

Architectural Drivers Document: Smart Home IoT

1. System Description

System Description

ระบบ Smart Home เป็นระบบที่ออกแบบมาเพื่อช่วยให้ผู้ใช้งานสามารถควบคุมอุปกรณ์ภายในบ้านได้อย่างสะดวก ปลอดภัย และมีประสิทธิภาพ ผ่านสมาร์ทโฟนหรืออุปกรณ์เชื่อมต่ออื่น ๆ ผู้ใช้สามารถสั่งเปิด-ปิด อุปกรณ์ไฟฟ้า ตรวจสอบสถานะอุปกรณ์ รับการแจ้งเตือนเหตุผิดปกติแบบ Real-time และตั้งเวลาการทำงานอัตโนมัติได้จากทุกที่ทุกเวลา ระบบยังรองรับการทำงานร่วมกับผู้ช่วยอัจฉริยะ เช่น Google Assistant หรือ Siri เพื่อเพิ่มความสะดวกในการใช้งานด้วยคำสั่งเสียง

ระบบนี้ออกแบบมาเพื่อรองรับผู้ใช้งานทุกกลุ่ม ไม่ว่าจะเป็นเจ้าของบ้าน ผู้อยู่อาศัยทั่วไป ผู้ดูแลบ้าน หรือแม้แต่ครอบครัวที่มีสมาชิกหลายคน โดยมีฟีเจอร์จัดการสิทธิ์ผู้ใช้งาน (Permission Management) เพื่อกำหนดว่าบุคคลแต่ละคนสามารถใช้งานอุปกรณ์ใดได้บ้าง นอกจากนี้ ระบบยังช่วยแก้ปัญหาการใช้พลังงานสิ้นเปลือง การลืมปิดอุปกรณ์ การขาดความปลอดภัยในบ้าน และการขาดข้อมูลย้อนหลังเกี่ยวกับกิจกรรมภายในบ้าน

ระบบ Smart Home ยังมาพร้อมความสามารถด้านความปลอดภัย เช่น การตรวจจับการบุกรุก กล้องวงจรปิด และเซนเซอร์ตรวจจับความผิดปกติต่าง ๆ เพื่อแจ้งเตือนผู้ใช้งานทันที และสามารถทำงานบางส่วนในโหมดออฟไลน์เมื่ออินเทอร์เน็ตล่ม เพื่อให้ฟังก์ชันพื้นฐานยังคงใช้งานได้ สุดท้าย ระบบถูกออกแบบให้รองรับการขยายตัวในอนาคต สามารถเพิ่มอุปกรณ์ใหม่ได้ง่าย และรองรับเทคโนโลยี IoT รุ่นใหม่ ๆ อย่างต่อเนื่อง ทำให้ระบบมีความยืดหยุ่นและเหมาะสมต่อการใช้งานในระยะยาว.

2. Target Users

- เจ้าของบ้าน (Home Owner)
- สมาชิกครอบครัว (Family Member)
- ผู้ดูแลบ้านหรือแม่บ้าน (Housekeeper)
- ผู้พัฒนาอุปกรณ์ภายนอก (External Device Developer)

3. Key Features (Top 10)

1. ควบคุมระยะไกล (Remote Control / Mobile App Control)
2. ควบคุมด้วยเสียง (Voice Control / Smart Assistant Integration)
3. ระบบรักษาความปลอดภัยเมื่อมีบุคคลภายนอก (Intrusion Security System)
4. การจัดการพลังงาน (Energy Monitoring & Efficiency)
5. ระบบแจ้งเตือนแบบ Real Time (Real-Time Notifications)
6. ระบบตั้งเวลาการควบคุมล่วงหน้า Scheduled Control
7. ระบบการจัดการสิทธิ์ Management & Permission
8. ระบบเก็บบันทึกกิจกรรมผู้ใช้ Activity Log (ประวัติการใช้งาน)
9. ระบบจัดการภัยพิบัติ (Disaster Management System)
10. ระบบทำงานแบบออฟไลน์ (Offline Mode)

