



ระบบสมarthomeสั่งงานผ่านแอปพลิเคชัน

Smart home system controlled via application

นาย วิทยา คนที

65122250008

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
สาขาวิชาการคอมพิวเตอร์และนวัตกรรมข้อมูล คณะวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยราชภัฏสวนสุนันทา

ปีการศึกษา 2568

หัวข้อโครงการ : ระบบสมarthomeสั่งงานผ่านแอปพลิเคชัน



ระบบสมarthomeสั่งงานผ่านแอปพลิเคชัน

Smart home system controlled via application

นาย วิทยา คนที

65122250008

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
สาขาวิชาการคอมพิวเตอร์และนวัตกรรมข้อมูล คณะวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยราชภัฏสวนสุนันทา

ปีการศึกษา 2568

หัวข้อโครงการ : ระบบสมarthomeสั่งงานผ่านแอปพลิเคชัน

ชื่อนักศึกษา : ชื่อ นาย วิทยา คนที รหัสนักศึกษา

สาขาวิชา : วิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏสวนสุนันหานุเมตติให้รับโครงการวิจัยนี้
เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาหลัก

กรรมการ

(ผศ. นิศานาณ เตชะเพชรไพบูลย์)

()

ชื่อโครงการ : ระบบสมarthomeสั่งงานผ่านแอปพลิเคชัน

ชื่อผู้ดูแล : ชื่อ นาย วิทยา คนที่ รหัสนักศึกษา 65122250008

สาขาวิชา : วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา : ผศ. นิศานาถ เตชะเพชรไพบูลย์

ปีการศึกษา : 2568

บทคัดย่อ

งานวิจัยนี้นำเสนอระบบ Smart Home ที่สามารถควบคุมและตรวจสอบอุปกรณ์ภายในบ้านผ่านแอปพลิเคชันบนมือถือแบบเรียลไทม์ โดยใช้ ESP32 เป็นไมโครคอนโทรลเลอร์หลักในการเชื่อมต่อกับเซนเซอร์วัดอุณหภูมิ ความชื้น แสงสว่าง และการเคลื่อนไหว ระบบยังควบคุม LED, Buzzer และอุปกรณ์ไฟฟ้าอื่น ๆ ผ่าน Firebase Realtime Database เพื่อให้สามารถเปิด-ปิดอุปกรณ์ได้ทั้งแบบอัตโนมัติและด้วยการสั่งงานผ่านแอปพลิเคชัน

นอกจากนี้ ระบบรองรับฟังก์ชัน แจ้งเตือนเหตุการณ์สำคัญ เช่น อุณหภูมิหรือความชื้นเกินค่าเกณฑ์และการตรวจพบการเคลื่อนไหว พร้อมจัดเก็บ ประวัติการแจ้งเตือน เพื่อให้ผู้ใช้งานตรวจสอบย้อนหลังได้ ระบบยังรวมฟีเจอร์ สั่งงานด้วยเสียง เพื่อความสะดวกและเพิ่มความทันสมัยในการควบคุมอุปกรณ์

ผลลัพธ์ที่ได้แสดงให้เห็นว่าสามารถ ควบคุมและตรวจสอบอุปกรณ์ได้อย่างมีประสิทธิภาพและปลอดภัย พร้อมทั้งลดความเสี่ยงจากสถานการณ์อันตรายภายในบ้านได้อย่างมีนัยสำคัญ

Title : Smart home system controlled via application

Author : Wittaya khonthee ID 65122250008

Curriculum : Computer Science

Suan Sunandha Rajabhat University

Advisor : Nisanat Techaphetphaiboon

Year : 2025

Abstract

This research presents a Smart Home system that enables real-time monitoring and control of household devices via a mobile application. The system utilizes ESP32 as the main microcontroller, connected to sensors measuring temperature, humidity, light, and motion. It also controls LEDs, buzzers, and other electrical devices through Firebase Realtime Database, allowing both automated and manual device operation via the app.

Additionally, the system supports event notifications, such as when temperature or humidity exceeds thresholds or when motion is detected, and it stores a notification history for users to review past events. Voice control functionality is integrated to enhance convenience and modernize device management.

The results demonstrate that the system can effectively and safely monitor and control home devices, significantly reducing potential hazards within the household environment.

กิตติกรรมประกาศ

โครงงาน ระบบสมาร์ทโฮมสั่งงานผ่านแอปพลิเคชัน โครงงานนี้จะสำเร็จลุล่วงได้ด้วยความกรุณาให้ความช่วยเหลือ รับฟัง ให้คำปรึกษา และตรวจสอบแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่อย่างยิ่งจาก

ผศ.นิสานาถ เตชะเพชรไพบูลย์ อาจารย์ที่ปรึกษารายงาน ผู้เขียนขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณคณาจารย์สาขาวิชาการคอมพิวเตอร์ทุกท่าน ที่ร่วมกันถ่ายทอดและสร้างองค์ความรู้ด้านเทคโนโลยีและระบบสมาร์ทโฮม ซึ่งเป็นพื้นฐานสำคัญสำหรับการทำโครงงานและการพัฒนาระบบควบคุมผ่านแอปพลิเคชัน

สุดท้ายนี้ ขอขอบคุณรุ่นพี่วิชาการคอมพิวเตอร์และเพื่อนทุกคน ที่ได้ช่วยส่งเสริม สนับสนุน กระตุ้นเตือน และเป็นกำลังใจให้ผู้เขียนดำเนินโครงงานสมาร์ทโฮมและจัดทำรายงานวิจัยตลอดมา

คณะผู้จัดทำ
นายวิทยา คนที
9 ตุลาคม 2568

คำนำ

เทคโนโลยีสมาร์ทโฮมเป็นหนึ่งในแนวโน้มสำคัญของโลกยุคปัจจุบันที่มุ่งเน้นการเพิ่มความสะดวกสบาย และความปลอดภัยภายในบ้าน การพัฒนาระบบสมาร์ทโฮมที่สามารถตรวจสอบสภาพแวดล้อม เช่น อุณหภูมิ ความชื้น แสงสว่าง และการเคลื่อนไหว พร้อมกับควบคุมอุปกรณ์ไฟฟ้าต่าง ๆ ผ่านแอปพลิเคชันมือถือ จะช่วยให้ผู้ใช้งานสามารถบริหารจัดการบ้านได้อย่างมีประสิทธิภาพและปลอดภัยยิ่งขึ้น

โครงการนี้จึงเน้นการออกแบบและพัฒนาระบบสมาร์ทโฮมโดยใช้ ESP32 ร่วมกับเซนเซอร์หลายประเภท และจัดการข้อมูลผ่าน Firebase Realtime Database ที่ยังมีฟังก์ชันแจ้งเตือนเหตุการณ์สำคัญ การเก็บประวัติการแจ้งเตือน เพื่อเพิ่มความสะดวกสบายและความทันสมัยให้กับผู้ใช้งาน

คณะผู้จัดทำ
นายวิทยา คนที
9 ตุลาคม 2568

สารบัญ

บทคัดย่อ	๑
Abstract	๒
กิตติกรรมประกาศ	๓
คำนำ	๔
สารบัญ	๕
บทที่ 1 บทนำ	๑
1.1 ความเป็นมาและความสำคัญของปัญหา	๑
1.2 ที่มาและแรงบันดาลใจ	๒
1.3 ขอบเขตของโครงการ	๒
1.4 แผนการดำเนินโครงการ	๓
1.5 วัตถุประสงค์ของโครงการ	๓
บทที่ 2	๔
2.1 เอกสารและเทคโนโลยีที่เกี่ยวข้อง	๔
2.2 งานวิจัยที่เกี่ยวข้อง	๑๐
บทที่ 3	๑๑
3.1 การวิเคราะห์ความต้องการของระบบ	๑๑
3.2 การออกแบบระบบ	๑๓
3.3 การพัฒนาโปรแกรม	๒๐
บทที่ 4 ผลการดำเนินงาน	๒๐
4.1 แผนการทดสอบ (Testing Plan)	๒๐
4.2 กรณีตัวอย่างการทดสอบ (Test Case)	๒๒

4.3 พิมพ์ชั้นการทำงานของเว็บไซต์	23
4.4 การทดสอบการยอมรับของผู้ใช้ระบบ (Acceptance Testing)	26
บทที่ 5 สรุปผล อภิปรายผล และข้อเสนอแนะ	27
5.1 สรุปผลการดำเนินงาน	27
5.2 อภิปรายผล	27
5.3 ข้อเสนอแนะในการพัฒนาระบบ	28
5.4 สรุปภาพรวมของโครงการ	28
บรรณานุกรม	29
ภาคผนวก ก	31
ภาคผนวก ข	39
ประวัติผู้จัดทำโครงการ	41

สารบัญรูปภาพ

ภาพ	หน้า
ภาพตารางที่ 1.1 ตารางแผนการดำเนินโครงการ	3
ภาพประกอบที่ 2.1 บอร์ดไมโครคอนโทรลเลอร์ ESP32	4
ภาพประกอบที่ 2.2 วัดอุณหภูมิและความชื้น	5
ภาพประกอบที่ 2.3 เซนเซอร์ตรวจจับแสง	5
ภาพประกอบที่ 2.4 เซนเซอร์ตรวจจับการ เคลื่อนไหว	6

ภาพประกอบที่ 2.5 อุปกรณ์แสดงผลและแจ้งเตือน ^(LED และ Buzzer)	6
ภาพประกอบที่ 2.6 ฐานข้อมูลแบบเรียลไทม์ ^{Firebase Realtime Database}	7
ภาพประกอบที่ 2.7 React Native	8
ภาพประกอบที่ 2.8 JavaScript	8
ภาพประกอบที่ 2.9 Expo	9
ภาพประกอบที่ 3.1 FISHBONDE DIAGRAM	11
ภาพประกอบที่ 3.2 Work Flow Diagram	13
ภาพประกอบที่ 3.3 Use Case Diagram	14
ภาพประกอบที่ 3.4 Class Diagram	15
ภาพประกอบที่ 3.5 Activity Diagram	16
ภาพประกอบที่ 3.6 Sequence Diagram	17
ภาพประกอบที่ 3.7 หน้าหลักของแอปพลิเคชัน	18
ภาพประกอบที่ 3.8 ตั้งค่าอุปกรณ์	19
ภาพประกอบที่ 3.9 หน้าการแจ้งเตือน	19
ภาพประกอบที่ 4.1 หน้าแรกของระบบ	23
ภาพประกอบที่ 4.2 หน้าตั้งค่า	24
ภาพประกอบที่ 4.3 หน้าการแจ้งเตือน	25

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบัน เทคโนโลยี Internet of Things (IoT) ได้เข้ามายึด主导ในชีวิตประจำวันมากขึ้น โดยเฉพาะในด้าน “สมาร์ทโฮม (Smart Home)” ซึ่งช่วยให้ผู้ใช้งานสามารถควบคุมอุปกรณ์ไฟฟ้าภายในบ้านได้อย่างสะดวกสบายผ่านอินเทอร์เน็ต ไม่ว่าจะอยู่ที่ใดก็ตาม การใช้ระบบสมาร์ทโฮมช่วยเพิ่มความปลอดภัย ประหยัดพลังงาน และยกระดับคุณภาพชีวิตของผู้ใช้ให้สูงมากยิ่งขึ้น

1.2 ที่มาและแรงบันดาลใจ

ระบบสมาร์ทโฮมที่มีขึ้นในท้องตลาดส่วนใหญ่มีราคาสูงและติดตั้งซับซ้อน ผู้จัดทำจึงมีแนวคิดพัฒนา “ระบบสมาร์ทโฮมต้นแบบ” ที่ควบคุมผ่าน แอปพลิเคชันโทรศัพท์มือถือ โดยใช้บอร์ด ESP32 เป็นตัวกลางในการเชื่อมต่อ กับอุปกรณ์ไฟฟ้าภายในบ้าน เช่น หลอดไฟ และเซนเซอร์ โดยแอปพลิเคชันถูกออกแบบให้ใช้งานง่าย สามารถสั่งเปิด-ปิดอุปกรณ์และดูสถานะได้แบบเรียลไทม์

