CS350 Safehome Project Software Requirement Specification

Team 1

20190813 Minjun Kim
20210145 Jehyung Kim
20210482 Onejune Lee
20210484 Yunje Lee

Table of Contents

I. Overview

1. Introduction

This document specifies the software requirements for '**SafeHome**', an integrated home automation system designed to provide comprehensive security and surveillance capabilities for residential properties. 'Safehome' is an intelligent platform engineered to protect a user's home from both physical threats and environmental hazards while maximizing daily convenience. The ultimate objective of this system is to provide users with comprehensive security monitoring, convenience, and peace of mind by empowering them with the ability to fully monitor and control their home's status from anywhere, at any time.

The need for effective home security solutions has grown as property owners increasingly require flexible, accessible methods to protect their homes while away. The concept of residential security has expanded beyond simple intrusion detection to encompass a broader scope, including the prevention of environmental disasters like fires and gas leaks, as well as the management of the indoor living environment. In response to these evolving needs, 'Safehome' establishes a single, cohesive ecosystem that seamlessly integrates intelligent security, live surveillance, system and user management, remote access, and indoor environmental monitoring.

To achieve this, the system is composed of a suite of high-sensitivity sensors (detecting motion, door/window entry, environmental hazards, etc.), high-definition IP cameras, a central control hub, and an intuitive mobile and web application for user interaction. Through this application, users can monitor their home in real-time, receive instant notifications in the event of an emergency, effortlessly switch between security modes, and efficiently manage all devices connected to the system.

This Software Requirements Specification (SRS) document defines all functional and non-functional requirements for the SafeHome system's first increment, which focuses on security and surveillance functions. It will serve as the foundational reference for all teams involved in the project—including planning, design, development, and quality assurance (QA)—providing clear direction for development and acting as the definitive standard to ensure the final product meets and exceeds user expectations.

2. Goal

The ultimate goal of the 'Safehome' system development is to leverage technology to provide comprehensive home security through integrated monitoring and automated response capabilities. To achieve this vision, the system is driven by the following core objectives:

Establish a Proactive and Comprehensive Security Framework: To move beyond reactive, post-incident responses, aiming instead to proactively predict and defend against potential threats. This means providing a multi-layered shield of protection against all threat types, from physical intrusions to environmental hazards. We will achieve this by integrating various sensors for real-time threat detection, minimizing false positives through AI-driven analytics, and establishing a swift, automated response framework that includes user notifications and automatic dispatch to emergency services.

Provide a Seamless and Intuitive User Experience: We prioritize usability through an intuitive mobile-first interface that enables users of varying technical skill levels to monitor and control their system effectively. We provide clear status indicators, straightforward controls, and guided setup processes. This will be accomplished through a unified, mobile-first interface, context-aware automation based on user patterns and location, and an accessible design that empowers anyone to manage their security easily, thereby providing confidence in home security.

Ensure High System Reliability and Data Security: To uphold 'trust' as the core value of our security system, we make it our highest priority to guarantee 24/7/365 operational stability and protect user data with the highest level of security. We will ensure system integrity through real-time health monitoring of all hardware components. Furthermore, all user data, including video streams, will be protected with robust end-to-end encryption (E2EE) both in transit and at rest, while access to the system will be strictly governed by mechanisms like Two-Factor Authentication (2FA) and Role-Based Access Control (RBAC).

Expand into a Healthy and Smart Living Environment: To evolve beyond a mere security system into a smart home platform that actively enhances the user's quality of life. Our objective is to support a healthier and more efficient lifestyle, with security as its foundation. The system will monitor indoor air quality (IAQ) and real-time energy consumption, automatically controlling connected devices to maintain a comfortable environment and promote energy conservation. Ultimately, we aim to foster an open ecosystem, enabling 'Safehome' to serve as the central hub for a truly unified smart home experience.

3. Major Functionalities

The 'Safehome' system is composed of five core functional groups that work together organically to achieve the defined goals. Each function is designed with a focus on the user's security, convenience, and control.

Intelligent Security: As the core function of the system, this actively detects and responds to all potential threats surrounding the home. This

includes physical intrusion detection for monitoring doors and windows, and multi-sensor monitoring for environmental hazard detection, such as fires or gas leaks. It also features an automated incident management system that sends instant notifications to the user upon threat detection and, if necessary, automatically contacts emergency services. Users have flexible control, including the ability to set one-touch security modes (e.g., Away, Home, Sleep) and temporarily bypass specific sensors based on their situation.

Live Surveillance: This function enables users to visually and audibly check their home's status directly from anywhere, at any time. It allows for real-time video streaming of the home's interior and exterior through high-definition cameras and supports two-way audio communication via built-in speakers and microphones. All footage is securely recorded to the cloud or local storage, and users can easily search and play back past recordings when needed. The system also provides a simple way to export and share important video clips for evidentiary purposes.

System and User Management: These are the administrative functions that maintain the stability and integrity of the entire 'Safehome' ecosystem. Users can easily add and place new devices, like sensors and cameras, into the system, and all device firmware is kept up-to-date through automatic over-the-air (OTA) updates. The system provides a status dashboard and activity timeline for an at-a-glance overview of system health, battery levels, and connectivity issues. Furthermore, it allows for systematic and secure user management by assigning different role-based access permissions (e.g., administrator, standard user) to family members or guests.

Remote Access and Account: This foundational function ensures that users can securely and reliably access the 'SafeHome' system while away from home. It provides a secure user account system protected by strong security policies, including Two-Factor Authentication (2FA), to prevent unauthorized access. Users can log in from anywhere via the mobile app or a web browser to control all functions and can securely manage their personal information, such as passwords and profile details.

Indoor Monitoring and Device Control: This is an expanded feature set that goes beyond security to create a more comfortable and efficient living environment. The system monitors indoor air quality (IAQ) and real-time power consumption, providing valuable insights to the user. Based on this data, it can automatically control connected indoor smart devices, such as ventilation systems or smart plugs, to consistently maintain a pleasant environment and help conserve energy.

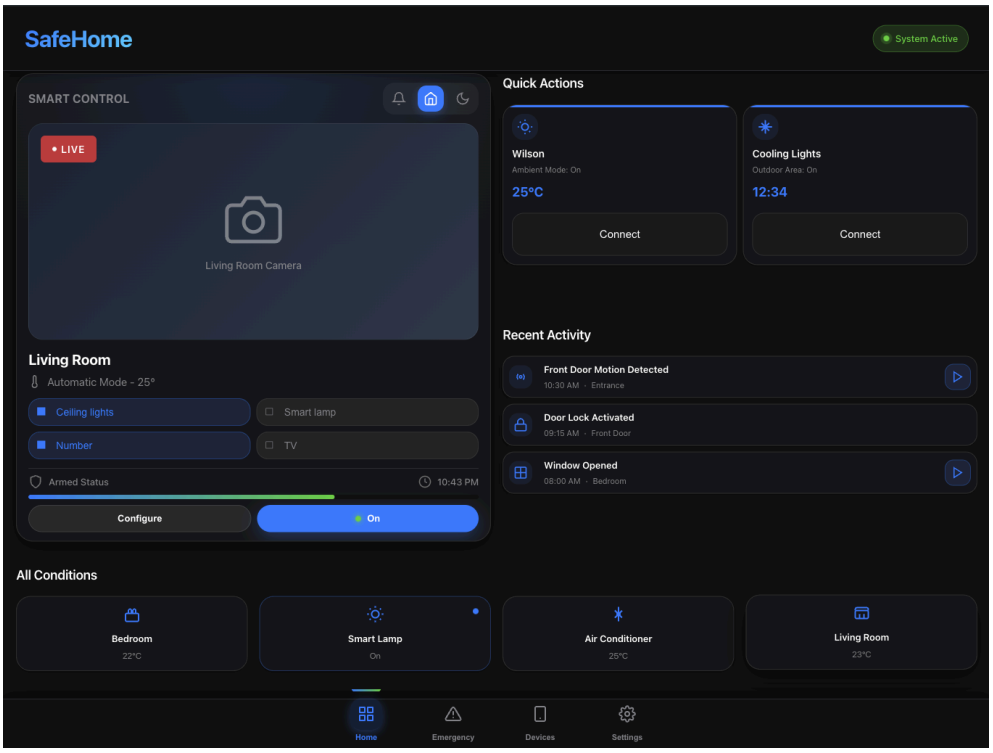
II. Project Schedule

This project will follow the concept of an incremental software development model. The first increment will focus on the core security and surveillance functions of the Safehome product. For this first release, we will also include extended accessibility through a mobile app, along with basic lighting control capabilities. Other functions will be developed in subsequent increments.

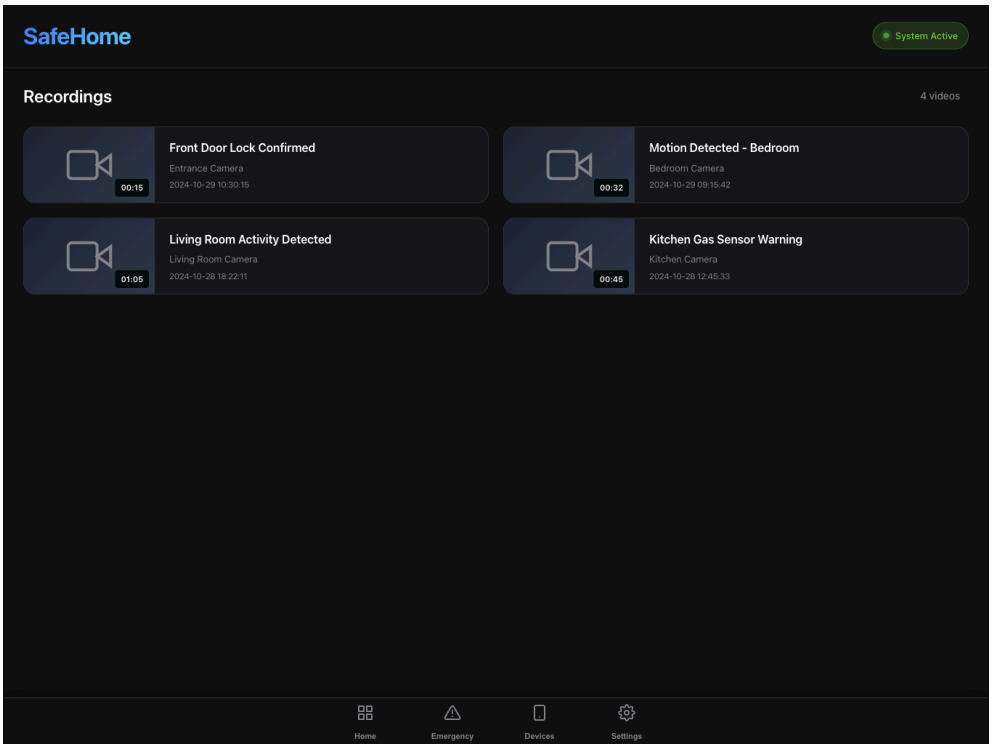
Phase	Period
Beginning of the project	Oct 20, 2025
Initial requirement gathering	Oct 20 – Oct 23, 2025
Requirement analysis and modeling	Oct 24 – Oct 27, 2025
Finalizing requirement specification	Oct 28 – Oct 31, 2025
Planning and creating analysis model	Nov 1 – Nov 4, 2025
Creating system design model	Nov 5 – Nov 11, 2025
Design review and refinement	Nov 12 – Nov 14, 2025
Module implementation (Security, Device, Camera)	Nov 15 – Nov 26, 2025
Integration and unit testing	Nov 27 – Nov 30, 2025
Debugging and internal verification	Dec 1, 2025
Peer testing and feedback collection	Dec 2 – Dec 8, 2025
Bug fixing and optimization	Dec 9 – Dec 15, 2025
Preparing final demo and documentation	Dec 16 – Dec 19, 2025
Final project deployment	Dec 20, 2025

III. Prototype UI/UX

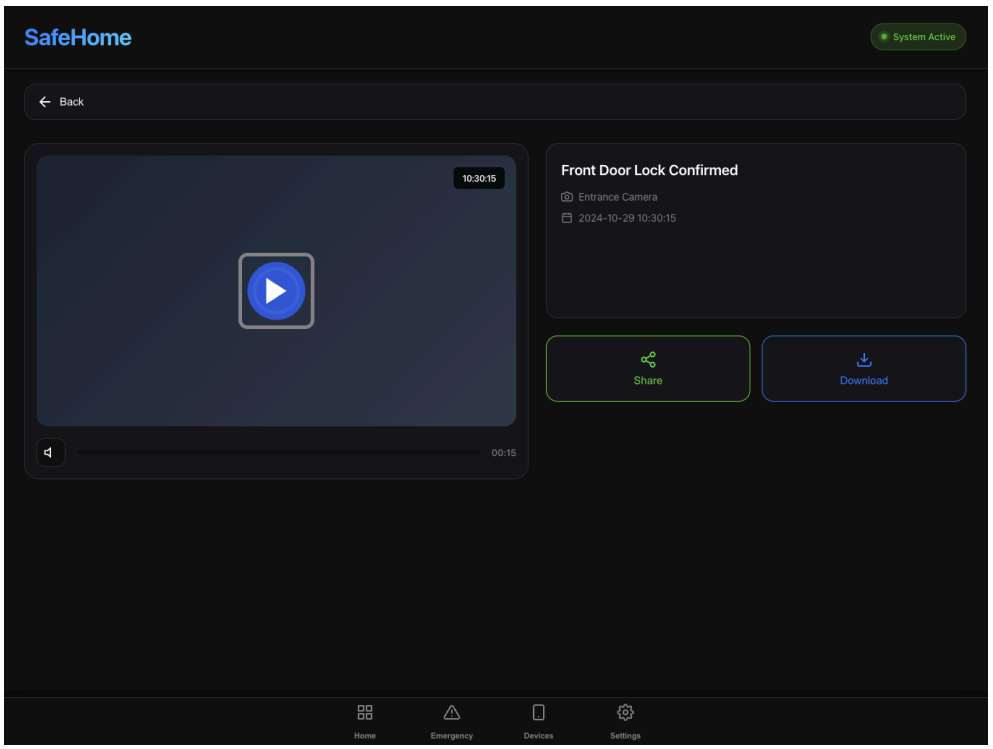
1. SafeHome Hub



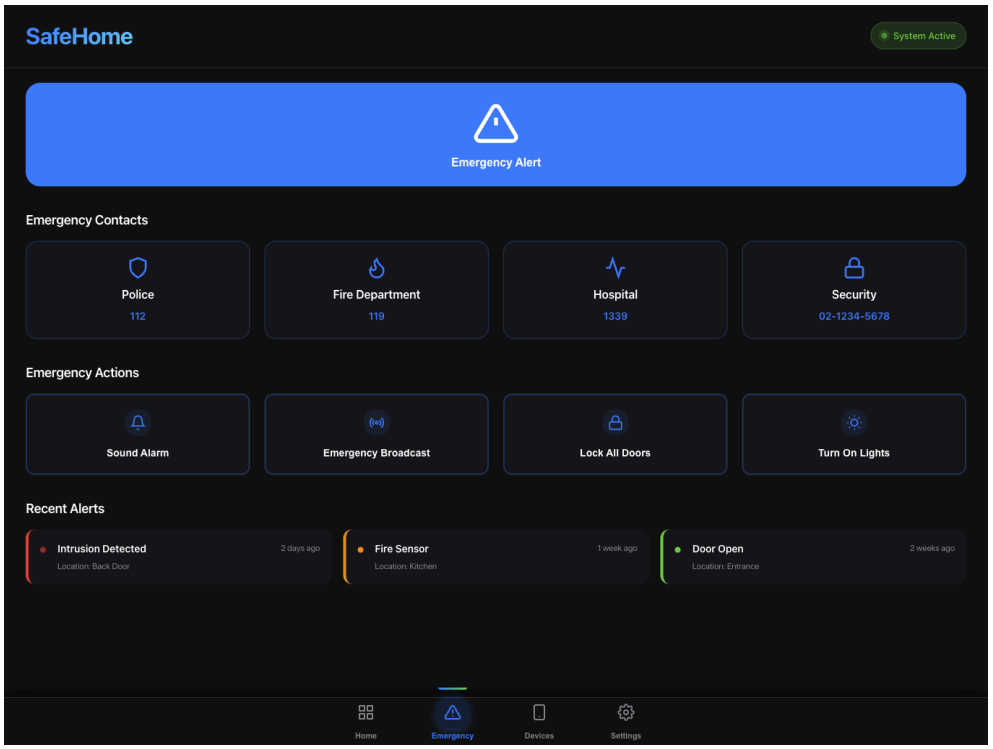
<Fig 1.Dashboard>



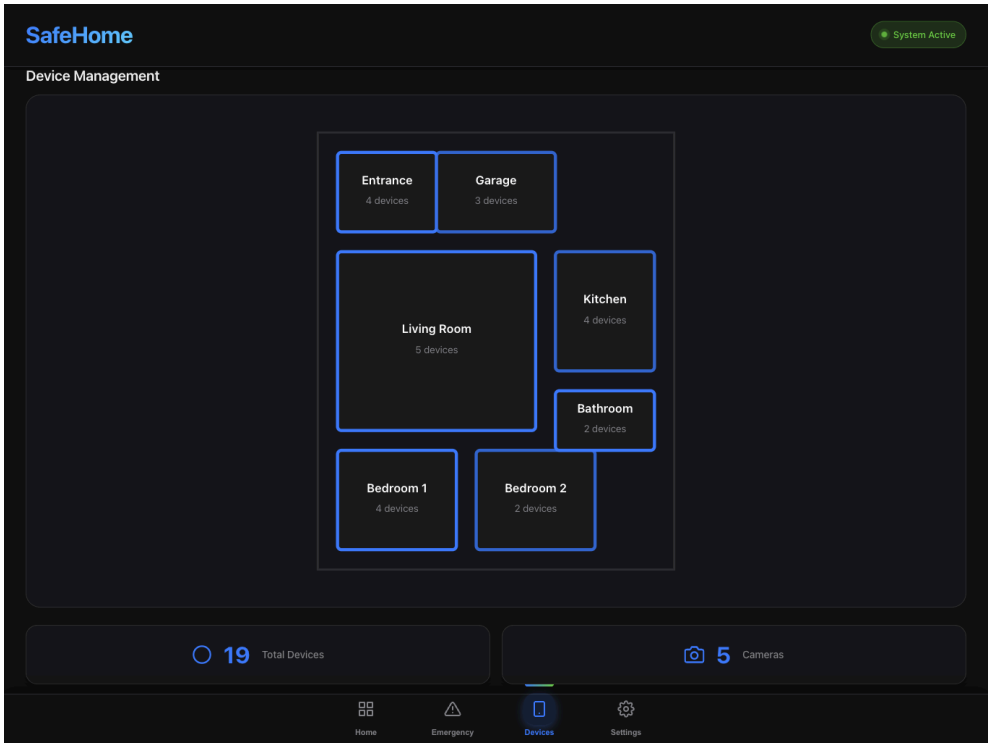
<Fig 2.Recordings>



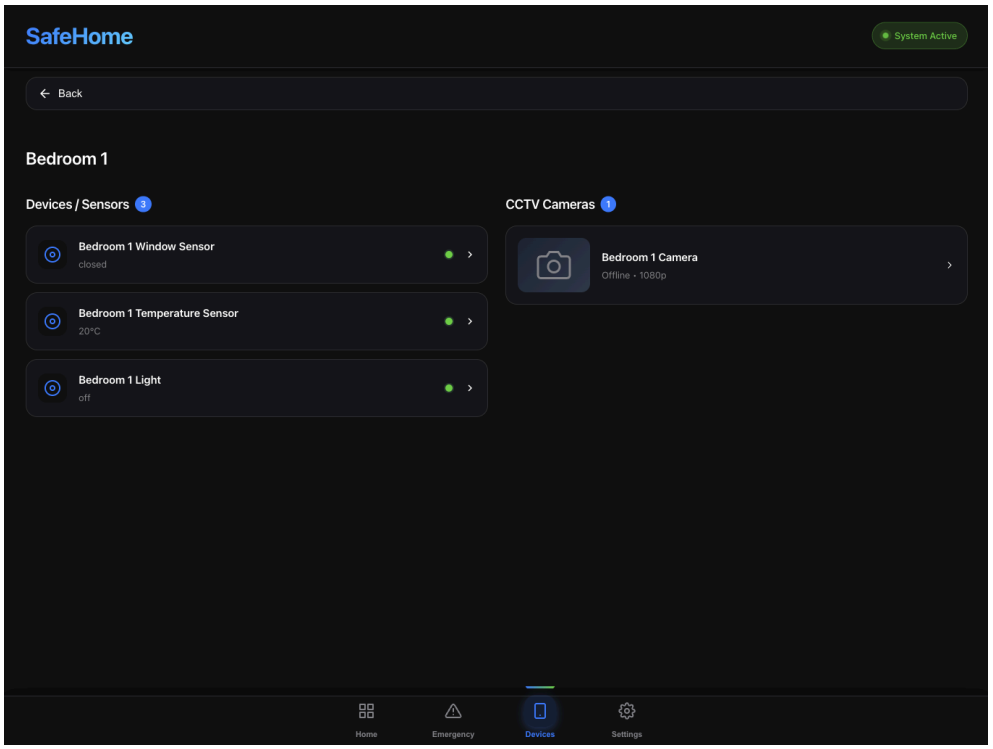
<Fig 3.Recording Detail>



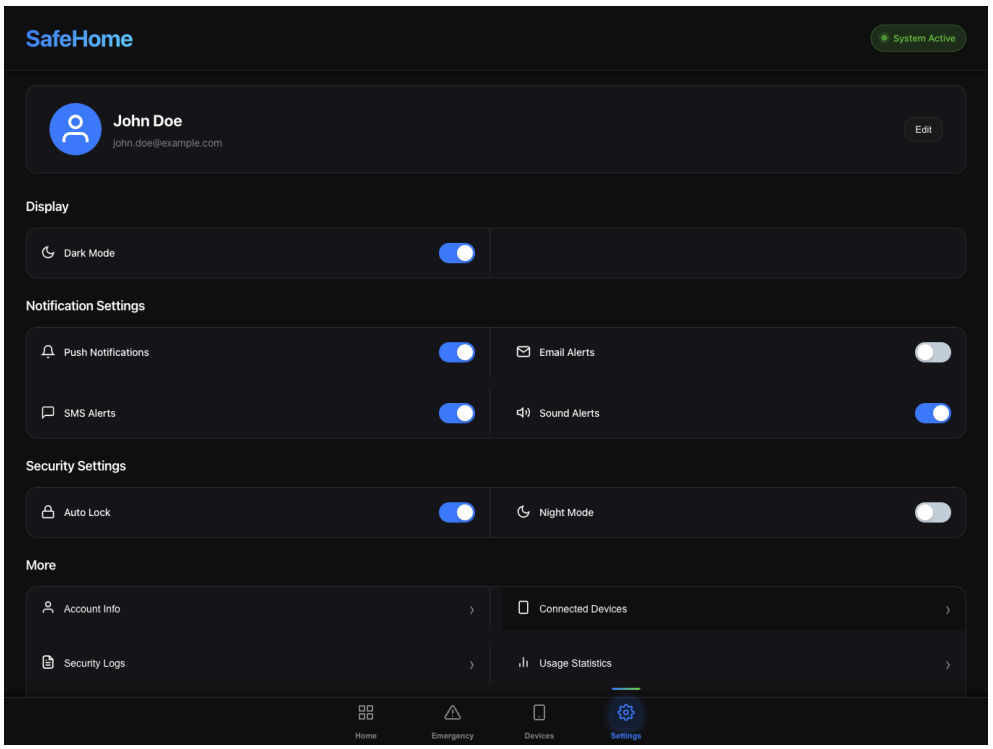
<Fig 4.Emergency>



<Fig 5.Devices>

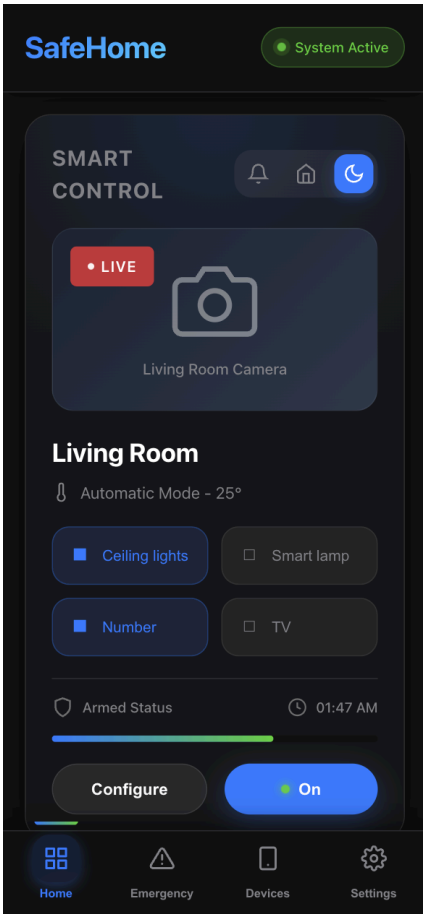


<Fig 6.Device Detail>

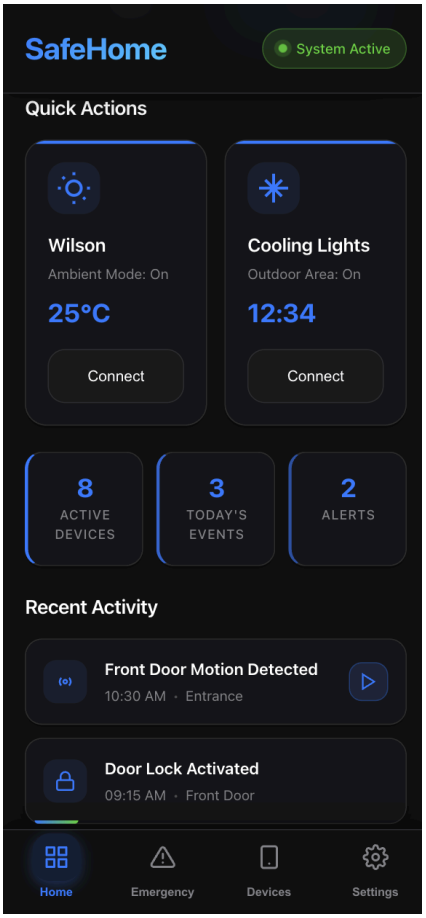


<Fig 7.Preferences and Settings>

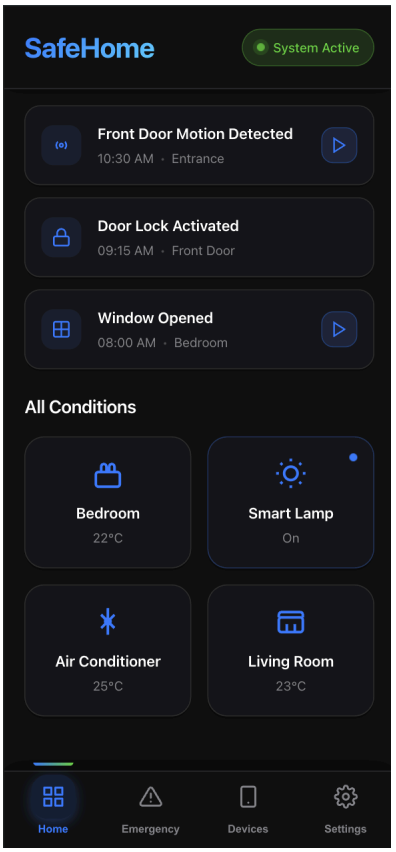
2. Mobile Application



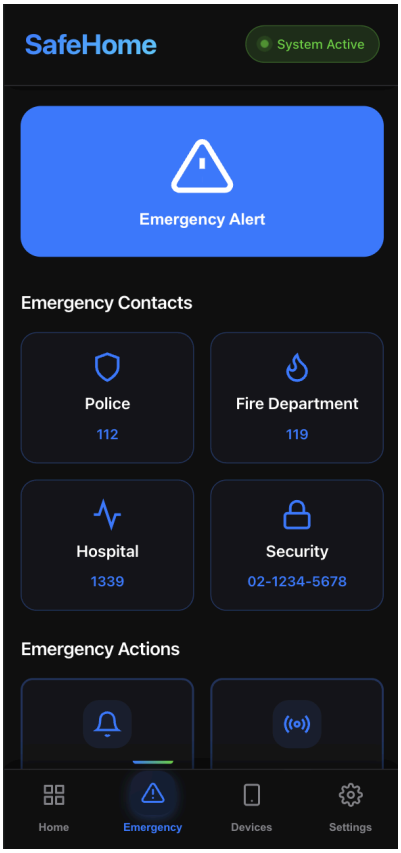
<Fig 8. Mobile Dashboard 1>



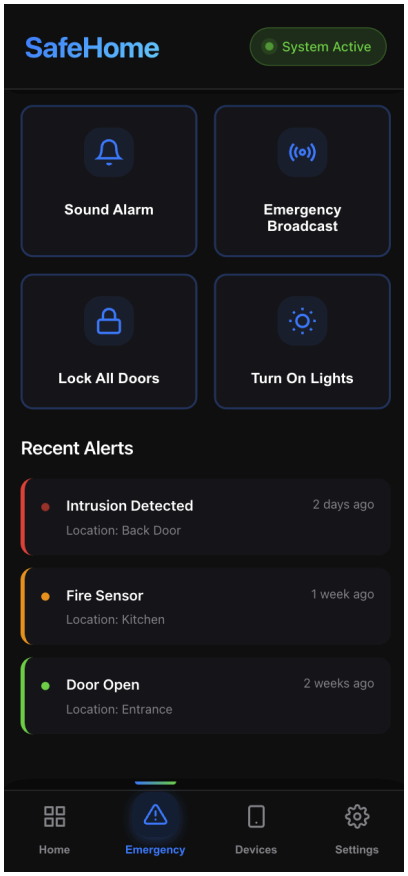
<Fig 9. Mobile Dashborad 2>



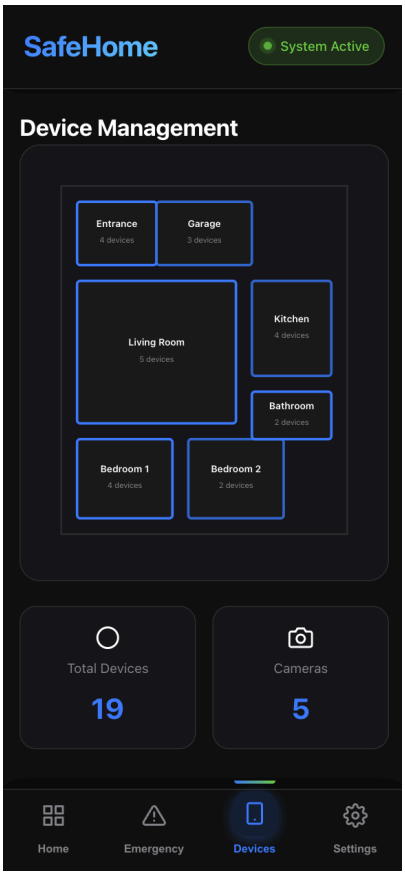
<Fig 10. Mobile Dashboard>



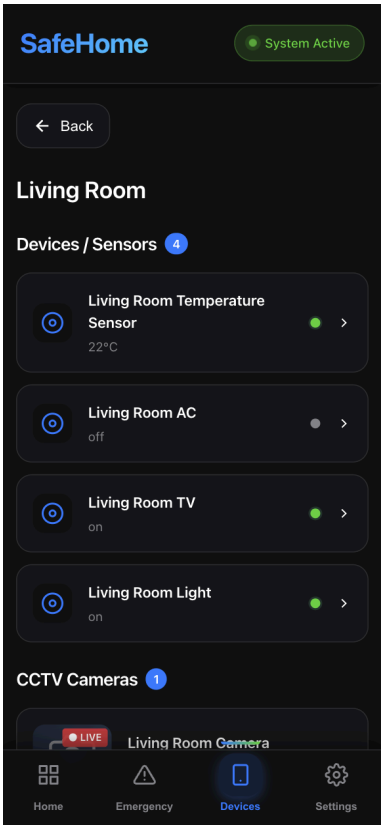
<Fig 11. Emergency Tab 1>



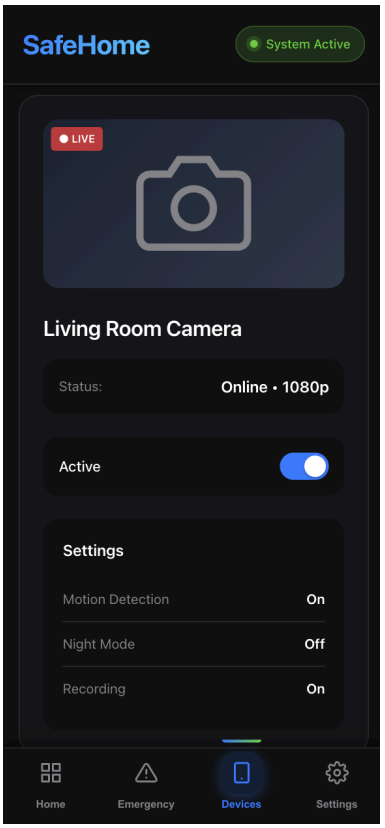
<Fig 12. Emergency Tab 2>



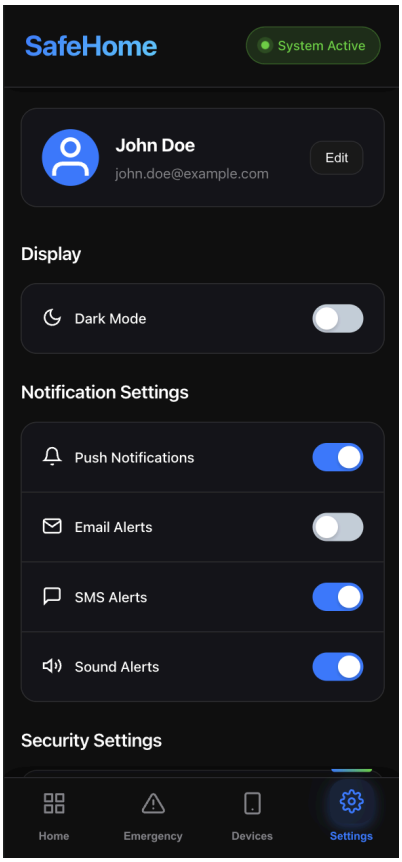
<Fig 13. Device management>



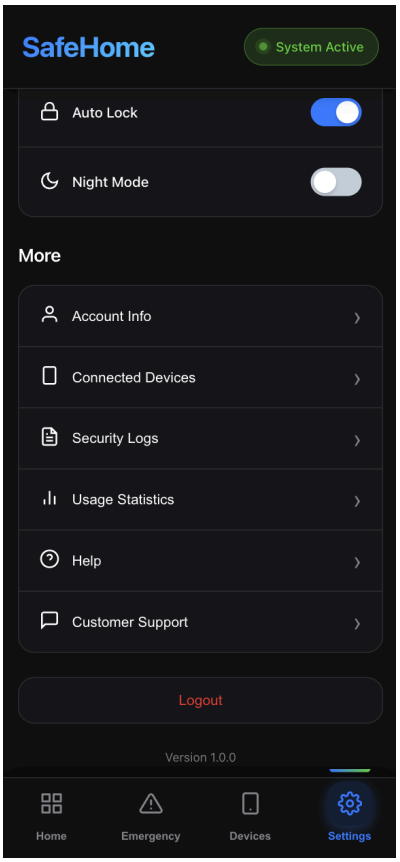
<Fig 14. Device Setting>



<Fig 15. Camera Live View>



<Fig 16. Settings 1>



<Fig 17. Settings 2>

IV. Assumptions

1. Pet Sensor Function(1.1.4) is defined in safehome dialog slide 58-59
2. Alarm Trigger and Instant Notification Function(1.2.1) is defined in safehome dialog slide 5
3. Emergency Service Integration and Auto Call Function(1.2.3) is defined in safehome dialog slide 6, 7
4. One-Touch Modes (Away, Home, Sleep) Function(1.3.1) is defined in safehome dialog slide 9
5. Sensor Bypass Function(1.3.2) is defined in safehome dialog slide 10
6. Single Camera Live View Function(2.1.1) is defined in safehome dialog slide 29-31
7. Two-Way Audio Function(2.1.2) is defined in safehome dialog slide 16
8. Camera Lock and Unlock Function(2.1.3) is defined in safehome dialog slide 19-31
9. Search and Playback Recordings Function(2.2.1) is defined in safehome dialog slide 29-31
10. Recording Settings Function(2.3.1) is defined in safehome dialog slide 29-31
11. Add and Place New Devices Function(3.1.1) is defined in safehome dialog slide 58
12. Activity Logs and Timeline Function(3.2.2) is defined in safehome dialog slide 39
13. User Role and Access Control Function(3.3.1) is defined in safehome dialog slide 70
14. Sign Up Function(4.1.1) is defined in safehome dialog slide 41
15. Log In Function(4.1.2) is defined in safehome dialog slide 42
16. Log Out Function(4.1.3) is defined in safehome dialog slide 43
17. Password Recovery and Reset Function(4.1.4) is defined in safehome dialog slide 44
18. Edit Profile Information Function(4.1.5) is defined in safehome dialog slide 45
19. Change Password Function(4.1.6) is defined in safehome dialog slide 46
20. Two-Factor Authentication Management Function(4.1.7) is defined in safehome dialog slide 47
21. Indoor Device Control Function(5.1.1) is defined in safehome dialog slide 39
22. Indoor Air Quality Monitoring and Ventilation Integration Function(5.2.1) is defined in safehome dialog slide 27
23. Real-Time Power Consumption Monitoring and Reporting Function(5.2.2) is defined in safehome dialog slide 29
24. Secure Onboarding (Device Registration Security) Function (Originally 3.1.2) was added because we determined it is necessary for improving the overall quality of

the service. However, we have assumed that it should be developed in the next iteration, as the current SRS is for the first release.