ส่วนที่ 2: Functional Requirements

Functional Requirements:

ควบคุมระยะไกล (Remote Control / Mobile App Control)

FR-01: ระบบต้องรองรับการควบคุมอุปกรณ์ไฟฟ้าและอุปกรณ์ IoT ภายในบ้านผ่าน Mobile Application หรือ Web Interface ได้

- **Input:** ผู้ใช้กดปุ่มเปิด-ปิด หรือปรับระดับการทำงานของอุปกรณ์
- **Process:** ระบบส่งคำสั่งผ่าน Internet → Gateway → อุปกรณ์
- **Output:** อุปกรณ์ตอบสนองตามคำสั่ง เช่น เปิดไฟ ปรับอุณหภูมิ เปิดกล้อง ฯลฯ
- **เงื่อนไข:** ต้องตอบสนองภายในเวลาที่กำหนด (น้อยกว่า 4-5 วินาที)

ควบคุมด้วยเสียง (Voice Control / Smart Assistant Integration)

FR-02: ระบบต้องเชื่อมต่อกับผู้ช่วยอัจฉริยะ เช่น Google Assistant, Siri, Alexa เพื่อรับคำสั่งเสียง

- **Input:** คำสั่งเสียง เช่น “เปิดไฟห้องนั่งเล่น”
- **Process:** Assistant → API → Smart Home System
- **Output:** อุปกรณ์ทำงานตามคำสั่ง และแสดงผลสำเร็จ/ไม่สำเร็จ

ระบบรักษาความปลอดภัยเมื่อมีบุคคลภายนอก (Intrusion Security System)

FR-03: ระบบต้องตรวจจับการเคลื่อนไหวหรือการรบกวนจาก Sensor ต่าง ๆ

FR-04: เมื่อพบการบุกรุก ระบบต้องแจ้งเตือนผู้ใช้ทันทีผ่าน App/Push Notification

FR-05: ระบบต้องบันทึก Event และภาพ/วิดีโอจากกล้อง (ถ้ามี)

- **Input:** Motion sensor, door/window sensor
- **Output:** Alarm, แจ้งเตือน Real-time, บันทึกหลักฐาน
- **เงื่อนไข:** แจ้งเตือนภายใน < 5 วินาทีหลังพบเหตุการณ์

การจัดการพลังงาน (Energy Monitoring & Efficiency)

FR-06: ระบบต้องแสดงข้อมูลการใช้พลังงานของอุปกรณ์แต่ละชนิดแบบ Real-time

FR-07: ระบบต้องสรุปการใช้พลังงานรายวัน/รายเดือน

FR-08: ระบบต้องแจ้งเตือนเมื่อการใช้ไฟฟ้าผิดปกติหรือเกิน Threshold

- **Output:** Dashboard, การแจ้งเตือน Over Usage
- **Process:** เก็บข้อมูลจาก Smart Meter / IoT Devices

ระบบแจ้งเตือนแบบ Real Time (Real-Time Notifications)

FR-09: ระบบต้องแจ้งเตือนเหตุการณ์สำคัญทันที เช่น เปิดประตู, ไฟดับ, อุณหภูมิสูง

FR-10: ผู้ใช้สามารถตั้งค่าเปิด/ปิด ประเภทแจ้งเตือนได้

- **Output:** Push Notification / SMS / Email

ระบบตั้งเวลาการควบคุมล่วงหน้า (Scheduled Control)

FR-11: ผู้ใช้สามารถตั้งเวลาให้ระบบเปิด-ปิดอุปกรณ์ล่วงหน้าได้

- เช่น เปิดแอร์เวลา 18:00 ทุกวัน

FR-12: ระบบต้องทำงานตามกำหนดเวลาแม้ผู้ใช้จะไม่ได้เชื่อมต่อกับอินเทอร์เน็ตในขณะนั้น

- **Process:** Scheduler Service → Local execution

ระบบการจัดการสิทธิ (Management & Permission)