1.3 ขอบเขตของโครงการ

1.3.1 ขอบเขตด้านเนื้อหา

โครงการนี้มุ่งเน้นการพัฒนาระบบสมาร์ทโฮมโดยใช้ไมโครคอนโทรลเลอร์ ESP32 เป็นอุปกรณ์หลักในการควบคุมและตรวจสอบอุปกรณ์ภายในบ้าน โดยสามารถควบคุมอุปกรณ์ไฟฟ้าผ่านแอปพลิเคชันบนสมาร์ทโฟน รวมถึงตรวจวัดอุณหภูมิ ความชื้น และการตรวจจับการเคลื่อนไหวภายในบ้าน ข้อมูลที่ได้จากเซนเซอร์จะถูกส่งและจัดเก็บแบบเรียลไทม์ผ่าน Firebase เพื่อให้ผู้ใช้งานสามารถติดตามสถานะและควบคุมระบบได้จากระยะไกลผ่านเครือข่ายอินเทอร์เน็ต

1.3.2 ซอฟต์แวร์ที่ใช้

- 1.3.2.1 Arduino IDE — สำหรับเขียนและอัปโหลดโค้ดลงในบอร์ด ESP32
- 1.3.2.2 Firebase Console — สำหรับสร้างฐานข้อมูลและจัดการข้อมูลแบบเรียลไทม์
- 1.3.2.3 Visual Studio Code / ExpoGo — สำหรับพัฒนาแอปพลิเคชันบนโทรศัพท์มือถือ
- 1.3.2.4 Visual Studio Code / Android Studio — สำหรับพัฒนาแอปพลิเคชันบนโทรศัพท์มือถือ

1.3.3 ภาษาที่ใช้ในการพัฒนา

- 1.3.3.1 ภาษา C++
- 1.3.3.2 ภาษา JavaScript
- 1.3.3.3 ภาษา HTML / CSS

1.3.4 ฮาร์ดแวร์ที่ใช้

- 1.3.4.1 ESP32 Board — บอร์ดหลักสำหรับประมวลผลและเชื่อมต่ออินเทอร์เน็ต
- 1.3.4.2 DHT11 Sensor — สำหรับวัดอุณหภูมิและความชื้น
- 1.3.4.3 LDR Sensor — สำหรับวัดค่าความสว่างของแสง
- 1.3.4.4 PIR Motion Sensor — สำหรับตรวจจับการเคลื่อนไหว
- 1.3.4.5 LED และ RGB LED — สำหรับแสดงสถานะของระบบ
- 1.3.4.6 Buzzer — สำหรับแจ้งเตือนเสียง
- 1.3.4.7 สาย usb to micro usb
- 1.3.4.8 โทรศัพท์มือถือ (Android) — สำหรับควบคุมและแสดงผลระบบ

1.4 แผนการดำเนินโครงการ

การดำเนินงาน	ระยะเวลา(เดือน)							
	ม.ค	ก.พ	มี.ค	เม.ย	ก.ค	ส.ค	ก.ย	ต.ค
ศึกษาดูแลข้อมูลที่เก็บรวบรวมบนสมาร์ทโฟน, ESP32, Firebase, และ IoT								
วิเคราะห์และออกแบบระบบ (การเชื่อมต่อ, และการท่า่งงานของแอปพลิเคชัน)								
ออกแบบและสร้างฐานข้อมูลบน Firebase								
เขียนโปรแกรมบนบอร์ด ESP32 สำหรับควบคุมอุปกรณ์และรับdataจากเซ็นเซอร์								
พัฒนาแอปพลิเคชันบนโทรศัพท์มือถือ (UI/UX และเชื่อมต่อ กับ Firebase)								
ทดสอบการทำงานของระบบ (การเชื่อมต่อ Wi-Fi, การส่งค่าผ่าน Firebase, การควบคุมอุปกรณ์)								
ปรับปรุงและแก้ไขข้อผิดพลาดจากการทดสอบ								
จัดทำคู่มือการใช้งานระบบและสรุปผลการทดสอบ								
จัดทำรายงานฉบับสมบูรณ์และเตรียมหน้าเสนอผลงาน								

ภาพตารางที่ 1.1 ตารางแผนการดำเนินโครงการ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

โครงการระบบสมาร์ทโฟนโดยใช้ ESP32 ควบคุมอุปกรณ์ผ่านแอปพลิเคชัน และใช้ Firebase ในการจัดการข้อมูล ช่วยเพิ่มความสะดวกสบายในการควบคุมอุปกรณ์ไฟฟ้าภายในบ้านจากระยะไกลผ่านสมาร์ทโฟน ผู้ใช้งานสามารถตรวจสอบสภาพแวดล้อมภายในบ้านได้แบบเรียลไทม์ ไม่ว่าจะเป็นอุณหภูมิ ความชื้น และการตรวจจับการเคลื่อนไหว ซึ่งช่วยเพิ่มระดับความปลอดภัยให้กับที่อยู่อาศัย อีกทั้งโครงการนี้ยังเป็นต้นแบบในการประยุกต์ใช้เทคโนโลยี Internet of Things (IoT) กับระบบบ้านอัจฉริยะ ผู้จัดทำโครงการได้รับความรู้และทักษะด้านการพัฒนาไมโครคอนโทรลเลอร์ ESP32 การเชื่อมต่อฐานข้อมูล Firebase และการพัฒนาแอปพลิเคชัน ซึ่งสามารถนำไปต่อยอดพัฒนาและประยุกต์ใช้งานในอนาคตได้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การจัดทำโครงงานระบบสมาร์ทโฮมโดยใช้ ESP32 ควบคุมอุปกรณ์ผ่านแอปพลิเคชัน และใช้ Firebase ในการจัดการข้อมูล จำเป็นต้องศึกษาเอกสาร แนวคิด และงานวิจัยที่เกี่ยวข้อง เพื่อใช้เป็นแนวทางในการออกแบบและพัฒนาระบบให้สามารถทำงานได้อย่างมีประสิทธิภาพ ซึ่งสามารถแบ่งเนื้อหาที่เกี่ยวข้องได้ดังต่อไปนี้

2.1 เอกสารและเทคโนโลยีที่เกี่ยวข้อง

เอกสารในส่วนนี้อธิบายถึงคุณสมบัติและหลักการทำงานของอุปกรณ์และเทคโนโลยีที่ใช้ในการพัฒนาระบบสมาร์ทโฮม

2.1.1 บอร์ดไมโครคอนโทรลเลอร์ ESP32



ภาพประกอบที่ 2.1 บอร์ดไมโครคอนโทรลเลอร์ ESP32
ที่มา <https://www.cybertice.com/article/226/>

ESP32 โดยเด่นด้วยการมีโมดูล Wi-Fi (802.11 b/g/n) และ Bluetooth (BLE) ในตัว ทำให้เป็นที่นิยมอย่างยิ่งในงาน Internet of Things (IoT)

- สถาปัตยกรรม: ใช้หน่วยประมวลผล Dual-core ที่มีความเร็วสูงสุดถึง 240 MHz
- หน่วยความจำ: ขนาด 4MB (หรือมากกว่า) สำหรับเก็บโปรแกรม และ SRAM สำหรับการประมวลผล
- การเชื่อมต่อ: รองรับ Wi-Fi (2.4GHz) และ Bluetooth ทำให้สามารถเชื่อมต่อกับแอปพลิเคชัน มือถือได้
- แรงดันไฟฟ้า: ใช้งานที่แรงดันไฟ 3.3V

2.1.2 เซนเซอร์วัดอุณหภูมิและความชื้น DHT11



ภาพประกอบที่ 2.2 วัดอุณหภูมิและความชื้น
ที่มา <https://www.cybertice.com/article/632/>

DHT11 เป็นเซนเซอร์ดิจิตอลที่ออกแบบมาเพื่อวัดค่า อุณหภูมิ และ ความชื้นสัมพัทธ์ ในตัวเดียวgan

- หลักการทำงาน: สำหรับวัดความชื้น และ Thermistor (NTC) สำหรับวัดอุณหภูมิ
- ช่วงการวัด: อุณหภูมิ 0–50°C และความชื้น 20–90% RH
- การสื่อสาร: สื่อสารข้อมูลแบบดิจิตัลผ่านขาเดียว ซึ่งง่ายต่อการเชื่อมต่อกับไมโครคอนโทรลเลอร์

2.1.3 เซนเซอร์ตรวจจับแสง (LDR)



ภาพประกอบที่ 2.3 เซนเซอร์ตรวจจับแสง
ที่มา <http://suwitkiravittaya.eng.chula.ac.th/>

เป็นตัวต้านทานชนิดหนึ่งที่มีค่าความต้านทานเปลี่ยนแปลงตามความเข้มของแสงที่ตกกระทบ

- หลักการทำงาน: เมื่อได้รับแสง ค่าความต้านทานจะลดลง และในที่มีดี ค่าความต้านทานจะเพิ่มขึ้น
- การประยุกต์ใช้: ใช้ร่วมกับวงจร Voltage Divider เพื่อแปลงค่าความต้านทานเป็นสัญญาณแรงดันไฟฟ้าที่สามารถอ่านได้ด้วยขา Analog-to-Digital Converter (ADC) ของ ESP32 เพื่อนำไปควบคุมระบบเปิด-ปิดไฟอัตโนมัติ

2.1.4 เซนเซอร์ตรวจจับการเคลื่อนไหว (PIR Sensor)



ภาพประกอบที่ 2.4 เซนเซอร์ตรวจจับการเคลื่อนไหว
ที่มา <https://www.cybertice.com/article/43/>

ทำหน้าที่ตรวจจับการเปลี่ยนแปลงของพลังงานความร้อนอินฟราเรดที่แผ่ออกมายาวตัวที่มีชีวิต เช่น ร่างกายมนุษย์

- หลักการทำงาน: เซนเซอร์จะอ่านค่าอินฟราเรดจากสภาพแวดล้อม และเมื่อตรวจพบการเปลี่ยนแปลงอย่างรวดเร็ว (เช่น มีคนเดินผ่าน) จะส่งสัญญาณเอาต์พุตเป็นสถานะ HIGH
- การประยุกต์ใช้: เหมาะสมสำหรับงานรักษาความปลอดภัย (Security) และการควบคุมแสงไฟอัตโนมัติ

2.1.5 อุปกรณ์แสดงผลและแจ้งเตือน (LED และ Buzzer)



ภาพประกอบที่ 2.5 อุปกรณ์แสดงผลและแจ้งเตือน (LED และ Buzzer)
ที่มา <https://www.cybertice.com/article/577>

ที่มา <https://esp32io.com/tutorials/esp32-button-piezo-buzzer>

- LED: ใช้สำหรับแสดงสถานะ การทำงานของระบบ เช่น สถานะเตือนภัยเบื้องต้น
- Buzzer: ทำหน้าที่ ส่งเสียงเตือน เมื่อเกิดเหตุการณ์สำคัญที่ต้องแจ้งเตือนผู้ใช้งานทันที เช่น อุณหภูมิสูงเกินค่าเกณฑ์หรือมีการตรวจพบผู้บุกรุก

2.1.6 ฐานข้อมูลแบบเรียลไทม์ Firebase Realtime Database



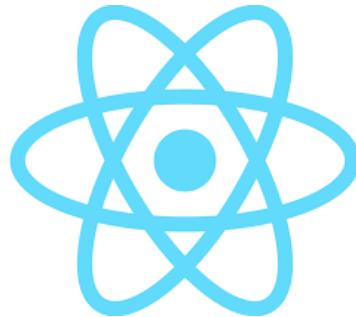
ภาพประกอบที่ 2.6 ฐานข้อมูลแบบเรียลไทม์ Firebase Realtime Database

ที่มา <https://medium.com/firebase-thailand>

Firebase เป็นแพลตฟอร์มบริการของ Google สำหรับการพัฒนาแอปพลิเคชันมือถือและเว็บไซต์ โดยส่วนของ Realtime Database เป็นฐานข้อมูลแบบ NoSQL ที่เก็บข้อมูลในรูปแบบ JSON และมีการซิงโครไนซ์ข้อมูลแบบเรียลไทม์