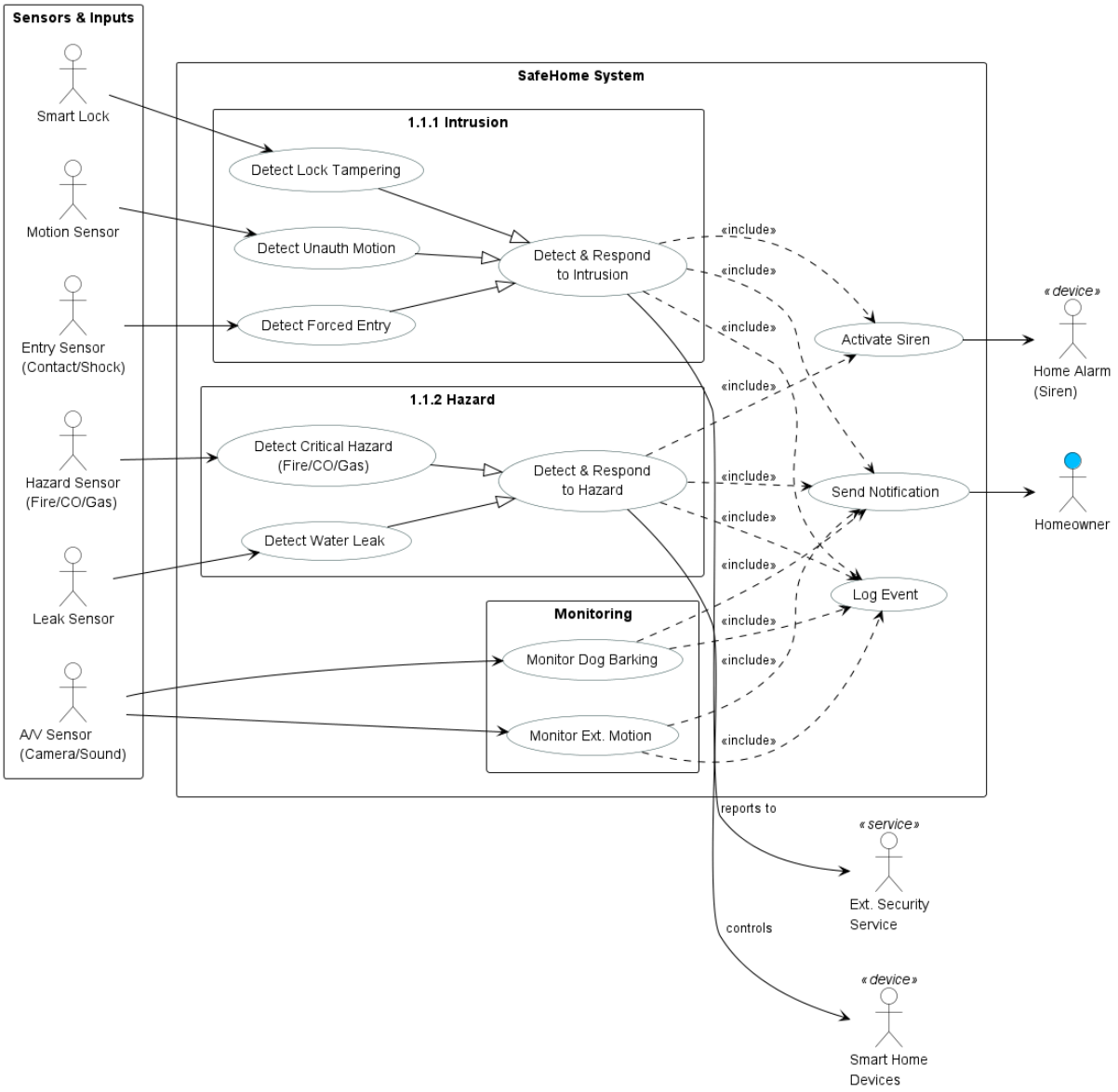
25. OTA Firmware Update Function (Originally 3.1.3) was added because we determined it is necessary for improving the overall quality of the service. However, we have assumed that it should be developed in the next iteration, as the current SRS is for the first release.
26. Health & Tamper Monitoring (Device Anomaly Detection) Function (Originally 3.2.3) was added because we determined it is necessary for improving the overall quality of the service. However, we have assumed that it should be developed in the next iteration, as the current SRS is for the first release.
27. Global Priority & Version Policy Function (Originally 3.3.2) was added because we determined it is necessary for improving the overall quality of the service. However, we have assumed that it should be developed in the next iteration, as the current SRS is for the first release.
28. Policy & Compliance Function (Originally 3.4) was added because we determined it is necessary for improving the overall quality of the service. However, we have assumed that it should be developed in the next iteration, as the current SRS is for the first release.
29. Data Retention & Deletion Policy Function (Originally 3.4.1) was added because we determined it is necessary for improving the overall quality of the service. However, we have assumed that it should be developed in the next iteration, as the current SRS is for the first release.
30. Encryption/Transmission Security Policy Function (Originally 3.4.2) was added because we determined it is necessary for improving the overall quality of the service. However, we have assumed that it should be developed in the next iteration, as the current SRS is for the first release.
31. Privacy Notice & Consent Function (Originally 3.4.3) was added because we determined it is necessary for improving the overall quality of the service. However, we have assumed that it should be developed in the next iteration, as the current SRS is for the first release.
32. Time Synchronization (NTP) Policy Function (Originally 3.4.4) was added because we determined it is necessary for improving the overall quality of the service. However, we have assumed that it should be developed in the next iteration, as the current SRS is for the first release.
33. It was decided to add the Physical Intrusion Detection Function(1.1.1) in the meeting on 10.29 after discussion in the meeting on 10.26.
34. It was decided to add the Environmental Hazard Detection Function(1.1.2) in the meeting on 10.29 after discussion in the meeting on 10.26.
35. It was decided to add the Outdoor Motion Detection Function(1.1.3) in the meeting on 10.29 after discussion in the meeting on 10.26.

36. It was decided to add the Alarm Verification Step Function(1.2.2) in the meeting on 10.29.
37. It was decided to add the Panic Button Function(1.2.4) in the meeting on 10.29.
38. It was decided to add the Sensor Activation and Deactivation Function(1.3.3) in the meeting on 10.29 after discussion in the meeting on 10.26.
39. It was decided to add the Camera Activation and Deactivation Function(2.1.4) in the meeting on 10.29 after discussion in the meeting on 10.26.
40. It was decided to add the Evidence Sharing and Export Function(2.2.2) in the meeting on 10.29.
41. It was decided to add the Notification Policy and Cooldown Function(2.3.2) in the meeting on 10.29.
42. It was decided to add the System Status Dashboard Function(3.2.1) in the meeting on 10.29.
43. Floor plan configuration and hardware deployment is complete and out of the scope of our project.
44. “System administrator” in our use case scenarios is not a person who is in charge of managing the system. It is the system itself acting as a facilitator for the use of system functionalities.
45. Between mobile and web, we have decided to make the mobile app our first release.

V. Use Case Diagram

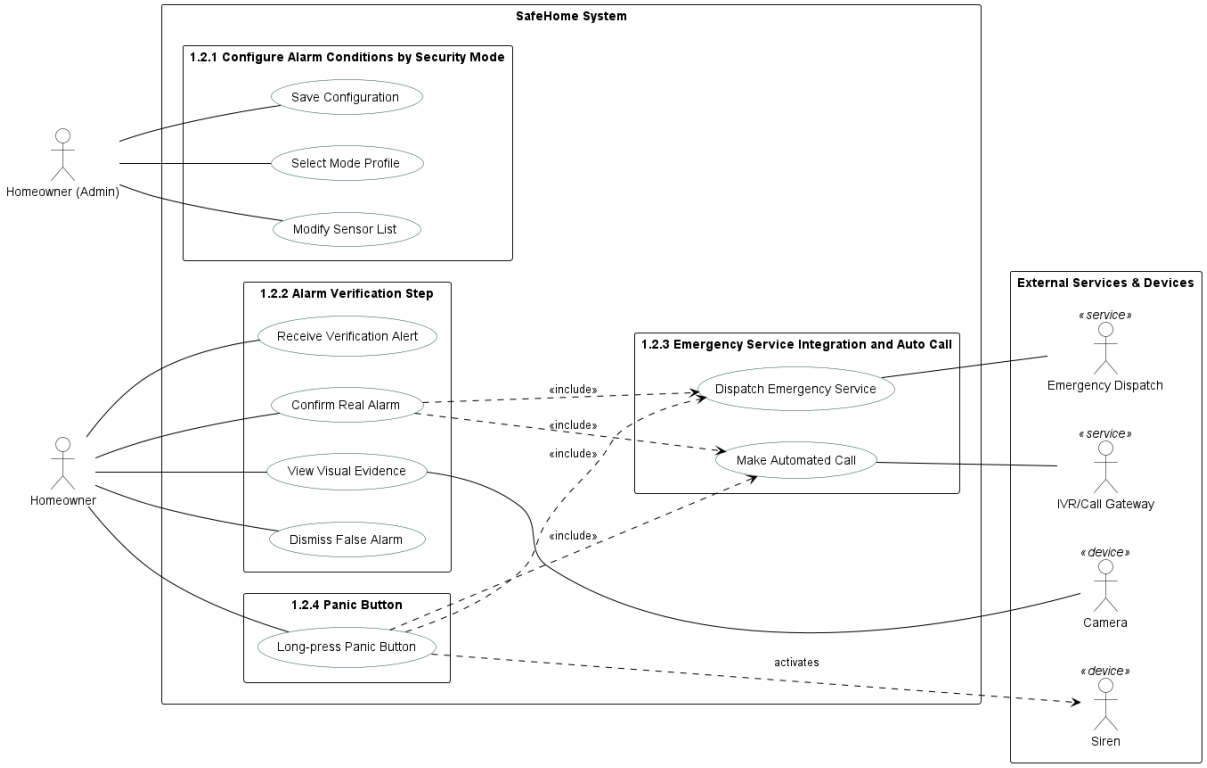
1. Intelligent Security

1.1. Sensor Monitoring - Use Case



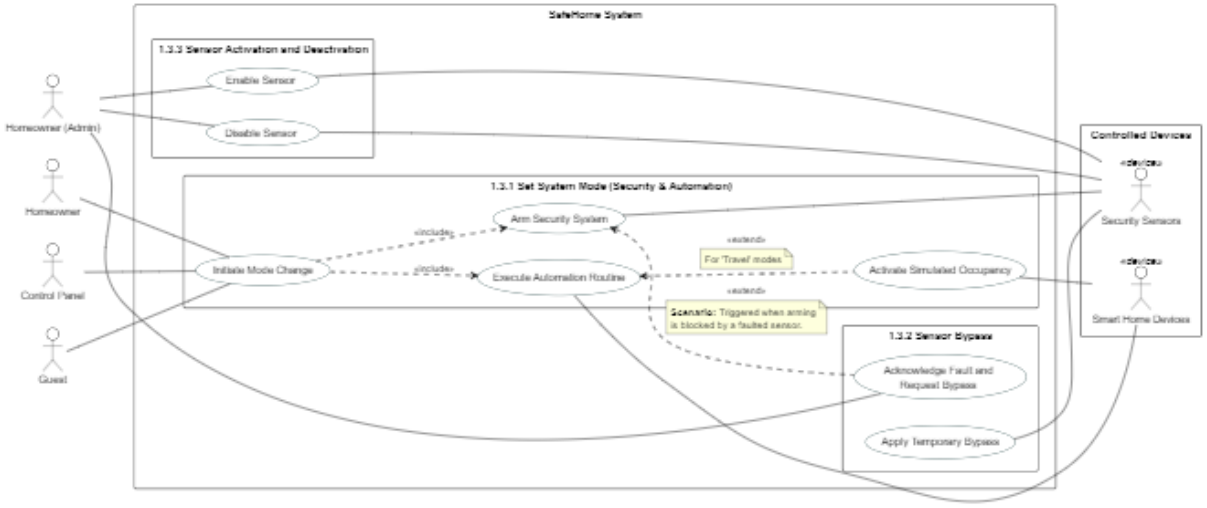
References on SEPA safehome dialog slide 58-59, 2025.10.26 / 2025.10.29 Meeting

1.2. Incident Management - Use Case



References on SEPA safehome dialog slide 5, 6, 7, 2025.10.29 Meeting

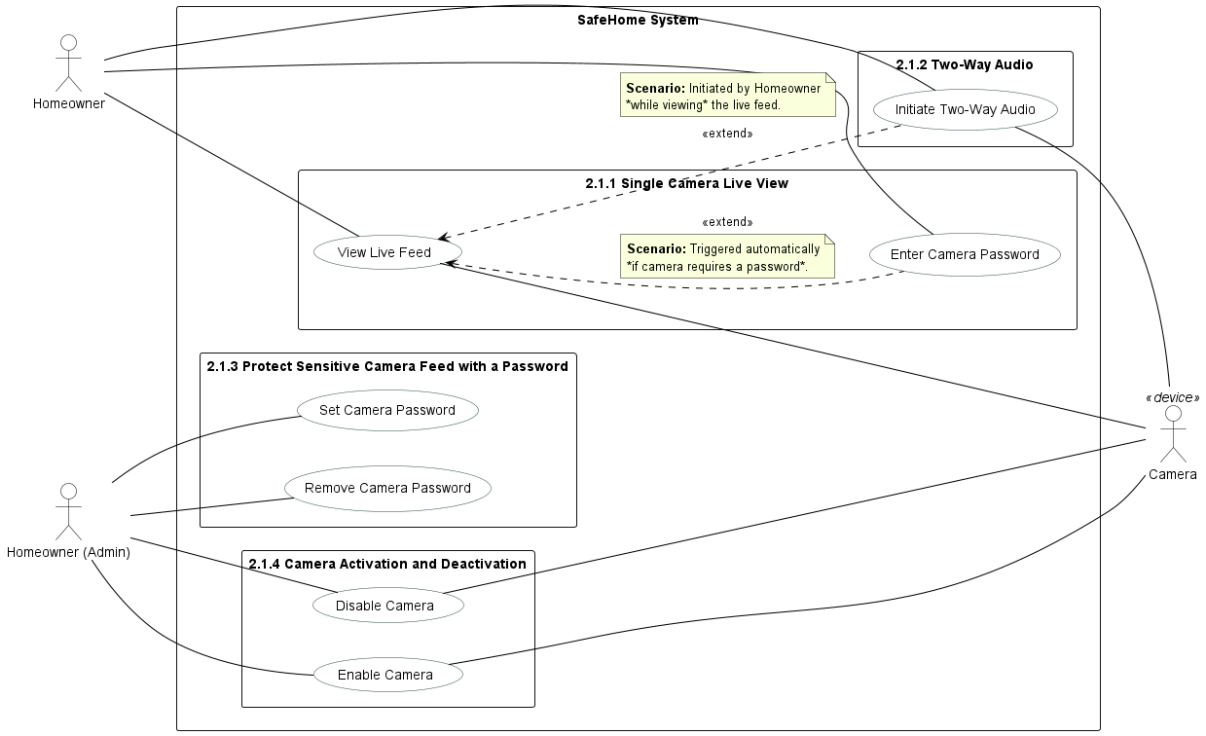
1.3 Security Mode Control - Use Case



References on SEPA safehome dialog slide 9, 10, 2025.10.26 / 2025.10.29 Meeting

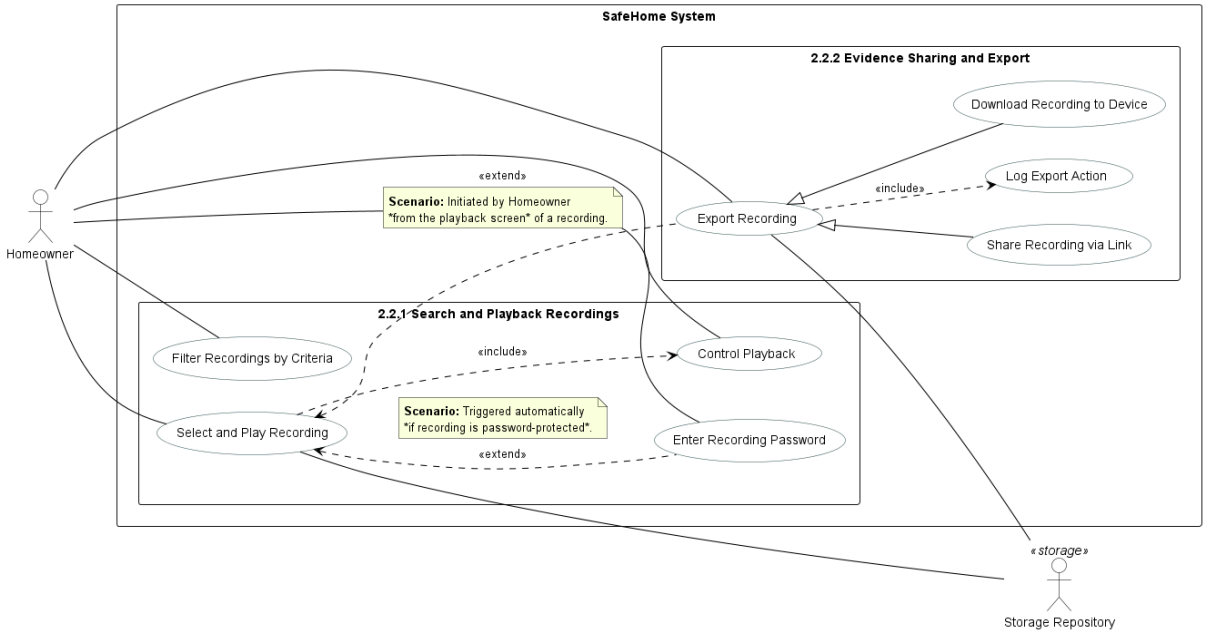
2. Live Surveillance

2.1 Camera Viewing and Control - Use Case



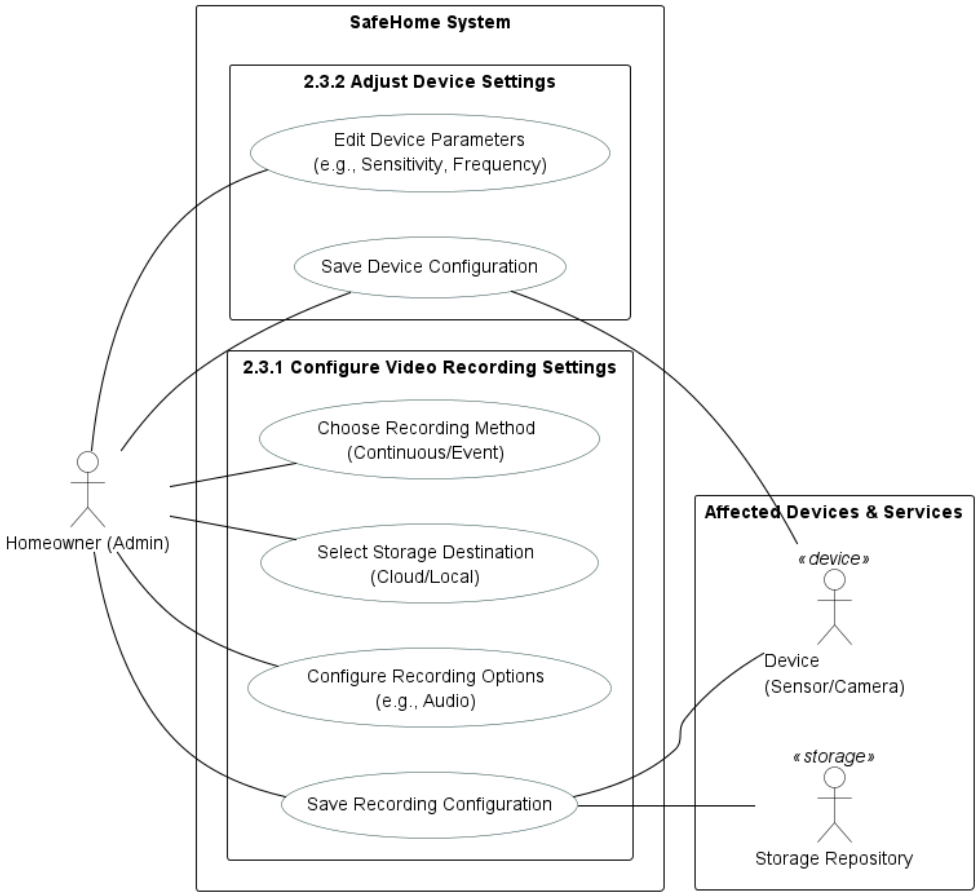
References on SEPA safehome dialog slide 16, 19-31, 29-31, 2025.10.26 / 2025.10.29 Meeting

2.2 Recording and Evidence Management - Use Case



References on SEPA safehome dialog slide 29-31, 2025.10.29 Meeting

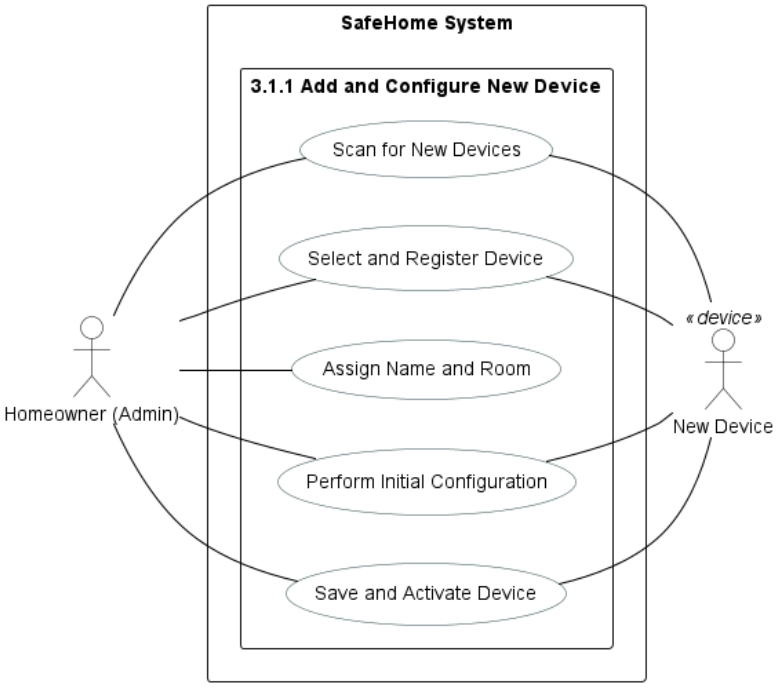
2.3 Surveillance Settings - Use Case



References on SEPA safehome dialog slide 29-31, 2025.10.29 Meeting

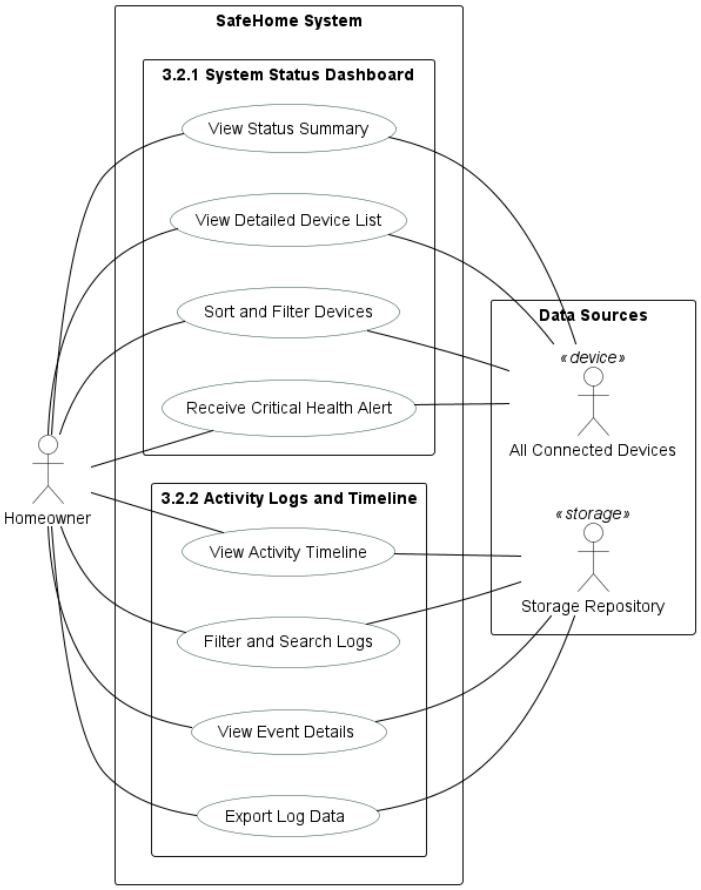
3. System and User Management

3.1 Device Management - Use Case



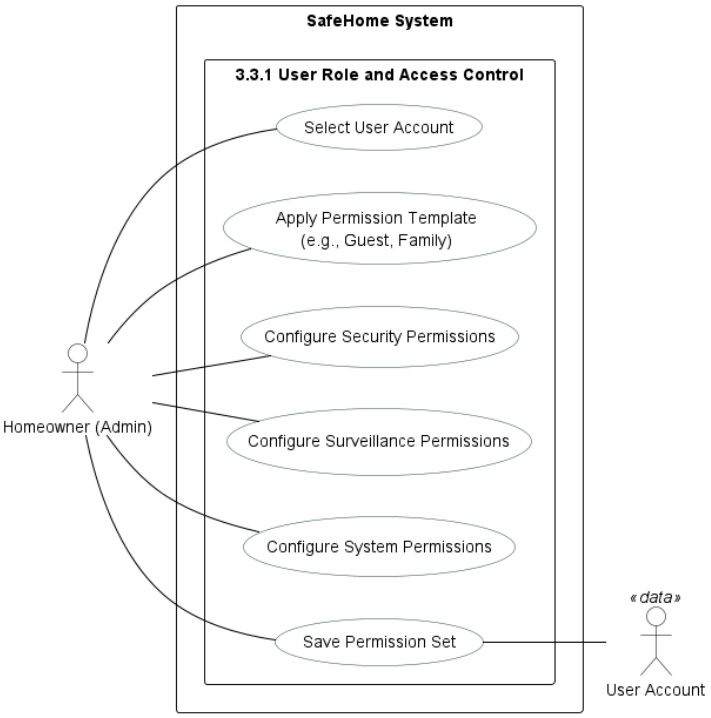
References on SEPA safehome dialog slide 58

3.2 System Status and Logs - Use Case



References on SEPA safehome dialog slide 39, 2025.10.29 Meeting

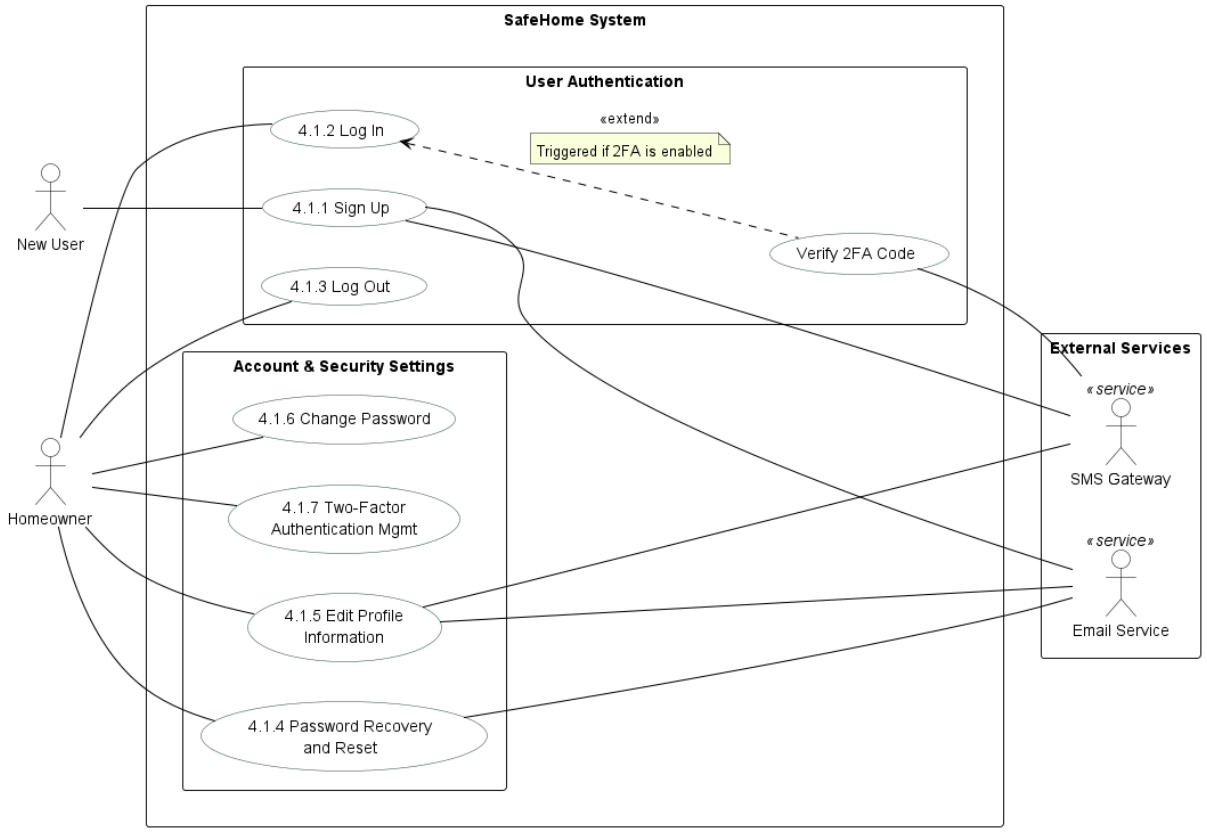
3.3 User and Permission Management - Use Case



