดังกล่าว โดยมุ่งหวังที่จะเพิ่มอรรถประโยชน์ให้แก่ข้อมูลสุขภาพที่ถูกจัดเก็บไว้ สร้างสะพานเชื่อมระหว่างข้อมูลดิบทางเทคนิคกับผู้ใช้งานทั่วไปและบุคลากรทางการแพทย์

## บทที่3

### วิธีการดำเนินงาน

ในการพัฒนา "แอปพลิเคชันสำหรับรวบรวมข้อมูลจากอุปกรณ์สวมใส่อัจฉริยะ" ผู้จัดทำได้วางแผนขั้นตอนการดำเนินงานอย่างเป็นระบบ เพื่อให้การพัฒนามีประสิทธิภาพและบรรลุตามวัตถุประสงค์ที่กำหนดไว้ในบทที่ 1 โดยบทนี้จะมุ่งเน้นการอธิบายถึงแนวคิดในการออกแบบสถาปัตยกรรม แผนผังการทำงาน และกระบวนการทางเทคนิคที่ใช้ในการพัฒนาเพื่อให้ได้มาซึ่งซอฟต์แวร์ต้นแบบ ซึ่งเป็นการนำเสนอในมุมมองของผู้พัฒนา ไม่ใช่การอธิบายวิธีการใช้งานสำหรับผู้ใช้งานปลายทาง โดยมีรายละเอียดของวิธีการดำเนินงานดังต่อไปนี้

#### 3.1 การวิเคราะห์ระบบเป้าหมาย (Gadgetbridge Application Analysis)

ขั้นตอนแรกของการดำเนินงานคือการวิเคราะห์ระบบเป้าหมายซึ่งเป็นแหล่งข้อมูลหลักของโครงการ นั่นคือ แอปพลิเคชัน Gadgetbridge เพื่อทำความเข้าใจกลไกการทำงาน, วิธีการจัดเก็บข้อมูล, และโครงสร้างของข้อมูลอย่างละเอียด

##### 3.1.1 การติดตั้งและทดสอบฟังก์ชันการทำงาน

ผู้จัดทำได้เริ่มต้นด้วยการติดตั้งแอปพลิเคชัน Gadgetbridge เวอร์ชัน 0.87.1 จากแหล่ง F-Droid ลงบนอุปกรณ์ทดสอบ โทรศัพท์เคลื่อนที่ยี่ห้อซัมซุง รุ่นกาแลคซีเอห้าศูนย์ จากนั้นจึงทำการจับคู่ กับอุปกรณ์สวมใส่อัจฉริยะ เสียวมีสมาร์ทวatches แบบดเจ็ด และทดสอบฟังก์ชันการทำงานพื้นฐาน เช่น การซิงโครไนซ์ข้อมูล, การวัดอัตราการเต้นของหัวใจ เพื่อให้มั่นใจว่าแอปพลิเคชัน Gadgetbridge สามารถรวบรวมและบันทึกข้อมูลจากอุปกรณ์ได้อย่างถูกต้องครบถ้วน

##### 3.1.2 การสำรวจและระบุตำแหน่งฐานข้อมูล

หลังจากยืนยันว่า Gadgetbridge สามารถเก็บข้อมูลได้แล้ว ขั้นตอนต่อไปคือการระบุตำแหน่งของไฟล์ฐานข้อมูลที่ใช้ในการจัดเก็บข้อมูลดังกล่าว จากการศึกษเอกสารสำหรับนักพัฒนาของ Gadgetbridge และการใช้เครื่องมือ File Explorer บนระบบปฏิบัติการแอนดรอยด์ พบว่าข้อมูลทั้งหมดถูกจัดเก็บไว้ในฐานข้อมูล SQLite ในไฟล์ชื่อ Gadgetbridge.db ซึ่งมีที่อยู่ที่เริ่มต้นที่ /storage/emulated/0/Android/data/nodomain.freeyourgadget.gadgetbridge/files ภายใต้อัปเดตด้านความปลอดภัยของระบบปฏิบัติการแอนดรอยด์ ที่ทำให้แอปพลิเคชันอื่นไม่สามารถเข้าถึงไฟล์นี้ได้โดยตรง