- คุณสมบัติหลัก: ข้อมูลจะถูกอัปเดตไปยังอุปกรณ์ที่เชื่อมต่อทั้งหมดทันทีที่ข้อมูลมีการเปลี่ยนแปลง ทำให้เหมาะสมอย่างยิ่งสำหรับการเป็นสื่อกลางในการสื่อสารระหว่าง ESP32 และ Mobile Application ในระบบ IoT
- การเข้าถึง: สามารถเข้าถึงและจัดการข้อมูลผ่าน API ซึ่ง ESP32 สามารถเชื่อมต่อเพื่ออ่านและเขียนข้อมูลได้อย่างง่ายดาย

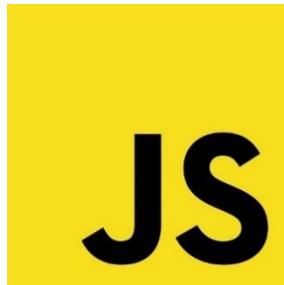
2.1.7 แอปพลิเคชันควบคุม (Mobile Application)



ภาพประกอบที่ 2.7 React Native
ที่มา <https://reactnative.dev/>

เป็นส่วนต่อประสานกับผู้ใช้ ที่พัฒนาขึ้นเพื่อให้ผู้ใช้งานสามารถ สั่งการ และ ตรวจสอบ สถานะของ อุปกรณ์スマาร์ทโฟนได้จากระยะไกล โดยมีรายละเอียดของภาษาและเครื่องมือที่ใช้ดังนี้
ภาษาและเฟรมเวิร์กหลัก:

- React Native: เป็นเฟรมเวิร์กโอเพนซอร์ซที่พัฒนาโดย Meta (Facebook) ใช้ภาษา



ภาพประกอบที่ 2.8 JavaScript
ที่มา <https://www.java.com/en/>

- JavaScript (หรือ TypeScript) ในการพัฒนาแอปพลิเคชันมือถือ โดยสามารถเขียนโค้ดชุดเดียว เพื่อสร้างแอปพลิเคชันที่รันได้ทั้งบนระบบปฏิบัติการ Android และ iOS
- JavaScript: เป็นภาษาโปรแกรมมิ่งหลักที่ใช้ในการเขียนล็อกิคและส่วนต่อประสานผู้ใช้ของแอป พลิเคชัน

expo

ภาพประกอบที่ 2.9 Expo
ที่มา <https://expo.dev/go>

- เครื่องมือช่วยพัฒนา (Expo):
 - Expo: เป็นชุดเครื่องมือและเฟรมเวิร์กที่สร้างอยู่บน React Native อีกชั้นหนึ่ง ช่วยให้การพัฒนาแอปพลิเคชันเร็วขึ้นและง่ายขึ้น โดยเฉพาะการจัดการกับไฟล์ และการเข้าถึง API ของอุปกรณ์ต่างๆ โดยไม่ต้องติดตั้งเครื่องมือพัฒนา ที่ซับซ้อน
 - Expo Go: เป็นแอปพลิเคชันสำหรับนักพัฒนาที่ติดตั้งอยู่บนมือถือจริง (Android/iOS) ใช้สำหรับการทดสอบแบบเรียลไทม์ (Live Preview) โดยนักพัฒนาสามารถสแกน QR Code และดูผลลัพธ์ของการเปลี่ยนแปลงโค๊ดได้ทันทีโดยไม่ต้องทำการ Build แอปพลิเคชันใหม่ทุกรอบ ทำให้การแก้ไขและพัฒนาทำได้รวดเร็วมาก

2.2 งานวิจัยที่เกี่ยวข้อง

Koushal, A., Gupta, R., et al. (2022). Home Automation System Using ESP32 and Firebase. In *Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC)* (pp. 37-42). IEEE.

- ลิงก์: <https://www.semanticscholar.org/paper/Home-Automation-System-Using-ESP32-and-Firebase-Koushal-Gupta/8ed264cd9790d67fdf08f205e56b222ba6e5e79e>

Mulla, S. S., Pawar, S. V., et al. (2020). IoT Based Smart Home Automation System Using ESP32. *International Journal of Engineering Research & Technology (IJERT)*, 9(09).

ลิงก์:

<https://penerbit.uthm.edu.my/periodicals/index.php/eeee/article/download/18942/6601/124979>

Prakash, S. S., Usha, R., et al. (2025). Smart Home and Security Systems: An IoT-Based Approach Utilizing ESP 32 and Multi-sensor Integration. *E3S Web of Conferences*, 616, 02004.

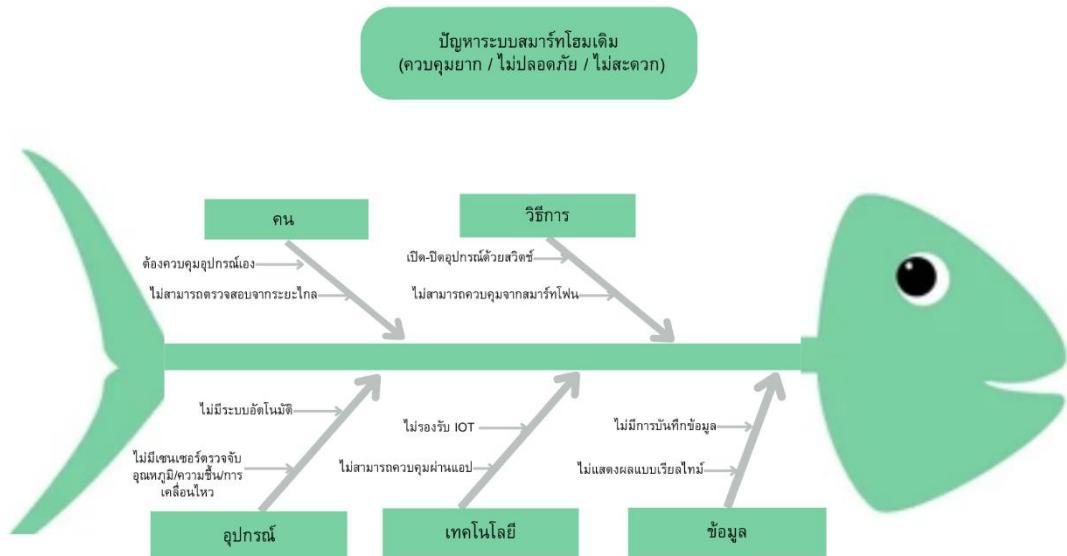
- ลิงก์: https://www.e3s-conferences.org/articles/e3sconf/abs/2025/16/e3sconf_icregcsd2025_02004/e3sconf_icregcsd2025_02004.html

บทที่ 3

วิธีการดำเนินงานและการพัฒนาซอฟต์แวร์

3.1 การวิเคราะห์ความต้องการของระบบ

3.1.1 ปัญหาของระบบงานเดิม



ภาพประกอบที่ 3.1 FISHBONDE DIAGRAM

ระบบงานเดิมของที่อยู่อาศัยทั่วไปยังคงอาศัยการควบคุมอุปกรณ์ไฟฟ้าด้วยวิธีแบบดั้งเดิม เช่น การเปิด-ปิดสวิตช์ด้วยตนเอง ทำให้ขาดความสะดวกสบายในการใช้งาน ผู้ใช้งานไม่สามารถควบคุมอุปกรณ์ไฟฟ้าจากระยะไกลได้ อีกทั้งยังไม่สามารถตรวจสอบสภาพแวดล้อมภายในบ้าน เช่น อุณหภูมิ ความชื้น และการเคลื่อนไหว ได้แบบเรียลไทม์ ส่งผลให้ขาดข้อมูลในการตัดสินใจและลดระดับความปลอดภัยของที่อยู่อาศัย นอกจากนี้ ระบบงานเดิมยังไม่มีการจัดเก็บข้อมูลอย่างเป็นระบบ และไม่สามารถแจ้งเตือนผู้ใช้งานเมื่อเกิดเหตุการณ์ผิดปกติได้

3.1.2 แนวทางการแก้ปัญหา

3.1.2.1 การใช้เทคโนโลยี IoT นำเทคโนโลยี Internet of Things (IoT) มาประยุกต์ใช้เพื่อเชื่อมต่ออุปกรณ์ภายในบ้านและควบคุมการทำงานผ่านอินเทอร์เน็ต

3.1.2.2 การใช้ไมโครคอนโทรลเลอร์ ESP32 ใช้ ESP32 เป็นอุปกรณ์หลักในการควบคุมและประมวลผลการทำงานของระบบ

3.1.2.3 การติดตั้งเซนเซอร์ตรวจวัดสภาพแวดล้อมติดตั้งเซนเซอร์วัดอุณหภูมิ ความชื้น และเซนเซอร์ตรวจจับการเคลื่อนไหว เพื่อใช้ในการตรวจสอบสภาพแวดล้อมและความปลอดภัย

3.1.2.4 การจัดเก็บข้อมูลด้วย Firebase ใช้ Firebase Realtime Database ในการจัดเก็บและรับ–ส่งข้อมูลของระบบแบบเรียลไทม์

3.1.2.5 การพัฒนาแอปพลิเคชันควบคุมระบบพัฒนาแอปพลิเคชันบนสมาร์ทโฟนสำหรับควบคุมอุปกรณ์และแสดงผลข้อมูลจากระบบ

3.1.2.6 การออกแบบระบบให้ขยายได้ออกแบบระบบให้มีความยืดหยุ่น สามารถพัฒนาและเพิ่มฟังก์ชันในอนาคตได้

3.1.3 ความต้องการของระบบใหม่

3.1.3.1 ความต้องการด้านการควบคุมอุปกรณ์ระบบต้องสามารถควบคุมผ่านแอปพลิเคชันบนสมาร์ทโฟนได้จากระยะไกล

3.1.3.2 ความต้องการด้านการตรวจสอบสภาพแวดล้อมระบบต้องสามารถตรวจวัดและแสดงผลอุณหภูมิและความชื้นภายในบ้านได้แบบเรียลไทม์

3.1.3.3 ความต้องการด้านความปลอดภัยระบบต้องสามารถตรวจจับการเคลื่อนไหวภายในบ้านและแสดงสถานะผ่านแอปพลิเคชัน

3.1.3.4 ความต้องการด้านการจัดการข้อมูลระบบต้องสามารถจัดเก็บและแสดงผลข้อมูลจากเซนเซอร์และสถานะอุปกรณ์ผ่าน Firebase แบบเรียลไทม์

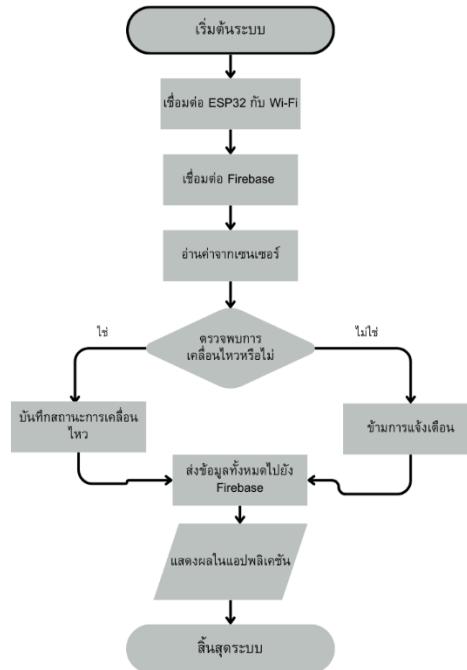
3.1.3.5 ความต้องการด้านการใช้งานระบบต้องมีรูปแบบการใช้งานที่เข้าใจง่ายและสะดวกต่อผู้ใช้งาน

3.1.3.6 ความต้องการด้านความเสถียรระบบต้องสามารถทำงานได้อย่างต่อเนื่องและมีความเสถียรในการรับ–ส่งข้อมูล

3.1.3.7 ความต้องการด้านการพัฒนาต่อยอดระบบต้องสามารถรองรับการขยายและพัฒนาเพิ่มเติมในอนาคต