FR-13: ผู้ดูแลระบบสามารถกำหนดสิทธิผู้ใช้ เช่น

- Admin / Member / Housekeeper

FR-14: ระบบต้องควบคุมการเข้าถึงตามสิทธิ เช่น Guest เปิดได้แค่ไฟ แต่อ่านข้อมูลพลังงานไม่ได้

FR-15: ระบบต้องรองรับการเพิ่ม/ลบ/แก้ไขสิทธิผู้ใช้งาน

ระบบเก็บบันทึกกิจกรรมผู้ใช้ (Activity Log)

FR-16: ระบบต้องบันทึกทุกกิจกรรมสำคัญ เช่น

- การเปิด/ปิดอุปกรณ์
- การล็อกอิน
- การแจ้งเตือน

FR-17: ผู้ดูแลต้องสามารถค้นหา/กรอง Log ตามช่วงเวลา / ประเภทเหตุการณ์ / ผู้ใช้ได้

ระบบจัดการภัยพิบัติ (Disaster Management System)

FR-18: ระบบต้องตรวจจับเหตุการณ์เช่น ควั่น ไฟไหม้ น้ำรั่ว แผ่นดินไหว จาก Sensor ที่รองรับ

FR-19: ระบบต้องแจ้งเตือนผู้ใช้ทันทีเมื่อมีสัญญาณผิดปกติ

FR-20: ระบบต้องสั่งปิดอุปกรณ์บางอย่างอัตโนมัติ เช่น ตัดไฟหลักหรือปิดวาล์วน้ำ

ระบบทำงานแบบออฟไลน์ (Offline Mode)

FR-21: ระบบต้องยังควบคุมอุปกรณ์พื้นฐานได้แม้อินเทอร์เน็ตล่ม เช่น ผ่าน Bluetooth, Local LAN

FR-22: งานที่เกี่ยวข้องกับ Safety เช่น Alarm ต้องทำงานได้โดยไม่ต้องใช้ Internet

FR-23: เมื่อระบบกลับมาออนไลน์ ต้อง Sync ข้อมูลอัตโนมัติ เช่น Log, การตั้งเวลา

ส่วนที่ 3: Quality Attributes & Scenarios

Quality Attributes

QA-1: Performance

Scenario:

ส่วน	รายละเอียด
Source	ผู้ใช้
Stimulus	สั่งเปิดไฟผ่าน Mobile App
Artifact	Device Controller Service
Environment	ระบบออนไลน์ปกติ
Response	อุปกรณ์เปิดไฟและสถานะในแอปอัปเดต
Response Measure	ระบบต้องตอบสนองภายใน ≤ 4 วินาที

เหมาะกับ Feature: Remote Control, Real-Time Notification

QA-2: Reliability— ระบบทำงานแม้เน็ตล่ม (Offline Mode):

Scenario: Local Execution When Internet Is Down

ส่วน	รายละเอียด
Source	ผู้ไม่ประสงค์ดี
Stimulus	พยายามล็อกอินด้วยรหัสผ่านผิดซ้ำ
Artifact	Authentication & Permission Management
Environment	เมื่อมีการเชื่อมต่อจากอุปกรณ์ภายนอก
Response	ระบบล็อกบัญชีชั่วคราวและแจ้งเตือนเจ้าของ
Response Measure	ต้องบล็อกบัญชีภายใน 1 วินาที และส่งแจ้งเตือนภายใน 5 วินาที

เหมาะกับ Feature: Permission Management, Intrusion Security

QA-3: Security — ป้องกันบุกรุกและเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต

Scenario: Unauthorized Access Attempt

ส่วน	รายละเอียด
Source	ผู้ใช้ใหม่
Stimulus	ต้องการเพิ่มอุปกรณ์ใหม่เข้าระบบ
Artifact	Mobile Application UI
Environment	เมื่อเข้าใช้งานครั้งแรก
Response	ระบบให้คำแนะนำ Step-by-Step พร้อม UI ที่เข้าใจง่าย (Wizard)
Response Measure	ผู้ใช้อ้างเพิ่มอุปกรณ์ได้สำเร็จภายใน ≤ 60 วินาที และมี error rate $< 5\%$