##### 3.1.3 การวิเคราะห์โครงสร้างฐานข้อมูล (Database Schema Analysis)

เมื่อทราบตำแหน่งของไฟล์ฐานข้อมูลแล้ว ผู้จัดทำได้ทำการคัดลอกไฟล์ Gadgetbridge.db จากอุปกรณ์ทดสอบมายังเครื่องคอมพิวเตอร์ และใช้เครื่องมือ DB Browser for SQLite เพื่อเปิดและวิเคราะห์โครงสร้างภายในอย่างละเอียด จากการตรวจสอบพบว่าข้อมูลกิจกรรมจากอุปกรณ์ เสียวมี

สมาร์ทโฟน แบนด์เจ็ต จะถูกจัดเก็บในตารางชื่อ MI\_BAND\_ACTIVITY\_SAMPLE ซึ่งจากการวิเคราะห์ มีโครงสร้างและรายละเอียดของแต่ละคอลัมน์ที่สำคัญต่อโครงงานดังนี้

ชื่อคอลัมน์	ประเภทข้อมูล	รายละเอียดข้อมูลที่จัดเก็บ	หมายเหตุ
TIMESTAMP	INTEGER	เวลาที่ข้อมูลถูกบันทึกในรูปแบบ Unix Timestamp	Primary Key (ร่วม)
DEVICE_ID	INTEGER	รหัสเฉพาะของอุปกรณ์ Smart Band ที่ทำการวัด	Primary Key (ร่วม)
USER_ID	INTEGER	รหัสของผู้ใช้งานที่ผูกกับอุปกรณ์ในแอป Gadgetbridge	Foreign Key
HEART_RATE	INTEGER	อัตราการเต้นของหัวใจ	ข้อมูลหลักของโครงงาน

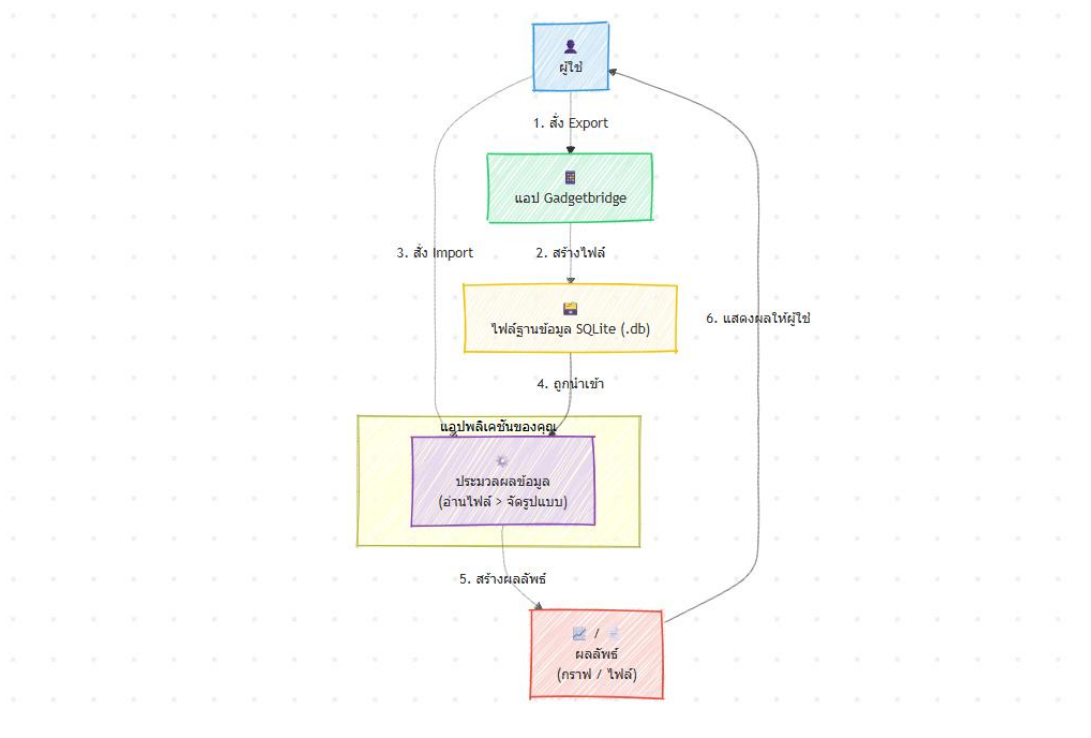
การวิเคราะห์ในขั้นตอนนี้ทำให้ทราบว่าตารางดังกล่าวใช้ Composite Primary Key ซึ่งคือการใช้คอลัมน์ TIMESTAMP และ DEVICE\_ID ร่วมกันเพื่อรับประกันความเป็นเอกลักษณ์ของแต่ละแถวข้อมูล และที่สำคัญคือทำให้ทราบชนิดและรูปแบบของข้อมูลที่ต้องใช้ในการพัฒนาแอปพลิเคชันต่อไป การทำความเข้าใจโครงสร้างและตำแหน่งของฐานข้อมูลนี้อย่างละเอียดเป็นพื้นฐานที่สำคัญที่สุดในการออกแบบ ส่วนการเข้าถึงข้อมูล (Data Access Layer) ของแอปพลิเคชันที่จะพัฒนาขึ้นในขั้นตอนถัดไป