References on SEPA safehome dialog slide 70

4. Remote Access and Account

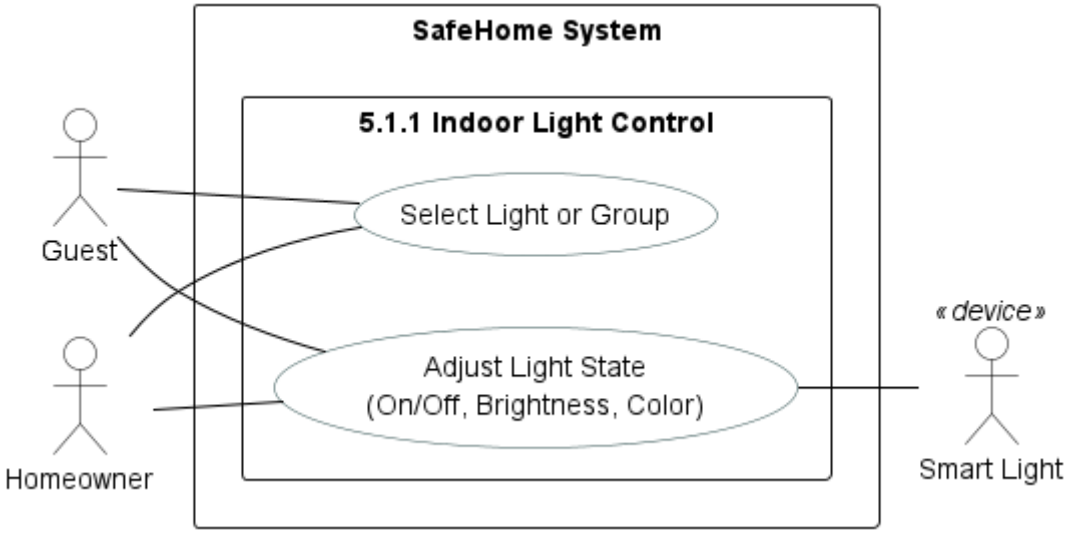
4.1 Account Management - Use Case



References on SEPA safehome dialog slide 41, 42, 43, 44, 45, 46, 47

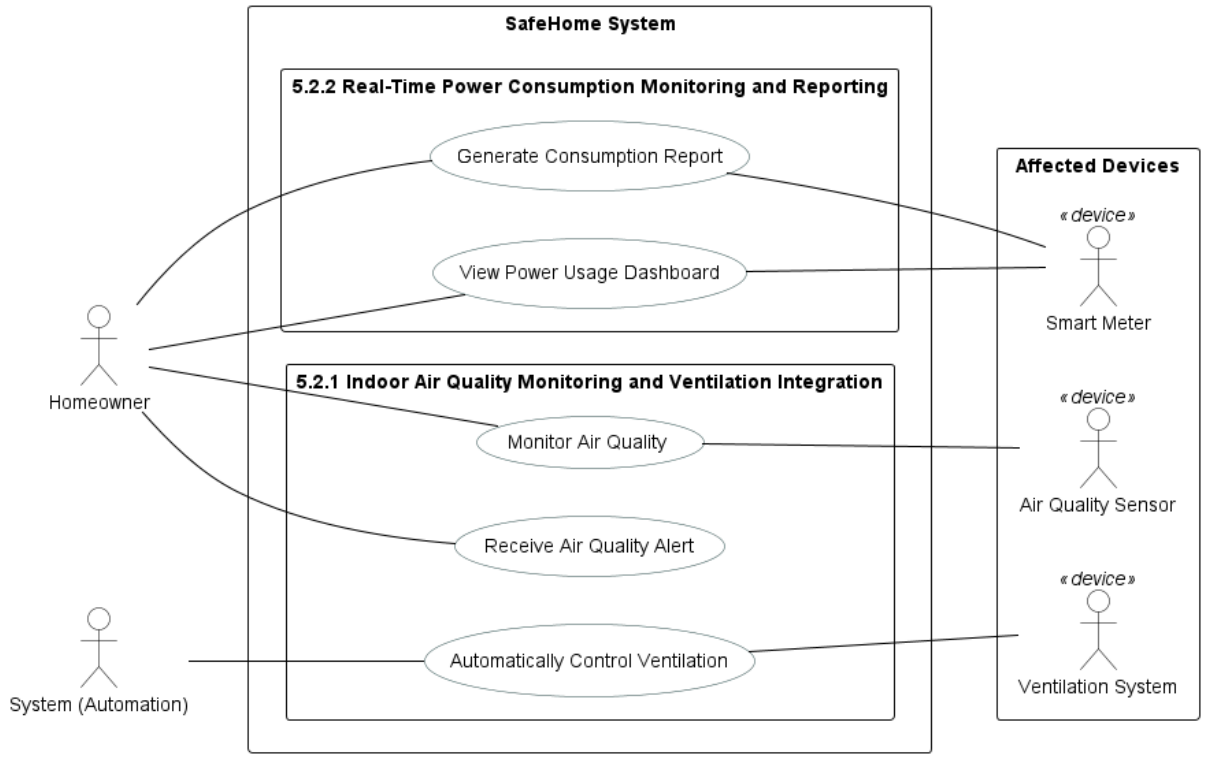
5. Indoor Monitoring and Device Control

5.1 Device Control - Use Case



References on SEPA safehome dialog slide 39

5.2 Indoor Monitoring System - Use Case



References on SEPA safehome dialog slide 27, 29

VI. Use Case

1. Intelligent Security

1.1. Sensor Monitoring

1.1.1 Physical Intrusion Detection - Use Case Diagram, Sequence Diagram

Use Case	1.1.1 Detect and Alert Physical Intrusion Attempt
Primary Actor	Homeowner
Goal in context	When the system is in 'Away' or 'Sleep' mode, to promptly and accurately detect a defined 'physical intrusion attempt', immediately notify the Homeowner, sound an internal alarm, and facilitate an immediate response through external reporting.
Precondition	1. The SafeHome system hub and required sensors are operating normally (at least one intrusion sensor active). 2. The security system is armed in 'Away' or 'Sleep' mode. 3. The hub has Internet connectivity for cloud-based notifications and dispatch. Local siren and event logging must operate without cloud connectivity. 4. The Homeowner is logged into the mobile app and has consented to receive emergency notifications.
Trigger	A 'Physical Intrusion Attempt' is detected when the system is armed. This includes one or more of the following sensor events: 1. Forced Opening: A 'Contact Sensor' on a locked door/window changes state from 'Closed' to 'Open'. 2. Lock Tampering: A 'Smart Door Lock' registers 10 consecutive password failures within a 5-minute window. 3. Forced Impact: A 'Shock/Vibration Sensor' on a door/window detects an impact exceeding its configured threshold. 4. Unauthorized Motion: In 'Away' mode, a 'Motion Sensor' detects movement in a designated internal area. All triggers are subject to a 60-second deduplication window to prevent alert storms.
Scenario	(Based on Trigger 3: Window Impact Attempt): 1. A 'Shock Sensor' on the living room window detects a breach event and transmits it to the SafeHome system hub. 2. The system hub verifies the current security state is 'Away' and identifies the event as a 'physical intrusion attempt'. 3. The system immediately sounds the internal siren (NFR: within 3 seconds). 4. In parallel, the hub sends the event to the cloud. The cloud service dispatches emergency push notifications (NFR: ≤5s) and SMS messages (NFR: ≤10s) to all registered Homeowners. 5. The system automatically reports the intrusion to a pre-configured external security service (NFR: ≤15s). A dispatch transaction ID is stored upon

	success. 6. When the Homeowner opens the mobile app via the notification, it displays the alarm status and the live feed from the relevant camera. 7. All outcomes (siren, notifications, dispatch results) and the initial detection event are permanently recorded in the system's activity log.
Exception	- (Scenario 3a) Siren Offline: → .1: The system logs the siren failure and proceeds with Homeowner notifications (step 4) and external reporting (step 5). - (Scenario 4a) Homeowner Notification Failure: → .1: If a Homeowner's device is offline, the system logs the push notification failure. SMS fallback is attempted, while the internal siren (step 3) and external reporting (step 5) proceed normally. - Higher-priority event: → .1: If a life-safety event (e.g., fire) occurs during an intrusion alarm, the system prioritizes the life-safety alarm. The intrusion event is logged and handled via push notifications only. - Concurrency: multi-user commands: → .1: If conflicting commands (e.g., 'Disarm' vs 'Dispatch Now') arrive simultaneously, they are resolved in the following priority order: Control Panel > Mobile (Owner) > Mobile (Guest) > Automation. → .2: A 'Disarm' command will cancel the dispatch only if it arrives before the dispatch request is committed. - Event storm: → .1: Multiple intrusion triggers within the 60-second window are consolidated into a single incident to rate-limit follow-up alerts (e.g., ≤3 per minute).
Priority	Essential
Frequency of use	Low (Occurs only during an intrusion attempt)
Channel to actor	Mobile app (push notification, in-app screen), SMS, Home alarm (siren)
Secondary actors	1. Sensors (Contact, Shock, Motion) 2. SafeHome Cloud Server 3. Home Alarm (Siren) 4. External Security Service 5. Camera
Channels to secondary actors	1. Sensor -> System Hub: LAN/Wi-Fi 2. System Hub <-> Cloud Server: Internet 3. Cloud Server -> External Security Service: Secure API
Open issues	1. Define a dispatch grace period (e.g., 30 seconds) to prevent false alarm fees and clarify cancellation authority for different Homeowner roles. 2. Finalize the lockout duration and Homeowner verification workflow for the '10 consecutive password failures' trigger.

	3. Confirm jurisdictional requirements and address validation for external reporting services.

References on 2025.10.26 / 2025.10.29 Meeting

1.1.2 Environmental Hazard Detection - Use Case Diagram, Sequence Diagram

Use Case	1.1.2 Environmental Hazard Detection
Primary Actor	Homeowner, Resident
Goal in context	To promptly detect hazardous environmental conditions (e.g., fire, CO, gas leaks), and execute a severity-based response, ensuring immediate local alarms for life-safety events while providing appropriate alerts for less critical issues.
Precondition	1. The SafeHome system hub and at least one environmental sensor are operating normally. 2. The Home Alarm (Speaker/Siren) device is registered and online. 3. For life-safety events (Fire, CO, Gas), the local alarm and event logging must operate without cloud connectivity. 4. The Homeowner is logged into the mobile app and has consented to receive emergency notifications.
Trigger	An 'Environmental Hazard' is detected, categorized by severity: - Critical (Life-Safety): 1. A 'Fire/Smoke Sensor' detects smoke or a rapid temperature rise. 2. A 'CO Sensor' detects a carbon monoxide concentration above a critical ppm threshold. 3. A 'Gas Sensor' detects a gas concentration above its explosive limit threshold. - Warning (Property/Health): 4. A 'Leak Sensor' detects water. 5. An 'Air Quality Sensor' detects VOC or CO2 levels above a warning threshold. For Critical hazards, a deduplication window applies to cloud notifications only, not the local alarm. Sensor thresholds use hysteresis to avoid rapid state changes.
Scenario	(Based on Trigger 2: Critical CO Detection): 1. A 'CO Sensor' detects a critical level of carbon monoxide and transmits the event to the SafeHome hub. 2. The hub identifies the event as a "Critical" life-safety hazard based on its internal policy. 3. The system immediately activates the internal Home Alarm with a distinct sound pattern or voice alert (NFR: ≤3s, ≥85dB @3m). 4. In parallel, the hub sends the event to the cloud, which dispatches urgent push notifications (NFR: ≤5s) and SMS messages (NFR: ≤10s) to all

	Homeowners. SMS serves as a fallback for push notification failures. 5. If configured, the hub commands linked smart home devices to execute a safety routine (e.g., shut off a smart gas valve). 6. The hazard event and all subsequent actions are permanently recorded in the system's activity log. 7. The local alarm persists until the hazard condition clears. For Critical events, a Homeowner acknowledgement may trigger a temporary 'Hush' period (e.g., 10 minutes).
Exception	- Home Alarm Offline: → .1: The hub logs the alarm device failure but proceeds with cloud notifications and any configured smart device actions. - Cloud Connectivity Failure: → .1: The local alarm and logging proceed without interruption. Queued notifications are sent once connectivity is restored. - Smart Device Command Failure: → .1: If a command to a smart device (e.g., gas valve) fails or times out, the system logs the failure and notifies the Homeowner, suggesting manual action. - Sensor Fault Detected: → .1: If a sensor reports a fault (e.g., low battery, tamper), the system sends a non-urgent 'Warning' notification to the Homeowner and flags the device for maintenance. - Concurrent Hazard Events: → .1: If multiple hazards occur simultaneously, alarms and resources are prioritized in the order of: Fire > Gas > CO. All events are still logged and notified.
Priority	Essential for Critical events; High for Warning events. The priority for Warning events may be elevated when the system is in 'Away' mode.
Frequency of use	Low (Occurs only during a hazard event)
Channel to actor	Home Alarm (Siren/Speaker), Mobile app (push notification, in-app screen), SMS
Secondary actors	1. Environmental Sensors (Fire, CO, Gas, Leak) 2. SafeHome Cloud Server 3. Home Alarm (Siren/Speaker) Device 4. Smart Home Devices (e.g., Smart Valves, Ventilation Fans)
Channels to secondary actors	1. Sensor -> System Hub: LAN/Wi-Fi 2. System Hub <-> Cloud Server: Internet 3. System Hub -> Smart Home Devices: LAN/Wi-Fi/Zigbee/Z-Wave/Thread
Open issues	1. Define Homeowner-configurable policies for Warning events, while ensuring Critical events retain non-disableable minimum guarantees (local alarm, logging, urgent notification). 2. Specify the recommended actions for smart devices, such as the automatic shutoff of a smart water valve upon leak detection.

	3. Finalize the hazard priority matrix, including preemption timing and rules for siren/voice alert takeover in multi-hazard scenarios.

References on 2025.10.26 / 2025.10.29 Meeting

1.1.3 Outdoor Motion Detection - Use Case Diagram, Sequence Diagram

Use Case	1.1.3 Outdoor Motion Detection
Primary Actor	Homeowner
Goal in context	To provide timely notifications with visual context (snapshots) for external motion events, regardless of the security mode, while minimizing notification fatigue through a Homeowner-configurable cooldown period.
Precondition	1. The SafeHome system hub and at least one external camera are registered and online. 2. The camera's 'Always-on Motion Detection' setting is enabled in the app. 3. The Homeowner is logged into the mobile app and has granted notification permissions.
Trigger	An external camera detects motion that meets a Homeowner-defined sensitivity threshold. Motion detection algorithms use dwell time and hysteresis to prevent false triggers. A per-camera cooldown period is applied to prevent notification storms.
Scenario	(Based on motion detected at the front door): 1. The 'Front Door Camera' detects motion, captures a high-resolution snapshot, and includes a short pre-roll clip (e.g., 3-5 seconds) if supported. 2. The camera transmits the event and associated media to the SafeHome hub. 3. The hub verifies that the configured cooldown period for this specific camera has elapsed. 4. The hub immediately writes the event (timestamp, camera name, snapshot) to the local activity log. 5. In parallel, the hub instructs the cloud server to send a rich push notification. The cloud first sends a notification with a low-resolution thumbnail (NFR: ≤5s), followed by a link to the full-resolution media once the upload is complete. 6. The Homeowner taps the notification, and the app opens directly to the event details page or the camera's live view. 7. The event is permanently logged, and the cooldown timer for the 'Front Door Camera' is reset.
Exception	- Poor Network Connectivity: → .1: If the snapshot upload is slow or fails, the cloud sends a text-only notification first (e.g., "Motion detected at Front Door"). - Cloud Connectivity Failure:

	→ .1: Local event logging continues without interruption. Queued media and notifications are sent once connectivity is restored. - Frequent Events during Cooldown: → .1: The system logs all detected motion events locally but sends only one notification to the Homeowner per configured cooldown period to avoid alert fatigue.
Priority	High. Priority may be elevated based on system mode (e.g., 'Away') or time-of-day schedules.
Frequency of use	High (Depends on the level of activity outside the home)
Channel to actor	Mobile app (rich push notification, in-app screen), Activity Log
Secondary actors	1. External Camera 2. SafeHome System Hub 3. SafeHome Cloud Server
Channels to secondary actors	1. Camera -> System Hub: LAN/Wi-Fi/RTSP 2. System Hub <-> Cloud Server: Internet
Open issues	1. Define the specifics of the notification 'Mute' or 'Snooze' functionality (e.g., mute for 1 hour, based on a schedule). 2. Evaluate the feasibility of a "smart summary" feature that aggregates multiple events from the same camera into a single notification digest.

References on 2025.10.26 / 2025.10.29 Meeting

1.1.4 Dog Barking Detection - Use Case Diagram, Sequence Diagram

Use Case	1.1.4 Dog Barking Detection
Primary Actor	Homeowner
Goal in context	To detect prolonged dog barking that may indicate distress or cause a nuisance, promptly notify the Homeowner, and provide tools for remote assessment and intervention (e.g., live camera view, two-way audio).
Precondition	1. The SafeHome system hub is online. 2. A camera or sensor with sound detection capabilities is registered and enabled for 'Dog Bark Detection'. 3. The Homeowner is logged into the mobile app with notification permissions granted.
Trigger	A sound sensor or camera detects audio patterns matching a 'dog bark' signature that continues for a Homeowner-defined duration (e.g., more than 60 seconds within a 90-second window). The trigger logic uses hysteresis to avoid resets from brief pauses in barking. A

	cooldown period applies after a notification is sent to prevent alert storms.
Scenario	(Based on a dog barking while the system is in 'Away' mode): 1. The 'Living Room Camera' detects a dog barking pattern and starts a duration timer. 2. The barking continues, and the duration exceeds the configured threshold (e.g., 60 seconds). 3. The hub checks the Homeowner's policy and confirms that barking notifications are active for the current 'Away' mode. 4. The hub immediately writes the event (timestamp, device name) to the local activity log. 5. In parallel, the hub instructs the cloud server to send a push notification to the Homeowner's mobile device (NFR: ≤5s). 6. The Homeowner taps the notification, and the app opens directly to the 'Living Room Camera's' live view, with the two-way audio feature readily accessible. 7. The barking event is permanently logged, and the notification cooldown timer for this event type is reset.
Exception	- Barking Stops Prematurely: → .1: If the barking stops before the duration threshold is met, the timer is reset. The event is logged locally for informational purposes, but no notification is sent. - Cloud Connectivity Failure: → .1: Local logging continues. The notification is queued and will be sent when the connection is restored. - Policy Mute Active: → .1: If the system is in 'Home' mode and the policy is set to disable barking alerts, the hub logs the event silently but does not send a notification. - Sound Misclassification: → .1: If a loud, non-barking noise occurs (e.g., TV sound), the sound signature does not match 'dog bark' with sufficient confidence, and the trigger is not activated.
Priority	Medium (Important convenience and pet wellness feature)
Frequency of use	Medium to High (Depends on pet's behavior)
Channel to actor	Mobile app (push notification, in-app camera view)
Secondary actors	1. Camera or Sound Sensor 2. SafeHome System Hub 3. SafeHome Cloud Server
Channels to secondary actors	1. Camera/Sensor -> System Hub: LAN/Wi-Fi 2. System Hub <-> Cloud Server: Internet

Open issues	1. Should the barking duration threshold (e.g., 1 min, 3 min, 5 min) and sound sensitivity be Homeowner-configurable? 2. Should the notification include quick-action buttons, such as 'View Camera' or 'Start Two-Way Audio'? 3. How should the system's policy differentiate between 'Home' and 'Sleep' modes for this alert?

References on SEPA safehome dialog slide 58-59

1.2. Incident Management

1.2.1 Configure Alarm Conditions by Security Mode - Use Case Diagram, Sequence Diagram

Use Case	1.2.1 Configure Alarm Conditions by Security Mode
Primary Actor	Homeowner (with Admin privileges)
Goal in context	To provide a robust and intuitive interface for an Admin Homeowner to customize which sensors trigger an alarm for each security mode ('Home', 'Away', 'Sleep'), ensuring changes are saved atomically and synchronized reliably to the hub.
Precondition	1. The Homeowner is logged into the mobile app or web interface with Admin privileges. 2. The SafeHome hub is registered, and at least one sensor is paired with the system. 3. If the hub is offline, the app permits edits but displays a 'Sync Pending' status until the hub confirms the update.
Trigger	The Admin Homeowner navigates to the 'Settings' > 'Security Modes' configuration screen. The system fetches the current configuration profile from the cloud, along with a version identifier to manage concurrent edits. An edit session lease is obtained.
Scenario	(Based on an Admin Homeowner modifying the 'Sleep' mode profile): 1. The Homeowner selects the 'Sleep' mode to edit. 2. The Homeowner modifies the sensor activation list in the UI (e.g., enables the 'Living Room Motion Sensor'). 3. The Homeowner taps the 'Save' button. 4. The app sends the modified sensor list for the 'Sleep' profile, along with the original version identifier and an idempotent request ID, to the cloud. 5. The cloud server validates the request and verifies that the provided version identifier matches the current one. 6. The server atomically updates the profile, increments the version identifier, and records a field-level difference in the audit log. 7. The cloud server pushes the updated profile to the SafeHome hub, which applies the new profile and acknowledges the update with the new version number.

	8. The configuration change is saved, audited, and applied immediately or upon the next mode transition, as per system policy.
Exception	- Concurrent Edit Conflict: → .1: If another Admin saves a change in the meantime, the version identifier will not match. The cloud server rejects the save request with a conflict error. → .2: The app may offer a comparison view to allow the Homeowner to merge changes and retry the save. - Hub Offline during Sync: → .1: The cloud queues the profile update. The hub will pull the latest profile upon reconnection. - Hub Sync Timeout: → .1: If the cloud does not receive an acknowledgement from the hub within a set time, it marks the profile as 'Hub Sync Pending' and shows this status in the app. - Invalid Sensor Reference: → .1: If the request references a sensor that was removed during the editing session, the server rejects the change with a descriptive error, prompting the Homeowner to reload. - Outdated Client Schema: → .1: If the app uses an outdated configuration schema, the server rejects the request and prompts for an app refresh or update.
Priority	High
Frequency of use	Low
Channel to actor	Mobile app, Web interface (Settings screen)
Secondary actors	1. SafeHome Cloud Server 2. SafeHome System Hub
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server -> System Hub: Internet (Secure Push). The hub also pulls for updates periodically as a fallback.
Open issues	1. Define which sensor parameters can be configured per mode. Life-safety sensors should have limited, non-disableable settings. 2. Establish the default activation states for newly paired sensors (e.g., life-safety sensors default to enabled in all modes, others default to disabled). 3. Specify the version history policy, including the number of versions to retain and the functionality to revert to a previous configuration.

References on SEPA safehome dialog slide 5

1.2.2 Alarm Verification Step - Use Case Diagram, Sequence Diagram

Use Case	1.2.2 Alarm Verification Step
Primary Actor	Homeowner
Goal in context	To allow a Homeowner to quickly assess a potential security event using visual evidence (snapshot/clip) and confirm whether it is a genuine threat (Confirm Alarm) or a false alarm (Dismiss Alarm), thereby preventing unnecessary escalations.
Precondition	1. An initial, non-life-safety alarm condition has been triggered by a sensor. 2. The system is in a security mode that requires verification (e.g., 'Away'). 3. The Homeowner's device is online and capable of receiving rich push notifications. 4. A camera is associated with the zone where the sensor was triggered.
Trigger	The system detects a potential alarm event and initiates the verification workflow instead of proceeding to immediate full escalation.
Scenario	(Based on a motion sensor trigger in 'Away' mode): 1. The 'Living Room Motion Sensor' is triggered. 2. The hub immediately captures a snapshot or short clip from the associated 'Living Room Camera'. 3. The hub sends an urgent "Verification Required" event to the cloud, which sends a rich push notification to all authorized Homeowners, including a snapshot preview. 4. The Homeowner taps the notification, opening the app to a dedicated verification screen. 5. The screen displays the visual evidence, the trigger source ("Living Room Motion"), and two prominent buttons: "Confirm Real Alarm" and "Dismiss False Alarm". A countdown timer (e.g., 60 seconds) shows the time remaining before automatic escalation. 6. If the Homeowner sees an intruder and taps "Confirm Real Alarm", the system immediately proceeds to full escalation (e.g., activating the main siren and initiating emergency service integration). 7. If the Homeowner sees their pet and taps "Dismiss False Alarm", the system immediately cancels the alarm sequence, silences any local pre-alerts, and logs the event as "False Alarm, dismissed by Homeowner." 8. The event is resolved as either a confirmed incident or a dismissed false alarm, and an audit log records the Homeowner's action and the outcome.
Exception	- Homeowner Does Not Respond: → .1: If the verification timer expires without input from a Homeowner, the system assumes the threat is real and automatically proceeds to full escalation. - Life-Safety Event Occurs: → .1: This verification workflow is SKIPPED for life-safety triggers (e.g., Fire/CO). The system escalates immediately and unconditionally. - No Associated Camera:

	→ .1: If the triggered sensor's zone has no camera, the system may skip verification and proceed to a standard alarm, or send a text-only notification allowing the Homeowner to remotely disarm (policy decision required). - Concurrent Action by Multiple Homeowners: → .1: The first Homeowner's action ('Confirm' or 'Dismiss') is processed. All other Homeowners' screens are updated in real-time to reflect the resolved state.
Priority	Essential
Frequency of use	Low (Occurs only when a potential alarm is triggered)
Channel to actor	Mobile app (rich push notification, verification screen)
Secondary actors	1. SafeHome System Hub 2. SafeHome Cloud Server 3. Camera 4. Siren (upon escalation)
Channels to secondary actors	1. App <-> Cloud Server: Internet (HTTPS) 2. System Hub <-> Cloud Server: Internet 3. System Hub <-> Camera/Siren: LAN/Wi-Fi
Open issues	1. What is the optimal default duration for the verification timer (e.g., 30s, 60s, 90s)? Should it be Homeowner-configurable? 2. When dismissing a false alarm, should the Homeowner be prompted to select a reason (e.g., "Pet", "Family Member") to provide feedback for future system improvements? 3. What is the pre-alert behavior during the verification window (e.g., no siren, quiet local chime, flashing lights)?

References on 2025.10.29 Meeting

1.2.3 Emergency Service Integration and Auto Call - Use Case Diagram, Sequence Diagram

Use Case	1.2.3 Emergency Service Integration and Auto Call
Primary Actor	SafeHome System (acting on a confirmed alarm)
Goal in context	Upon confirmation of a genuine alarm, to execute a pre-configured emergency response plan by systematically notifying Homeowners via automated phone calls and, if enabled, dispatching professional emergency services.
Precondition	1. A genuine alarm has been confirmed (manually by a Homeowner or automatically after a timeout). 2. The Homeowner has configured an emergency contact list and/or enabled integration with an external emergency service. 3. The system's main siren is already active.

	4. For external dispatch, the Homeowner's home address is accurately registered and validated.
Trigger	The system's incident management workflow receives a "Confirm Real Alarm" event.
Scenario	(Based on a confirmed intrusion alarm): 1. The system receives the "Confirm Real Alarm" event and enters the 'Escalation' phase. 2. The hub instructs the cloud to initiate an automated call sequence via an external IVR/Call Gateway service. 3. The IVR service begins calling the Homeowner's pre-defined emergency contact numbers in order. When answered, an automated voice message is played (e.g., "This is an alert from your SafeHome system. An intrusion has been confirmed at [Home Address]. Please check your app for details."). 4. In parallel with the calls, if Emergency Service Integration is enabled, the cloud sends a dispatch request to the configured service. 5. The dispatch request payload includes the home address, event type ("Intrusion"), trigger sensor, and event timestamp. 6. The external emergency service receives the request, acknowledges it, and returns a dispatch transaction ID upon success. 7. The cloud logs the outcome of all call attempts and the result of the dispatch request, including the transaction ID. 8. All configured emergency actions are executed. The system status in the app is updated to "Escalated: Emergency Services Dispatched," and the entire sequence is recorded in the audit log.
Exception	- Homeowner Disarms During Escalation: → .1: The cloud immediately sends a 'Cancel' command to the IVR Gateway to stop further calls and to the Emergency Service to cancel the dispatch. The success or failure of the cancellation is logged. - IVR / Call Gateway Failure: → .1: The cloud logs the failure and sends a push notification to the Homeowner ("Automated calls could not be completed."). The emergency dispatch, if enabled, proceeds independently. - Emergency Dispatch Failure: → .1: The cloud logs the failure, retries per policy, and sends an urgent push notification to the Homeowner ("CRITICAL: Automated dispatch failed. Please contact emergency services manually."). - No Contacts or Services Configured: → .1: If the alarm is confirmed but no call numbers or dispatch services are configured, the system logs this state. The main siren continues to operate, but no external notifications are sent.
Priority	Essential
Frequency of use	Low (Occurs only during a confirmed, critical event)

Channel to actor	Automated Phone Call (IVR), API Call (to emergency services)
Secondary actors	1. SafeHome Cloud Server 2. IVR/Call Gateway Service 3. Emergency Service Dispatch System
Channels to secondary actors	1. Cloud Server <-> External Services: Internet (Secure API)
Open issues	1. What is the detailed logic for the call tree (e.g., call in sequence vs. simultaneously? What happens if a call goes to voicemail?)? 2. What are the specific cancellation policies for different emergency services (e.g., is there a point after which dispatch cannot be canceled?)? 3. Should the Homeowner be able to trigger this escalation manually via a Panic Button without a sensor event?

References on SEPA safehome dialog slide 6, 7

1.2.4 Panic Button - Use Case Diagram, Sequence Diagram

Use Case	1.2.4 Panic Button
Primary Actor	Homeowner (any authenticated Homeowner)
Goal in context	To allow a Homeowner to manually and immediately trigger the highest-priority system alarm in a perceived emergency, bypassing all other system states or verification steps to ensure an immediate response.
Precondition	1. The Homeowner is logged into the SafeHome mobile app. 2. The app has a persistent, easily accessible Panic Button in its main UI. 3. For remote activation, the Homeowner's device must have an internet connection.
Trigger	The Homeowner long-presses the Panic Button in the app for a pre-defined duration (e.g., 3 seconds) to prevent accidental activation.
Scenario	(Based on a Homeowner activating the panic button): 1. The Homeowner perceives an emergency and long-presses the Panic Button in the mobile app. The app provides haptic and visual feedback to indicate the activation is in progress. 2. After the 3-second hold is complete, the app immediately sends a "Panic" command with the highest priority level to the cloud. 3. The cloud receives the command and immediately instructs the hub to activate the main siren at maximum volume with a distinct panic sound pattern. 4. The cloud simultaneously initiates the full emergency escalation workflow as defined in UC 1.2.3 (Escalate Alarm & Notify Emergency Contacts). This includes automated calls and dispatching emergency services without any verification delay.

	5. The cloud sends a push notification to all other Homeowners on the account, stating, "[Homeowner Name]'s Panic Button has been activated." 6. The system is in its highest alarm state, emergency contacts and services are being notified, and the event is logged in the audit trail with the highest severity, including which Homeowner activated it.
Exception	- Accidental Press / Cancellation: → .1: If the Homeowner releases the Panic Button before the 3-second hold is complete, the activation is cancelled and no alarm is triggered. - Hub is Offline: → .1: The cloud still proceeds with cloud-based escalation (automated calls, emergency service dispatch). → .2: The app receives a notification: "Panic Alert sent to contacts, but local siren could not be activated as the Hub is offline." - App is Offline: → .1: If the Homeowner's device has no internet connection, the app displays an immediate error: "No internet connection. Cannot send Panic Alert." - Concurrent Alarm: → .1: If a standard sensor alarm is in progress, the Panic event overrides it, escalating the alarm to the highest priority and triggering the immediate response defined in this use case.
Priority	Critical
Frequency of use	Very Low (Emergency situations only)
Channel to actor	Mobile app (dedicated button)
Secondary actors	1. SafeHome System Hub 2. SafeHome Cloud Server 3. All other registered Homeowners
Channels to secondary actors	1. App -> Cloud Server: Internet (HTTPS) 2. Cloud Server -> System Hub: Internet
Open issues	1. Should there be different types of Panic buttons (e.g., a "Silent Panic" that dispatches police without a local siren vs. an "Audible Panic")? 2. Should a physical, standalone Panic Button accessory be offered as part of the product line? 3. What is the cancellation process for an accidental Panic Button activation? (It should require a PIN/password and immediately send a "False Alarm" signal to dispatched services if possible).

References on 2025.10.29 Meeting

1.3 Security Mode Control

1.3.1 One-Touch Modes (Away, Home, Sleep) - Use Case Diagram, Sequence Diagram

Use Case	1.3.1 One-Touch Modes (Away, Home, Sleep)
Primary Actor	Homeowner, Guest
Goal in context	To allow a Homeowner or Guest to quickly and reliably change the system's mode, which simultaneously adjusts the security state (e.g., Arm, Disarm) and controls smart home devices (e.g., lights, HVAC) according to predefined settings.
Precondition	1. The Homeowner or Guest is logged into the mobile app or is physically present at the Control Panel. 2. The SafeHome hub, relevant sensors, and smart home devices (lights, HVAC, etc.) are online and operational. 3. Mode configurations are predefined. The Control Panel can change modes locally without cloud connectivity.
Trigger	A Homeowner or Guest initiates a mode change from the mobile app or Control Panel. The system supports four primary modes: Home, Away, Overnight Travel, and Extended Travel.
Scenario	(Based on a Homeowner setting the 'Away' mode from the mobile app): 1. The Homeowner taps the 'Away' button in the app. 2. The app sends the 'SetMode=Away' command to the cloud, which validates it and forwards it to the hub. 3. The hub initiates two parallel processes: a. Security Arming: It validates security sensors for faults and begins the 'Exit Delay' timer, updating its state to 'Arming'. b. Automation Execution: It retrieves the 'Away' mode automation settings and sends commands to linked devices (e.g., turn off all lights, set thermostat to energy-saving temperature). 4. The hub reports the new 'Arming' state back to the cloud, which synchronizes it to all clients. 5. After the Exit Delay completes, the hub's security state transitions to 'Armed-Away', and this final state is synchronized. 6. For 'Overnight Travel' and 'Extended Travel' modes, the automation execution (step 3b) also includes activating a "simulated occupancy" routine that turns designated lights on and off at random intervals. 7. The entire mode change, including the user, client, and all device state

	changes, is recorded in the audit log.
Exception	- Concurrent Command Conflict: → .1: If a 'Disarm' command arrives while the system is 'Arming', the 'Disarm' command takes precedence and cancels the entire mode change sequence. - Security Sensor Fault: → .1: If a security sensor is faulted (e.g., a window is open), the hub rejects the arming part of the sequence and notifies the Homeowner, providing an option to bypass the sensor. - Smart Device Fails to Respond: → .1: If an automation device (e.g., a light) is unresponsive, the hub logs the failure and notifies the Homeowner, but proceeds with the security arming and the rest of the automation. - Hub Offline: → .1: If an app command is sent while the hub is offline, the cloud informs the Homeowner that the command could not be delivered. Mode changes via the local Control Panel will still function.
Priority	Essential
Frequency of use	Very High
Channel to actor	Mobile app, Control Panel, Web interface
Secondary actors	1. SafeHome System Hub 2. SafeHome Cloud Server 3. All Sensors 4. Connected Smart Home Devices (Lights, HVAC, etc.)
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server <-> System Hub: Internet 3. System Hub <-> Devices: LAN/Wi-Fi/Zigbee
Open issues	1. Should Homeowners be able to create, customize, or delete their own modes (e.g., "Movie Mode")? 2. How should the "simulated occupancy" routine be configured by the Homeowner (e.g., which lights to include, active time windows)? 3. Should the behavior of a 'Duress PIN' be defined, which would appear to disarm the system but would silently trigger a panic event? 4. How should the system visually represent the status of bypassed sensors

	when a mode is active?