3.2 การออกแบบระบบ

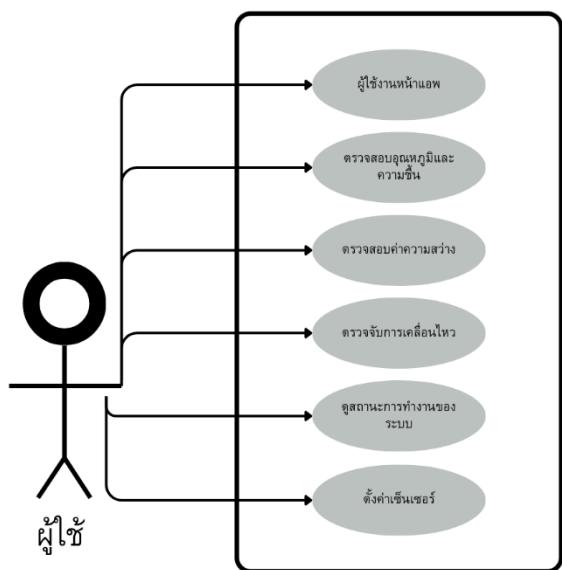
3.2.1 ภาพรวมของระบบใหม่



ภาพประกอบที่ 3.2 Work Flow Diagram

ภาพที่ 3.2 แสดงแผนภาพการทำงาน (Work Flow Diagram) ของระบบสมาร์ทโฮมโดยใช้ ESP32 ตรวจจับการเคลื่อนไหว วัดอุณหภูมิ ความชื้น และความเข้มแสง พร้อมแสดงผลและควบคุมอุปกรณ์ผ่าน Firebase

3.2.2 ขอบเขตของระบบใหม่

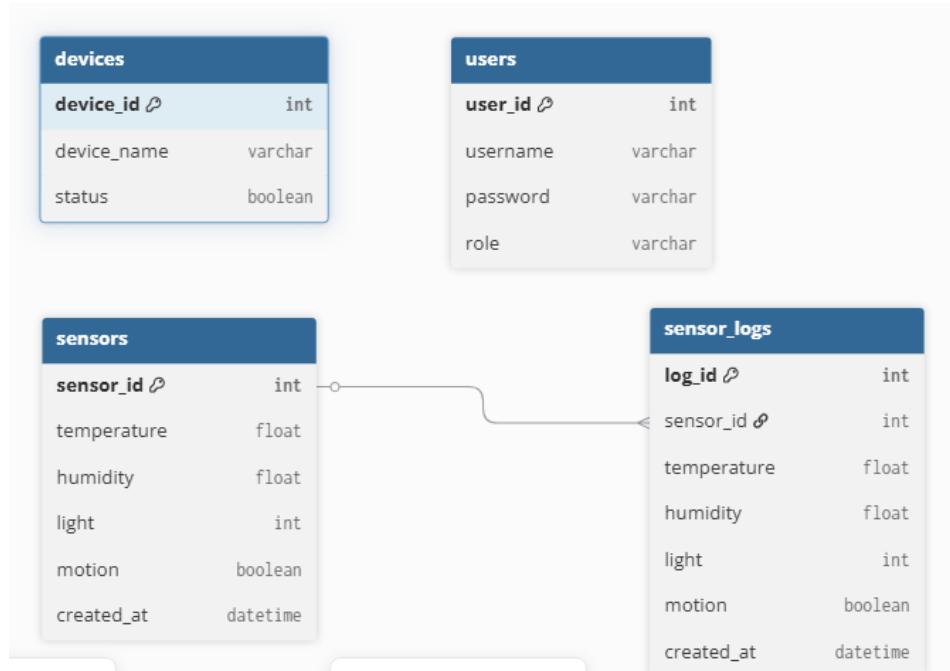


ภาพประกอบที่ 3.3 Use Case Diagram

Use Case Diagram ของระบบสมาร์ทໂໂມที่พັນນາຂຶ້ນ ແສດຂອບເຂດການທຳງານຂອງຮະບບແລະຄວາມສັນພັນນີ້ ຮະຫວ່າງຜູ້ໃຊ້ງານກັບຮະບບ ໂດຍມີຜູ້ໃຊ້ງານໜັກ (User) ເປັນຜູ້ຄວບຄຸມແລະຕຽບສອບການທຳງານຜ່ານແອປພິເຄີນນັນ ສມາർຖໂໂນ ສ່ວນຮະບບສມາർຖໂໂມທຳການທີ່ຮັບຂໍ້ມູນຈາກເຊັນເຊົ່ວໆ ປະມາລັດ ຝາຍໃນບ້ານຜ່ານ ESP32 ແລະ Firebase

ผู้ใช้งานสามารถเข้าสู่ระบบเพื่อใช้งานแอปพลิเคชัน ตรวจสอบข้อมูลอุณหภูมิ ความชื้น ค่าความสว่าง และสถานะการตรวจจับการเคลื่อนไหวแบบเรียลไทม์ ผ่านแอปพลิเคชัน โดยคำสั่งจะถูกส่งไปยัง Firebase และส่งต่อไปยัง ESP32 เพื่อควบคุมอุปกรณ์ตามที่กำหนด ระบบยังสามารถบันทึกและแสดงสถานะการทำงานทั้งหมดให้ผู้ใช้งานทราบได้อย่างต่อเนื่อง

3.2.3 ฐานข้อมูลของระบบใหม่



ภาพประกอบที่ 3.4 Class Diagram

ภาพที่ 3.4 แสดงแผนภาพฐานข้อมูลของระบบสมาร์ทโฮม ซึ่งประกอบด้วยตารางผู้ใช้งาน ตารางข้อมูลเซนเซอร์ ตารางอุปกรณ์ไฟฟ้า และตารางบันทึกข้อมูลเซนเซอร์ โดยฐานข้อมูลถูกออกแบบให้รองรับการจัดเก็บและเรียกใช้งานข้อมูลแบบเรียลไทม์

อธิบายตาราง

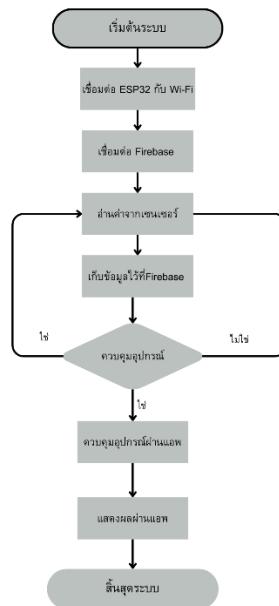
Users ใช้จัดเก็บข้อมูลผู้ใช้งานระบบสมาร์ทโฮม

Sensors ใช้เก็บค่าล่าสุดจากเซนเซอร์อุณหภูมิ ความชื้น ความเข้มแสง และการตรวจจับการเคลื่อนไหว

Devices ใช้เก็บข้อมูลอุปกรณ์ไฟฟ้าและสถานะการเปิด-ปิด

sensor_logs ใช้บันทึกประวัติค่าที่อ่านได้จากเซนเซอร์ เพื่อนำไปวิเคราะห์ย้อนหลัง

3.2.4 ขั้นตอนการทำงานของระบบใหม่



ภาพประกอบที่ 3.5 Activity Diagram

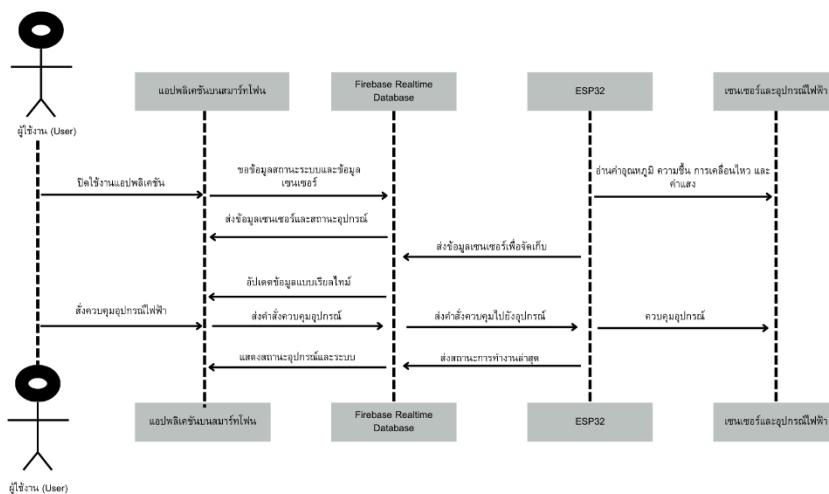
ภาพที่ 3.5 แสดงแผนภาพกิจกรรม (Activity Diagram) ของระบบสมาร์ทโคม ซึ่งอธิบายขั้นตอนการทำงานตั้งแต่การเชื่อมต่อ ESP32 กับเครือข่าย การอ่านข้อมูลจากเซ็นเซอร์ การจัดเก็บข้อมูลใน Firebase และการควบคุมอุปกรณ์ผ่านแอปพลิเคชัน

ขั้นตอนการทำงานของระบบใหม่

1. ระบบเริ่มต้นการทำงานเมื่อมีการเปิดใช้งานอุปกรณ์ ESP32
2. ESP32 ทำการเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตผ่าน Wi-Fi
3. หลังจากเชื่อมต่ออินเทอร์เน็ตสำเร็จ ระบบจะเชื่อมต่อกับ Firebase Realtime Database
4. ESP32 ทำการอ่านค่าจากเซ็นเซอร์ต่าง ๆ ได้แก่เซ็นเซอร์วัดอุณหภูมิ

- เช่นเชอร์วัดความชื้น
- เช่นเชอร์ตรวจจับการเคลื่อนไหว
- เช่นเชอร์วัดความเข้มแสง

- 5.ข้อมูลที่อ่านได้จากเซนเซอร์ทั้งหมดจะถูกส่งไปจัดเก็บใน Firebase แบบเรียลไทม์
- 6.ผู้ใช้งานเปิดแอปพลิเคชันเพื่อทำการตรวจสอบข้อมูลจากระบบ
- 7.ระบบตรวจสอบว่าผู้ใช้งานมีการส่งคำสั่งควบคุมอุปกรณ์หรือไม่
- 8.หากมีคำสั่งจากผู้ใช้งาน ระบบจะส่งคำสั่งจาก Firebase ไปยัง ESP32
- 9.ESP32 ทำการควบคุมอุปกรณ์ไฟฟ้าภายในบ้านผ่านรีเลย์ตามคำสั่งที่ได้รับ
- 10.ระบบแสดงผลสถานะการทำงานและข้อมูลจากเซนเซอร์บนแอปพลิเคชัน
- 11.ระบบจะทำงานวนซ้ำอย่างต่อเนื่องตลอดเวลาที่มีการเปิดใช้งาน

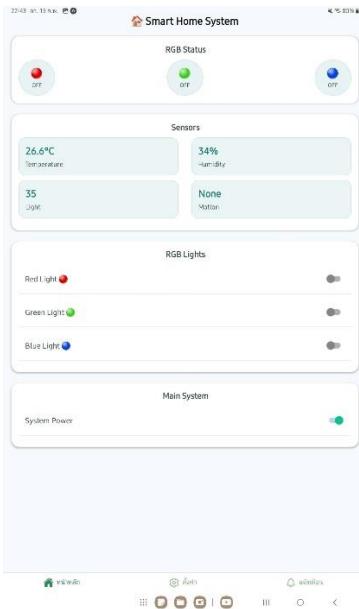


ภาพประกอบที่ 3.6 Sequence Diagram

Sequence Diagram ใช้แสดงลำดับการติดต่อสื่อสารระหว่างองค์ประกอบต่าง ๆ ของระบบสมาร์ทโฮม ได้แก่ ผู้ใช้งาน แอปพลิเคชัน Firebase และอุปกรณ์ ESP32 เพื่อให้เห็นลำดับการทำงานของระบบอย่างชัดเจน ตามช่วงเวลา