เหมาะกับ Feature: Permission Management, Intrusion Security

QA-4: Usability — ใช้งานง่ายและไม่สร้างภาระให้ผู้ใช้

Scenario: Simple Control Interaction

ส่วน	รายละเอียด
Source	ผู้ใช้ใหม่
Stimulus	ต้องการเพิ่มอุปกรณ์ใหม่เข้าระบบ
Artifact	Mobile Application UI
Environment	เมื่อเข้าใช้งานครั้งแรก
Response	ระบบให้คำแนะนำ Step-by-Step พร้อม UI ที่เข้าใจง่าย (Wizard)
Response Measure	ผู้ใช้อาจเพิ่มอุปกรณ์ได้สำเร็จภายใน ≤ 60 วินาที และมี error rate $< 5\%$

เหมาะกับ Feature: Remote Control, Voice Control, Scheduled Control

QA-5: Scalability — รองรับการเพิ่มอุปกรณ์จำนวนมากในบ้าน

Scenario: Device Load Expansion

ส่วน	รายละเอียด
Source	ผู้ใช้งานที่อยู่ใน
Stimulus	เชื่อมต่ออุปกรณ์ใหม่พร้อมกัน 100 ตัว
Artifact	Device Registry / MQTT Broker
Environment	Network และ Cloud ทำงานปกติ
Response	ระบบต้องลงทะเบียนและซิงค์สถานะได้ปกติ
Response Measure	ลงทะเบียนอุปกรณ์ใหม่ ≤ 4 วินาที/อุปกรณ์ และ latency < 4 วินาที สำหรับคำสั่งควบคุม

เหมาะกับ Feature: Activity Log, Permission, Energy Monitoring

QA-6: Modifiability — แก้ไขระบบง่ายและปลอดภัย

Scenario: Update Firmware or Add New Device Type

ส่วน	รายละเอียด
Source	ทีมพัฒนา
Stimulus	เพิ่ม Device Type ใหม่
Artifact	Device Plug-in Architecture / Firmware Module
Environment	ระบบกำลังให้บริการปกติ
Response	ระบบต้องรองรับอุปกรณ์ใหม่โดยไม่กระทบอุปกรณ์เก่า
Response Measure	ต้อง deploy update โดยไม่ downtime (Zero Downtime Deployment) เวลา integration ≤ 2 วันทำงาน

เหมาะกับ Feature: Remote Control, Energy Monitoring, Activity Log

ส่วนที่ 4: Constraints

Technical Constraints:

ข้อจำกัดด้านเทคนิคที่เกิดจากเทคโนโลยี อุปกรณ์ หรือระบบที่เลือกใช้

- **TC-01:** ระบบต้องรองรับเฉพาะโปรโตคอลมาตรฐาน (Wi-Fi, Zigbee, Z-Wave, Bluetooth LE) ทำให้ไม่สามารถรองรับอุปกรณ์เฉพาะทางที่ใช้โปรโตคอลปิดได้
- **TC-02:** ความสามารถของ Gateway/Hub ถูกจำกัดตามสเปกฮาร์ดแวร์ เช่น ประมวลคำสั่งได้สูงสุด X ครั้ง/วินาที
- **TC-03:** ระบบ Cloud ที่ใช้ขึ้นอยู่กับ SLA ของผู้ให้บริการ ทำให้ Availability ไม่สามารถรับประกัน 100%
- **TC-04:** แอปพลิเคชันต้องพัฒนาให้รองรับทั้ง Android และ iOS ซึ่งมีข้อจำกัดด้าน API และสิทธิ์เข้าถึงต่างกัน
- **TC-05:** ความเร็วอินเทอร์เน็ตของผู้ใช้เป็นข้อจำกัดที่ควบคุมไม่ได้ โดยเฉพาะพีเจอร์ Real-Time และ กล้องวงจรปิด
- **TC-06:** บางอุปกรณ์ IoT ไม่รองรับ OTA Firmware Update ทำให้ความสามารถในการอัปเดตถูกจำกัด