References on SEPA safehome dialog slide 9

1.3.2 Sensor Bypass - Use Case Diagram, Sequence Diagram

Use Case	1.3.2 Sensor Bypass
Primary Actor	Homeowner (with Admin privileges)
Goal in context	To allow an Admin Homeowner to intentionally and temporarily exclude a specific faulted sensor from monitoring when arming, ensuring the rest of the system can be armed while clearly acknowledging the security exception.
Precondition	1. The Homeowner is in the process of arming the system, and this action has been blocked by a faulted sensor. 2. The Homeowner has Admin privileges. 3. The SafeHome hub is online, as the bypass command is a local function.
Trigger	The system rejects an arming attempt due to a faulted sensor and presents the Homeowner with an error message and a 'Bypass' option. If multiple sensors are faulted, the UI lists them with options to bypass individually or all at once, requiring explicit risk acknowledgment.
Scenario	(Based on a Homeowner bypassing an open window to arm the system): 1. The Homeowner attempts to arm, but the command fails due to an open window sensor. 2. The app or control panel displays the error and a "Bypass" button, which the Homeowner selects. 3. The client sends a new arming command that includes a list of sensors to bypass. 4. The hub receives the command and validates the Homeowner's privileges and the sensor type (ensuring it's not a life-safety sensor). 5. The hub adds the specified sensor to a temporary bypass list for the current arming session. 6. The hub proceeds with the arming sequence (e.g., Exit Delay), ignoring input from the bypassed sensor. 7. The hub reports its new state ('Arming') and the list of bypassed sensors to the cloud, which synchronizes this information to all clients. 8. The system becomes 'Armed' with a visible indication of the bypass. The bypass is automatically cleared on the next Disarm event and the action is logged.
Exception	- Attempt to Bypass Life-Safety Sensor: → .1: If the Homeowner attempts to bypass a Fire/CO sensor, the system rejects the command with an error. - Concurrent Disarm: → .1: If another Homeowner disarms the system while a bypass command is being issued, the bypass is voided. - Bypass Limit Exceeded:

	→ .1: If the Homeowner attempts to bypass more sensors than the system limit allows, the command is rejected. - Privilege Re-validation Failure: → .1: If the system requires a PIN re-entry to confirm the bypass and the validation fails, the command is rejected. - Hub Unreachable from App: → .1: If an app-initiated bypass command cannot reach the hub, the app displays an error, and the system is not armed.
Priority	High
Frequency of use	Medium
Channel to actor	Mobile app, Control Panel
Secondary actors	1. SafeHome System Hub 2. SafeHome Cloud Server 3. The specific Sensor being bypassed
Channels to secondary actors	1. App/Panel -> System Hub (via Cloud or Local LAN) 2. System Hub -> Cloud Server: Internet
Open issues	1. Should a bypass be a one-time action for the current arming period, or can it be made persistent (e.g., for a scheduled duration)? 2. Should there be a system-wide limit on how many sensors can be bypassed simultaneously? 3. How should a bypassed sensor's status be visually represented in the app (e.g., banner, list) and mentioned in notifications?

References on SEPA safehome dialog slide 10

1.3.3 Sensor Activation and Deactivation - Use Case Diagram, Sequence Diagram

Use Case	1.3.3 Sensor Activation and Deactivation
Primary Actor	Homeowner (with Admin privileges)
Goal in context	To allow an Admin Homeowner to deliberately deactivate a sensor from the system's monitoring, either for a fixed duration or indefinitely, ensuring the change is clearly communicated and safely managed.
Precondition	1. The Homeowner is logged into the mobile app or web interface with Admin privileges. 2. The sensor to be modified is registered with the system. 3. The system is in a 'Disarmed' state. 4. The Homeowner may be required to acknowledge a risk or re-enter their PIN to confirm the action.
Trigger	The Admin Homeowner navigates to 'Settings' > 'Device Management',

	selects a specific sensor, and initiates a state change. The system ensures it has the latest sensor status before allowing changes to prevent conflicts.
Scenario	(Based on a Homeowner disabling a sensor for 1 hour): 1. The Homeowner selects a sensor and toggles its state to 'Disabled'. 2. The app presents duration options (e.g., "Disable for 1 hour," "Disable until I re-enable"). The Homeowner selects "1 hour" and confirms. 3. The app sends the disable command, specifying the duration, to the cloud. 4. The cloud validates the Homeowner's privileges, updates the sensor's state, and schedules its automatic re-enabling in exactly 1 hour. 5. The cloud pushes the new state ('Disabled') to the hub. 6. The disabled state is synchronized across all clients. The sensor is now inactive for monitoring purposes. 7. The action is recorded in the audit log, including which Homeowner made the change, the reason, and the duration. A disabled sensor is always overridden by a Panic Button event.
Exception	- Attempt to Disable Life-Safety Sensor: → .1: If the Homeowner attempts to disable a Fire/CO sensor, the system rejects the command with an error stating it cannot be disabled. - Attempt While System is Armed: → .1: If the Homeowner attempts this action while the system is armed, the command is rejected. - Concurrent Edit Conflict: → .1: If the sensor's configuration was changed by another session, the save command is rejected, and the Homeowner is prompted to refresh. - Sensor Has Dependencies: → .1: If the sensor is used by other system automations, the app displays a warning summarizing the impact, and the Homeowner must confirm before proceeding. - Non-Disableable Alerts: → .1: Even when a sensor is disabled, critical device-health alerts (e.g., tamper, low battery) remain active and are logged.
Priority	High
Frequency of use	Low
Channel to actor	Mobile app, Web interface. A persistent 'Disabled' badge with a countdown timer (if applicable) is shown on the sensor in the UI.
Secondary actors	1. SafeHome Cloud Server 2. SafeHome System Hub 3. The specific Sensor being disabled
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server <-> System Hub: Internet (Secure Push/Pull)

Open issues	1. How should persistently disabled sensors be managed to prevent them from being forgotten (e.g., weekly summary notifications, an in-app banner after 30 days)? 2. What is the full list of non-disableable sensors or signal types (beyond life-safety and health alerts)? 3. Should there be a system-wide limit on how many sensors can be disabled at one time?

References on 2025.10.26 / 2025.10.29 Meeting

2. Live Surveillance

2.1 Camera Viewing and Control

2.1.1 Single Camera Live View - Use Case Diagram, Sequence Diagram

Use Case	2.1.1 Single Camera Live View
Primary Actor	Homeowner
Goal in context	To allow the Homeowner to assess the current situation inside or outside the home by viewing a clear, real-time video feed from a selected camera via the mobile app or web.
Precondition	1. The SafeHome system hub and cloud server are operating normally. 2. At least one camera is properly installed, connected to the system, and online. 3. The Homeowner is logged into the mobile app or web interface.
Trigger	The Homeowner initiates a request to view the live feed from a specific camera to check the status of the home.
Scenario	1. The Homeowner selects the desired camera from the app or web interface. 2. The system checks if the camera requires a password for access. 3. (If password is required) The system prompts the Homeowner to enter the password, and the Homeowner enters the correct password. 4. The system requests a live video stream from the selected camera. 5. The camera begins transmitting the video data to the system hub. 6. The system hub securely forwards the video stream via the cloud server to the Homeowner's device. 7. The Homeowner views the live video feed on their device screen (NFR: stream starts within 5 seconds).
Exception	- Incorrect Password Entry: → .1: If the Homeowner enters an incorrect password, the system displays an "Incorrect password" message and denies access. The scenario ends. - Camera Connection Failure: → .1: If the selected camera is offline (e.g., powered off, network disconnected), the system displays a message to the Homeowner, such as "Cannot connect to camera" or "Camera is offline."

	- Poor Homeowner Network Connection: → .1: If the Homeowner's device has an unstable network connection, the video stream may lag, suffer from reduced quality, or fail to load. → .2: The system attempts to automatically adjust the stream quality (e.g., resolution, bitrate) based on network conditions to ensure continuous playback.
Priority	Essential
Frequency of use	High (Used as needed by the Homeowner)
Channel to actor	Mobile app, Web browser
Secondary actors	1. Camera 2. SafeHome System Hub 3. SafeHome Cloud Server
Channels to secondary actors	1. Camera -> System Hub: Wi-Fi/LAN 2. System Hub <-> Cloud Server: Internet
Open issues	1. What security policies (e.g., end-to-end encryption) will be implemented to protect the video stream from unauthorized access? 2. How will system load and performance be managed when multiple Homeowners access the same camera feed simultaneously (e.g., stream replication, session limits)?

References on SEPA safehome dialog slide 29-31

2.1.2 Two-Way Audio - Use Case Diagram, Sequence Diagram

Use Case	2.1.2 Two-Way Audio
Primary Actor	Homeowner
Goal in context	To enable the Homeowner to have a real-time, two-way conversation with people or pets at the camera's location by using the camera's built-in microphone and speaker.
Precondition	1. The Homeowner is currently viewing a live feed from a specific camera (as per UC 2.1.1). 2. The selected camera supports two-way audio (has an integrated microphone and speaker). 3. The SafeHome mobile app or web browser has been granted permission to use the microphone on the Homeowner's device.
Trigger	While viewing a live feed, the Homeowner initiates the 'Talk' or 'Two-Way Audio' feature to communicate.
Scenario	1. The Homeowner taps the 'Microphone' or 'Talk' icon on the live video screen.

	2. The system activates the microphone on the Homeowner's device and begins capturing their voice. 3. The captured audio is transmitted in real-time through the cloud and hub to the selected camera. 4. The camera plays the Homeowner's voice through its built-in speaker. 5. Simultaneously, the camera's microphone captures audio from its location, which is transmitted back to the Homeowner's device. 6. The Homeowner ends the session by tapping the 'Microphone' or 'Talk' icon again.
Exception	- Camera Does Not Support Two-Way Audio: → .1: The 'Talk' icon is disabled or displays a message ("This camera does not support two-way audio") if selected. - No Microphone Permission: → .1: The system prompts the Homeowner to grant microphone permission. The feature will not activate until permission is granted. - Poor Network Connection: → .1: If the network connection is unstable for either the Homeowner or the camera, the audio may be delayed, choppy, or fail entirely. - Concurrent Usage Conflict: → .1: If another Homeowner is already using the two-way audio or PTZ features of the camera, the system rejects the new request with a message ("Camera is currently in use by another user.").
Priority	High
Frequency of use	High (Used as needed by the Homeowner)
Channel to actor	Mobile app, Web browser (on the live view screen)
Secondary actors	1. Camera (with two-way audio) 2. SafeHome System Hub 3. SafeHome Cloud Server
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server <-> System Hub: Internet 3. System Hub <-> Camera: LAN/Wi-Fi
Open issues	1. Should noise cancellation and echo reduction algorithms be implemented to improve audio clarity, especially in noisy environments or when the Homeowner's device is near the camera? 2. What is the policy for audio stream priority and quality of service (QoS) to minimize latency for a natural conversation flow?

References on SEPA safehome dialog slide 16

2.1.3 Protect Sensitive Camera Feed with a Password - Use Case Diagram, Sequence Diagram

Use Case	2.1.3 Protect Sensitive Camera Feed with a Password
Primary Actor	Homeowner (with Admin privileges)
Goal in context	To allow an Admin Homeowner to set a unique password for a specific camera, restricting its live view and recordings to only those who know the password, thereby enhancing privacy.
Precondition	1. The Homeowner is logged into the mobile app or web interface with Admin privileges. 2. The target camera is registered with the system and is online.
Trigger	The Admin Homeowner navigates to a specific camera's settings menu to manage its privacy settings.
Scenario	To Set a Password: 1. The Homeowner navigates to the target camera's settings and enables the 'Password Lock' option. 2. The Homeowner enters and confirms a new password that meets complexity requirements. 3. The Homeowner saves the changes. 4. The system securely stores the password requirement for the camera. Subsequently, any Homeowner attempting to view this camera's feed will be prompted for this password. To Remove a Password: 1. The Homeowner navigates to the locked camera's settings and disables the 'Password Lock' option. 2. The system prompts for confirmation, potentially requiring the Homeowner to re-enter their main account password for security. 3. The Homeowner confirms the action. 4. The system removes the password requirement for the camera.
Exception	- Password Mismatch During Setup: → .1: If the confirmation password does not match the initial password entered, the system displays a "Passwords do not match" message and rejects the change. - Incorrect Password for Viewing: → .1: If a Homeowner enters the wrong password when trying to access the locked camera, the system displays an "Incorrect password" message and blocks access. Multiple failed attempts may trigger a temporary lockout. - Password Removal Cancelled: → .1: If the Homeowner cancels the confirmation prompt when attempting to remove a password, the operation is aborted, and the password lock remains active.
Priority	High

Frequency of use	Low (Primarily for initial setup or occasional changes)
Channel to actor	Mobile app, Web interface (in camera settings)
Secondary actors	1. SafeHome Cloud Server 2. SafeHome System Hub
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server <-> System Hub: Internet
Open issues	1. What are the password complexity requirements (e.g., minimum length, character types)? 2. What is the secure process for a Homeowner to recover or reset a forgotten camera password? 3. Should a Homeowner be required to re-enter their main account password before removing a camera's password lock for added security?

References on SEPA safehome dialog slide 19-31

2.1.4 Camera Activation and Deactivation - Use Case Diagram, Sequence Diagram

Use Case	2.1.4 Camera Activation and Deactivation
Primary Actor	Homeowner (with Admin privileges)
Goal in context	To allow an Admin Homeowner to temporarily stop (disable) and resume (enable) a specific camera's video/audio streaming and recording functions for privacy management, without physically disconnecting it.
Precondition	1. The Homeowner is logged into the mobile app or web interface with Admin privileges. 2. The target camera is registered with the system and is online.
Trigger	The Admin Homeowner navigates to a specific camera's settings or control interface and decides to change its operational state.
Scenario	To Disable a Camera: 1. The Homeowner selects a camera and activates the 'Disable Camera' or 'Privacy Mode' toggle. 2. The system sends a disable command to the selected camera via the hub and cloud. 3. The camera hardware stops transmitting all video and audio data. 4. The system immediately stops all recording activities (both continuous and event-based) for this camera. 5. The system updates the camera's status to 'Disabled' across all interfaces, confirming the action to the Homeowner. To Enable a Camera: 1. The Homeowner navigates to the disabled camera's status screen.

	2. The Homeowner activates the 'Enable Camera' or 'Turn Off Privacy Mode' toggle. 3. The system sends an enable command to the selected camera. 4. The camera hardware resumes transmitting video and audio data. 5. The system resumes recording activities for this camera according to its saved settings. 6. The system updates the camera's status to 'Enabled' or 'Online' and confirms the action to the Homeowner.
Exception	- Command Failure: → .1: If the system is unable to communicate with the camera during either an enable or disable attempt due to a network issue, the operation is cancelled and an error message is displayed (e.g., "Failed to change camera status. Please check its connection.").
Priority	High
Frequency of use	Low to Medium (Used whenever privacy is temporarily needed)
Channel to actor	Mobile app, Web browser (in camera settings or control interface)
Secondary actors	1. Camera 2. SafeHome System Hub 3. SafeHome Cloud Server
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server <-> System Hub: Internet 3. System Hub <-> Camera: LAN/Wi-Fi
Open issues	1. How should a 'Disabled' camera state be visually represented in the UI to clearly distinguish it from an 'Offline' or 'Locked' state? 2. Should disabling the camera also deactivate its microphone for sound-based event detection (e.g., dog barking, glass breaking)? 3. Following a system-wide reboot or power outage, should a camera that was previously disabled by a Homeowner remain disabled for privacy protection?

References on 2025.10.26 / 2025.10.29 Meeting

2.2 Recording and Evidence Management

2.2.1 Search and Playback Recordings - Use Case Diagram, Sequence Diagram

Use Case	2.2.1 Search and Playback Recordings
Primary Actor	Homeowner
Goal in context	To allow the Homeowner to quickly and accurately find and play back a specific past video clip by setting criteria like camera, date, or event type, instead of manually scrubbing through all recorded footage.

Precondition	1. The SafeHome system hub and cloud server are operating normally. 2. The Homeowner is logged into the mobile app or web interface. 3. At least one video has been previously recorded and is stored in a connected repository (local or cloud). 4. Recorded videos are saved with associated metadata, such as timestamps and event types.
Trigger	The Homeowner initiates a search for a past event, such as 'what happened at the front door last night' or 'what time the kids came home from school'.
Scenario	1. The Homeowner navigates to the 'View Recordings' or 'Playback' menu. 2. The Homeowner selects search criteria, such as a specific camera, a date from a calendar, and/or an event filter (e.g., 'Person Detected'). 3. The system queries the storage repository and displays the resulting recordings in a timeline or event list (NFR: search results appear within 3 seconds). 4. The Homeowner selects a specific video clip from the results to watch. 5. The system checks if the recording belongs to a password-protected camera. 6. (If protected) The system prompts for the password, which the Homeowner enters correctly. 7. The system retrieves the selected video file from storage and begins streaming it to the Homeowner's device. 8. The Homeowner uses playback controls (play, pause, fast-forward) to review the video.
Exception	- No Recordings Found: → .1: If no recordings match the specified criteria, the system displays a message such as "No recordings found for the selected date and filters." - Incorrect Password Entry: → .1: If the Homeowner enters the wrong password for a protected recording, the system denies playback and displays an "Incorrect password" message. - Video File Corrupted or Missing: → .1: If the selected video file cannot be played, the system displays an error message like "Unable to play video" or "File is corrupt." - Poor Network Connection (Playback Lag): → .1: If the Homeowner's network connection is unstable during cloud playback, the system may reduce stream quality to prevent buffering, or playback may fail.
Priority	Essential
Frequency of use	Medium (Used whenever a review of past footage is needed)
Channel to actor	Mobile app, Web browser

Secondary actors	1. SafeHome System Hub 2. SafeHome Cloud Server 3. Storage Repository (Local on Hub or Cloud Storage)
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server <-> System Hub: Internet 3. System Hub <-> Storage Repository: LAN/SATA
Open issues	1. How can the responsiveness of thumbnail previews be optimized to help Homeowners quickly identify the desired video? 2. What format and procedure should be provided for downloading or sharing specific video clips? 3. How should the playback UI visually distinguish between recordings that include audio and those that are video-only?

References on SEPA safehome dialog slide 29-31

2.2.2 Evidence Sharing and Export - Use Case Diagram, Sequence Diagram

Use Case	2.2.2 Evidence Sharing and Export
Primary Actor	Homeowner
Goal in context	To allow the Homeowner to securely download recorded evidence (e.g., video clips, images) or share it via a secure link for review, reporting, or backup.
Precondition	1. The Homeowner is logged into the mobile app or web interface. 2. The evidence (video clip, image, etc.) exists in the system's storage repository. 3. The SafeHome cloud server and storage repository are operational. 4. The Homeowner has the necessary permissions to access and export the evidence.
Trigger	The Homeowner selects a "Download" or "Share" action for a specific piece of recorded evidence from the event history or playback screen.
Scenario	1. The Homeowner finds a specific event clip they wish to save or share. 2. The Homeowner selects the 'Export' or 'Share' option for that clip. 3. The system presents available options, such as "Download to Device" and "Share Link." 4. (If "Download") The cloud server retrieves the video file from the storage repository, prepares it (e.g., as an MP4 file), and initiates a secure download to the Homeowner's device. 5. (If "Share Link") The cloud server generates a secure, time-limited access link for the video file. 6. The system confirms that the download has started or that the share link has been successfully created and copied. 7. The export or share action is recorded in the system's audit log for security traceability.

Exception	- File Unavailable or Corrupt: → .1: If the requested evidence file cannot be found or is corrupted in the storage repository, the system displays an error message (e.g., "File unavailable or corrupted."). - Network or Permission Error: → .1: If the download or link generation fails due to a network issue or insufficient permissions, the system logs the failure and notifies the Homeowner ("Export failed. Please check your connection or access rights."). - Link Generation Failure: → .1: If the cloud server is temporarily unable to generate a shareable link, the system displays a message ("Unable to generate share link. Please try again later.").
Priority	High
Frequency of use	Low to Medium (Used when reviewing or exporting important events)
Channel to actor	Mobile app, Web interface
Secondary actors	1. SafeHome Cloud Server 2. Storage Repository (Cloud or Local)
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server <-> Storage Repository: Secure internal network connection or API
Open issues	1. Should exported evidence be encrypted or watermarked with a timestamp and device ID to ensure authenticity? 2. Should the expiration times and access permissions (e.g., password protection) for shared links be Homeowner-configurable? 3. Should the system provide an option to automatically compress large video files before download to reduce file size?

References on 2025.10.29 Meeting

2.3 Surveillance Settings

2.3.1 Recording Settings - Use Case Diagram, Sequence Diagram

Use Case	2.3.1 Recording Settings
Primary Actor	Homeowner (with Admin privileges)
Goal in context	To allow an Admin Homeowner to configure how and where a camera's video is recorded (e.g., continuously or based on events) in order to secure evidence and review past events.
Precondition	1. The Homeowner is logged into the mobile app or web interface with Admin privileges.

	2. The target camera is registered with the system and is online. 3. A storage location (e.g., a local SD card is installed or a cloud subscription is active) is available.
Trigger	The Admin Homeowner navigates to a specific camera's settings menu to define its recording rules.
Scenario	1. The Homeowner selects a camera and navigates to its 'Recording Settings' menu. 2. The Homeowner chooses a recording method: a. Continuous Recording: Records video 24/7. b. Event Recording: Records video only when a specific event (e.g., motion) is detected. 3. The Homeowner selects the storage destination (e.g., Cloud, Local SD card). 4. The Homeowner configures additional options, such as enabling or disabling audio recording. 5. The Homeowner saves the settings. 6. The cloud server validates and stores the new recording configuration and pushes the update to the hub. 7. The system now automatically records video from that camera according to the saved configuration.
Exception	- Local Storage Full: → .1: During recording, if local storage runs out of space, the system will either overwrite the oldest footage or stop recording and send a "Storage Full" notification to the Homeowner, based on system policy. - Cloud Connection Failure: → .1: If a video upload to the cloud fails due to a network outage, the system may buffer footage locally (if possible) and retry the upload when the connection is restored. A "Cloud Upload Failed" notification may be sent. - Cloud Subscription Expired: → .1: If the Homeowner's cloud storage subscription expires, the system stops cloud recording and sends a notification to the Homeowner to renew the subscription.
Priority	Essential
Frequency of use	Low (for configuration); the resulting recording process is constant.
Channel to actor	Mobile app, Web browser (in camera settings)
Secondary actors	1. Camera 2. SafeHome System Hub 3. SafeHome Cloud Server 4. Storage Repository (Local SD card or Cloud Storage)

Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server <-> System Hub: Internet 3. System Hub <-> Camera/Storage Repository: LAN/Wi-Fi/SATA
Open issues	1. For event recording, what duration of pre-roll (before event) and post-roll (after event) video should be captured? Should this be Homeowner-configurable? 2. What is the video retention policy for cloud storage (e.g., 7 days, 30 days), and how is it tied to subscription plans? 3. How can the system mitigate the risk of video loss if a local storage device fails or is stolen (e.g., offer a cloud backup option for critical events)?

References on SEPA safehome dialog slide 29-31

2.3.2 Notification Policy and Cooldown - Use Case Diagram, Sequence Diagram

Use Case	2.3.2 Notification Policy and Cooldown
Primary Actor	Homeowner (with Admin privileges)
Goal in context	To allow an Admin Homeowner to customize the configuration of a registered sensor or camera (e.g., sensitivity, alert interval, power mode) to fine-tune its behavior.
Precondition	1. The Homeowner is logged into the mobile app or web interface with Admin privileges. 2. The target device is registered with the system and is online.
Trigger	The Admin Homeowner navigates to a specific device's settings menu to modify its operational parameters.
Scenario	1. The Homeowner navigates to a specific device's 'Settings' menu. 2. The system displays the current, adjustable parameters for that device (e.g., Motion Sensitivity, Notification Frequency, Power Mode). 3. The Homeowner edits the desired values within the valid ranges presented in the UI. 4. The Homeowner saves the changes. 5. The app sends the updated configuration to the cloud server, which validates the request and forwards it to the hub. 6. The hub applies the new configuration directly to the physical device. 7. The system confirms that the settings were saved successfully and updates the UI to reflect the new state. The change is recorded in the audit log.
Exception	- Device is Offline: → .1: If the device is not reachable when the Homeowner attempts to save, the system rejects the change and displays an error ("Device not reachable. Settings could not be applied.")." - Invalid Input: → .1: If the Homeowner enters a value outside the permitted range for a

	setting, the UI provides real-time validation feedback and prevents saving until the value is corrected. - Concurrent Edit Conflict: → .1: If the device's configuration was changed by another session, the save command is rejected, and the Homeowner is prompted to refresh to get the latest settings.
Priority	Medium
Frequency of use	Low to Medium (Used when device behavior needs fine-tuning)
Channel to actor	Mobile app, Web interface (in device settings)
Secondary actors	1. The specific Device being configured 2. SafeHome System Hub 3. SafeHome Cloud Server
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server <-> System Hub: Internet 3. System Hub -> Device: LAN/Wi-Fi/Zigbee
Open issues	1. Should the system provide setting "presets" (e.g., Low, Medium, High sensitivity) in addition to custom values to simplify configuration? 2. Should changing certain critical settings be restricted while the security system is in an 'Armed' state? 3. Should device setting changes be logged in the main, user-visible activity feed, or only in a more detailed, admin-level audit log?

References on 2025.10.29 Meeting

3. System and User Management

3.1 Device Management

3.1.1 Add and Configure New Devices - Use Case Diagram, Sequence Diagram

Use Case	3.1.1 Add and Configure New Devices
Primary Actor	Homeowner (with Admin privileges)
Goal in context	To allow an Admin Homeowner to seamlessly discover, register, and perform the initial configuration of a new sensor or camera, integrating it into the SafeHome system.
Precondition	1. The Homeowner is logged into the mobile app or web interface with Admin privileges. 2. The new device is powered on and in a pairing or discovery mode, within wireless range of the SafeHome hub. 3. The SafeHome hub is online and operational.
Trigger	The Admin Homeowner navigates to the 'Devices' section of the application and initiates the 'Add New Device' process.

Scenario	1. The Homeowner selects 'Add New Device' in the app. 2. The app instructs the hub to enter device discovery mode and scan for nearby compatible devices. 3. The system detects the new device and displays it in the app for the Homeowner to select. 4. The Homeowner selects the device to add. 5. The hub securely registers the device, and this registration is synchronized with the cloud server. 6. The app prompts the Homeowner to assign the device a name (e.g., "Living Room Camera") and a room. 7. After naming, the Homeowner is guided to an initial configuration screen to set basic parameters (e.g., sensitivity, alert modes), as per UC 2.3.2 (Adjust Device Settings). 8. The Homeowner saves the configuration, and the system confirms that the "Device was successfully added and is now active."
Exception	- No New Devices Detected: → .1: If the system cannot find any new devices in discovery mode, the app displays troubleshooting tips (e.g., "Ensure the device is powered on and in pairing mode."). - Incompatible Device: → .1: If a detected device is not compatible with the SafeHome system, the app displays an error message indicating incompatibility. - Registration or Configuration Failure: → .1: If the device fails to register with the hub or apply the initial configuration, the system displays an error and prompts the Homeowner to retry the process.
Priority	High
Frequency of use	Low (Used when setting up or expanding the system)
Channel to actor	Mobile app, Web interface
Secondary actors	1. The new Device (Sensor/Camera) 2. SafeHome System Hub 3. SafeHome Cloud Server
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server <-> System Hub: Internet 3. System Hub -> New Device: LAN/Wi-Fi/Zigbee
Open issues	1. Should the pairing process support alternative methods like QR code scanning or NFC for faster and more secure device identification? 2. What are the default settings and security mode assignments for a new device before the Homeowner configures it? 3. How does the system handle mandatory firmware updates for a newly added device as part of the setup process?

References on SEPA safehome dialog slide 58

3.2 System Status and Logs

3.2.1 System Status Dashboard - Use Case Diagram, Sequence Diagram

Use Case	3.2.1 System Status Dashboard
Primary Actor	Homeowner
Goal in context	To allow the Homeowner to view the current operational status of all connected devices in a unified dashboard—including network connectivity, battery level, and signal strength—for real-time system monitoring and maintenance awareness.
Precondition	1. The Homeowner is logged into the mobile app or web interface. 2. All devices (sensors, cameras, etc.) are registered with the system. 3. The SafeHome hub and cloud server are operational and collecting device status data.
Trigger	The Homeowner navigates to the 'Device Status' or 'Health Dashboard' section of the application.
Scenario	1. The app requests the latest device status data from the cloud server. 2. The cloud server aggregates the most recent information from the hub for all connected devices, including connection state, battery level, and signal strength. 3. The app displays a 'Status Dashboard' with summary information (e.g., "Total devices online/offline," "Devices with low battery"). 4. The Homeowner can drill down into a detailed list view of all devices. 5. This detailed view shows a card for each device with its name, location, and specific health metrics. 6. The Homeowner can sort or filter this list (e.g., "Show only offline devices," "Sort by battery level"). 7. The system automatically sends a notification to the Homeowner when a device's health becomes critical (e.g., "Front Door Sensor battery is low" or "Living Room Camera is offline").
Exception	- Device Data Unavailable: → .1: If the cloud server has not received a recent status update from a specific device, it will display the last known data with a timestamp and may be marked as 'Status Unavailable.' - Cloud Connectivity Failure: → .1: If the app cannot reach the cloud server, it displays an error message ("Dashboard temporarily unavailable.") and retries the connection.
Priority	High
Frequency of use	High (for passive monitoring); Medium (for active checking)
Channel to actor	Mobile app, Web interface (Dashboard view)

Secondary actors	1. All Connected Devices (providing telemetry) 2. SafeHome System Hub 3. SafeHome Cloud Server
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server <-> System Hub: Internet 3. System Hub <-> Connected Devices: LAN/Wi-Fi/Zigbee
Open issues	1. Should the dashboard support historical trend graphs for key metrics, such as a device's battery level over the past 30 days? 2. Should critical devices (e.g., smoke detectors) have a higher priority for alerts and be displayed more prominently on the dashboard? 3. Should the Homeowner be able to export a system health report in a common format like PDF or CSV for maintenance records?

References on 2025.10.29 Meeting

3.2.2 Activity Logs and Timeline - Use Case Diagram, Sequence Diagram

Use Case	3.2.2 Activity Logs and Timeline
Primary Actor	Homeowner
Goal in context	To allow the Homeowner to review a chronological history of all system events—including sensor detections, device controls, user actions, and state changes—to monitor activity, investigate incidents, or verify system operation.
Precondition	1. The Homeowner is logged into the mobile app or web interface. 2. The SafeHome hub and cloud server are operational and have been recording events. 3. Sufficient log data exists in the system's database.
Trigger	The Homeowner navigates to the 'Activity Log' or 'Timeline' section of the application.
Scenario	1. The app requests the most recent activity logs from the cloud server. 2. The cloud server retrieves the logs and sends them to the app. 3. The app displays the activity timeline in reverse chronological order, showing key details for each event (e.g., timestamp, event type, device name, and user who performed the action). 4. The Homeowner uses the filter and search functions to narrow down the results by date range, event type (e.g., 'Alarms only'), specific device, or user. 5. The system displays the filtered results. 6. The Homeowner selects a specific event to view more details. 7. The system displays an expanded view with the full event description, links to any associated media (e.g., a camera snapshot), and context of related events. 8. The Homeowner can choose to export the filtered log data in a common

	format like PDF or CSV.
Exception	- No Activity Records Found: → .1: If no events match the selected filter criteria, the system displays a message and suggests adjusting the search. - Large Data Set: → .1: If a query returns a very large number of records, the system uses pagination to display the results efficiently. - Detailed Information Unavailable: → .1: If a log entry is corrupted or has been purged due to retention policies, the system displays all available information with a note indicating that some details are missing. - Export Fails: → .1: If a report fails to generate, the system displays an error and may offer an alternative format or suggest narrowing the scope.
Priority	High
Frequency of use	Medium to High
Channel to actor	Mobile app, Web interface
Secondary actors	1. SafeHome System Hub 2. SafeHome Cloud Server 3. Storage Repository (for logs and media)
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server <-> System Hub: Internet 3. Cloud Server <-> Storage Repository: Secure internal network connection or API
Open issues	1. What is the log retention policy (e.g., 30 days, 90 days), and should it differ based on event type or subscription level? 2. How can the system ensure log integrity to prevent tampering? 3. What level of detail in the activity log should be visible to non-Admin Homeowners or Guests? 4. Should video clips be automatically and permanently linked to their corresponding sensor events in the log?