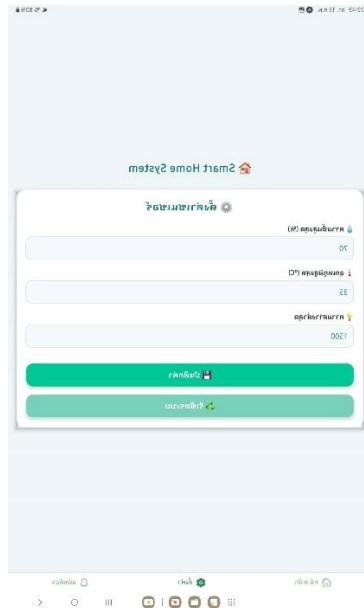
3.2.5 ต้นแบบส่วนต่อประสานกับผู้ใช้

3.2.5.1 เมนู ลิงก์ และปุ่มต่างๆ



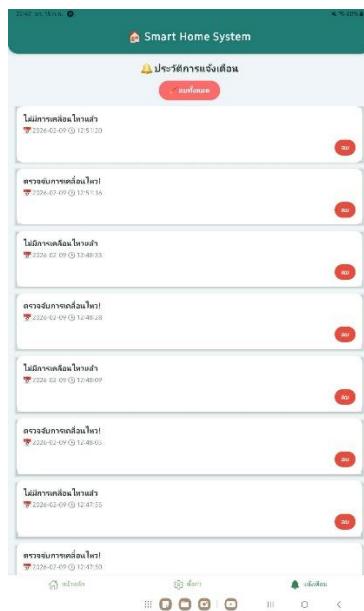
ภาพที่ 3.7 หน้าหลักของแอปพลิเคชัน

แสดงค่าของอุณหภูมิ ความชื้น แสง และสถานะของการเคลื่อนไหวแบบเรียลไทม์และผู้ใช้สามารถสั่งเปิด–ปิดระบบ RGB หรือไฟภายในบ้านได้จากปุ่มในหน้าจอ



ภาพที่ 3.8 ตั้งค่าอุปกรณ์

ผู้ใช้สามารถตั้งค่าระบบได้ในหน้านี้



ภาพที่ 3.9 หน้าการแจ้งเตือน

เมื่อเซ็นเซอร์ตรวจพบอุณหภูมิสูงหรือมีการเคลื่อนไหว ระบบจะส่งข้อความแจ้งเตือนไปยังหน้านี้

3.3 การพัฒนาโปรแกรม

3.3.1 ภาษาและเครื่องมือที่ใช้พัฒนา

- 3.3.1.1 ใช้ภาษา C/C++ สำหรับพัฒนาโปรแกรมบนไมโครคอนโทรลเลอร์ ESP32
- 3.3.1.2 ใช้ Arduino IDE เป็นเครื่องมือในการเขียนและอป์โหลดโปรแกรมไปยัง ESP32
- 3.3.1.3 ใช้ภาษา HTML, CSS และ JavaScript สำหรับพัฒนาเว็บอินเทอร์เฟซของระบบ
- 3.3.1.4 ใช้ Firebase SDK สำหรับเชื่อมต่อและสื่อสารกับฐานข้อมูล Firebase

บทที่ 4 ผลการดำเนินงาน

4.1 แผนการทดสอบ (Testing Plan)

4.1.1 วัตถุประสงค์ของการทดสอบ

- 4.1.1.1 เพื่อทดสอบความถูกต้องในการทำงานของระบบสมาร์ทโคม
- 4.1.1.2 เพื่อทดสอบการรับ-ส่งข้อมูลระหว่าง ESP32 และ Firebase
- 4.1.1.3 เพื่อทดสอบความสามารถในการควบคุมอุปกรณ์ไฟฟ้าผ่านแอปพลิเคชัน
- 4.1.1.4 เพื่อประเมินความเสถียรและความพร้อมใช้งานของระบบ

4.1.2 ขอบเขตการทดสอบ

- 4.1.2.1 การเชื่อมต่อเครือข่ายอินเทอร์เน็ตของ ESP32
- 4.1.2.2 การอ่านค่าจากเซนเซอร์อุณหภูมิและความชื้น
- 4.1.2.3 การตรวจสอบการเคลื่อนไหว
- 4.1.2.4 การอ่านค่าความเข้มแสง
- 4.1.2.5 การจัดเก็บและเรียกใช้ข้อมูลจาก Firebase
- 4.1.2.6 การแสดงผลข้อมูลบนแอปพลิเคชัน

4.1.3 วิธีการทดสอบ

- 4.1.3.1 ทดสอบการเชื่อมต่อ Wi-Fi และ Firebase
- 4.1.3.2 ทดสอบการอ่านค่าจากเซนเซอร์แต่ละชนิด
- 4.1.3.3 ทดสอบการส่งข้อมูลจาก ESP32 ไปยัง Firebase
- 4.1.3.4 ทดสอบการแสดงผลข้อมูลแบบเรียลไทม์
- 4.1.3.5 ทดสอบการสั่งเปิด–ปิดอุปกรณ์ไฟฟ้า

4.1.4 เกณฑ์การประเมินผล

- 4.1.4.1 ระบบสามารถเชื่อมต่อเครือข่ายได้อย่างถูกต้อง
- 4.1.4.2 ระบบสามารถอ่านและแสดงค่าจากเซนเซอร์ได้ถูกต้อง
- 4.1.4.3 ระบบสามารถรับ–ส่งข้อมูลกับ Firebase ได้แบบเรียลไทม์
- 4.1.4.4 ระบบสามารถควบคุมอุปกรณ์ไฟฟ้าได้ตามคำสั่ง
- 4.1.4.5 ระบบทำงานได้อย่างเสถียรและต่อเนื่อง

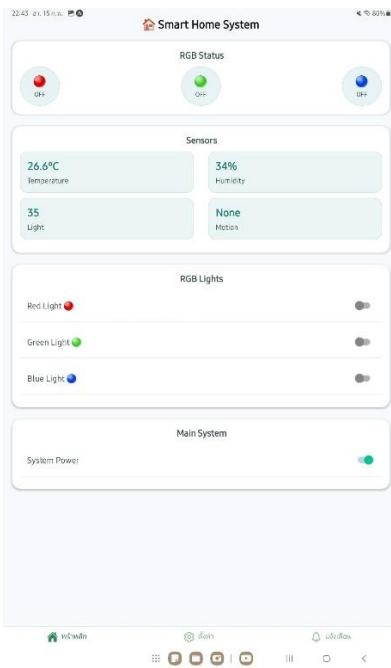
4.1.5 เครื่องมือที่ใช้ในการทดสอบ

- 4.1.5.1 อุปกรณ์ ESP32 และเซนเซอร์ที่ใช้ในโครงงาน
- 4.1.5.2 Firebase Console สำหรับตรวจสอบข้อมูล
- 4.1.5.3 เว็บเบราว์เซอร์สำหรับทดสอบการแสดงผล
- 4.1.5.4 สมาร์ทโฟนหรือคอมพิวเตอร์สำหรับควบคุมระบบ

4.2 กรณีตัวอย่างการทดสอบ (Test Case)

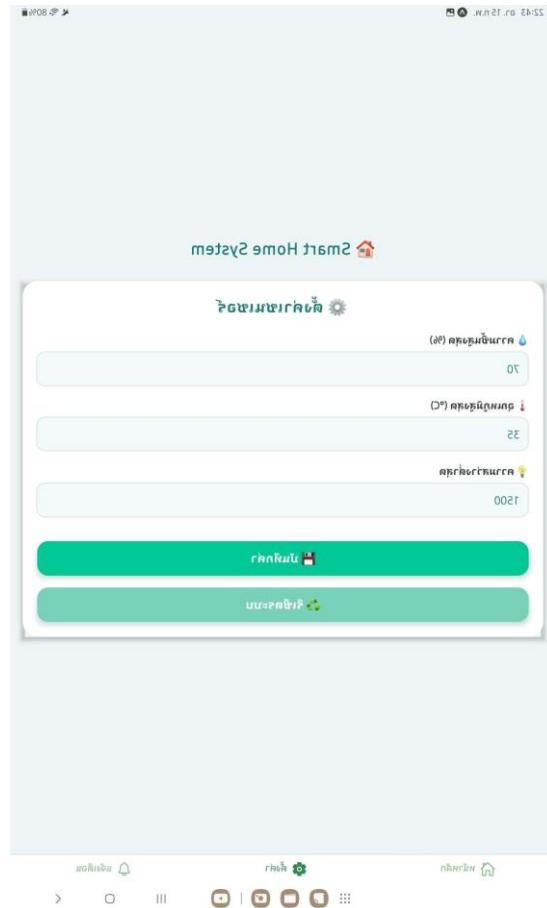
Test Case ID	ฟังก์ชันที่ทดสอบ	เงื่อนไขการทดสอบ	ผลลัพธ์ที่คาดหวัง	สถานะ
TC-01	การเชื่อมต่อ Wi-Fi	เปิดใช้งาน ESP32 และเชื่อมต่อเครือข่ายอินเทอร์เน็ต	ESP32 เชื่อมต่อ Wi-Fi ได้สำเร็จ	ผ่าน
TC-02	การเชื่อมต่อ Firebase	ตรวจสอบการเชื่อมต่อระหว่าง ESP32 กับ Firebase	ระบบเชื่อมต่อ Firebase ได้อย่างถูกต้อง	ผ่าน
TC-03	การอ่านค่าอุณหภูมิ	อ่านค่าจากเซนเซอร์อุณหภูมิ	แสดงค่าอุณหภูมิถูกต้องบนแอป	ผ่าน
TC-04	การอ่านค่าความชื้น	อ่านค่าจากเซนเซอร์ความชื้น	แสดงค่าความชื้นถูกต้องบนแอป	ผ่าน
TC-05	การตรวจจับการเคลื่อนไหว	ทดสอบเซนเซอร์ตรวจจับการเคลื่อนไหว	ระบบตรวจจับการเคลื่อนไหวได้	ผ่าน
TC-06	การอ่านค่าความเข้มแสง	ทดสอบเซนเซอร์วัดความเข้มแสง	แสดงค่าความเข้มแสงถูกต้อง	ผ่าน
TC-07	การส่งข้อมูลไป Firebase	ส่งข้อมูลจาก ESP32 ไปยัง Firebase	ข้อมูลถูกจัดเก็บใน Firebase	ผ่าน
TC-08	การแสดงผลข้อมูล	เปิดแอปพลิเคชันเพื่อตรวจสอบข้อมูล	ข้อมูลแสดงผลแบบเรียลไทม์	ผ่าน
TC-09	การเปิดอุปกรณ์ไฟฟ้า	สั่งเปิดอุปกรณ์ผ่านแอปพลิเคชัน	อุปกรณ์เปิดทำงานตามคำสั่ง	ผ่าน
TC-10	การปิดอุปกรณ์ไฟฟ้า	สั่งปิดอุปกรณ์ผ่านแอปพลิเคชัน	อุปกรณ์ปิดทำงานตามคำสั่ง	ผ่าน
TC-11	การแสดงสถานะอุปกรณ์	ตรวจสอบสถานะอุปกรณ์บนแอป	แสดงสถานะถูกต้อง	ผ่าน
TC-12	ความเสถียรของระบบ	เปิดระบบทำงานต่อเนื่องเป็นเวลานาน	ระบบทำงานต่อเนื่องไม่เกิดข้อผิดพลาด	ผ่าน