Time Constraints:

ข้อจำกัดด้านเวลาในการพัฒนา ส่งมอบ หรือดำเนินการ

- **TMC-01:** ระยะเวลาโครงการต้องเสร็จภายใน X เดือน ทำให้ไม่สามารถพัฒนาฟีเจอร์ขั้นสูงทั้งหมดได้
- **TMC-02:** การทดสอบอุปกรณ์ IoT ใช้เวลานานเนื่องจากต้องรอ Hardware ชุดจริง ทำให้เวลาทดสอบจำกัด
- **TMC-03:** การรอ API จาก Third-Party เช่น Google Assistant, Alexa อาจทำให้กำหนดการล่าช้า
- **TMC-04:** การพัฒนา Mobile App บนสองแพลตฟอร์มต้องใช้รอบการทดสอบแยก ทำให้ใช้เวลามากขึ้น
- **TMC-05:** ต้องดำเนินการติดตั้ง/ทดสอบภายในสถานที่ของผู้ใช้ซึ่งต้องเผื่อเวลาเตรียมหน้างาน

Budget Constraints:

ข้อจำกัดด้านงบประมาณ ทั้งอุปกรณ์, Cloud, บุคลากร, การพัฒนา และการติดตั้ง

- BC-01: งบประมาณจำกัดทำให้ไม่สามารถใช้เซิร์ฟเวอร์ประสิทธิภาพสูงหรือ Cloud Tier สูงสุดได้
- BC-02: ต้นทุนอุปกรณ์ IoT (Sensor/Hub/Camera) จำกัดจำนวนอุปกรณ์ที่สามารถติดตั้งในโครงการ
- BC-03: งบประมาณบุคลากรจำกัด ทำให้ทีมพัฒนา/ทดสอบมีขนาดเล็ก
- BC-04: ต้องเลือกใช้ Open-source และ Free-tier Services ในบางส่วนเพื่อลดค่าใช้จ่าย
- BC-05: การออกแบบให้รองรับ Scalability ระดับสูงอาจเกินงบประมาณโครงการ

Legal/Policy Constraints:

- LPC-01: การเก็บข้อมูลผู้ใช้ต้องปฏิบัติตาม PDPA/GDPR ซึ่งจำกัดรูปแบบการจัดเก็บและประมวลผลข้อมูล
- LPC-02: การใช้งานกล้องวงจรปิดต้องปฏิบัติตามกฎหมาย CCTV และการบันทึกภาพผู้เยี่ยมชม
- LPC-03: ข้อมูลการใช้งานพลังงานและกิจกรรมในบ้านถือเป็นข้อมูลส่วนบุคคล ต้องมีมาตรการป้องกัน
- LPC-04: อุปกรณ์ไฟฟ้าและ IoT ต้องผ่านมาตรฐาน เช่น มอก., CE, FCC ตามที่กฎหมายกำหนด
- LPC-05: ห้ามเก็บข้อมูลระบุตัวบุคคลที่ไม่จำเป็น เช่น เสียง รูปภาพ โดยไม่ได้รับความยินยอม
- LPC-06: ต้องปฏิบัติตามนโยบายบริษัทเกี่ยวกับความปลอดภัยเครือข่ายและการเข้ารหัส

ส่วนที่ 5: Assumptions

Assumptions ด้านผู้ใช้งาน (User-related Assumptions)

1. ผู้ใช้งานมีสมาร์ทโฟนที่รองรับการติดตั้งแอปพลิเคชันทั้งบน iOS หรือ Android
2. ผู้ใช้มีความรู้พื้นฐานด้านเทคโนโลยี เช่น การเชื่อมต่อ Wi-Fi, การยืนยันตัวตน และการจัดการอุปกรณ์
3. ผู้ใช้จะตั้งค่ารหัสผ่านและข้อมูลความปลอดภัยตามคำแนะนำของระบบ
4. ผู้ใช้จะยินยอมให้สิทธิ์การเข้าถึงบางอย่าง เช่น การแจ้งเตือน หรือ Location เพื่อให้ฟีเจอร์ทำงานสมบูรณ์
5. ผู้ใช้สามารถระบุพื้นที่หรือห้องในบ้านได้เพื่อให้ฟีเจอร์ Automation และ Scheduling ทำงานได้ถูกต้อง