References on SEPA safehome dialog slide 39

3.3 User and Permission Management

3.3.1 User Role and Access Control - Use Case Diagram, Sequence Diagram

Use Case	3.3.1 User Role and Access Control
Primary Actor	Homeowner (with Admin privileges)

Goal in context	To allow an Admin Homeowner to define what each user (e.g., family members, guests) can and cannot do by assigning specific permissions for security, surveillance, and system configuration.
Precondition	1. The Homeowner is logged into the mobile app or web interface with Admin privileges. 2. At least one other user account exists in the system.
Trigger	The Admin Homeowner navigates to 'User Account Management' to configure the permissions for a specific user account.
Scenario	1. The Homeowner selects a user account to manage from the user list. 2. The system displays a permission dashboard organized by categories: Security, Surveillance, and Configuration. 3. The Homeowner can either select a predefined template (e.g., 'Family Member', 'Guest') to apply a standard set of permissions, or manually configure them. 4. For manual configuration: a. Security: The Homeowner grants or denies access to actions like arming/disarming the system, viewing security logs, or using the Panic Button. b. Surveillance: The Homeowner specifies which cameras the user can view (e.g., all cameras or only specific ones like the 'Front Door Camera') and whether they can access recordings. c. Configuration: The Homeowner determines if the user can change system settings, such as adding new devices or editing other user accounts. 5. The Homeowner saves the new permission set. 6. The cloud server validates and atomically applies the permissions to the user's account, logging the change for auditing purposes. 7. The system confirms the update and may invalidate the affected user's active sessions to enforce the new rules immediately.
Exception	- Attempt to Modify Own Privileges: → .1: If an Admin Homeowner attempts to remove their own administrative permissions in a way that would leave no admins on the account, the system rejects the change with a warning. - Permission Conflict Detected: → .1: If the selected permissions create a logical conflict (e.g., allowing a user to manage recordings for a camera they are not allowed to view), the system highlights the conflict and requires correction before saving. - Invalid Configuration: → .1: If the Homeowner assigns permissions related to a non-existent device or zone, the system displays a warning. - Save Fails: → .1: If the system fails to save the permissions due to a server error, it rolls back any partial changes and displays an error message.
Priority	Essential

Frequency of use	Low (Used when adding new users or changing roles)
Channel to actor	Mobile app, Web interface (in User Management settings)
Secondary actors	1. SafeHome Cloud Server 2. SafeHome System Hub
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server <-> System Hub: Internet
Open issues	1. Should permissions support temporary access with an automatic expiration date (e.g., for a temporary guest)? 2. What is the secure process for an emergency access override if an Admin Homeowner is unavailable? 3. Should permission changes trigger a mandatory notification to the affected user, explaining what has changed? 4. Should the system support location-based permissions (e.g., a user can only disarm the system when their phone is detected at home)?

References on SEPA safehome dialog slide 70

4. Remote Access and Account

4.1 Account Management

4.1.1 Sign Up - Use Case Diagram, Sequence Diagram

Use Case	4.1.1 Sign Up
Primary Actor	New User
Goal in context	To create a new, secure SafeHome account by verifying a valid email address and phone number, in order to use SafeHome products and services.
Precondition	1. The New User has access to the SafeHome mobile app or website with an active internet connection. 2. The New User has a valid, accessible email address and a mobile phone. 3. The New User provides consent for data processing and communications.
Trigger	The New User selects the 'Sign Up' option, which is protected by human verification (e.g., CAPTCHA) and rate limits to prevent abuse. The client generates a unique request ID to ensure the account creation step is idempotent.
Scenario	1. The New User enters their details. Passwords must meet complexity rules (e.g., ≥12 characters) and are checked against lists of commonly breached passwords. 2. The New User agrees to the Terms of Service and Privacy Policy; this

	acceptance is audit-logged. 3. The New User initiates phone verification, and the cloud sends a time-limited (e.g., 5-minute) one-time password (OTP) via SMS. 4. The New User enters the received OTP, which is validated by the cloud. 5. The New User selects 'Create Account'. The request is sent with the idempotent ID and human-verification token. 6. The cloud server performs final validation, checking that the email and phone number are unique. All passwords are stored as salted hashes (e.g., using Argon2 or bcrypt). 7. The server atomically creates the account in a 'Pending Verification' state. 8. The server sends a single-use, time-limited (e.g., 24-hour) activation link to the New User's email. 9. The account is created but remains inactive. Upon successful email verification, the account state transitions to 'Active', and this transition is logged.
Exception	- Email or Phone Already Exists: → .1: If the email or phone number is already registered, the request is rejected with a message indicating it is already in use. - Invalid OTP: → .1: If the New User enters an incorrect or expired OTP, the system rejects it and allows a new OTP request after a cooldown. - Email Verification Link Expired: → .1: If the New User clicks an expired activation link, the system informs them and provides an option to resend the link. - External Gateway Outage: → .1: If the SMS or email provider is unavailable, the system displays an error and allows the New User to retry after a short delay. - Disposable Email Domain: → .1: If the email address is from a disallowed domain, the system rejects it and asks for a different email. - Email Bounce: → .1: If the activation email cannot be delivered (hard bounce), the system marks the email as unverified and prompts the New User to correct it.
Priority	Essential
Frequency of use	Very Low (Once per user)
Channel to actor	Mobile app, Web interface
Secondary actors	1. SafeHome Cloud Server 2. SMS Gateway Service 3. Email Service
Channels to secondary	1. App/Web <-> Cloud Server: Internet (HTTPS/TLS) 2. Cloud Server -> External Services: Internet (Secure API)

actors	All communications use TLS, and security tokens like OTPs and activation links are signed and time-limited.
Open issues	1. Should social login options (e.g., Google, Apple) be supported? If so, phone verification should still be required for account recovery and critical notifications. 2. What is the data retention policy for incomplete sign-ups (e.g., purge unverified accounts after 72 hours)? 3. After email verification, should the New User be automatically logged in, or redirected to the login page?

References on SEPA safehome dialog slide 41

4.1.2 Log In - Use Case Diagram, Sequence Diagram

Use Case	4.1.2 Log In
Primary Actor	Homeowner
Goal in context	To securely authenticate a registered Homeowner and establish a session, granting them authorized access to monitor and control their SafeHome system.
Precondition	1. The Homeowner has a verified, 'Active' account that is not suspended or locked. 2. The device has an active internet connection.
Trigger	The Homeowner enters their email and password and initiates the 'Log In' action. Login attempts are rate-limited and may require human verification (e.g., CAPTCHA) after repeated failures.
Scenario	(Based on a Homeowner logging in with 2FA enabled): 1. The Homeowner enters their email and password. 2. The app securely sends the credentials to the cloud server over an encrypted connection. 3. The cloud server validates the password against the stored salted hash. 4. The server checks if Two-Factor Authentication (2FA) is enabled for the account. 5. The server challenges the Homeowner for a 2FA code (e.g., from an authenticator app or SMS). 6. The Homeowner enters the 2FA code, which is validated by the server. 7. Upon successful authentication, the server generates and returns secure session tokens (e.g., a short-lived access token and a long-lived refresh token). 8. The app securely stores the tokens and logs the Homeowner in. The system logs the successful login event and may notify the Homeowner of a new device login.
Exception	- Invalid Credentials:

	→ .1: The email address or password does not match the records. - Account Lockout: → .1: The account is temporarily locked due to too many failed login attempts. - Account Not Verified: → .1: The account is still in a 'Pending Verification' state and cannot be used to log in. - Invalid 2FA Code: → .1: The Homeowner enters an incorrect or expired 2FA code. - Account Suspended: → .1: The account is in a suspended state and cannot be accessed. The system provides guidance to contact support. - SMS Delivery Failure: → .1: The 2FA code sent via SMS fails to be delivered. The system allows a resend attempt after a short delay. - Rate Limit Exceeded: → .1: Further login attempts are temporarily blocked due to rate limiting.
Priority	Essential
Frequency of use	High
Channel to actor	Mobile app, Web interface
Secondary actors	1. SafeHome Cloud Server 2. SMS Gateway Service (for 2FA)
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server -> SMS Gateway: Internet (Secure API)
Open issues	1. What is the session management policy, including token rotation and expiration (e.g., 14 days for web, 30 for mobile)? How are sessions handled after a password reset? 2. What is the "Forgot Password" workflow, and when should it be presented to the Homeowner? 3. How are new device logins communicated to the Homeowner, and what tools are provided for managing active sessions (e.g., revoking access for a specific device)?

References on SEPA safehome dialog slide 42

4.1.3 Log Out - Use Case Diagram, Sequence Diagram

Use Case	4.1.3 Log Out
Primary Actor	Homeowner (Logged-in)
Goal in	To allow a Homeowner to securely terminate their current session on a

context	specific device, ensuring that both client-side and server-side tokens are invalidated to prevent unauthorized access.
Precondition	1. The Homeowner is currently logged into the app or web interface with a valid session. 2. The device has an active internet connection to communicate with the cloud server.
Trigger	The Homeowner selects the 'Log Out' option from the application's menu and confirms their intention in a confirmation dialog.
Scenario	1. The Homeowner confirms their decision to log out. 2. The app sends a secure logout request for the current session to the cloud server. 3. The cloud server validates the request and revokes the server-side session token, effectively ending the session on the server. 4. The server also invalidates any push notification tokens associated with that specific session to stop sending alerts to the logged-out device. 5. The cloud server returns a success response. 6. Upon receiving confirmation, the app securely deletes all local session data, including access tokens and cached credentials. 7. The session is terminated, and the Homeowner is redirected to the login screen. The logout event is recorded in the audit log.
Exception	- Homeowner Cancels Logout: → .1: If the Homeowner cancels the confirmation dialog, the process is aborted, and the session remains active. - Server is Unreachable: → .1: If the cloud server cannot be reached, the app clears all local session data to log the Homeowner out on the device. A queued logout request may be sent to the server upon reconnection to invalidate the server-side token. - Invalid Session Token: → .1: If the session token is already expired or invalid when the logout request is made, the server handles the request gracefully and returns a success response, as the session is already terminated.
Priority	Essential
Frequency of use	Low to Medium
Channel to actor	Mobile app, Web interface
Secondary actors	1. SafeHome Cloud Server
Channels to secondary actors	1. App/Web -> Cloud Server: Internet (HTTPS)
Open issues	1. Should a "Log Out From All Devices" feature be provided that revokes all

	active sessions for the Homeowner's account? 2. What is the policy for clearing sensitive cached data (e.g., camera snapshots) upon logout, versus retaining non-sensitive user preferences? 3. Should the app provide a "Manage Sessions" screen where a Homeowner can view and remotely revoke sessions on other devices? 4. Should all active sessions be automatically revoked upon a password reset or when an account compromise is reported?

References on SEPA safehome dialog slide 43

4.1.4 Password Recovery and Reset - Use Case Diagram, Sequence Diagram

Use Case	4.1.4 Password Recovery and Reset
Primary Actor	Homeowner
Goal in context	To allow a Homeowner who has forgotten their password to securely reset it and regain access to their account.
Precondition	1. The Homeowner has an existing, active account. 2. The Homeowner has access to the verified email address or phone number associated with the account.
Trigger	The Homeowner selects the 'Forgot Password?' link on the login screen. Password reset attempts are rate-limited per account and IP address to prevent abuse.
Scenario	1. The Homeowner enters their account email address. 2. The cloud server verifies the account exists and sends a password reset link to the registered email address. The link is single-use and time-limited (e.g., 1 hour). 3. The Homeowner clicks the link in the email and is directed to a secure password reset page. 4. The Homeowner enters a new password and confirms it. 5. The cloud server validates that the new password meets security requirements (e.g., complexity, not previously used) and updates the account. 6. Upon a successful password change, the system revokes all other active sessions for the account as a security measure. 7. The system notifies the Homeowner that their password has been changed successfully and prompts them to log in with the new password. The reset event is audit-logged.
Exception	- Account Not Found: → .1: If the entered email does not match an existing account, the system displays a generic message to prevent account enumeration (e.g., "If an account exists for this email, a reset link has been sent."). - Reset Link Expired or Invalid: → .1: If the Homeowner uses an expired or invalid link, the system informs them and provides an option to restart the process.

	- New Password Fails Security Requirements: → .1: If the new password is too weak or does not match the confirmation, the system displays a specific error message outlining the requirements.
Priority	Essential
Frequency of use	Low (Used only when a Homeowner forgets their password)
Channel to actor	Mobile app, Web browser
Secondary actors	1. SafeHome Cloud Server 2. Email Service 3. SMS Gateway Service (for notification)
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server -> External Services: Internet (Secure API)
Open issues	1. What is the optimal expiration time for password reset links (e.g., 1 hour, 24 hours)? 2. What is the specific password policy (e.g., minimum length, complexity, reuse prevention)? 3. Should the Homeowner be required to complete a 2FA challenge (if enabled) before being allowed to reset the password?

References on SEPA safehome dialog slide 44

4.1.5 Edit Profile Information - Use Case Diagram, Sequence Diagram

Use Case	4.1.5 Edit Profile Information
Primary Actor	Homeowner (Logged-in)
Goal in context	To allow a logged-in Homeowner to securely view and modify their personal account information, such as their name, phone number, or email address.
Precondition	1. The Homeowner is logged into the mobile app or web interface. 2. The Homeowner may be required to re-enter their password to access sensitive profile settings. 3. To change an email or phone number, the Homeowner must have access to the new email inbox or phone number to complete a verification step.
Trigger	The Homeowner navigates to the 'Account Settings' or 'Profile' screen and selects the option to edit their information.
Scenario	To Update Name: 1. The Homeowner edits the name field and saves the change. 2. The cloud server validates and updates the name in the database. 3. The system confirms the update was successful.

	To Update Email or Phone Number: 1. The Homeowner enters a new email address or phone number. 2. The cloud server sends a time-limited verification code (OTP) or link to the new address/number. 3. The Homeowner enters the received code or clicks the link to prove ownership. 4. Upon successful verification, the cloud server atomically updates the contact information in the Homeowner's profile. 5. As a security measure, a notification is sent to the old email address and/or phone number, informing them of the change. All changes are audit-logged.
Exception	- Invalid Input: → .1: If the Homeowner enters data in an invalid format (e.g., incorrect email structure), the system displays an error and prevents saving. - Verification Failure: → .1: If the Homeowner enters an incorrect or expired verification code, the change is rejected, and they are given the option to resend the code. - New Email/Phone Already in Use: → .1: If the new email or phone number is already registered to another account, the system rejects the change with an error message. - Database Update Failure: → .1: If the system is unable to save the changes due to a server-side issue, it displays an error message ("Unable to update profile. Please try again later.").
Priority	High (as it involves sensitive personal information)
Frequency of use	Low (Occasional)
Channel to actor	Mobile app, Web interface (in Account Settings)
Secondary actors	1. SafeHome Cloud Server 2. Email Service 3. SMS Gateway Service
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server -> External Services: Internet (Secure API)
Open issues	1. Should changing a primary email or phone number automatically revoke all other active sessions as a security precaution? 2. What is the policy on rate-limiting profile update attempts to prevent abuse? 3. Should there be a mandatory waiting period or "cooldown" after a critical profile change before other high-security actions (like password resets) are allowed?

References on SEPA safehome dialog slide 45

4.1.6 Change Password - Use Case Diagram, Sequence Diagram

Use Case	4.1.6 Change Password
Primary Actor	Homeowner (Logged-in)
Goal in context	To allow a logged-in Homeowner to securely change their account password from within the application settings.
Precondition	1. The Homeowner is logged into the mobile app or web interface. 2. The Homeowner knows their current password to authorize the change.
Trigger	The Homeowner navigates to 'Account Settings' and selects the 'Change Password' option.
Scenario	1. The Homeowner is prompted to enter their current password, their new password, and a confirmation of the new password. 2. The Homeowner enters the required information and confirms the action. 3. The app sends this data securely to the cloud server. 4. The cloud server first verifies that the provided 'current password' is correct. 5. The server then validates that the new password meets all security policy requirements (e.g., complexity, not previously used). 6. Upon successful validation, the server updates the password hash in the database. 7. As a security measure, the system revokes all other active sessions for the account. 8. The system confirms the password has been changed successfully and may terminate the current session, requiring the Homeowner to log in again with the new password. The event is audit-logged.
Exception	- Incorrect Current Password: → .1: If the Homeowner enters their current password incorrectly, the system rejects the change and displays an error message. - New Password Fails Security Requirements: → .1: If the new password does not meet the security policy or the confirmation does not match, the system displays a specific error message outlining the requirements. - Database Update Failure: → .1: If the system is unable to save the new password due to a server-side issue, it displays an error message ("Password change failed. Please try again later.").
Priority	High
Frequency of use	Low (Occasional)
Channel to actor	Mobile app, Web interface (in Account Settings)

Secondary actors	1. SafeHome Cloud Server
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS)
Open issues	1. Should an email and/or SMS notification be sent to the Homeowner after a successful password change as a security alert? 2. Should changing the password automatically invalidate all active sessions, including the current one, forcing a re-login everywhere? 3. Should a rate-limiting or temporary lockout mechanism be implemented after multiple failed attempts with an incorrect 'current password'?

References on SEPA safehome dialog slide 46

4.1.7 Two-Factor Authentication Management - Use Case Diagram, Sequence Diagram

Use Case	4.1.7 Two-Factor Authentication Management
Primary Actor	Homeowner (with Admin privileges)
Goal in context	To allow an Admin Homeowner to enable, configure, and manage Two-Factor Authentication (2FA) methods, including generating and viewing backup codes, to enhance account security.
Precondition	1. The Homeowner is logged into the mobile app or web interface with Admin privileges. 2. To modify existing 2FA settings, the Homeowner must be able to verify their identity using their current 2FA method or account password. 3. The Homeowner has access to the device (e.g., phone for authenticator app or SMS) they wish to configure.
Trigger	The Admin Homeowner navigates to the 'Security Settings' screen to enable or manage their 2FA configuration.
Scenario	To Enable 2FA: 1. The Homeowner selects "Enable 2FA" and chooses a method (e.g., Authenticator App, SMS Verification). 2. The system generates a secret key (presented as a QR code and manual entry key) for the chosen method. 3. The Homeowner uses their device to scan the QR code or enter the key, then enters the resulting verification code into the SafeHome app. 4. The cloud server validates the code and activates 2FA for the account. 5. The system generates a set of single-use backup codes and prompts the Homeowner to save them in a safe place, warning them that this is the only time they will be displayed. 6. The 2FA activation is confirmed, and the event is audit-logged.

	To Change 2FA Method or Manage Backup Codes: 1. The Homeowner navigates to the 2FA settings and selects an option like "Change Method" or "View Backup Codes." 2. The system requires the Homeowner to enter a code from their current 2FA method to authorize the change. 3. After successful verification, the Homeowner can either proceed with setting up a new 2FA method (which deactivates the old one upon completion) or view/regenerate their backup codes (which invalidates the old set).
Exception	- Invalid Verification Code: → .1: If the Homeowner enters an incorrect or expired code, the system rejects it and allows a limited number of retries before a temporary lockout. - 2FA Device or Setup Issue: → .1: If the Homeowner cannot scan the QR code, the system provides a manual entry key as an alternative. If SMS/email codes fail to deliver, a retry option is offered. - Backup Codes Not Saved: → .1: If the Homeowner attempts to complete setup without acknowledging they have saved their backup codes, the system displays a persistent warning about the risk of account lockout. - Maximum Attempts Exceeded: → .1: After too many failed verification attempts, the system temporarily locks the ability to set up or modify 2FA for a short period (e.g., 15 minutes) to prevent brute-force attacks.
Priority	High
Frequency of use	Low (Primarily for initial setup or when changing devices)
Channel to actor	Mobile app, Web interface (in Security Settings)
Secondary actors	1. SafeHome Cloud Server 2. SMS Gateway Service 3. Email Service
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server -> External Services: Internet (Secure API)
Open issues	1. Should the system support hardware security keys (e.g., YubiKey) as a premium 2FA method? 2. Should a 'trusted devices' feature be implemented to bypass 2FA for a set period (e.g., 30 days)? 3. What is the secure recovery process if a Homeowner loses access to all 2FA methods and their backup codes? 4. Should the system enforce mandatory 2FA for all Admin-level accounts?

References on SEPA safehome dialog slide 47

5. Indoor Monitoring and Device Control

5.1 Device Control

5.1.1 Indoor Device Control - Use Case Diagram, Sequence Diagram

Use Case	5.1.1 Indoor Device Control
Primary Actor	Homeowner, Guest
Goal in context	To allow a Homeowner or Guest to remotely control the state (on/off, brightness, color) of individual smart lights or groups of lights for convenience, security, or energy efficiency.
Precondition	1. The Homeowner or Guest is logged into the mobile app or web interface. 2. At least one smart light device is registered with the system and is online. 3. The SafeHome hub and cloud server are operational.
Trigger	The Homeowner or Guest selects a specific light, a pre-configured group of lights, or an "All Lights" option from the application to control it.
Scenario	1. The Homeowner selects a light (e.g., 'Living Room Lamp') from the app's interface. 2. The system displays the current status (e.g., On, 80% Brightness) and provides control options. 3. The Homeowner adjusts the controls (e.g., toggles it off, changes brightness to 50%). 4. The app sends the control command to the cloud server. 5. The cloud server relays the command to the hub. 6. The hub sends the command to the target light device via its wireless interface (e.g., Zigbee, Wi-Fi). 7. The light executes the command and reports its new state back to the hub. 8. The hub updates the cloud, which synchronizes the new status across all of the Homeowner's connected clients in real-time.
Exception	- Device Fails to Respond: → .1: If a light is unresponsive (e.g., due to a network error or power outage), the hub logs the command failure. → .2: The cloud server informs the Homeowner via the app that the device is unresponsive. → .3: The system may mark the light as 'Unavailable' in the UI until it comes back online.
Priority	High
Frequency of use	High

Channel to actor	Mobile app, Web interface
Secondary actors	1. Smart Light Device 2. SafeHome System Hub 3. SafeHome Cloud Server
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server <-> System Hub: Internet 3. System Hub -> Smart Light Device: LAN/Wi-Fi/Zigbee
Open issues	1. Should Homeowners be able to create and manage custom light groups (e.g., "Downstairs Lights") and scenes (e.g., "Movie Mode")? 2. How should the system handle and reflect status changes when a light is controlled by a physical wall switch? 3. Should the system support advanced scheduling features for lights (e.g., turn on porch light at sunset, "Wake-Up" lighting that gradually brightens)?

References on SEPA safehome dialog slide 39

5.2 Indoor Monitoring System

5.2.1 Indoor Air Quality Monitoring and Ventilation Integration - Use Case Diagram, Sequence Diagram

Use Case	5.2.1 Indoor Air Quality Monitoring and Ventilation Integration
Primary Actor	Homeowner, System (for automation)
Goal in context	To allow the Homeowner to monitor indoor air quality and to configure the system to automatically maintain a healthy environment by activating ventilation when pollutants (e.g., CO_2, VOCs) exceed predefined thresholds.
Precondition	1. The Homeowner has configured air quality thresholds or is using system defaults. 2. An Air Quality Sensor and a compatible Ventilation System are registered with the system and are online. 3. The SafeHome hub and cloud server are operational.
Trigger	An Air Quality Sensor reports a pollutant level that has exceeded the configured "unhealthy" threshold to the SafeHome hub.
Scenario	1. An Air Quality Sensor detects that CO_2 levels have exceeded the predefined threshold and reports the event to the hub. 2. The hub identifies the event as a trigger for the air quality automation. 3. The hub sends an 'activate' command to the linked ventilation system. 4. In parallel, the hub instructs the cloud server to send a notification to the Homeowner (e.g., "High CO_2 detected. Ventilation has been activated."). 5. The system continues to monitor the sensor. Once the CO_2 level drops back into the acceptable range, the hub sends a 'deactivate' command to the ventilation system.

	6. The hub reports the resolution to the cloud, and the system status (air quality level, ventilation state) is updated across all of the Homeowner's clients.
Exception	- Ventilation System Fails to Respond: → .1: If the ventilation system is unresponsive, the hub logs the command failure. → .2: The cloud sends a critical alert to the Homeowner ("Air quality is poor, and the ventilation system is unresponsive.")." → .3: The system may retry the command at defined intervals. - Air Quality Sensor is Offline: → .1: If the hub fails to receive periodic updates from the sensor, it flags the sensor as 'Offline' in the UI. → .2: A maintenance alert is sent to the Homeowner. - Cloud Connectivity Failure: → .1: If the cloud connection is lost, the hub continues to operate the automation locally based on sensor data. The UI may show a "Monitoring temporarily unavailable" status until the connection is restored.
Priority	High
Frequency of use	High (automatic and continuous background operation)
Channel to actor	Mobile app, Web interface (for notifications and monitoring)
Secondary actors	1. Air Quality Sensor 2. Ventilation System 3. SafeHome System Hub 4. SafeHome Cloud Server
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server <-> System Hub: Internet 3. System Hub -> Sensor/Ventilation System: LAN/Wi-Fi/Zigbee
Open issues	1. Should the system consider outdoor air quality data before activating ventilation to avoid bringing more polluted air inside? 2. Should air quality thresholds be Homeowner-configurable, or should they be based on fixed health standards (or a combination)? 3. Should the system provide an option to define quiet hours (e.g., "Do not activate ventilation between 12-6 AM") to avoid noise disruption?

References on SEPA safehome dialog slide 27

5.2.2 Real-Time Power Consumption Monitoring and Reporting - Use Case Diagram, Sequence Diagram

Use Case	5.2.2 Real-Time Power Consumption Monitoring and Reporting
Primary Actor	Homeowner

Goal in context	To allow the Homeowner to monitor real-time and historical power consumption patterns, enabling better energy management and potential cost savings.
Precondition	1. The Homeowner is logged into the mobile app or web interface. 2. A Smart Meter or other power monitoring sensor is registered with the system and is online. 3. The SafeHome hub and cloud server are operational.
Trigger	The Homeowner navigates to the 'Energy Management' section of the application to check current power usage or analyze historical consumption.
Scenario	1. The Homeowner opens the 'Energy' screen in the app. 2. The app requests the latest power usage data from the cloud server. 3. The cloud server retrieves the latest readings from the hub (which polls the Smart Meter). 4. The cloud server returns key metrics, such as current total power consumption (W) and historical usage trends (kWh). 5. The app displays a dashboard with real-time stats and charts for daily, weekly, and monthly usage. 6. The Homeowner selects an option to "View Report." 7. The cloud server generates a detailed report, including analysis of peak usage hours and comparisons with previous periods. 8. The Homeowner reviews the report in the app to identify energy-saving opportunities.
Exception	- Smart Meter Unresponsive: → .1: If the hub cannot retrieve data from the Smart Meter, it logs the failure. → .2: The cloud informs the Homeowner via the app that real-time data is unavailable and displays the last known reading with a timestamp. - Cloud Connectivity Failure: → .1: If the app cannot reach the cloud server, it displays an error message ("Real-time energy data temporarily unavailable."). Previously loaded historical data may still be viewable from a local cache. - Report Generation Failure: → .1: If the cloud server fails to generate a report due to a database error, it sends a notification to the app ("Report generation failed. Please try again later.").
Priority	Medium
Frequency of use	Medium to High
Channel to actor	Mobile app, Web interface (for monitoring and reports)
Secondary	1. Smart Meter (or power sensor)

actors	2. SafeHome System Hub 3. SafeHome Cloud Server
Channels to secondary actors	1. App/Web <-> Cloud Server: Internet (HTTPS) 2. Cloud Server <-> System Hub: Internet 3. System Hub -> Smart Meter: LAN/Wi-Fi/Zigbee
Open issues	1. Should the system support appliance-level analysis? (This would require additional sub-metering hardware). 2. What is the target refresh rate for "real-time" data (e.g., every 5 seconds, 15 seconds)? 3. Should the system support threshold-based alerts (e.g., "Notify me if my current usage exceeds 5kW")? 4. Should reports be exportable in common formats like PDF or CSV?

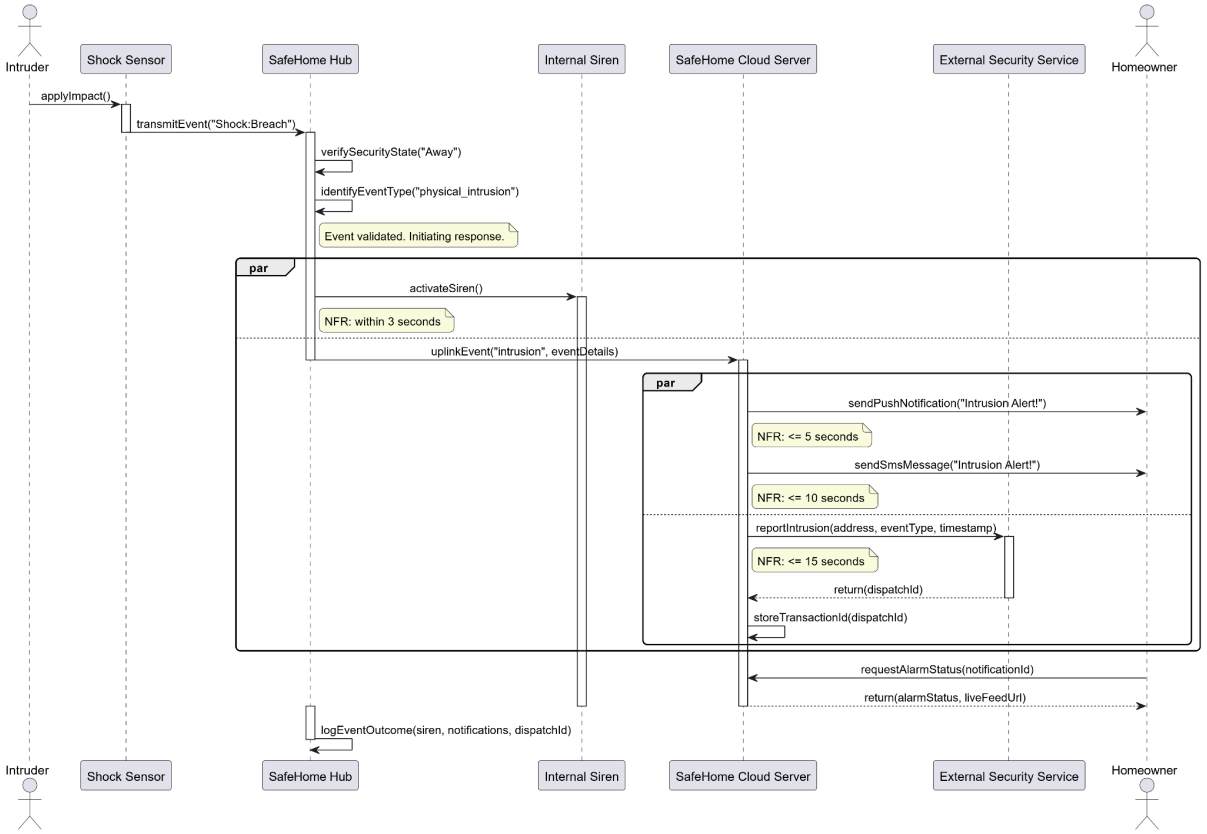
References on SEPA safehome dialog slide 29

VII. Sequence Diagram

1. Intelligent Security

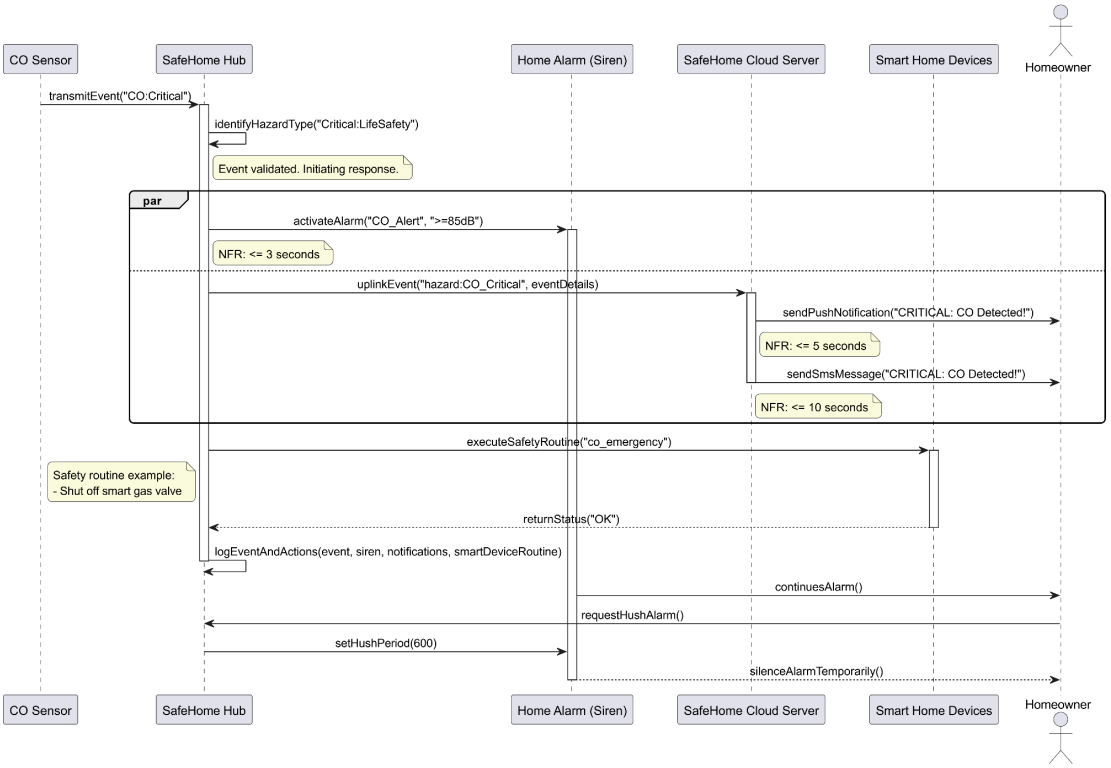
1.1. Sensor Monitoring

1.1.1 Physical Intrusion Detection - Use Case



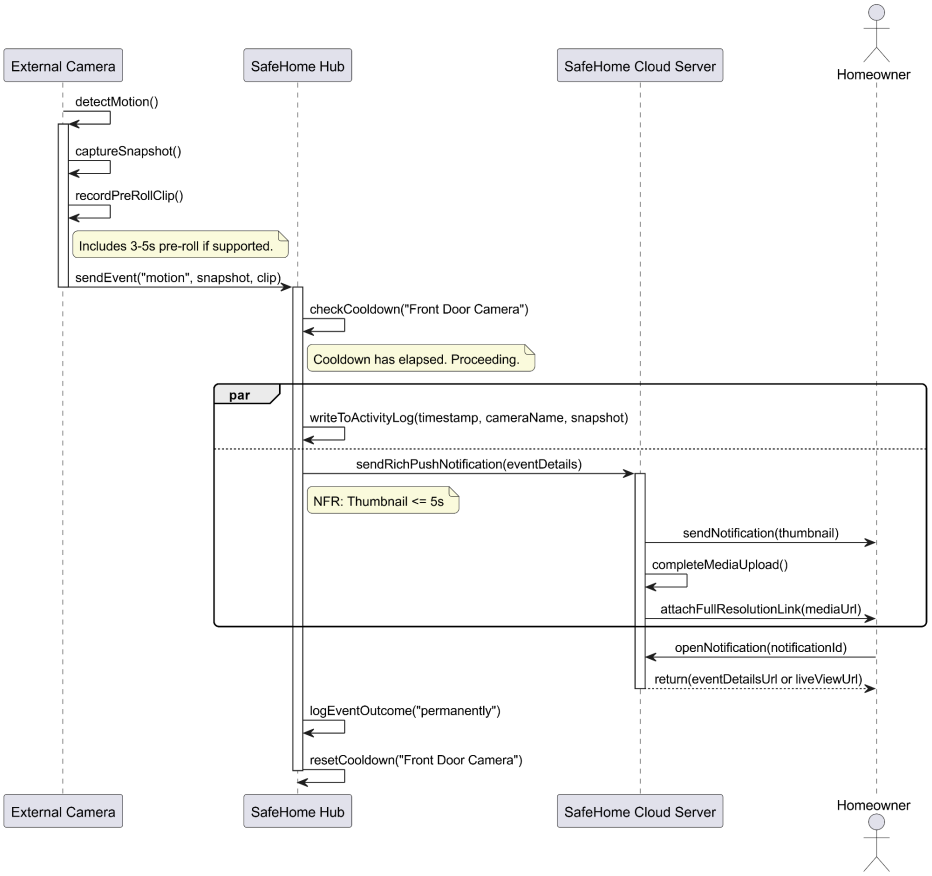
References on 2025.10.26 / 2025.10.29 Meeting