4.3 พังก์ชันการทำงานของเว็บไซต์



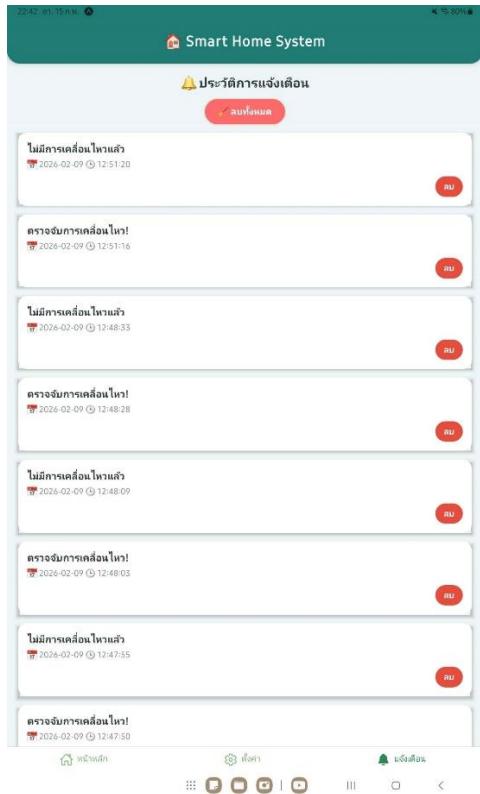
ภาพที่ 4.1 หน้าแรกของระบบ

หน้าแรกแสดงค่าของอุณหภูมิ ความชื้น แสง และสถานะของการเคลื่อนไหวแบบเรียลไทม์และผู้ใช้สามารถสั่งเปิด–ปิดระบบ RGB หรือไฟภายในบ้านได้จากปุ่มในหน้าจอ



ກາພທີ 4.2 ມີ້ນຕັ້ງຄ່າ

ແສດງມີ້ນຕັ້ງຄ່າຂອງຮະບບສມາർທໂໂມ ຈຶ່ງຄູກອອກແບບມາເພື່ອໃຫ້ຜູ້ໃຊ້ຈານສາມາດຮັດກຳທຳການທຳມະນຸດຂອງຮະບບໄດ້ຕາມຄວາມຕ້ອງການ
ມີ້ນຕັ້ງຄ່ານີ້ຊ່ວຍໃຫ້ຜູ້ໃຊ້ຈານສາມາດປັບປຸງການຕັ້ງຄ່າທີ່ເກີດຂຶ້ນກັບອຸປະກອນົມແລະຮະບບ



ภาพที่ 4.3 หน้าการแจ้งเตือน

แสดงหน้าการแจ้งเตือนของระบบสมาร์ทโฮม ซึ่งใช้สำหรับแสดงข้อมูลแจ้งเตือนเมื่อระบบตรวจพบเหตุการณ์ที่กำหนดไว้

4.4 การทดสอบการยอมรับของผู้ใช้ระบบ (Acceptance Testing)

Test Case ID	ฟังก์ชันที่ทดสอบ	เงื่อนไขการทดสอบ	ผลลัพธ์ที่คาดหวัง	สถานะ
TC-01	การเชื่อมต่อ Wi-Fi	เปิดใช้งาน ESP32 และเชื่อมต่อเครือข่ายอินเทอร์เน็ต	ESP32 เชื่อมต่อ Wi-Fi ได้สำเร็จ	ผ่าน
TC-02	การเชื่อมต่อ Firebase	ตรวจสอบการเชื่อมต่อระหว่าง ESP32 กับ Firebase	ระบบเชื่อมต่อ Firebase ได้อย่างถูกต้อง	ผ่าน
TC-03	การอ่านค่าอุณหภูมิ	อ่านค่าจากเซนเซอร์อุณหภูมิ	แสดงค่าอุณหภูมิถูกต้องบนแอป	ผ่าน
TC-04	การอ่านค่าความชื้น	อ่านค่าจากเซนเซอร์ความชื้น	แสดงค่าความชื้นถูกต้องบนแอป	ผ่าน
TC-05	การตรวจจับการเคลื่อนไหว	ทดสอบเซนเซอร์ตรวจจับการเคลื่อนไหว	ระบบตรวจจับการเคลื่อนไหวได้	ผ่าน
TC-06	การอ่านค่าความเข้มแสง	ทดสอบเซนเซอร์วัดความเข้มแสง	แสดงค่าความเข้มแสงถูกต้อง	ผ่าน
TC-07	การส่งข้อมูลไป Firebase	ส่งข้อมูลจาก ESP32 ไปยัง Firebase	ข้อมูลถูกจัดเก็บใน Firebase	ผ่าน
TC-08	การแสดงผลข้อมูล	เปิดแอปพลิเคชันเพื่อตรวจสอบข้อมูล	ข้อมูลแสดงผลแบบเรียลไทม์	ผ่าน
TC-09	การเปิดอุปกรณ์ไฟฟ้า	สั่งเปิดอุปกรณ์ผ่านแอปพลิเคชัน	อุปกรณ์เปิดทำงานตามคำสั่ง	ผ่าน
TC-10	การปิดอุปกรณ์ไฟฟ้า	สั่งปิดอุปกรณ์ผ่านแอปพลิเคชัน	อุปกรณ์ปิดทำงานตามคำสั่ง	ผ่าน
TC-11	การแสดงสถานะอุปกรณ์	ตรวจสอบสถานะอุปกรณ์บนแอป	แสดงสถานะถูกต้อง	ผ่าน
TC-12	ความสามารถของระบบ	เปิดระบบทำงานต่อเนื่องเป็นเวลานาน	ระบบทำงานต่อเนื่องไม่เกิดข้อผิดพลาด	ผ่าน

บทที่ 5 สรุปผล อภิรายผล และข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

โครงการเรื่อง “ระบบสมาร์ทโถมควบคุมอุปกรณ์ผ่านแอปพลิเคชันโดยใช้ ESP32” มีวัตถุประสงค์เพื่อพัฒนาระบบที่สามารถควบคุมและตรวจสอบสถานะของอุปกรณ์ภายในบ้าน เช่น หลอดไฟ พัดลม และระบบตรวจจับอุณหภูมิ ความชื้น และแสง ผ่านทางแอปพลิเคชันบนโทรศัพท์มือถือ โดยใช้บอร์ด ESP32 เป็นตัวควบคุมหลักในการรับส่งข้อมูลกับฐานข้อมูล Firebase Realtime Database

ระบบที่พัฒนาขึ้นสามารถทำงานได้ตามที่ออกแบบไว้ ผู้ใช้สามารถเปิด–ปิดอุปกรณ์ไฟฟ้าได้จากระยะไกล ตรวจสอบค่าจากเซนเซอร์แบบเรียลไทม์ และรับการแจ้งเตือนเมื่อเกิดเหตุการณ์ผิดปกติ เช่น อุณหภูมิสูงเกินกำหนด หรือมีการเคลื่อนไหวในบริเวณที่ติดตั้งเซนเซอร์ ทั้งนี้ระบบสามารถเชื่อมต่อกับอินเทอร์เน็ตได้อย่างเสถียร และตอบสนองต่อคำสั่งได้อย่างรวดเร็ว

ผลจากการทดสอบกับผู้ใช้จริง พบว่าระบบสามารถทำงานได้อย่างมีประสิทธิภาพ ผู้ใช้มีความพึงพอใจในด้านความสะดวก ความเข้าใจง่าย และการทำงานที่ตอบสนองต่อการใช้งานจริง

5.2 อภิรายผล

จากการดำเนินงานพบว่า

- ระบบสามารถเชื่อมต่อและสื่อสารกับฐานข้อมูล Firebase ได้อย่างถูกต้องและต่อเนื่อง
- เซนเซอร์ที่ใช้ เช่น DHT11, LDR, PIR สามารถตรวจจับค่าต่าง ๆ ได้แม่นยำและอัปเดตข้อมูลแบบเรียลไทม์
- การสั่งงานผ่านแอปพลิเคชันสามารถตอบสนองได้ทันทีภายในเวลาไม่เกิน 1–2 วินาที
- ผู้ใช้สามารถควบคุมอุปกรณ์และตรวจสอบสถานะได้จากทุกที่ที่มีอินเทอร์เน็ต ทำให้เกิดความสะดวกในการใช้งานจริง
- การแจ้งเตือนเหตุการณ์ผิดปกติช่วยเพิ่มความปลอดภัยในบ้านได้ในระดับหนึ่ง

อย่างไรก็ตาม ระบบยังมีข้อจำกัดบางประการ เช่น ความเสถียรของสัญญาณ Wi-Fi และการทำงานของแอปพลิเคชันที่ยังไม่รองรับทุกระบบปฏิบัติการ โดยเฉพาะ iOS ซึ่งต้องมีการปรับปรุงเพิ่มเติมในอนาคต

5.3 ข้อเสนอแนะในการพัฒนาระบบ

1. ควรพัฒนาแอปพลิเคชันให้รองรับทั้งระบบ iOS และ Android เพื่อให้ผู้ใช้สามารถใช้งานได้หลากหลายมากขึ้น
2. เพิ่มระบบ ล็อกอินสำหรับผู้ใช้ เพื่อความปลอดภัยและการจัดการผู้ใช้งานหลายคน
3. ปรับปรุงการออกแบบส่วนติดต่อผู้ใช้ (UI/UX) ให้สวยงามและใช้งานง่ายยิ่งขึ้น
4. เพิ่มการเชื่อมต่อกับระบบ Voice Assistant เช่น Google Assistant หรือ Alexa เพื่อสั่งงานด้วยเสียง
5. เพิ่มการเก็บข้อมูลในรูปแบบ สถิติหรือกราฟ เพื่อให้ผู้ใช้วิเคราะห์การใช้พลังงานย้อนหลังได้
6. พัฒนาให้ระบบสามารถทำงานร่วมกับอุปกรณ์สมาร์ทโฮมรุ่นอื่น ๆ ได้ (เช่น Smart Plug หรือ Smart Lamp)

5.4 สรุปภาพรวมของโครงการ

โดยสรุปแล้ว โครงการระบบสมาร์ทโฮมควบคุมอุปกรณ์ผ่านแอปพลิเคชันโดยใช้ ESP32 สามารถตอบโจทย์การใช้งานในชีวิตประจำวันได้อย่างมีประสิทธิภาพ ทั้งในด้านความสะดวก ความปลอดภัย และการตรวจสอบข้อมูลแบบเรียลไทม์ ถือเป็นต้นแบบที่สามารถนำไปต่อยอดพัฒนาให้เป็นระบบสมาร์ทโฮมเต็มรูปแบบในอนาคตได้

บรรณานุกรม

Espressif Systems. (2024). *ESP32 Technical Reference Manual*.

สืบค้นจาก: <https://www.espressif.com/en/products/socs/esp32/resources>

เข้าถึงเมื่อวันที่ 1 พฤศจิกายน 2568

Firebase. (2024). *Firebase Realtime Database Documentation*.

สืบค้นจาก: <https://firebase.google.com/docs/database>

เข้าถึงเมื่อวันที่ 1 พฤศจิกายน 2568

Nore, P., & Satyanarayana, P. (2023). *Internet of Things (IoT) Based Home Automation Using ESP-32*.

International Journal for Research in Applied Science and Engineering Technology (IJRASET).

สืบค้นจาก: <https://www.ijraset.com/research-paper/iot-based-home-automation-using-esp-32>

เข้าถึงเมื่อวันที่ 1 พฤศจิกายน 2568

Random Nerd Tutorials. (2024). *ESP32 with Firebase Realtime Database – Complete Guide*.

สืบค้นจาก: <https://randomnerdtutorials.com/esp32-firebase-realtime-database/>

เข้าถึงเมื่อวันที่ 1 พฤศจิกายน 2568

Shafie, N. A. Z. N., & Mohd Sam, S. (2022). *Real-Time Temperature and Humidity Monitoring Testing Approach Using DHT Sensor*.

Universiti Teknologi Malaysia.

สืบค้นจาก: <https://eprints.utm.my/104592/>

เข้าถึงเมื่อวันที่ 1 พฤศจิกายน 2568

Somboon, T. (2022). การพัฒนาโมดูลสมาร์ทโ Hern ต้นแบบโดยใช้บอร์ด ESP32 และระบบ Firebase.

มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี.

เข้าถึงเมื่อวันที่ 1 พฤศจิกายน 2568

Srithong, W. (2023). การออกแบบและพัฒนาระบบควบคุมอุปกรณ์ไฟฟ้าผ่านสมาร์ทโฟนโดยใช้เทคโนโลยี IoT.

วิทยาลัยเทคโนโลยีพระนครศรีอยุธยา.