Assumptions ด้านระบบเครือข่าย (Network Assumptions)

1. บ้านของผู้ใช้มีสัญญาณ Wi-Fi ที่ครอบคลุมพื้นที่ติดตั้งอุปกรณ์ทั้งหมด
2. อินเทอร์เน็ตของผู้ใช้มีความเสถียรเพียงพอสำหรับการควบคุมอุปกรณ์และรับการแจ้งเตือนแบบ Real-time
3. หากอินเทอร์เน็ตไม่เสถียร ระบบ Local Control เช่น Zigbee หรือ Bluetooth จะยังสามารถใช้งานได้
4. ความเร็ว Upload ของผู้ใช้เพียงพอสำหรับการใช้งานกล้องวงจรปิดแบบ Real-time

Assumptions ด้านอุปกรณ์และฮาร์ดแวร์ (Hardware & Device Assumptions)

1. อุปกรณ์ IoT ต่าง ๆ รองรับโปรโตคอลที่ระบบกำหนด เช่น Wi-Fi, Zigbee, Bluetooth LE
2. Gateway หรือ Hub กลางสามารถทำงานได้ปกติ และมีเสถียรภาพเพียงพอต่อการประมวลผล
3. เซนเซอร์ที่ใช้พลังงานแบตเตอรี่จะได้รับการเปลี่ยนแบตเตอรี่ตามกำหนด
4. อุปกรณ์ทั้งหมดสามารถเชื่อมต่อและ Sync ข้อมูลกับระบบ Cloud ได้
5. อุปกรณ์ IoT รองรับการสื่อสารผ่าน API หรือโปรโตคอลที่ระบบกำหนดไว้

Assumptions ด้านข้อมูลและระบบ Cloud (Data & Cloud Assumptions)

1. ระบบ Cloud ที่ใช้ให้บริการมีความพร้อมตาม SLA (เช่น 99.9% uptime)
2. ระบบสามารถเก็บและ Sync ข้อมูลต่าง ๆ เช่น Log, Energy Consumption, Activity Records ได้ตามข้อกำหนด
3. พื้นที่จัดเก็บข้อมูลที่จัดสรรเพียงพอต่อการเก็บข้อมูล Log ตามระยะเวลาที่กำหนด
4. ระบบรองรับการเข้ารหัสข้อมูลทั้งขณะส่ง (in transit) และขณะพัก (at rest)

Assumptions ด้านบริการภายนอก (Third-party Integration Assumptions)

1. บริการของผู้ช่วยอัจฉริยะ เช่น Google Assistant, Siri, และ Alexa ยังให้บริการตามปกติ
2. Payment Gateway, Notification Service หรือระบบ API ต่าง ๆ ของ Third-party มีความเสถียรเพียงพอ
3. นโยบายหรือ API ของบริการภายนอกไม่เปลี่ยนแปลงในลักษณะที่ทำให้ระบบทำงานผิดพลาด

Assumptions ด้านการพัฒนาและการดูแลรักษา (Development & Maintenance Assumptions)

1. ทีมพัฒนาและทีมทดสอบมีความเชี่ยวชาญด้าน IoT, Cloud และ Mobile App เพียงพอ
2. สามารถเข้าถึงอุปกรณ์จริงหรือสถานที่ติดตั้งเพื่อทดสอบระบบได้
3. ระบบมีการบำรุงรักษาและอัปเดตซอฟต์แวร์อย่างสม่ำเสมอ
4. การจัดการเวอร์ชันของระบบและอุปกรณ์เป็นไปอย่างถูกต้องและเป็นระบบ

Assumptions ด้านการใช้งานจริง (Operational Assumptions)