1.1.2 Environmental Hazard Detection - Use Case



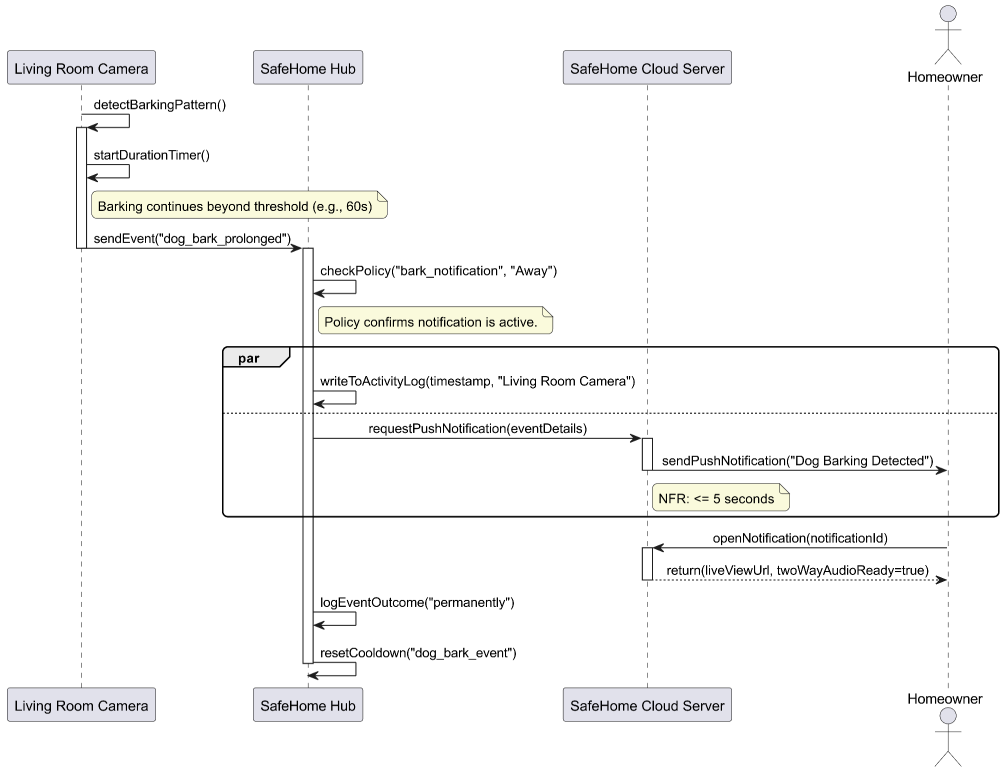
References on 2025.10.26 / 2025.10.29 Meeting

1.1.3 Outdoor Motion Detection - Use Case



References on 2025.10.26 / 2025.10.29 Meeting

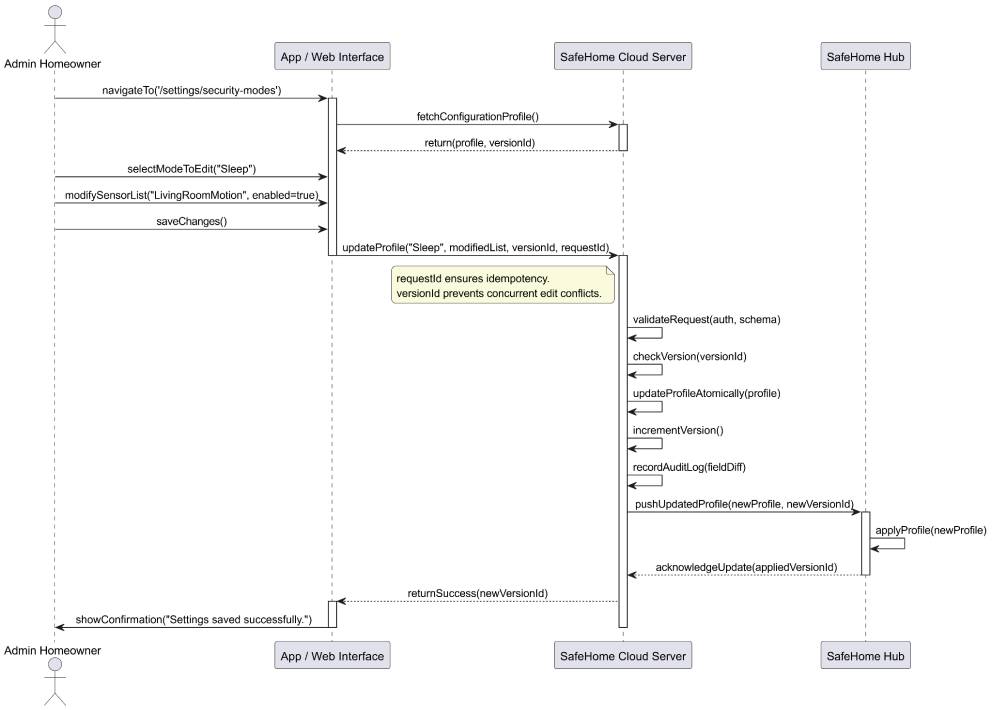
1.1.4 Dog Barking Detection - Use Case



References on SEPA safehome dialog slide 58-59

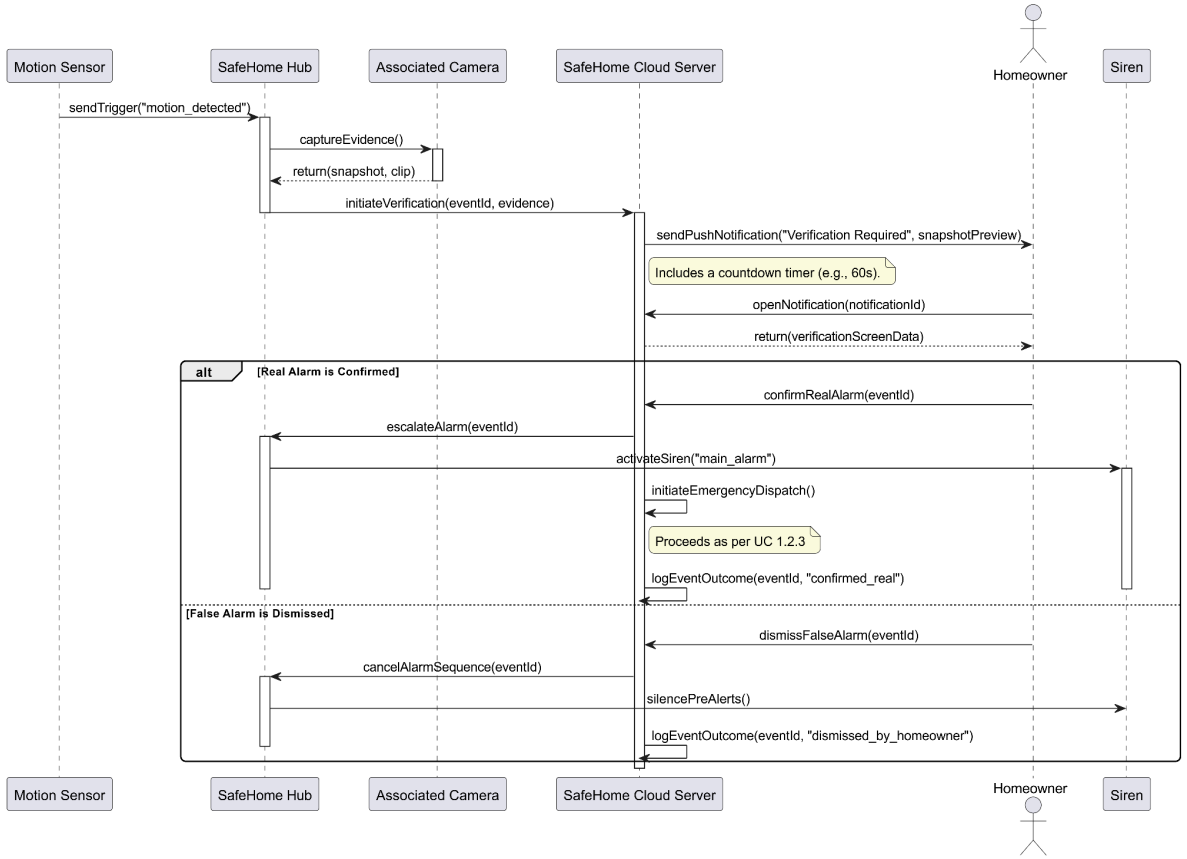
1.2. Incident Management

1.2.1 Configure Alarm Conditions by Security Mode - Use Case



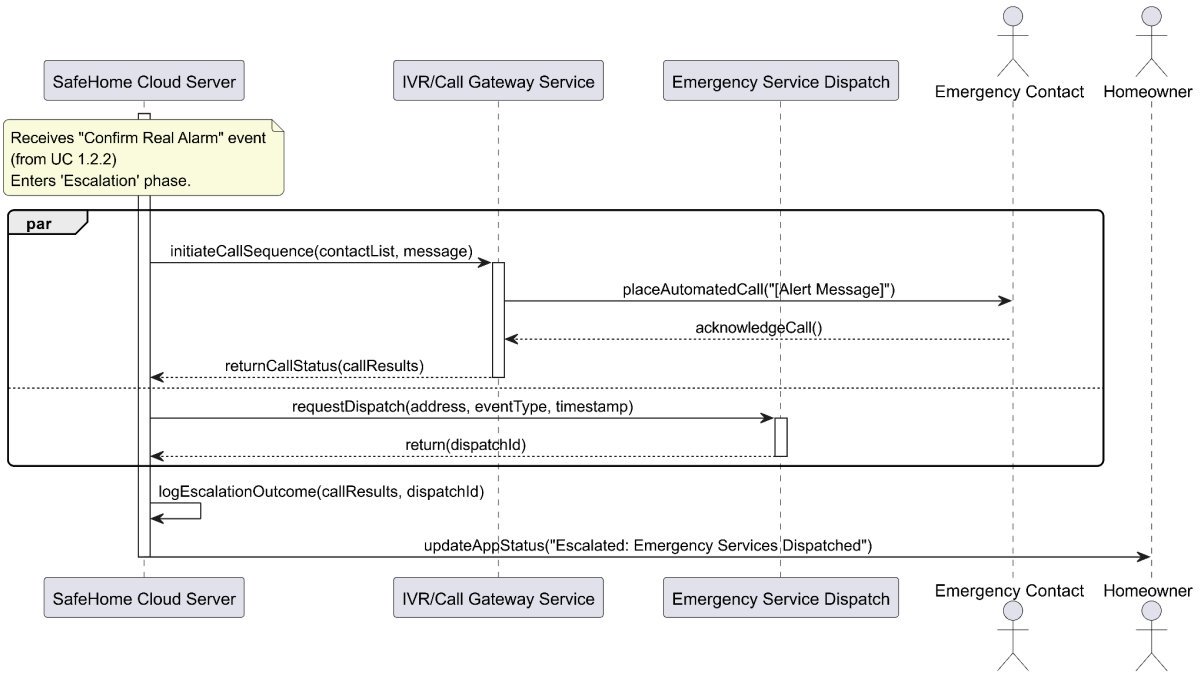
References on SEPA safehome dialog slide 5

1.2.2 Alarm Verification Step - Use Case



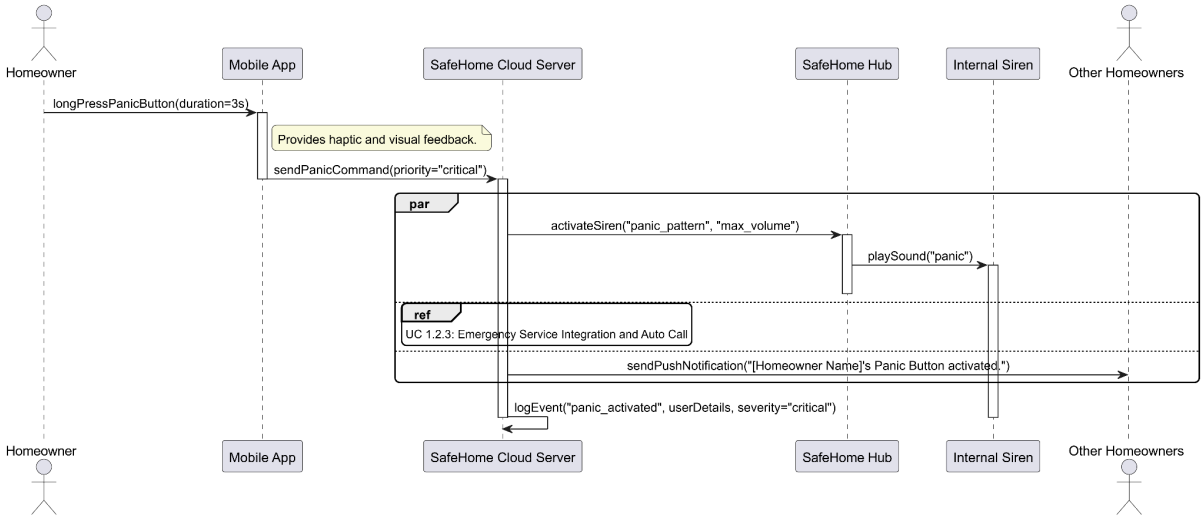
References on 2025.10.29 Meeting

1.2.3 Emergency Service Integration and Auto Call - Use Case



References on SEPA safehome dialog slide 6, 7

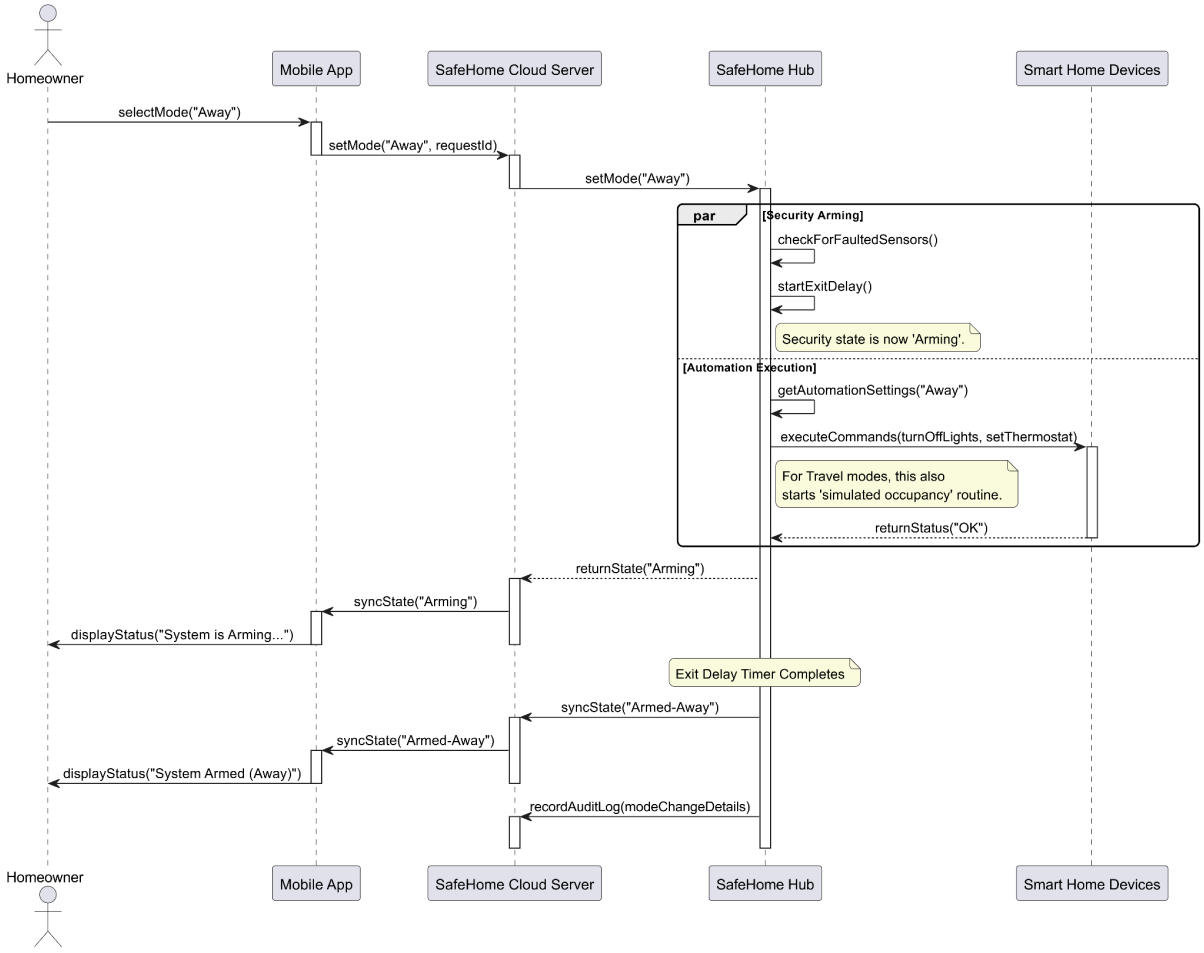
1.2.4 Panic Button - Use Case



References on 2025.10.29 Meeting

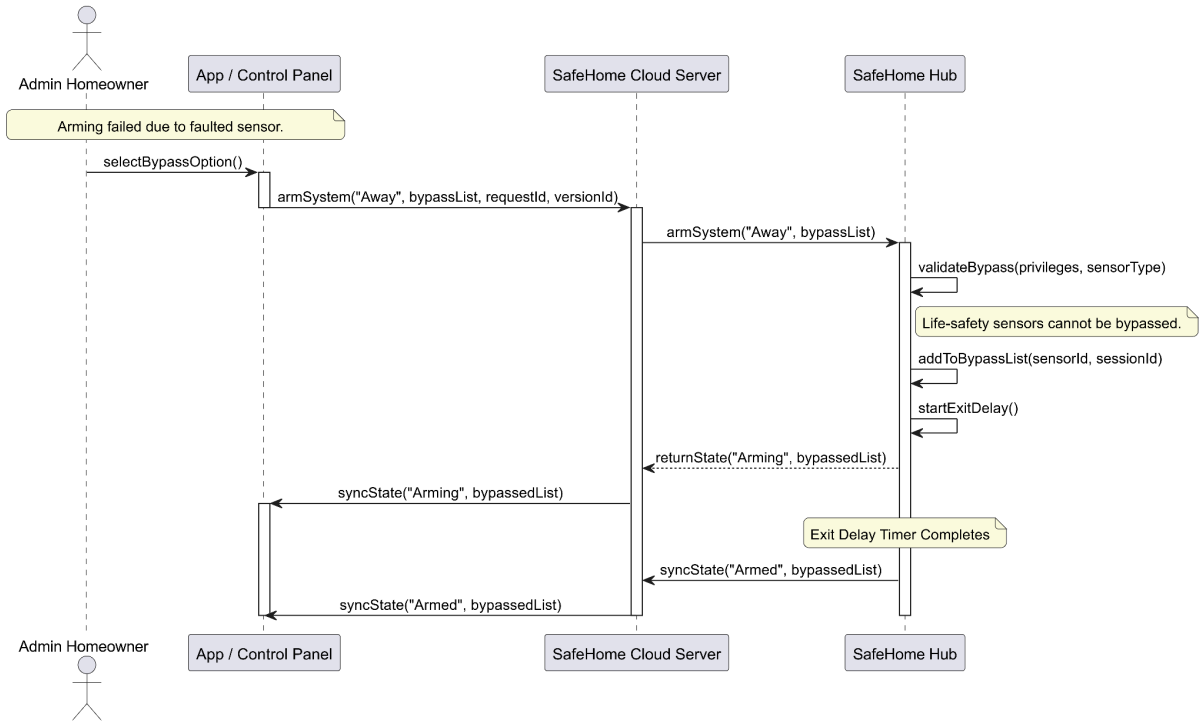
1.3 Security Mode Control

1.3.1 One-Touch Modes (Away, Home, Sleep) - Use Case



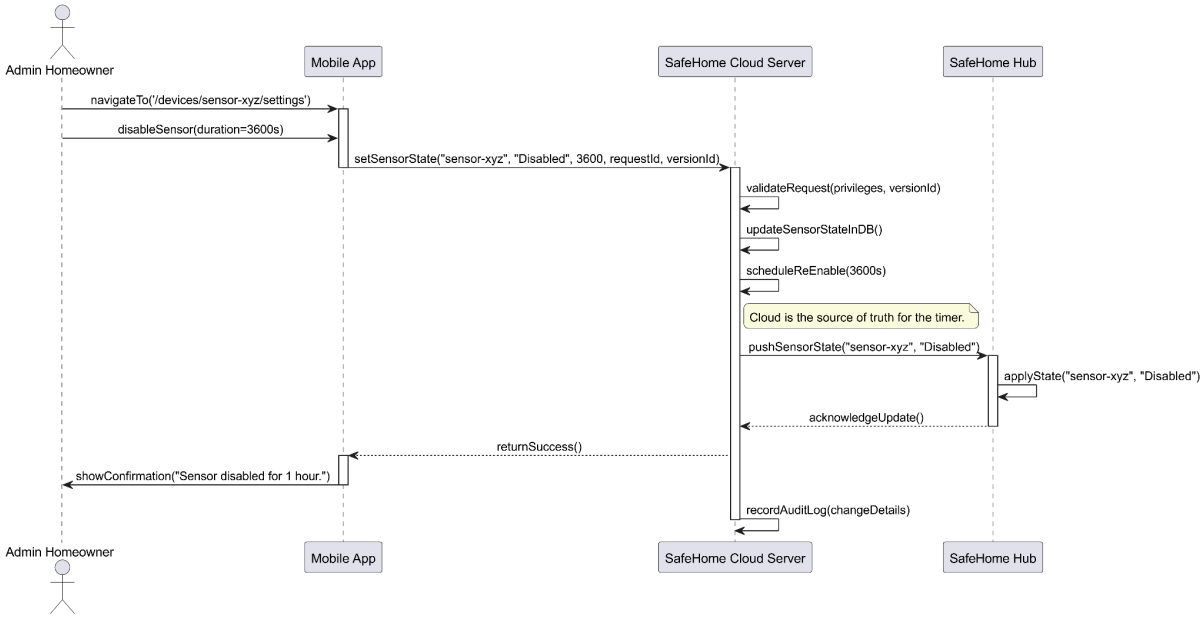
References on SEPA safehome dialog slide 9

1.3.2 Sensor Bypass - Use Case



References on SEPA safehome dialog slide 10

1.3.3 Sensor Activation and Deactivation - Use Case

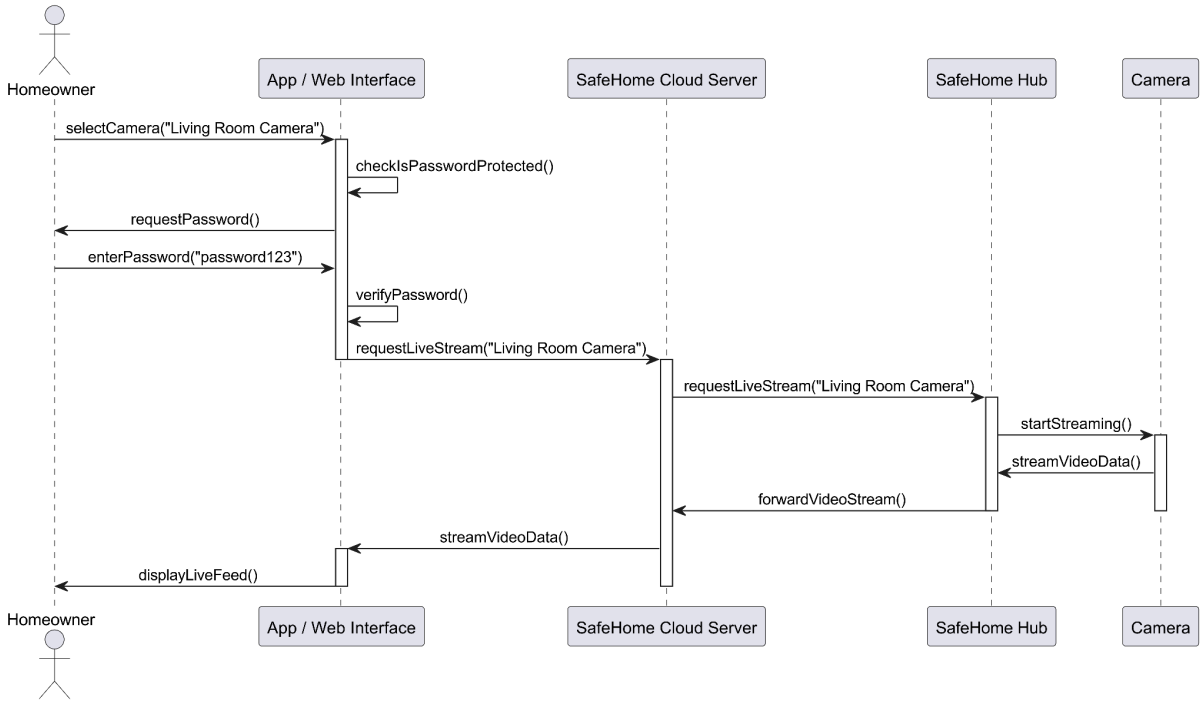


References on 2025.10.26 / 2025.10.29 Meeting

2. Live Surveillance

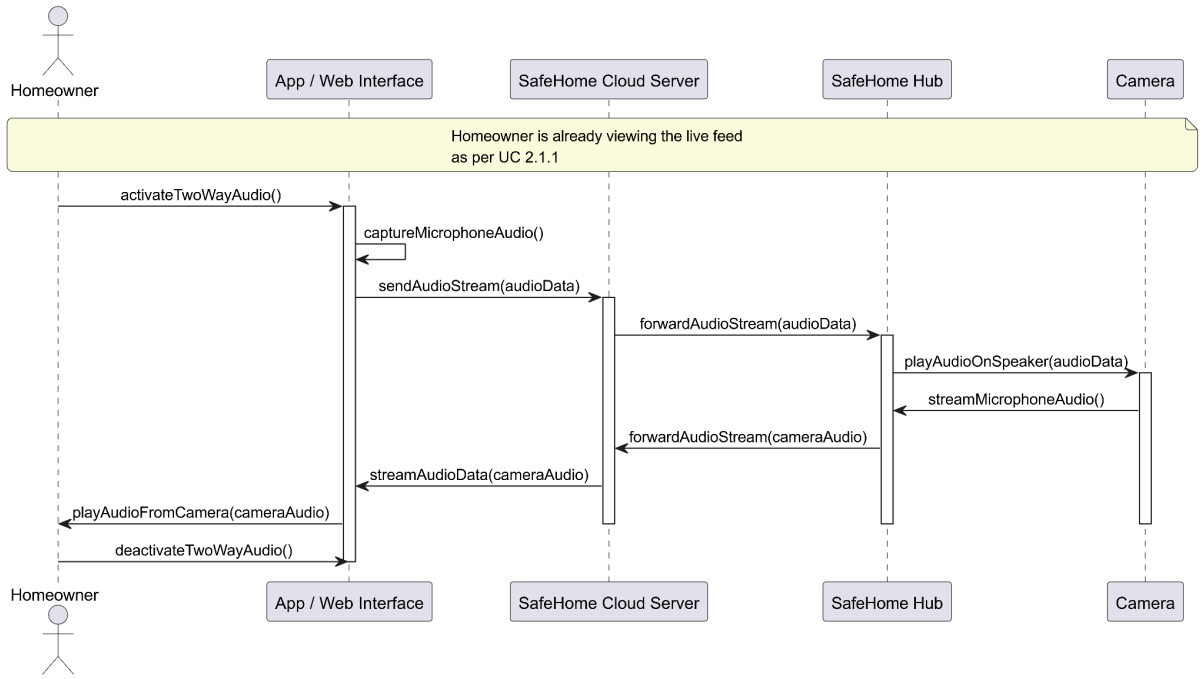
2.1 Camera Viewing and Control

2.1.1 Single Camera Live View - Use Case



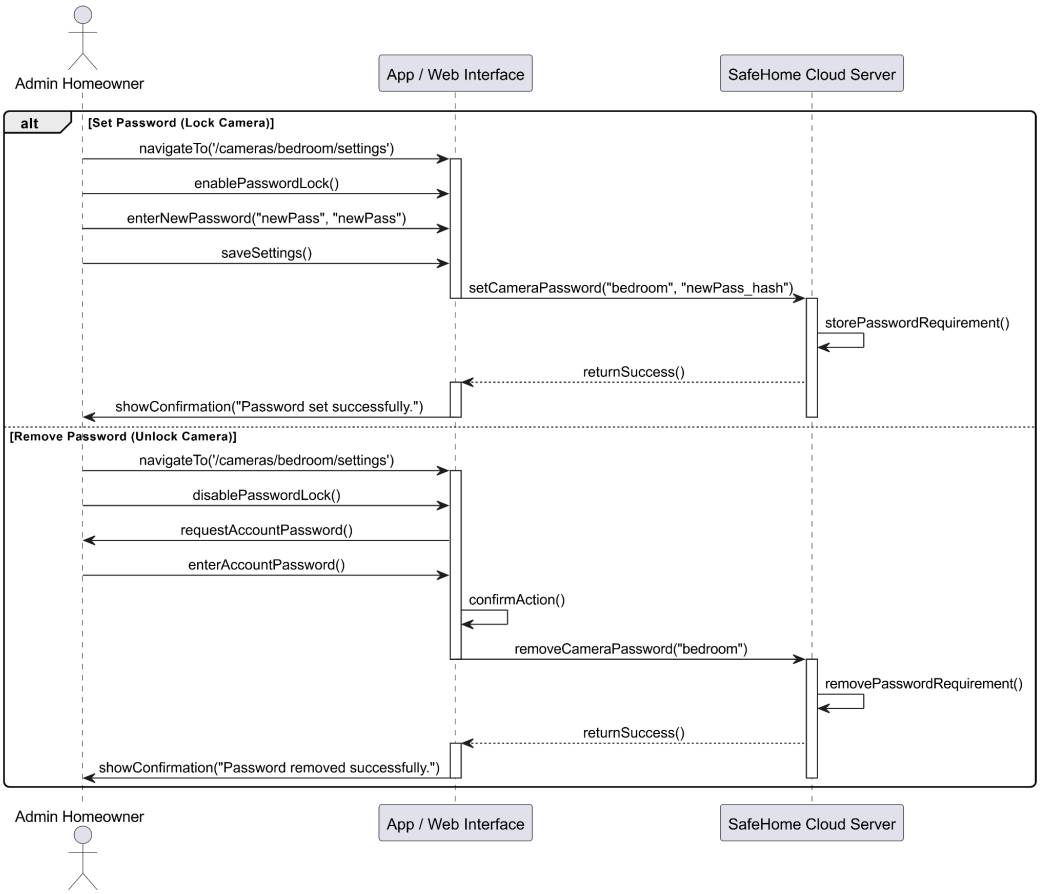
References on SEPA safehome dialog slide 29-31