### 3.2 การออกแบบสถาปัตยกรรมและพัฒนาระบบ (System Architecture and Development)

หลังจากที่ได้วิเคราะห์และทำความเข้าใจระบบเป้าหมายอย่างละเอียดในขั้นตอนก่อนหน้านี้แล้ว กระบวนการต่อไปคือการออกแบบสถาปัตยกรรมและลงมือพัฒนาแอปพลิเคชันต้นแบบให้บรรลุตามวัตถุประสงค์ที่กำหนดไว้

#### 3.2.1 การออกแบบสถาปัตยกรรมของระบบ

เพื่อให้การพัฒนามีโครงสร้างที่เป็นระเบียบ ง่ายต่อการบำรุงรักษาและต่อยอดในอนาคต ผู้จัดทำได้ออกแบบแอปพลิเคชันตาม สถาปัตยกรรมซอฟต์แวร์แบบ 3 ชั้น ที่ทำงานทั้งหมดบนอุปกรณ์ เพื่อรักษาความเป็นส่วนตัวสูงสุดของข้อมูลผู้ใช้ สถาปัตยกรรมนี้ช่วยแยกส่วนการทำงานออกจากกันอย่างชัดเจน



ภาพที่ 3.2 แผนผังสถาปัตยกรรมของระบบ

#### 3.2.1.1 ส่วนติดต่อผู้ใช้งาน

เป็นชั้นบนสุดที่ผู้ใช้ได้ตอบด้วยโดยตรง ถูกพัฒนาขึ้นด้วยเทคโนโลยี Jetpack Compose ซึ่งเป็นชุดเครื่องมือสมัยใหม่สำหรับสร้าง UI บนแอนดรอยด์ ชั้นนี้มีหน้าที่รับข้อมูลนำเข้าจากผู้ใช้ และแสดงผลข้อมูลที่ได้รับจากชั้นประมวลผลในรูปแบบของกราฟและตาราง

#### 3.2.1.2 ส่วนประมวลผลข้อมูล

เป็นชั้นกลางที่ทำหน้าที่เป็นสมองของแอปพลิเคชัน ประกอบด้วย ViewModel ซึ่งทำหน้าที่จัดการสถานะ ของ UI และเป็นตัวกลางในการสื่อสารระหว่าง UI Layer และ Data Access Layer ชั้นนี้รับผิดชอบการสร้างคำสั่ง SQL การกรองข้อมูลตามเงื่อนไข, และการแปลงข้อมูล ให้อยู่ในรูปแบบที่พร้อมสำหรับนำไปแสดงผลหรือส่งออก