1. บ้านหรือสถานที่ติดตั้งมีแหล่งจ่ายไฟที่เสถียรเพียงพอ
2. อุปกรณ์ตรวจจับภัยพิบัติ เช่น คิวหรือไฟไหม้ ได้รับการติดตั้งและบำรุงรักษาตามคำแนะนำ
3. ผู้ใช้ไม่แก้ไขหรือดัดแปลงฮาร์ดแวร์นอกกรอบแบบมาตรฐาน
4. การเชื่อมต่ออุปกรณ์ใหม่จะดำเนินการตามขั้นตอน Pairing ที่ออกแบบไว้
5. เมื่อระบบกลับมาออนไลน์หลังจาก Offline Mode จะ Sync ข้อมูลโดยอัตโนมัติ

ส่วนที่ 6: Priority & Trade-offs

Quality Attributes Priority

Rank	Quality Attribute	เหตุผล
1	Performance	ระบบ Smart Home ต้องตอบสนองคำสั่งควบคุมอุปกรณ์แบบทันที เช่น เปิดไฟ, ปิดประตู, แจ้งเตือนเหตุบุกรุก หากระบบมีความหน่วงสูงจะทำให้ประสบการณ์ใช้งานแย่และอาจเกิดความเสี่ยงด้านความปลอดภัย
2	Reliability	ระบบต้องทำงานได้อย่างถูกต้องและสม่ำเสมอ เช่น เซนเซอร์ต้องส่งข้อมูลได้ต่อเนื่อง และระบบต้องฟื้นตัวจากข้อผิดพลาดได้ เพื่อให้ Smart Home เชื่อถือได้ในสถานการณ์จริง เช่น ตรวจจับควันหรือการบุกรุก
3	Security	Smart Home เก็บข้อมูลอ่อนไหว เช่น พฤติกรรมการใช้งาน ภาพกล้อง และข้อมูลการเข้าออกบ้าน จึงต้องป้องกันการเข้าถึงโดยไม่ได้รับอนุญาต และจัดการสิทธิ์ผู้ใช้ให้ปลอดภัย
4	Usability	ผู้ใช้ทั่วไปอาจไม่มีความรู้ด้านเทคนิคสูง แอปพลิเคชันและกระบวนการเพิ่มอุปกรณ์ต้องใช้งานง่าย เพื่อให้ระบบ Smart Home ถูกใช้งานจริงในชีวิตประจำวัน และลดภาระผู้ใช้
5	Scalability	บ้านสมัยใหม่มีอุปกรณ์ Smart Home เพิ่มขึ้นเรื่อย ๆ เช่น ไฟ แอร์ ประตู เซนเซอร์หลายชนิด ระบบต้องรองรับจำนวนอุปกรณ์ที่เพิ่มขึ้นโดยไม่ลดประสิทธิภาพการทำงาน
6	Maintainability	ระบบ Smart Home ต้องมีการอัปเดตซอฟต์แวร์และ firmware อยู่เรื่อย ๆ รวมถึงการเพิ่มอุปกรณ์รุ่นใหม่ในอนาคต ดังนั้นระบบต้องออกแบบให้บำรุงรักษาและขยายฟีเจอร์ได้ง่าย ไม่กระทบส่วนอื่น

Trade-offs Analysis

Trade-off #1: Performance vs Security

Trade-off:

- Performance ต้องการความเร็วในการตอบสนองคำสั่ง (เช่น เปิดไฟภายใน $\leq 1-2$ วินาที)
- แต่ Security ต้องเข้ารหัสข้อมูล, ตรวจสอบตัวตน, ใช้ TLS/HTTPS ซึ่งเพิ่มภาระประมวลผลและทำให้ตอบสนองช้าลง

Analysis:

- หากเน้น Performance มากเกินไปจะลดความปลอดภัย
- หากเน้น Security มากเกินไปอาจทำให้การควบคุมอุปกรณ์หนึ่งและประสบการณ์ไม่ดี

Decision: เลือก Security และปรับปรุง Performance ผ่านการทำ Local Control / Caching

เพราะระบบ Smart Home เกี่ยวข้องกับความปลอดภัยในชีวิตและทรัพย์สิน จึงห้ามลดระดับ Security