2.1.2 Two-Way Audio - Use Case



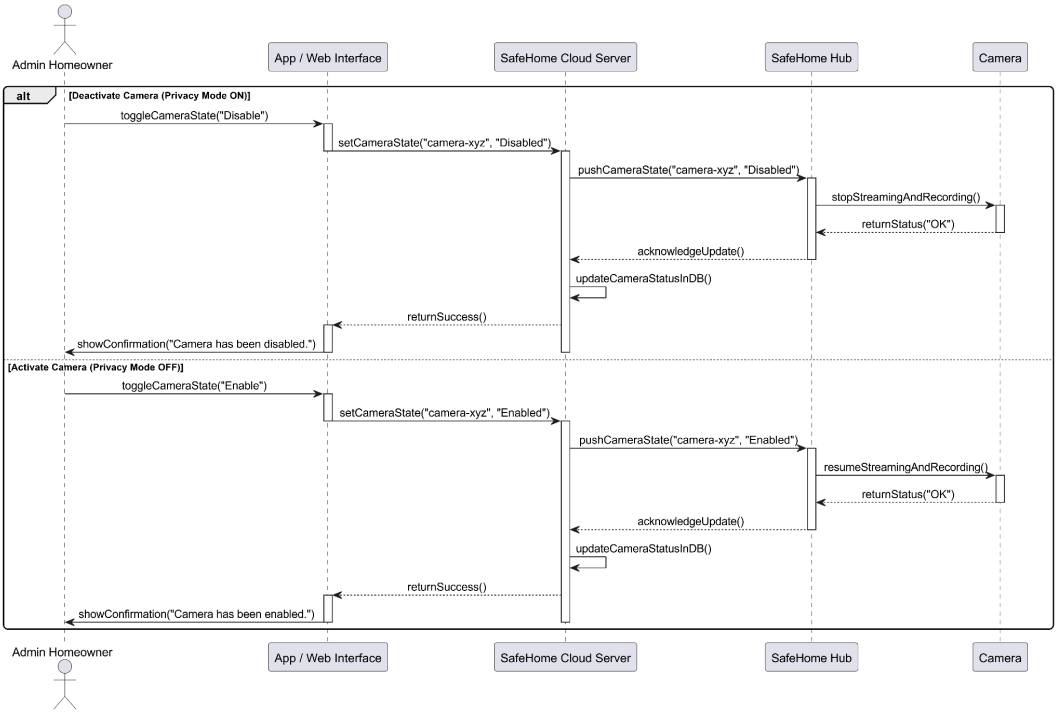
References on SEPA safehome dialog slide 16

2.1.3 Protect Sensitive Camera Feed with a Password - Use Case



References on SEPA safehome dialog slide 19-31

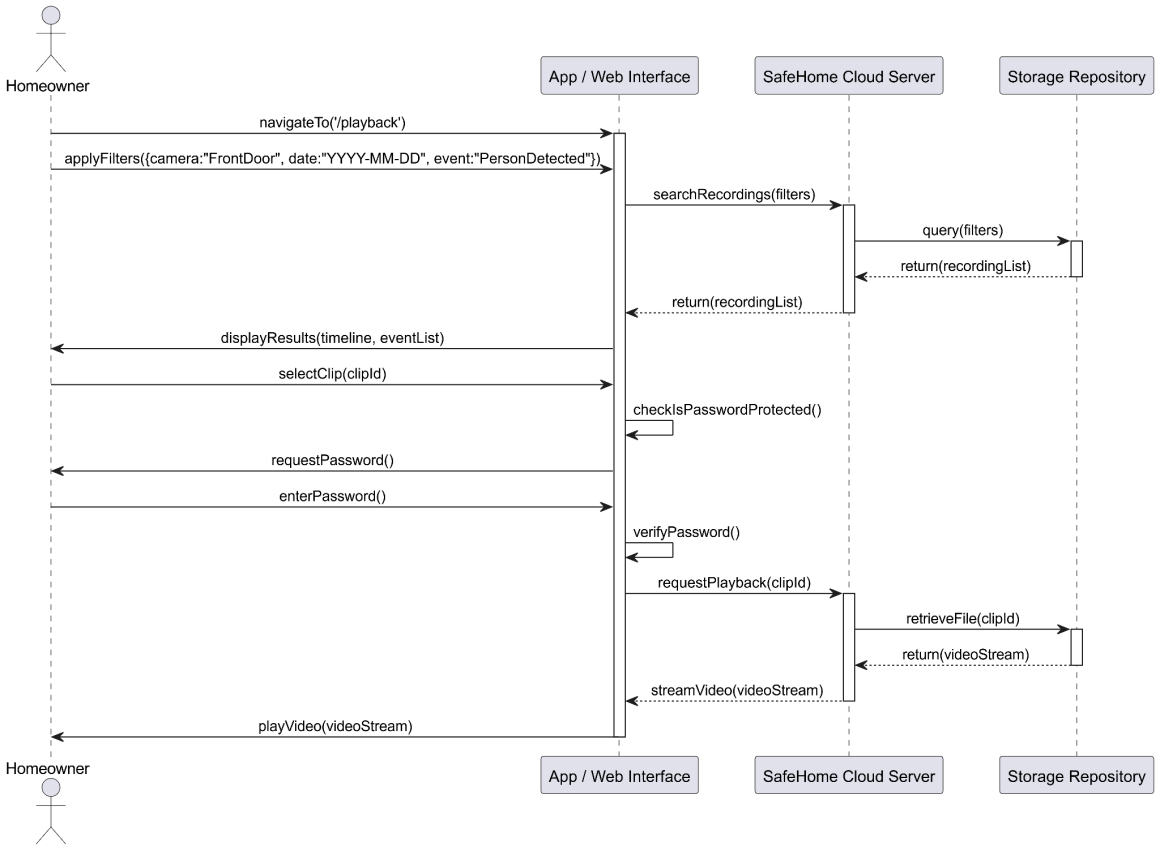
2.1.4 Camera Activation and Deactivation - Use Case



References on 2025.10.26 / 2025.10.29 Meeting

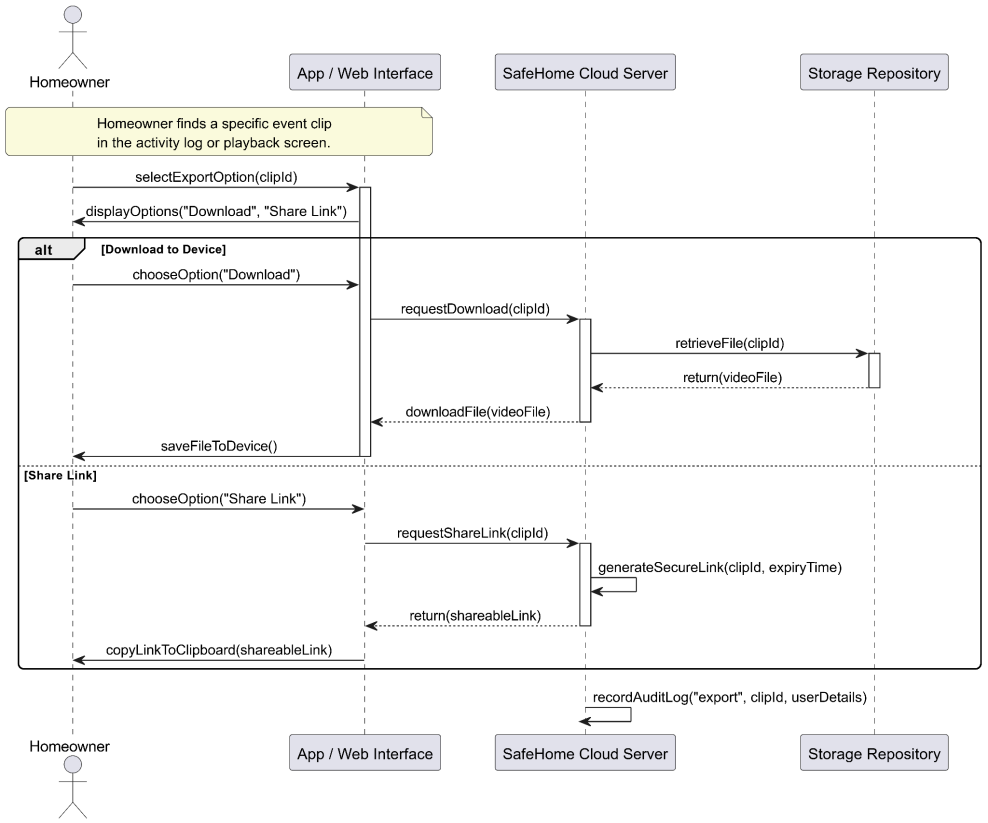
2.2 Recording and Evidence Management

2.2.1 Search and Playback Recordings - Use Case



References on SEPA safehome dialog slide 29-31

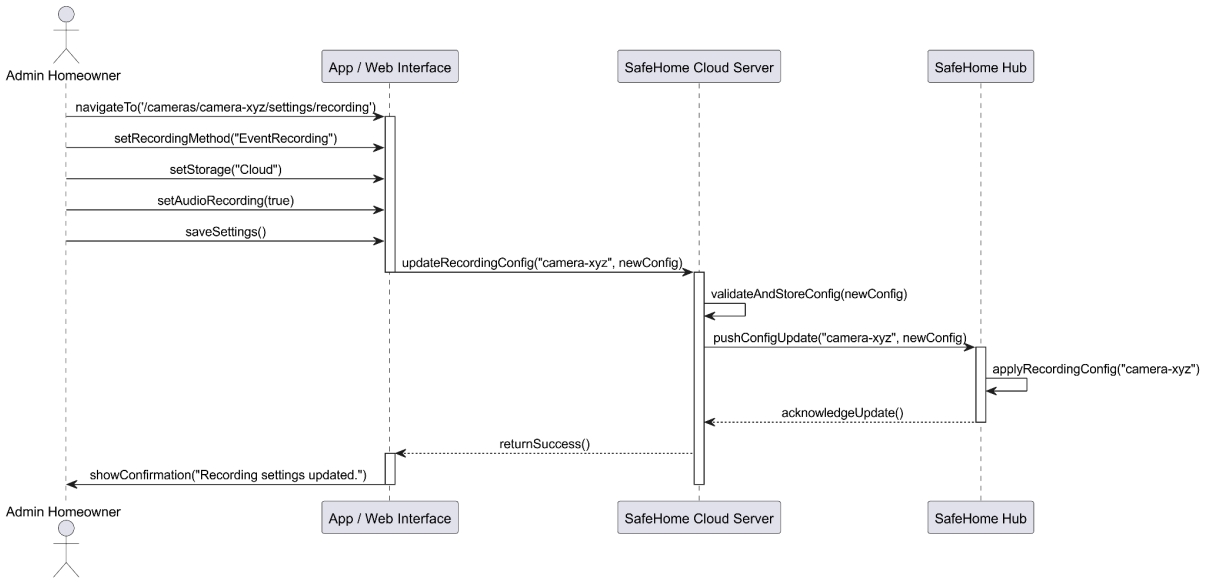
2.2.2 Evidence Sharing and Export - Use Case



References on 2025.10.29 Meeting

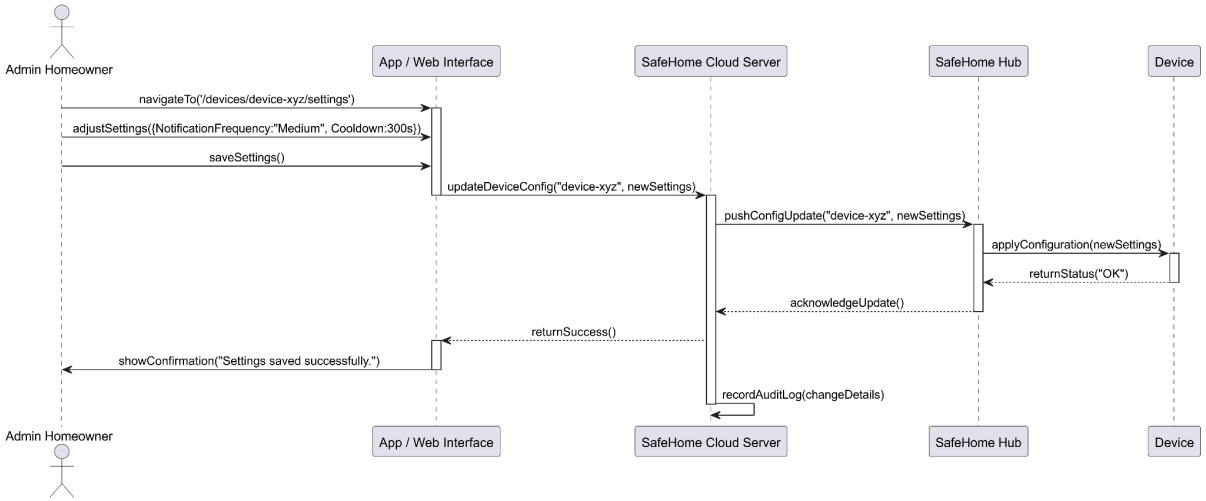
2.3 Surveillance Settings

2.3.1 Recording Settings - Use Case



References on SEPA safehome dialog slide 29-31

2.3.2 Notification Policy and Cooldown - Use Case

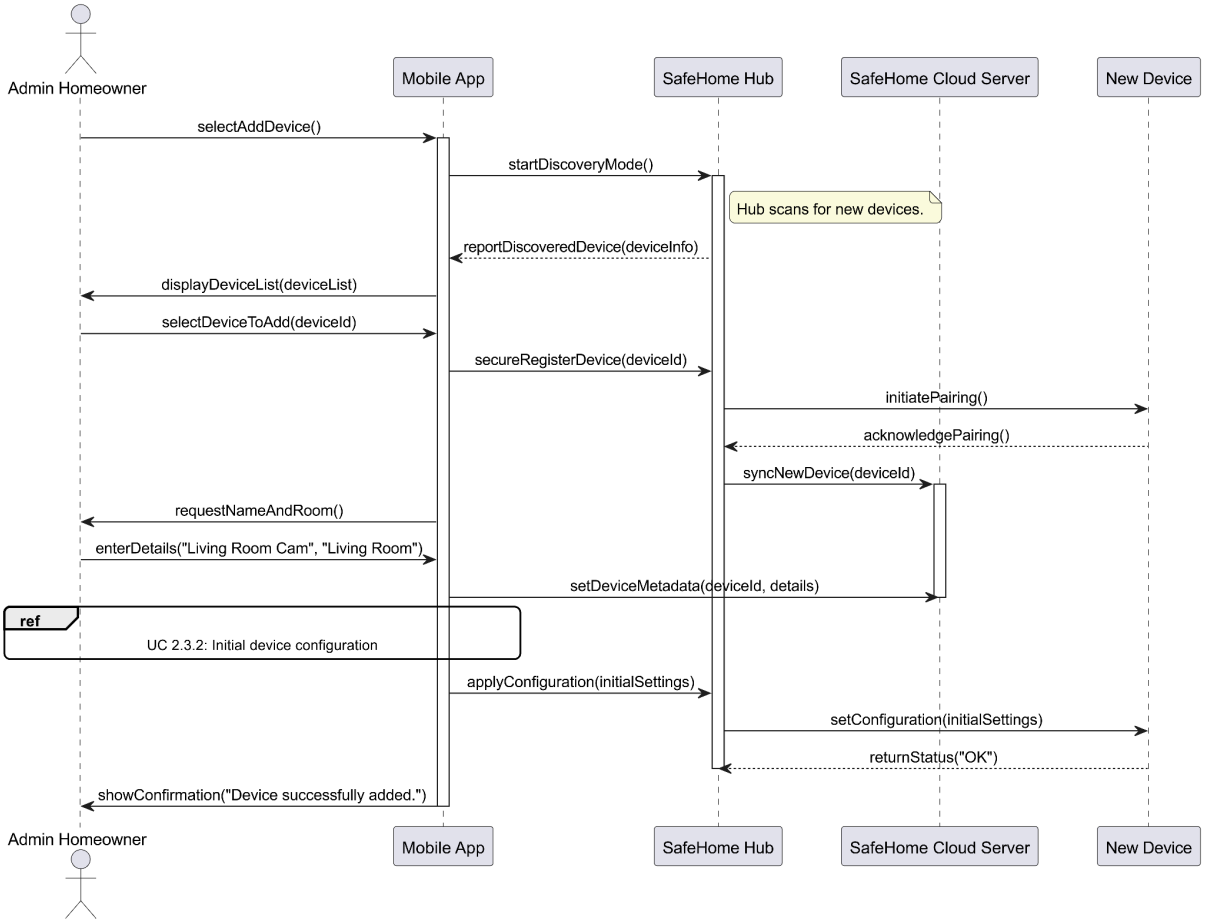


References on 2025.10.29 Meeting

3. System and User Management

3.1 Device Management

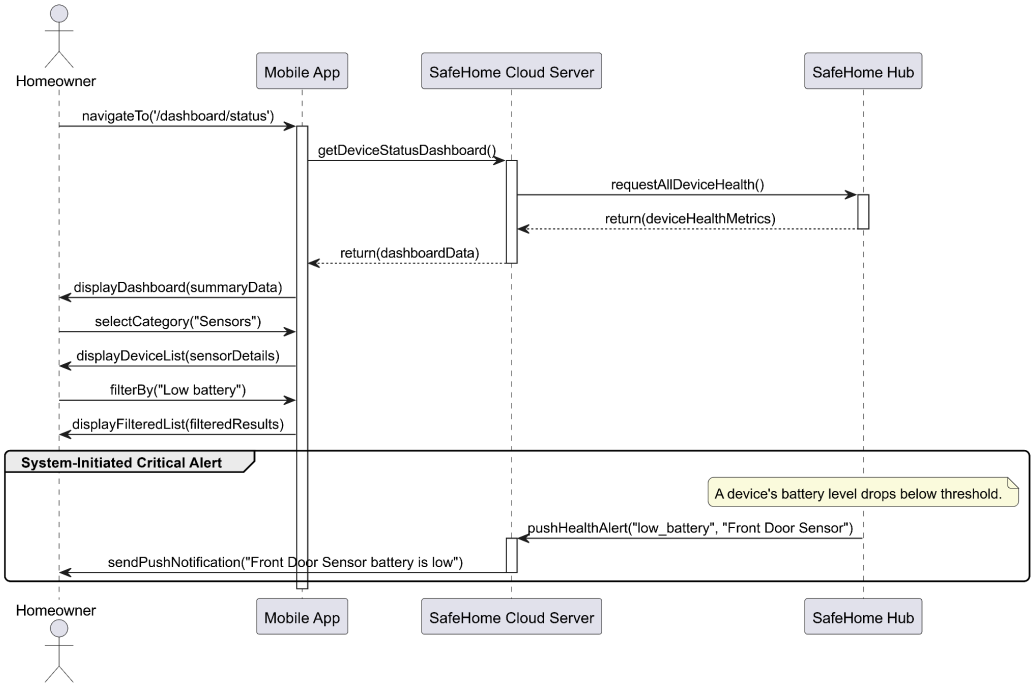
3.1.1 Add and Place New Devices - Use Case



References on SEPA safehome dialog slide 58

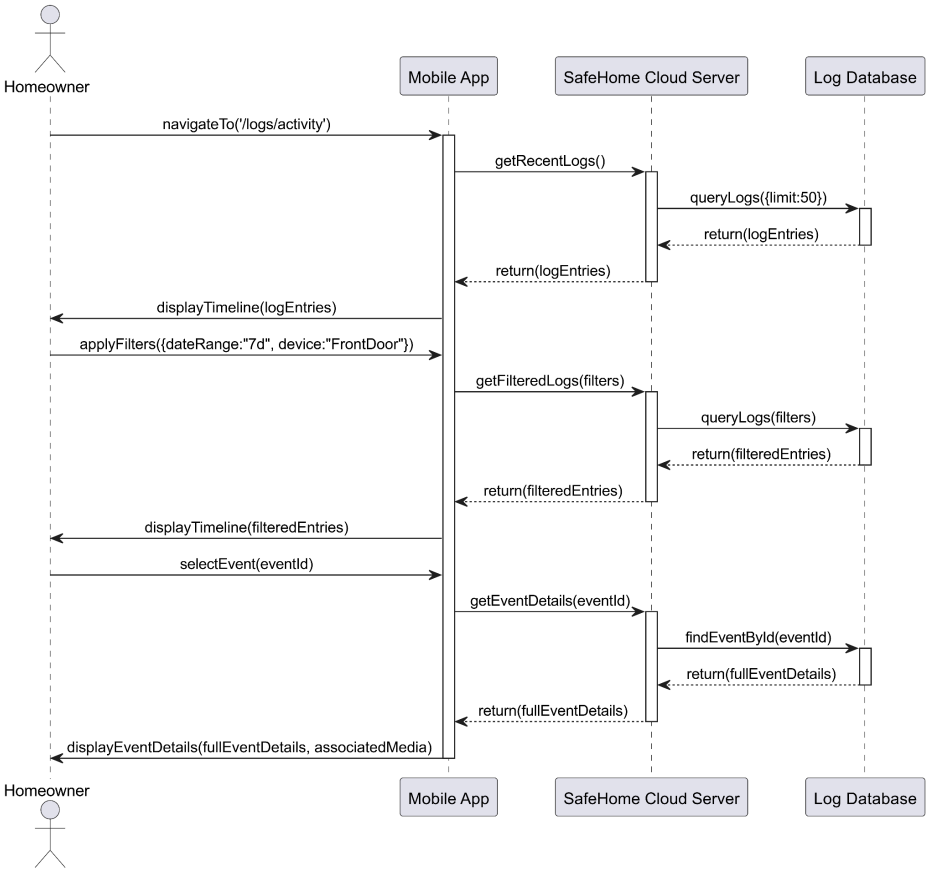
3.2 System Status and Logs

3.2.1 System Status Dashboard - Use Case



References on 2025.10.29 Meeting

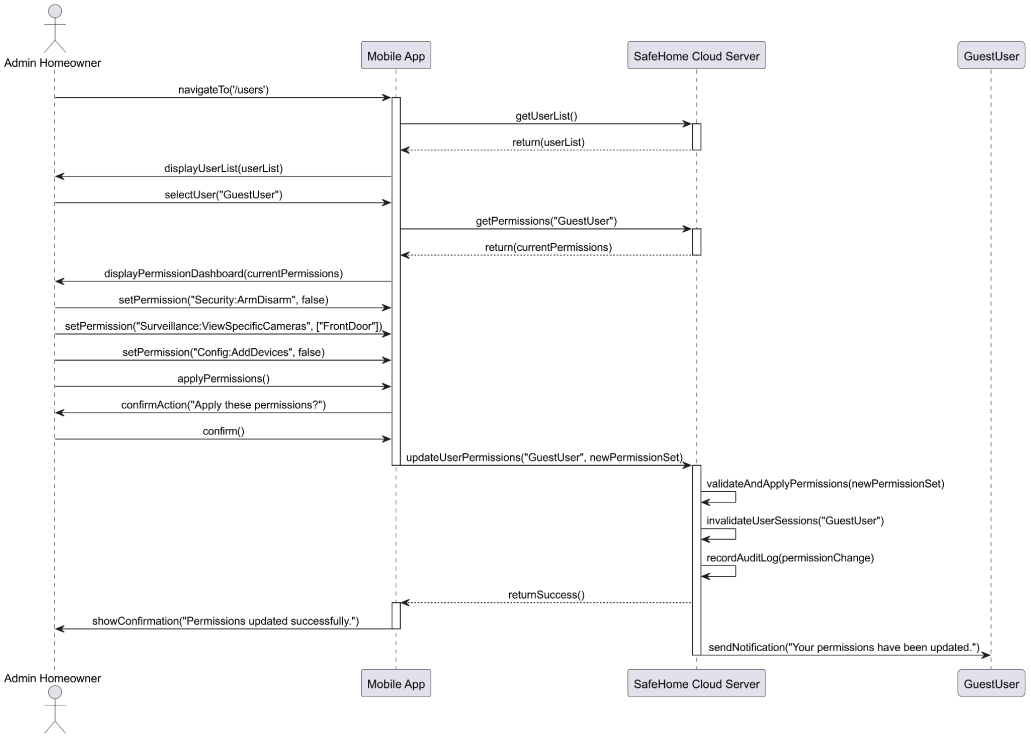
3.2.2 Activity Logs and Timeline - Use Case



References on SEPA safehome dialog slide 39

3.3 User and Permission Management

3.3.1 User Role and Access Control - Use Case

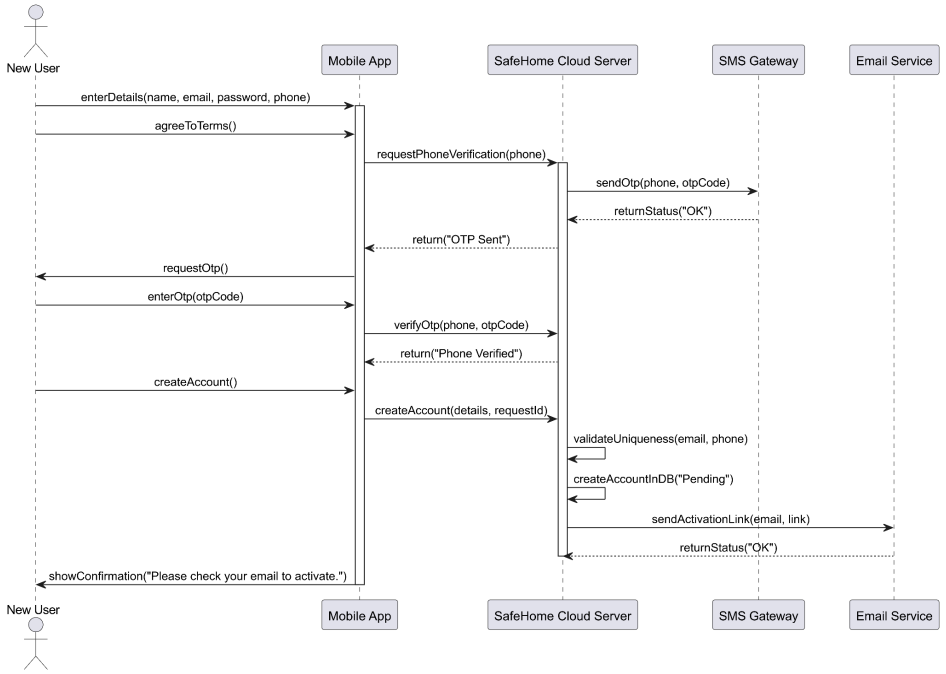


References on SEPA safehome dialog slide 70

4. Remote Access and Account

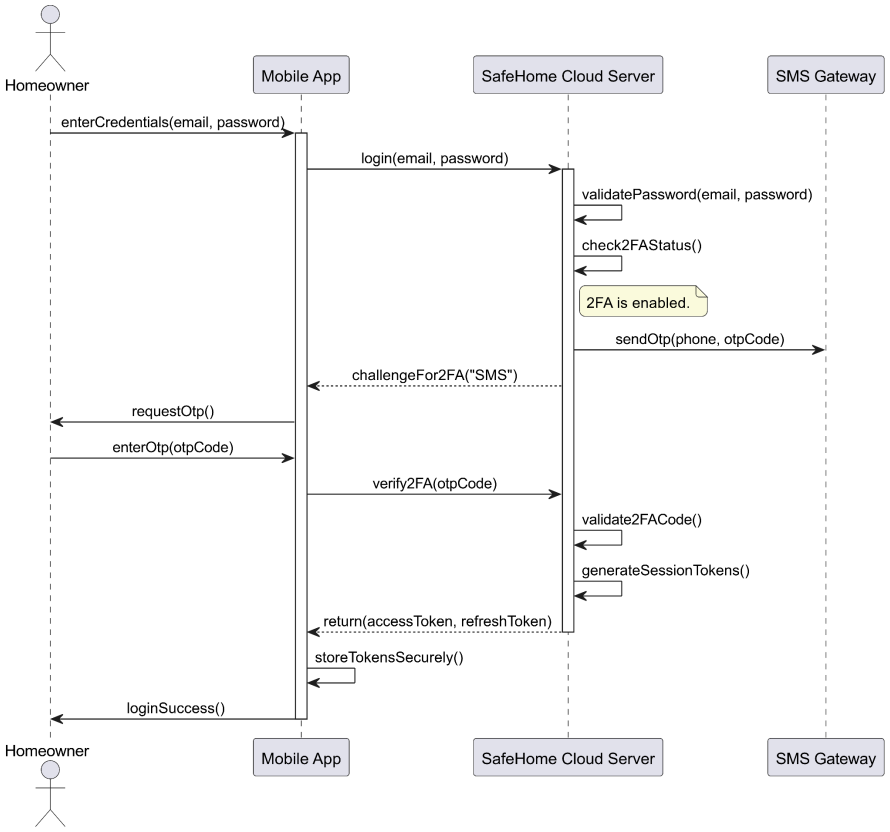
4.1 Account Management

4.1.1 Sign Up - Use Case



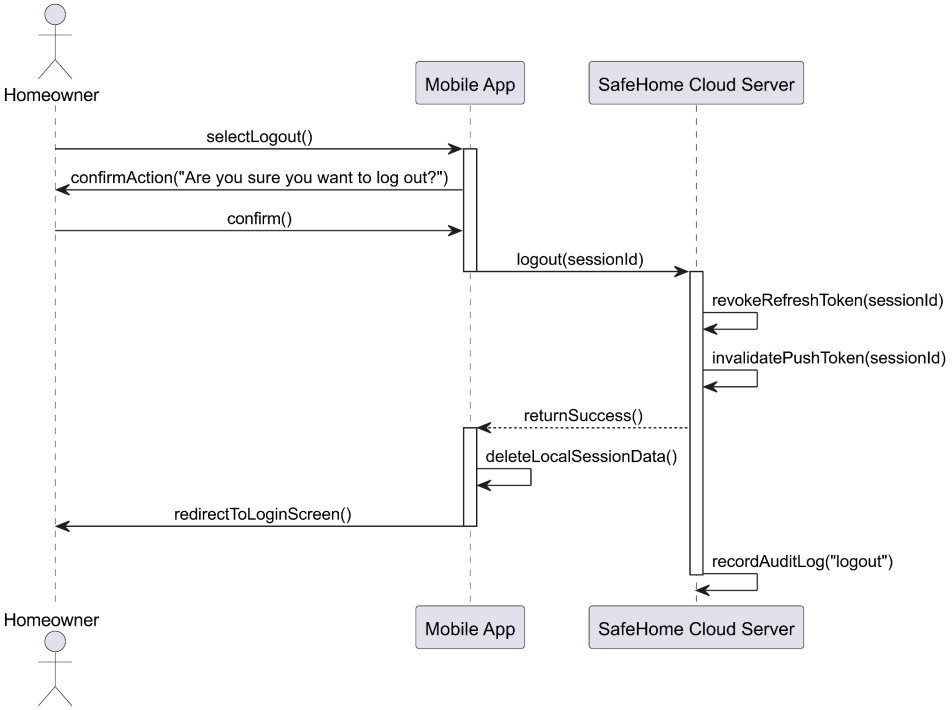
References on SEPA safehome dialog slide 41

4.1.2 Log In - Use Case



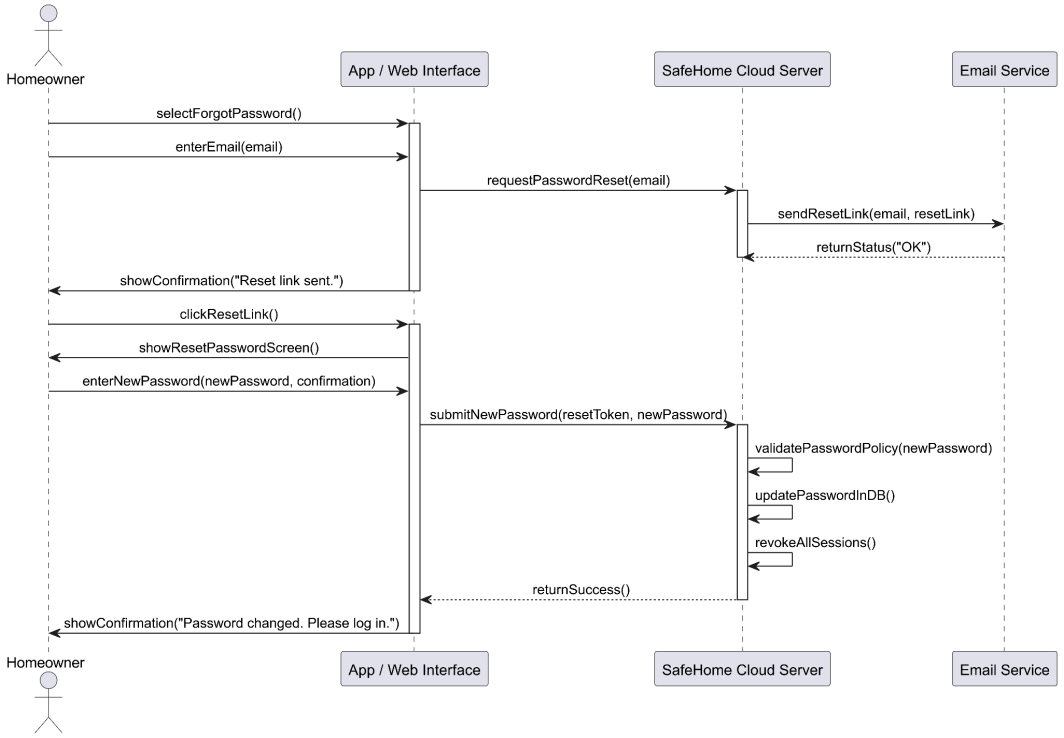
References on SEPA safehome dialog slide 42