#### 3.2.1.3 ส่วนการเข้าถึงข้อมูล

เป็นชั้นล่างสุดที่รับผิดชอบการโต้ตอบกับแหล่งข้อมูลภายนอกโดยตรง ในโครงงานนี้คือการเชื่อมต่อและอ่านข้อมูลจากไฟล์ฐานข้อมูล SQLite (Gadgetbridge.db) การออกแบบชั้นนี้แยกออกมาต่างหากช่วยให้หากในอนาคตมีการเปลี่ยนแหล่งข้อมูล ก็สามารถแก้ไขได้เฉพาะในชั้นนี้โดยไม่กระทบกับส่วนอื่นๆ ของแอปพลิเคชัน

### 3.2.2 การดำเนินงานพัฒนา (Implementation)

หลังจากที่ได้ออกแบบสถาปัตยกรรมของระบบแล้ว ผู้จัดทำได้ลงมือพัฒนาแอปพลิเคชันในแต่ละส่วนโดยใช้โปรแกรม Android Studio เป็นเครื่องมือหลัก โดยมีรายละเอียดการดำเนินงานในแต่ละชั้นสถาปัตยกรรมดังนี้

#### 3.2.2.1 การพัฒนาส่วนเข้าถึงข้อมูล (Data Access Layer)

ส่วนนี้เป็นส่วนที่มีความท้าทายทางเทคนิคมากที่สุด เนื่องจากต้องเข้าถึงไฟล์ฐานข้อมูลของแอปพลิเคชันอื่นภายใต้ข้อจำกัดด้านความปลอดภัยของระบบปฏิบัติการแอนดรอยด์ การพัฒนาเริ่มต้นด้วยการเขียนโค้ดเพื่อร้องขอสิทธิ์ ในการเข้าถึงหน่วยความจำภายนอกจากผู้ใช้ เมื่อได้รับสิทธิ์แล้ว จึงทำการสร้างการเชื่อมต่อไปยังไฟล์ฐานข้อมูล Gadgetbridge.db ตามที่อยู่ ที่ได้วิเคราะห์ไว้ในหัวข้อ 3.1.2 โดยเป็นการเปิดการเชื่อมต่อในโหมดอ่านอย่างเดียว เพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้นกับข้อมูลต้นฉบับ จากนั้นจึงสร้างฟังก์ชันสำหรับรับคำสั่ง SQL และคืนค่าผลลัพธ์ในรูปแบบที่ส่วนประมวลผลสามารถนำไปใช้ได้

#### 3.2.2.2 การพัฒนาส่วนประมวลผลข้อมูล (Data Processing Layer)

ส่วนนี้ถูกพัฒนาขึ้นโดยใช้ ViewModel เป็นศูนย์กลางในการจัดการตรรกะการทำงานทั้งหมด เมื่อ ViewModel ได้รับคำสั่งจากส่วนติดต่อผู้ใช้งาน (เช่น ผู้ใช้เลือกช่วงวันที่) จะทำหน้าที่สร้างคำสั่ง SQL SELECT แบบไดนามิก โดยมีเงื่อนไข WHERE เพื่อกรองข้อมูลตามช่วงของ TIMESTAMP ที่ผู้ใช้กำหนด จากนั้นจะส่งคำสั่งนี้ไปยัง Data Access Layer เมื่อได้รับผลลัพธ์กลับมา จะทำการวนลูปผ่านข้อมูลดิบที่ได้ และแปลงให้อยู่ในรูปแบบของคลาสข้อมูล ในภาษา Kotlin เพื่อให้ง่ายต่อการจัดการและส่งต่อไปยังส่วนอื่นๆ