เข้าถึงเมื่อวันที่ 1 พฤศจิกายน 2568

Tech Explorations. (2023). *DHT11 and DHT22 Temperature and Humidity Sensors Guide.*

สืบค้นจาก: <https://techexplorations.com/guides/dht11-dht22-sensors/>

เข้าถึงเมื่อวันที่ 1 พฤศจิกายน 2568

Tutorials Point. (2023). *PIR Sensor Working and Applications.*

สืบค้นจาก: <https://www.tutorialspoint.com/pir-sensor>

เข้าถึงเมื่อวันที่ 1 พฤศจิกายน 2568

Wemos Wiki. (2023). *LDR Sensor Module – Working Principle and Circuit Design.*

สืบค้นจาก: <https://components101.com/sensors/ldr-sensor>

เข้าถึงเมื่อวันที่ 1 พฤศจิกายน 2568

ภาคผนวก ก

แบบฟอร์มกรอกข้อมูล

แบบประเมินความพึงพอใจของผู้ใช้ระบบสมาร์ทໂໂມຄວບຄຸມອຸປະກຣນິ່ງແລ້ວພລິເຄີ້ນ

ตอนที่ 1 ข้อมูลทั่วไปของผู้ตอบแบบสอบถาม

1. เพศ

ชาย

หญิง

อื่น ๆ

2. อายุ

ต่ำกว่า 18 ปี

18-25 ปี

26-35 ปี

36 ปีขึ้นไป

3.อาชีพ

 นักเรียน / นักศึกษา พนักงานบริษัท ธุรกิจส่วนตัว อื่น ๆ _____

4.เคยใช้ระบบสมาร์ทโฟมมาก่อนหรือไม่

 เคย ไม่เคย

ระดับความพึงพอใจ คะแนน

พึงพอใจมากที่สุด 5

พึงพอใจมาก 4

พึงพอใจปานกลาง 3

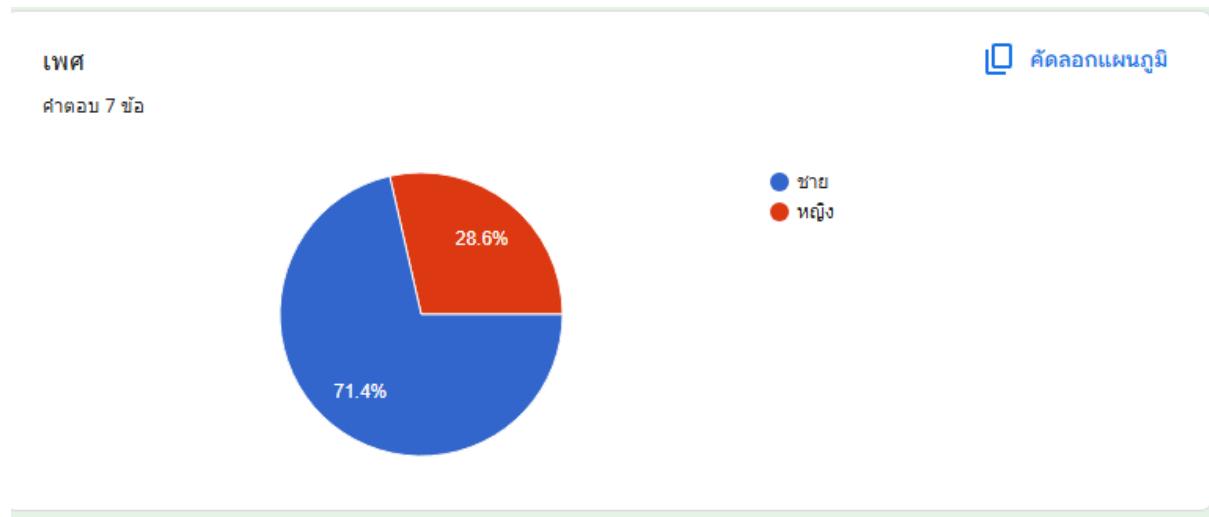
พึงพอใจน้อย 2

พึงพอใจน้อยที่สุด 1

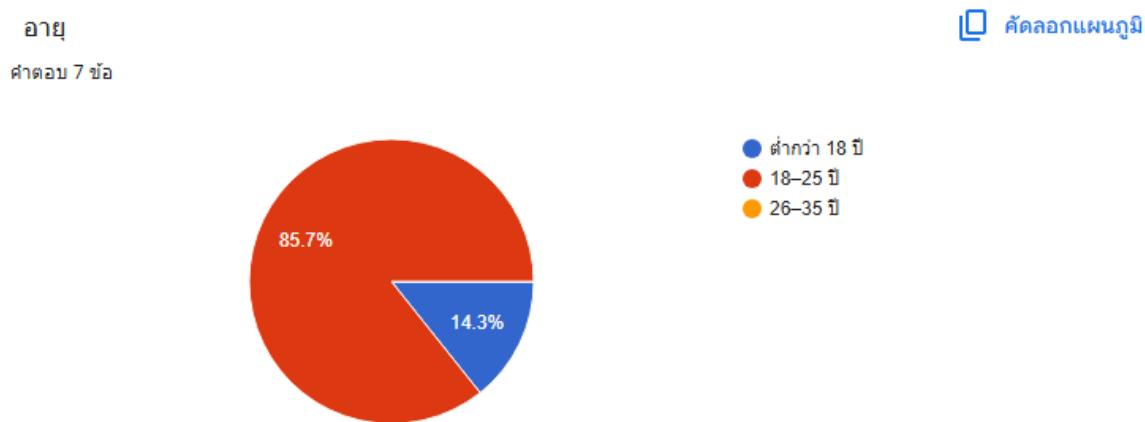
รายการประเมิน	1	2	3	4	5
1.1 แอปพลิเคชันมีรูปแบบสวยงามและเข้าใจง่าย	<input type="checkbox"/>				
1.2 โครงสร้างเมนูใช้งานสะดวก	<input type="checkbox"/>				
1.3 การแสดงข้อมูลของอุปกรณ์และเซนเซอร์ชัดเจน	<input type="checkbox"/>				
รายการประเมิน	1	2	3	4	5
2.1 ระบบสามารถเชื่อมต่อกับอุปกรณ์ได้เสถียร	<input type="checkbox"/>				
2.2 การสั่งงานผ่านแอปพลิเคชันทำได้รวดเร็ว	<input type="checkbox"/>				
2.3 ข้อมูลที่แสดงในแอปตรงกับค่าจริงจากอุปกรณ์	<input type="checkbox"/>				

ผลสรุปประเมินความพึงพอใจ

จากแบบทดสอบทั้งหมด 7 คนแบ่งเป็นผู้ชาย 5 คนและผู้หญิง 2 คน



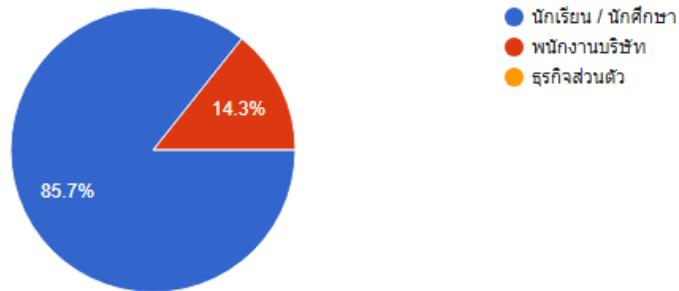
กราฟแสดงเพศที่ทำแบบสอบถาม



กราฟแสดงอายุ ที่ทำแบบสอบถาม

อาชีพ
ค่าตอบ 7 ข้อ

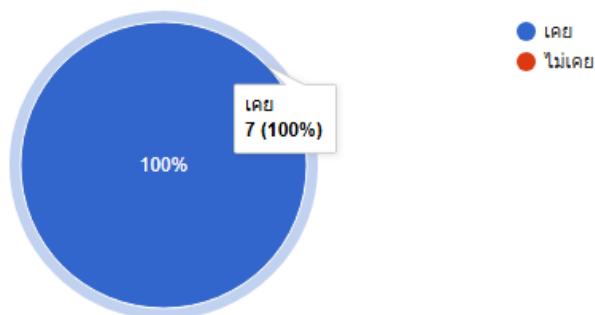
 คัดลอกแผนภูมิ



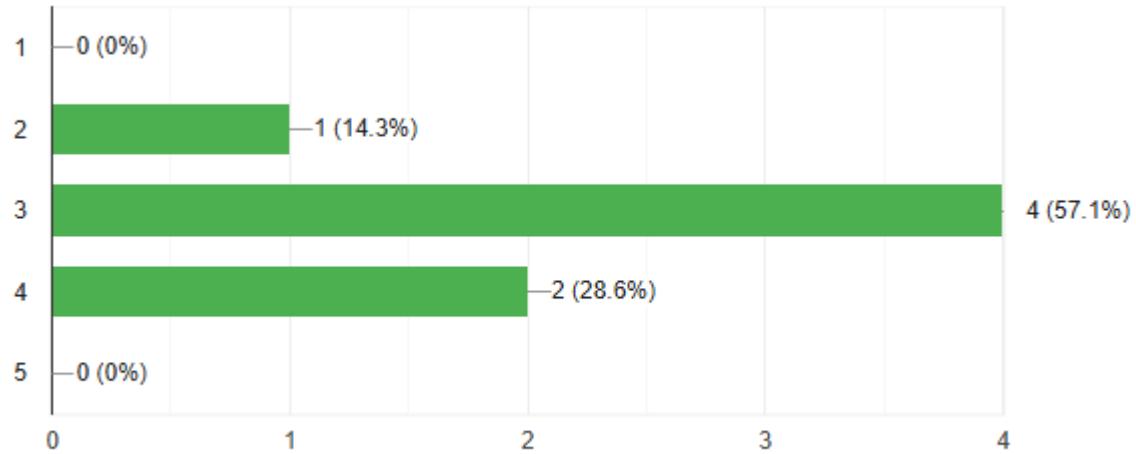
กราฟแสดงอาชีพ ที่ทำแบบสอบถาม

เคยใช้ระบบสมาร์ทโฟนมาก่อนหรือไม่
ค่าตอบ 7 ข้อ

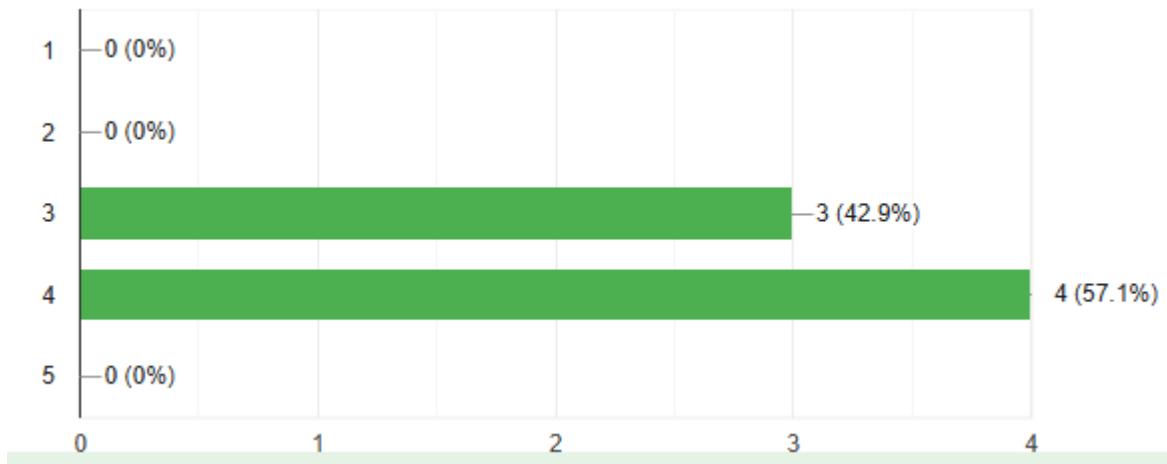
 คัดลอกแผนภูมิ



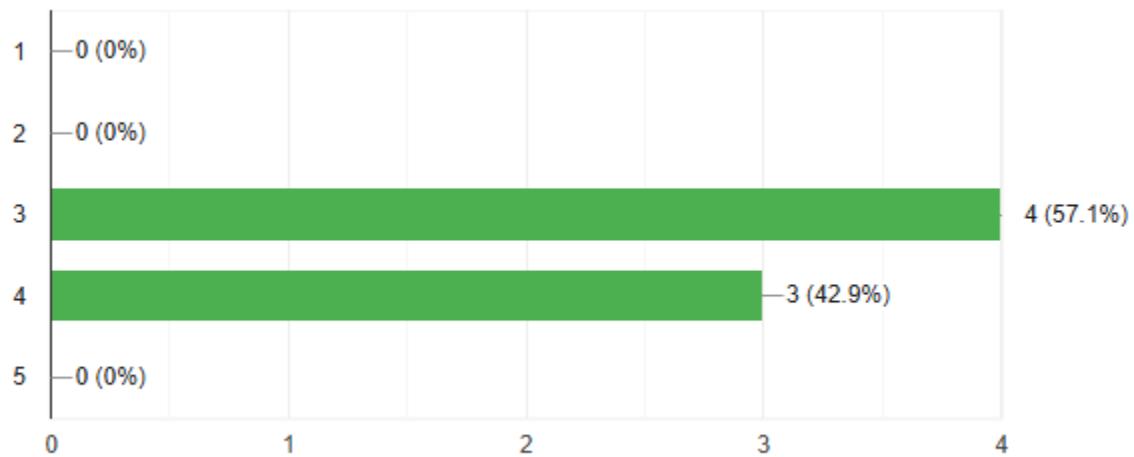
กราฟแสดงว่าเคยใช้ระบบหรือไม่ ที่ทำแบบสอบถาม