Trade-off #2: Reliability vs Maintainability

Trade-off:

- Reliability ต้องการระบบทำงานคงที่และไม่เปลี่ยนแปลง
- Maintainability ต้องการปรับปรุงบ่อย, อัปเดตรุ่นใหม่ได้ง่าย, เพิ่มความสามารถ

Analysis:

- การอัปเดตบ่อย ๆ อาจทำให้ความเสถียรลดลง
- แต่ไม่อัปเดตก็ทำให้ระบบล้าสมัยและมีช่องโหว่

Decision: Maintainability พร้อมคง Reliability ผ่าน Zero-downtime Update

เพราะระบบมีโอกาขยายและต้องอัปเดต firmware เสมอ

Trade-off #3: Usability vs Security

Trade-off:

- Usability ต้องการระบบใช้งานง่าย เช่น Login อัตโนมัติ, ไม่ต้องยืนยันหลายขั้นตอน
- Security ต้องการ MFA, การตรวจสอบสิทธิ์, การยืนยันตัวตนที่รัดกุม

Analysis:

- เพิ่มความปลอดภัยมากไป จะทำให้ผู้ใช้รู้สึกยุ่งยาก
- ลดความปลอดภัย จะเพิ่มความเสี่ยงบุกรุกบ้าน/ข้อมูล

Decision: เลือก แบบสมดุล โดยใช้

- Face ID / Fingerprint สำหรับ MFA แบบสะดวก
 - การกำหนดสิทธิ์แบบ Role-based (RBAC)
- เพราะต้องรักษาความปลอดภัยสูงสุด พร้อมให้ผู้ใช้รู้สึกสะดวก

Trade-off #4: Scalability vs Performance

Trade-off:

- Scalability ต้องเพิ่มบริการ Clustering, Load Balancer, Message Broker
- แต่ขั้นตอนนี้อาจเพิ่ม latency เล็กน้อยและทำให้ระบบซับซ้อนขึ้น

Analysis:

- รองรับอุปกรณ์หลายร้อยตัวต้องใช้ระบบที่ซับซ้อนกว่าระบบบ้านทั่วไป
- หากลด Scalability ระบบจะขยายไม่ได้ในอนาคต

Decision: เลือก Scalability เป็นหลัก เพราะ Smart Home แนวใหม่เพิ่มอุปกรณ์อย่างรวดเร็ว

Performance จะปรับโดยใช้ Local Processing และ Lightweight Protocol (MQTT)

Trade-off #5: Performance vs Maintainability

Trade-off:

- Maintainability ต้องการโค้ดอ่านง่าย, แยกบริการ, ใช้ microservice
- แต่ microservice เพิ่ม Overhead → อาจช้ากว่า monolith
- Performance ต้องการเรียกบริการให้น้อยที่สุด

Analysis:

- ระบบ Smart Home มี event มากมาย (sensor, automation, alert)
- ระบบขนาดใหญ่ควร maintain ง่ายกว่า maintain แบบ monolithic

Decision: เลือก Maintainability ผ่าน Modular Architecture

เพราะการเพิ่มอุปกรณ์/ฟีเจอร์เกิดขึ้นบ่อยกว่าการเพิ่ม Performance ที่ละจุด

Trade-off #6: Reliability vs Scalability

trade-off:

- Reliability ต้องการระบบคงที่ ไม่แก้ไขบ่อย
- Scalability ต้องเพิ่ม node/databases ซึ่งทำให้ระบบซับซ้อนและเสี่ยงต่อความผิดพลาดเพิ่มขึ้น

Analysis:

- การขยายระบบต้องวางแผนดี ไม่เช่นนั้น reliability จะลดลง

Decision: เลือก Scalability ควบคู่กับระบบตรวจสอบ (Monitoring + Auto-recovery)**

เพราะอุปกรณ์ Smart Home มีจำนวนเพิ่มขึ้นตามเวลา หากไม่รองรับ Scalability ตั้งแต่แรกจะเป็นปัญหาใหญ่