#### 3.2.2.3 การพัฒนาส่วนติดต่อผู้ใช้งาน (User Interface Layer)

ส่วนติดต่อผู้ใช้งานทั้งหมดถูกสร้างขึ้นด้วยเทคโนโลยี Jetpack Compose ซึ่งเป็นชุดเครื่องมือสมัยใหม่สำหรับสร้าง UI บนแอนดรอยด์ ในหน้าจอหลักของการแสดงผลกราฟ UI จะทำการสังเกตการณ์ ข้อมูลจาก ViewModel เมื่อมีข้อมูลใหม่เข้ามา จะทำการแปลงข้อมูลนั้นให้อยู่ในรูปแบบอ็อบเจกต์ Entry ที่ไลบรารี MPAndroidChart ต้องการ จากนั้นจึงนำไปสร้างเป็นกราฟเส้น พร้อมทั้งปรับแต่งการแสดงผลของแกน X และ Y ให้แสดงหน่วยของเวลาและอัตราการเต้นของหัวใจตามที่ได้ออกแบบไว้

#### 3.2.2.4 การพัฒนาฟังก์ชันส่งออกข้อมูล (Data Export Functionality)

ฟังก์ชันนี้ถูกพัฒนาขึ้นเพื่อตอบสนองต่อวัตถุประสงค์ของโครงการโดยตรง โดยกระบวนการจะเริ่มต้นเมื่อผู้ใช้งานผ่าน UI ส่วนประมวลผลข้อมูล จะรับข้อมูลชุดปัจจุบันที่แสดงผลอยู่มาทำการวนลูป และสร้างเป็นเนื้อหาข้อความ ที่มีโครงสร้างตามรูปแบบของไฟล์ .csv หรือ .xls จากนั้นจะใช้ MediaStore API ของแอนดรอยด์ในการสร้างไฟล์และบันทึกลงในไดเรกทอรี Downloads ของอุปกรณ์ พร้อมทั้งแสดงการแจ้งเตือน ให้ผู้ใช้ทราบเมื่อการบันทึกเสร็จสิ้น

## บรรณานุกรม

- กรมกิจการผู้สูงอายุ <https://www.dop.go.th/th/know/15/548>
- แหล่งข้อมูลและ Open source <https://gadgetbridge.org/>
- Freeyourgadget <https://codeberg.org/Freeyourgadget/Gadgetbridge/wiki>
- Sh., M., et al. (2023). Gadgetbridge Database Exporter. GitHub Repository. Retrieved from <https://github.com/melvin-sh/gadgetbridge-database-exporter>
- Vo, T., Nguyen, H., and Le, V. (2021). "A Systematic Review on the Security and Privacy of Internet-of-Things based Wearable Devices." 13th International Conference on Knowledge and Systems Engineering (KSE), 1-6.
- Allen, J. (2007). "Photoplethysmography and its application in clinical physiological measurement." Physiological Measurement, 28(3), R1–R39.
- Dounis, A. P., Karanasiou, G. S., & Koutsaftis, C. (2023). "A comprehensive review of smartwatch-based methods for arrhythmia detection." Sensors, 23(17), 7481.
- Bluetooth SIG. (n.d.). Bluetooth Low Energy. Retrieved from <https://www.bluetooth.com/learn-about-bluetooth/bluetooth-technology/le-audio/>
- This is ANT. (n.d.). What is ANT?. Retrieved from <https://www.thisisant.com/consumer/ant-101/what-is-ant>
- Google. (n.d.). Kotlin and Android. Android Developers. Retrieved from <https://developer.android.com/kotlin>
- Google. (n.d.). Layouts in Android. Android Developers. Retrieved from <https://developer.android.com/guide/topics/ui/declaring-layout>
- Hoare, T. (2009). "Null References: The Billion Dollar Mistake. QCon London. Retrieved from <https://www.infoq.com/presentations/Null-References-The-Billion-Dollar-Mistake-Tony-Hoare/>