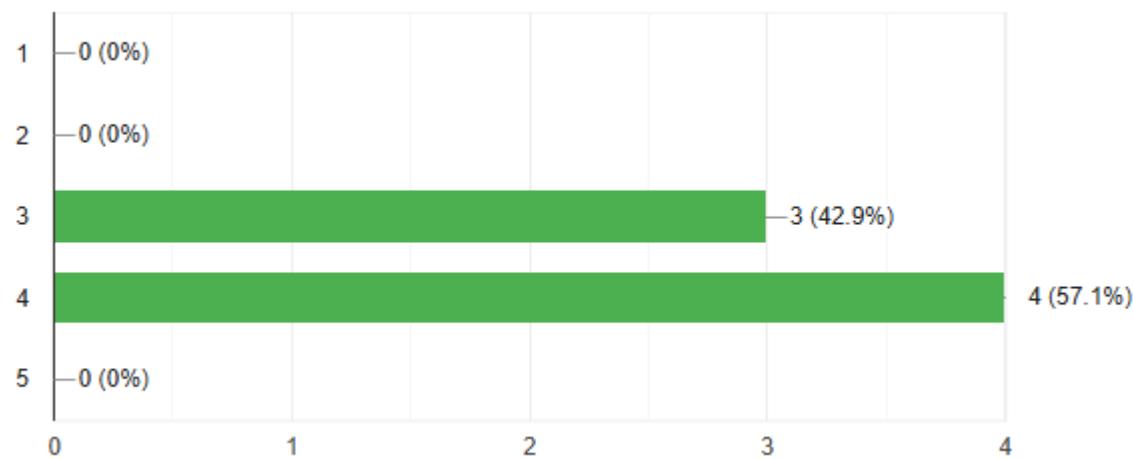
กราฟประเมินรูปแบบส่วยงาม



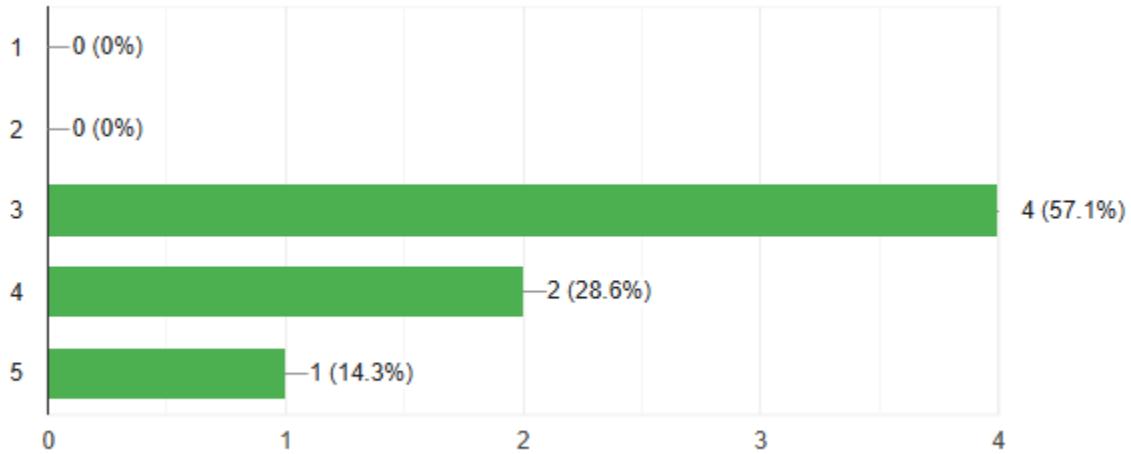
กราฟประเมินเมนูใช้งานสะดวก



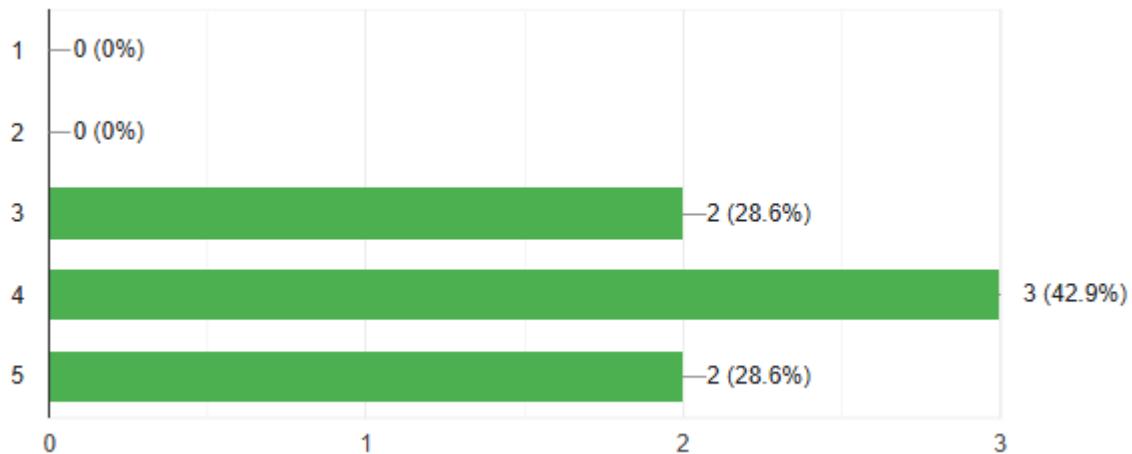
กราฟประเมินรูปแสดงข้อมูลและความซัดเจน



กราฟประเมินความเสถียร



กราฟประเมินการสั่งงานผ่านแอปพลิเคชัน



กราฟประเมินข้อมูลที่แสดงในแอป

ภาคผนวก ข

ดัชนีไฟล์ Source Code ของโปรแกรมและคำอธิบาย

ระบบจัดการหนังสือห้องสมุดสาขาประเทศไทย

โครงสร้างไฟล์ของโปรแกรมสามารถท่องถูกพัฒนาด้วย React Native และ Expo โดยมีการจัดแยกไฟล์ตามหน้าที่การทำงานเพื่อให้ง่ายต่อการพัฒนาและบำรุงรักษา รายละเอียดของไฟล์แต่ละส่วนมีดังนี้

1. โฟลเดอร์หลักของโปรแกรม

- **.expo**
ใช้สำหรับจัดเก็บไฟล์ที่เกี่ยวข้องกับการทำงานของ Expo ซึ่งเป็นเครื่องมือช่วยในการพัฒนาและรันแอปพลิเคชัน React Native
- **.vscode**
ใช้เก็บค่าการตั้งค่าของโปรแกรม Visual Studio Code เช่น การตั้งค่า Editor และ Extensions
- **assets**
ใช้จัดเก็บไฟล์สื่อ เช่น รูปภาพ ไอคอน หรือไฟล์อื่น ๆ ที่ใช้แสดงผลภายในแอปพลิเคชัน
- **node_modules**
ใช้จัดเก็บไลบรารีและแพ๊กเกจต่าง ๆ ที่ติดตั้งผ่าน npm ซึ่งจำเป็นต่อการทำงานของโปรแกรม

2. โฟลเดอร์ screens

โฟลเดอร์ screens ใช้สำหรับจัดเก็บไฟล์หน้าจอหลักของแอปพลิเคชัน โดยแต่ละไฟล์แทนหน้าจอการทำงานหนึ่งหน้าจอ

- **HomeScreen.js**
ใช้แสดงหน้าหลักของระบบสมาร์ทโฟน และข้อมูลจากเซนเซอร์ เช่น อุณหภูมิ ความชื้น การเคลื่อนไหว และความเข้มแสง รวมถึงแสดงสถานะของอุปกรณ์ภายในบ้าน

- **NotificationScreen.js**
ใช้แสดงหน้าการแจ้งเตือน เมื่อระบบตรวจพบเหตุการณ์ต่าง ๆ เช่น การตรวจจับการเคลื่อนไหวหรือค่าผิดปกติจากเซนเซอร์
- **SettingsScreen.js**
ใช้สำหรับหน้าตั้งค่าระบบ ให้ผู้ใช้งานสามารถกำหนดค่าการทำงานของระบบและอุปกรณ์ต่าง ๆ ได้

3. ไฟล์หลักของโปรแกรม

- **App.js**
เป็นไฟล์หลักของแอปพลิเคชัน ทำหน้าที่กำหนดโครงสร้างหลักของระบบและจัดการการนำทาง (Navigation) ระหว่างหน้าจอต่าง ๆ
- **index.js**
เป็นไฟล์เริ่มต้นของโปรแกรม ทำหน้าที่เรียกใช้งานแอปพลิเคชันหลักเพื่อเริ่มการทำงาน
- **firebaseConfig.js**
ใช้สำหรับกำหนดค่าการเชื่อมต่อกับ Firebase เช่น API Key, Database URL และการตั้งค่าการเขื่อมต่อฐานข้อมูล

4. ไฟล์กำหนดค่าและไฟล์ระบบ

- **app.json**
ใช้กำหนดค่าพื้นฐานของแอปพลิเคชัน เช่น ชื่อแอป ไอคอน และการตั้งค่าของ Expo
- **package.json**
ใช้ระบุรายละเอียดของโปรแกรม เช่น ชื่อโปรเจค เวอร์ชัน และไลบรารีที่ใช้งานในการพัฒนา
- **package-lock.json**
ใช้บันทึกเวอร์ชันของแพ็กเกจที่ติดตั้ง เพื่อให้การติดตั้งไลบรารีในแต่ละครั้งมีความสอดคล้องกัน
- **.gitignore**
ใช้กำหนดไฟล์หรือโฟลเดอร์ที่ไม่ต้องการให้ระบบควบคุมเวอร์ชัน (Git) ติดตาม

ประวัติผู้จัดทำโครงการ

ชื่อ-สกุล : นาย วิทยา คนที

วันเดือนปีเกิด : วันที่ 24 มีนาคม พ.ศ. 2547

โทรศัพท์ : 0649344494

อีเมล : s65122250008@ssru.ac.th

ประวัติการศึกษา

ประถมศึกษา โรงเรียนบางคูลัด

มัธยมศึกษาตอนต้น โรงเรียนเตรียมอุดมศึกษาน้อมเกล้า นนทบุรี

มัธยมศึกษาตอนปลาย โรงเรียนเตรียมอุดมศึกษาน้อมเกล้า นนทบุรี

ปริญญาตรี วิทยาการคอมพิวเตอร์ มหาวิทยาลัยราชภัฏสวนสุนันทา