4.1.3 Log Out - Use Case



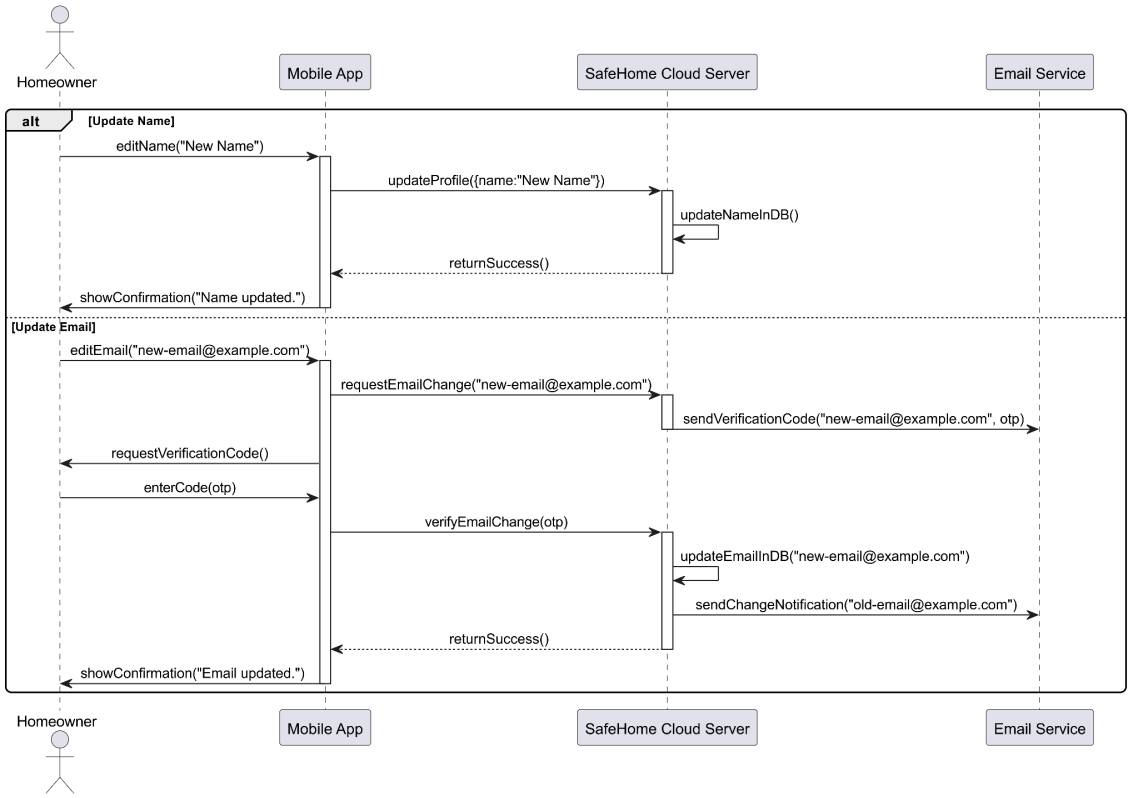
References on SEPA safehome dialog slide 43

4.1.4 Password Recovery and Reset - Use Case



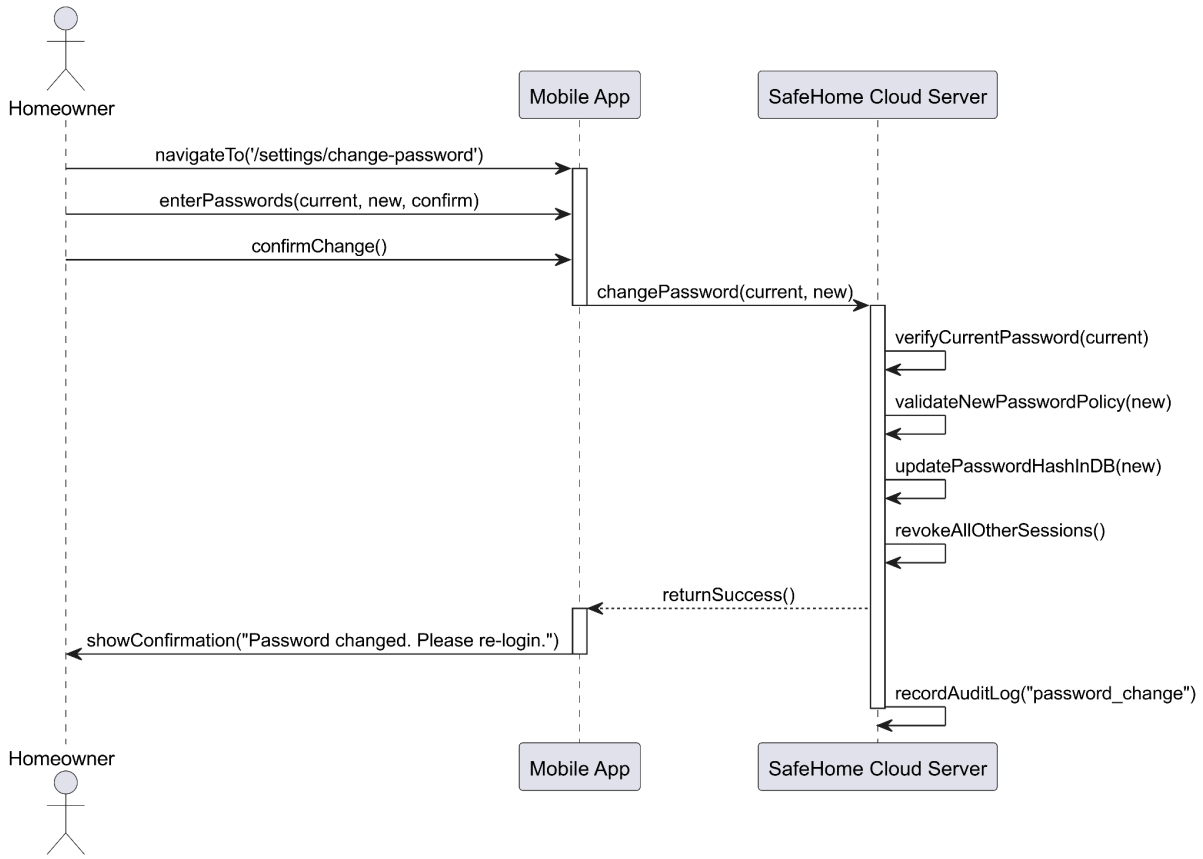
References on SEPA safehome dialog slide 44

4.1.5 Edit Profile Information - Use Case



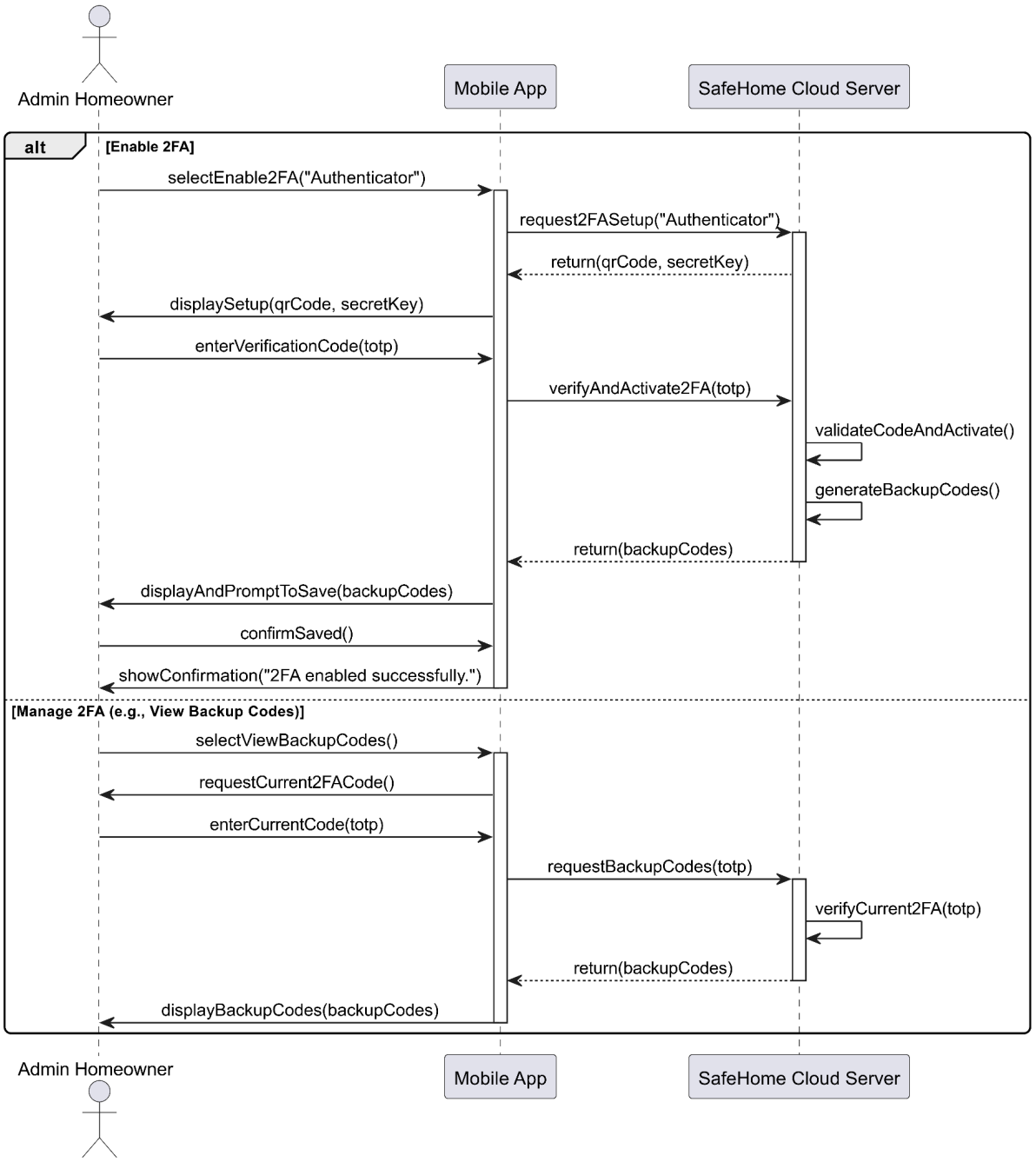
References on SEPA safehome dialog slide 45

4.1.6 Change Password - Use Case



References on SEPA safehome dialog slide 46

4.1.7 Two-Factor Authentication Management - Use Case

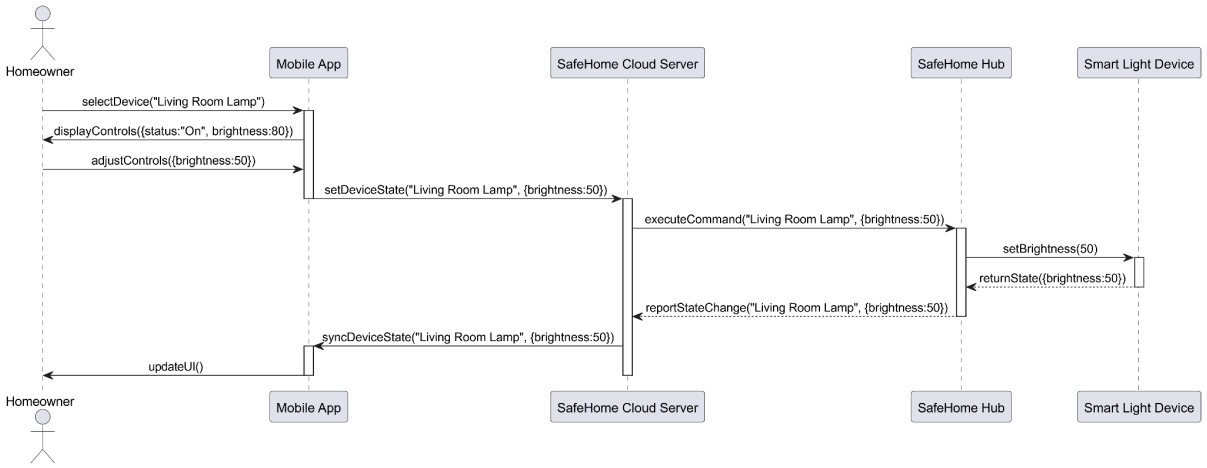


References on SEPA safehome dialog slide 47

5. Indoor Monitoring and Device Control

5.1 Device Control

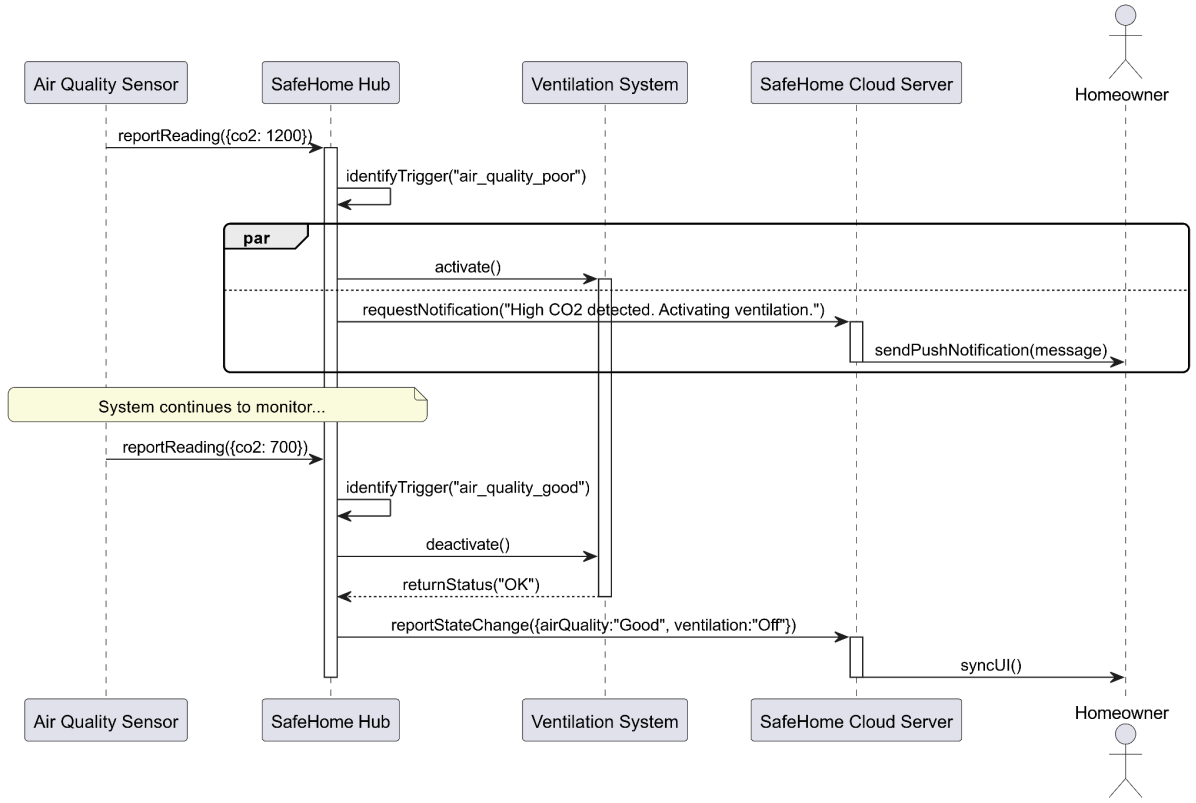
5.1.1 Indoor Device Control - Use Case



References on SEPA safehome dialog slide 39

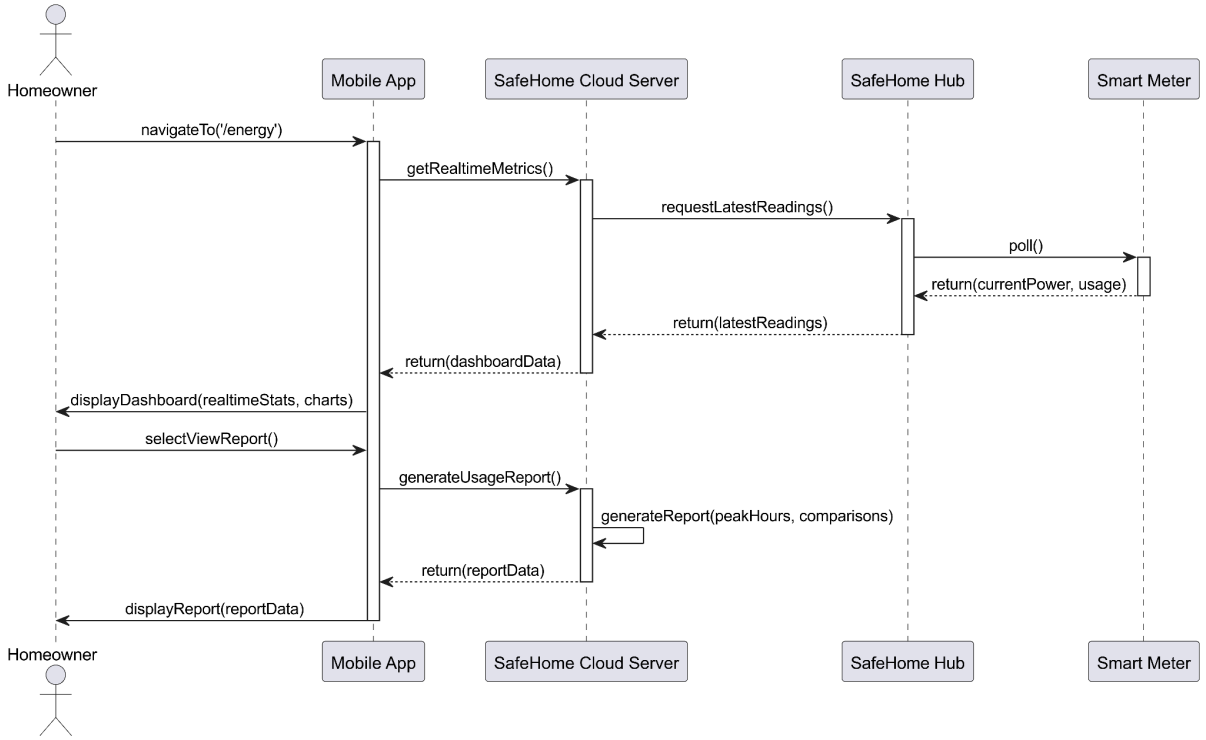
5.2 Indoor Monitoring System

5.2.1 Indoor Air Quality Monitoring and Ventilation Integration - Use Case



References on SEPA safehome dialog slide 27

5.2.2 Real-Time Power Consumption Monitoring and Reporting - Use Case



References on SEPA safehome dialog slide 29

VIII. Who did what

MinJun Kim

1. Complete UI/UX Design
    a. Created mobile UI design prototype for SafeHome
    b. Designed control panels, login screens, and security zone displays
    c. Developed modern smart home interface with glassmorphic styling
2. Use Case Scenarios Development
    a. Common functions use cases (Log onto system via control panel and web browser)
    b. Security functions use cases (arm/disarm, safety zones, alarm conditions)
    c. Surveillance functions use cases (camera view, pan/zoom, recording)
    d. Configuration functions use cases (system settings, floor plan, device management)
3. UML Diagrams Creation
    a. Use case diagrams for all major functions
    b. Sequence diagrams for common, security, and surveillance functions
    c. PlantUML implementation with proper UML standards (lifelines, activation bars, message types)
4. Document Writing and Refinement
    a. Refinement of overview section (Introduction, Goals, Major Functionalities)
    b. Creating comprehensive Glossary with categorized terms
    c. Refinement of technical writing (removing marketing language, improving clarity)
    d. Comprehensive SRS document evaluation and revision
5. Requirements Analysis
    a. Finding and documenting references from SEPA dialog slides
    b. Analyzing safehome_dialog.pdf for requirement extraction

Jehyung Kim

1. Use Case Scenarios Development
    a. Core Interactions: System login procedures via control panel and web.
    b. Surveillance Features: Live camera viewing, pan-tilt-zoom (PTZ) controls, and video recording functions.
    c. System Administration: Management of system settings, floor plan layouts, and individual devices.
    d. Combined all use cases our teammates generated to ensure a consistent style and context throughout.

2. UML Diagrams Creation

    a. Created UML sequence diagrams detailing the workflows for common, security, and surveillance features.
    b. Ensured standard-compliant implementation using PlantUML, with correct notation for lifelines, activations, and message semantics.

3. Document Writing and Refinement

    a. Enhanced the document's introductory sections (Introduction, Goals, Major Functionalities) to improve structural clarity and focus.
    b. Authored a comprehensive glossary with categorized terms to standardize project terminology and improve reader comprehension.
    c. Refined the document's technical language by eliminating marketing jargon and improving overall precision and clarity.
    d. Conducted an end-to-end evaluation and revision of the entire Software Requirements Specification (SRS) document to ensure quality and completeness.

4. Requirements Analysis
    a. Elicited and documented foundational requirements by analyzing the SEPA dialog slides for key reference points.
    b. Extracted detailed system specifications through a comprehensive analysis of the safehome_dialog.pdf document.

Onejune Lee

1. Use Case Scenarios Development
    a. Intelligent Security functions use cases
        i. Developed the complete set of scenarios for all functions within the Intelligent Security section (1.1.1 through 1.3.3). This included defining detailed steps, pre-conditions, triggers, and exceptions for:
            1. Sensor Monitoring: Physical Intrusion Detection, Environmental Hazard Detection, Outdoor Motion Detection, and Dog Barking Detection.
        - Incident Management: Alarm Trigger and Instant Notification, Alarm Verification Step, Emergency Service Integration, and the Panic Button feature.
        - Security Mode Control: One-Touch Modes (Away, Home, Sleep), Sensor Bypass, and Sensor Activation/Deactivation.
2. UML Diagrams Creation
    a. Designed and created the Use Case Diagrams and Sequence Diagrams for the entire Intelligent Security section, covering all sub-functions from Sensor Monitoring to Security Mode Control.
    b. Reviewed and standardized all UML diagrams submitted by the team, ensuring consistency in style, format, and notation across the entire project document.
3. Document Writing and Refinement
    a. Project Scope and Direction: Led project meetings, set the agenda, and guided discussions to define and refine the project's overall scope and final feature list.
    b. UI/UX Design and Prototyping: Proposed and designed the entire UI/UX structure for the mobile and tablet application, including the main tab layout (Dashboard, Emergency, Devices, Settings) and the detailed content for each screen.
    c. Document Standardization and Quality Assurance: Took responsibility for the final document's quality by reviewing all sections, ensuring a consistent format and writing style, and standardizing technical terms and diagram layouts.

4. Requirements Analysis
    a. Led Initial Requirements Definition: Initiated the requirements analysis process by using AI to extract a baseline feature list from the provided course materials and then led the team's effort to refine and finalize this list.
    b. Defined System Logic and Rules: Analyzed potential system conflicts and user interactions to define critical operational rules, such as the command priority order (Security > User Input > Automation) and specific development guidelines (e.g., password complexity).

Yunje Lee

1. Use Case Scenarios Development
    a. Remote Access and Account functions use cases
        i. Developed scenarios for Sign Up, Log In, Log Out, Password Recovery and Reset, Edit Profile Information, Change Password, and Two-Factor Authentication Management.
    b. Indoor Monitoring and Device Control functions use cases
        i. Developed scenarios for Indoor Device Control, Indoor Air Quality Monitoring and Ventilation Integration, and Real-Time Power Consumption Monitoring and Reporting.

2. UML Diagrams Creation

    a. Designed and created a complete set of use case diagrams that illustrate all major system functionalities. The diagrams cover the following core areas:

    Intelligent Security (including Sensor Monitoring, Incident Management, and Security Mode Control)

    Live Surveillance (including Camera Viewing, Recording Management, and Surveillance Settings)

    System and User Management (including Device Management, System Logs, User Permissions, and Policy)

    Remote Access and Account (including all aspects of Account Management)

    Indoor Monitoring and Device Control (including Device Control and Monitoring Systems)

3. Document Writing and Refinement
    c. Refinement of overview section (Introduction, Goals, Major Functionalities)
    d. Creating comprehensive Glossary with categorized terms

e. Refinement of technical writing (removing marketing language, improving clarity) f. Comprehensive SRS document evaluation and revision 4. Requirements Analysis g. Finding and documenting references from SEPA dialog slides h. Analyzing safehome_dialog.pdf for requirement extraction

IX. Meeting logs

Date	2025/10/26
Location	KAIST Library Group Project Room
Attendees	Attendee 1: Onejune Lee. Attendee 2: Jehyung Kim Attendee 3: Yunje Lee

Summary	• Meeting Objective: To finalize the functional requirements for the Safehome project and establish a plan for creating the Software Requirements Specification (SRS) document. • Key Discussion Points: 1. Defining Functional Requirements: ○ The initial list of functions, generated by GPT based on the professor's presentation slides, was reviewed. ○ Outdated functions from the textbook (e.g., converting voice messages to text) were deemed unsuitable for the current project and were excluded. ○ To finalize the list, the team will cross-reference the current draft with the professor's presentation to identify any missing features and map each function to its corresponding slide number. ○ It was agreed to add necessary modern features not explicitly mentioned in the presentation (e.g., enhanced security, one-touch system deactivation) based on reasonable assumptions. ○ Complex or out-of-scope features, such as advanced AI functions, community-wide security, and scalability, will be excluded. 2. Use Case and UML Diagram Planning: ○ The team will create detailed use case scenarios for each of the approximately 30 finalized functions. ○ A test using GPT to generate these scenarios yielded high-quality results, so the team decided to actively utilize it for this task. ○ The agreed-upon workflow is: (1) Define the overall system flow with a Use Case Diagram → (2) Document the scenario for each use case → (3) Create Sequence Diagrams based on the scenarios. 3. UI/UX and Development Strategy: ○ The UI examples in the textbook were considered obsolete. The team will create a new UI prototype using Figma instead. ○ Given the tight schedule, the project will follow the "Waterfall" model, as mentioned by the professor. ○ To minimize bugs during the implementation phase, the

	development focus will be on demonstrating screen flows rather than building a fully functional backend. ○ Google Docs will be used for collaborative work on the SRS document. • Decisions Made: ● The final list of functions will be confirmed by comparing the GPT draft with the professor's presentation. ● GPT will be actively used for creating use case scenarios and related documentation. ● All final deliverables will be written in English. ● The next meeting is scheduled for October 27th at 10:00 PM. • To-Do List (Before the next meeting): 1. Create Use Case Scenarios: Each member is to generate use case scenarios for their assigned functions using GPT. 2. Practice UML Diagrams: Each member will select one of their use cases and create one Use Case Diagram and one Sequence Diagram as practice. 3. Research UI/UX References: Each member is to find one relevant UI template or app design to serve as a reference.
AI Note	https://clovanote.naver.com/s/JAcqyrvWSvRxKaGb7uLpXLS password: ypupni

Date	2025/10/27
Location	KAIST Library Group Project Room
Attendees	Attendee 1: Onejune Lee. Attendee 2: Minjun Kim Attendee 3: Yunje Lee Attendee 4: Jehyung Kim
Summary	Meeting Objective: To simplify the project's scope, standardize diagramming methods, and decide on the report format and task

	distribution. Key Discussion Points: 1. Simplification of Functional Requirements: ○ Feature Exclusion: The team discussed removing features that are uncertain to implement (e.g., special sensors, IoT device integration) or are excessive for the project's scope (e.g., intelligent automation). ○ Clarifications for Professor: A decision was made to ask the professor for clarification on whether individual appliance control and a floor plan feature are within the project's scope. ○ Features to Retain: Essential and feasible functions, such as user-customized rules and situational modes, will be kept. It was decided to tentatively include special sensors in the function list as they were mentioned in the presentation slides. 2. UI/UX and Development Platform: ○ Platform Decision: Considering the convenience of testing and deployment, the team decided to develop a responsive website with a "mobile app view." The documentation will state that the system supports both web and mobile app environments. ○ AI Tool Utilization: The team agreed to actively use Figma and AI tools (e.g., Cursor) to efficiently create the UI prototype. 3. Standardization of Diagramming Methods: ○ Tool Unification: After testing several tools, the team concluded that the PlantUML extension in VS Code is the most accurate and convenient, and decided to use it as the standard tool. ○ Notation Agreement: The team agreed to adhere to standard notation for sequence diagrams, including activation boxes, synchronous/asynchronous messages, and explicit termination conditions. A concise, function-call format was preferred over descriptive sentences. 4. Use Case Scenario Authoring: ○ Quality Control: It was acknowledged that initial drafts of scenarios generated by GPT may contain inaccurate assumptions or out-of-scope elements. Team members will be responsible for carefully reviewing and revising their respective scenarios. ○ Authoring Guidelines: To minimize AI errors, a guideline

	was established to first write clear functional descriptions (e.g., in an if-then format) to provide better context for the AI. The team also committed to following documentation principles learned in class, such as clearly defining the subject (actor). Decisions and Action Items: ● UI: Develop a responsive website with a mobile-centric view. ● Diagramming Tool: Use PlantUML. ● Documentation: The final report will be written using Notion or Google Docs and submitted as a PDF. To-Do (Before the next meeting): 1. Finalize Use Case Scenarios: Each member is to write detailed use case scenarios for their assigned functions in Korean, ensuring all errors and unnecessary assumptions are removed. 2. Next Meeting: The next meeting is scheduled for tomorrow from 7:00 PM to 9:00 PM.
AI Note	https://clovanote.naver.com/s/NJenuULAgKcMdyXEgGYtV3S?t=807 비밀번호: djpgww

Date	2025/10/28
Location	KAIST Undergraduate Branch Library
Attendees	Attendee 1: Onejune Lee. Attendee 2: Minjun Kim Attendee 3: Jehyung Kim Attendee 4: Yunje Lee
Summary	Meeting Objective: To simplify the project's scope, standardize diagramming methods, and decide on the report format and task distribution. Key Discussion Points: 1. Simplification of Functional Requirements: ○ Feature Exclusion: The team discussed removing features that are uncertain to implement (e.g., special sensors, IoT device integration) or are excessive for the

project's scope (e.g., intelligent automation).
    - Clarifications for Professor: A decision was made to ask the professor for clarification on whether individual appliance control and a floor plan feature are within the project's scope.
    - Features to Retain: Essential and feasible functions, such as user-customized rules and situational modes, will be kept. It was decided to tentatively include special sensors in the function list as they were mentioned in the presentation slides.
2. UI/UX and Development Platform:
    - Platform Decision: Considering the convenience of testing and deployment, the team decided to develop a responsive website with a "mobile app view." The documentation will state that the system supports both web and mobile app environments.
    - AI Tool Utilization: The team agreed to actively use Figma and AI tools (e.g., Cursor) to efficiently create the UI prototype.
3. Standardization of Diagramming Methods:
    - Tool Unification: After testing several tools, the team concluded that the PlantUML extension in VS Code is the most accurate and convenient, and decided to use it as the standard tool.
    - Notation Agreement: The team agreed to adhere to standard notation for sequence diagrams, including activation boxes, synchronous/asynchronous messages, and explicit termination conditions. A concise, function-call format was preferred over descriptive sentences.
4. Use Case Scenario Authoring:
    - Quality Control: It was acknowledged that initial drafts of scenarios generated by GPT may contain inaccurate assumptions (e.g., 100-decibel sirens) or out-of-scope elements (e.g., Zigbee, smart water valves). Team members will be responsible for carefully reviewing and revising their respective scenarios.
    - Authoring Guidelines: To minimize AI errors, a guideline was established to first write clear functional descriptions (e.g., in an if-then format) to provide better context for the AI. The team also committed to following documentation principles learned in class, such as clearly defining the subject (actor).

	Decisions and Action Items: ● UI: Develop a responsive website with a mobile-centric view. ● Diagramming Tool: Use PlantUML. ● Documentation: The final report will be written using Notion or Google Docs and submitted as a PDF. To-Do (Before the next meeting): 1. Finalize Use Case Scenarios: Each member is to write detailed use case scenarios for their assigned functions in Korean, ensuring all errors and unnecessary assumptions are removed. 2. Next Meeting: The next meeting is scheduled for tomorrow from 7:00 PM to 9:00 PM.
AI Note	https://clovanote.naver.com/s/JnaqtSgY7oseKpj8NvcxdBS 비밀번호: hhv7tg

Date	2025/10/29
Location	Undergraduate Branch Library
Attendees	Attendee 1: Yunje Lee Attendee 2: Onejune Lee Attendee 3: Minjun Kim
Summary	Meeting Objective: To finalize the project's feature list based on feedback from the professor, define the UI/UX scope, and assign tasks for documentation and design. Key Discussion Points: 1. Feature List Finalization: ○ The team reviewed and revised the feature list following a meeting with the professor. The list was reorganized for clarity, with new features marked accordingly. ○ Features deemed out-of-scope or too complex for the project, such as firmware updates and advanced data privacy policies, will be handled as "assumptions" rather than implemented functionalities. ○ Some features were merged to avoid redundancy. ○ A discussion was held on whether to include "Home

	Automation" features (e.g., lighting control). Although the professor indicated it was outside the core "Safe Home" scope, the team decided to include a limited set of these features, such as lighting, to enhance the product's value, justifying this in the assumptions section. ○ The feature for monitoring device connection status and battery levels was reinstated, while the more complex "failure prediction" function was removed. 2. UI/UX Strategy and Screen Design: ○ The team decided to identify which features require a dedicated UI and which can be handled by push notifications or background processes (e.g., intrusion alerts). ○ The main mobile application will be structured with four main tabs: Dashboard, Emergency, Devices, and Settings. ○ Dashboard: Will serve as the home screen, displaying recent camera recordings, event logs, and status widgets for features like air quality and power consumption. It will also feature controls for one-touch modes (e.g., Home, Away). ○ Emergency: Will feature a large panic button and options for contacting emergency services. ○ Devices: This section will display a floor plan to show the location of sensors and cameras. Users can select a room from the plan to view and control the devices within it. The floor plan itself will be a static image, as its management is out of scope. ○ Settings & Logs: A dedicated screen for system logs will be created, allowing users to review events, filter by type, and export evidence. Other settings like user permissions and profile management will also be included in this section. ○ In-Home Panel (GUI): The team discussed the physical control panel. It was decided to design it as a modern tablet interface rather than replicating the outdated example provided. The design will be a responsive version of the web app, optimized for a tablet view. 3. Documentation and Task Division: ○ The final report and documentation will be compiled in Google Docs to facilitate features like internal hyperlinks, which are necessary for navigation. Notion was considered but deemed less suitable for the final PDF submission requirements. ○ Tasks were divided for writing the remaining sections of

	the SRS document, including the introduction and assumptions. ○ A plan was set to have one team member with experience in AI-powered front-end development generate the initial UI screens based on the agreed-upon structure and feature descriptions. The rest of the team will focus on finalizing the use cases and diagrams. --- Decisions Made: ● The feature list has been finalized, with out-of-scope items to be documented in the "Assumptions" section. ● The primary user interface will be a responsive web application designed with a mobile-first approach. An adapted tablet view will serve as the in-home control panel. ● The final project documentation will be created in Google Docs. Action Items: ● Finalize all new and revised use case scenarios and diagrams. ● One team member will generate the initial UI mockups for all required screens using an AI tool. ● Team members will write their assigned introductory sections of the SRS document, including the project scope and assumptions. ● All documentation (use cases, diagrams, introductory text) will be compiled into a single Google Docs file for final review and formatting.
AI Note	https://clovanote.naver.com/s/nGVtY6jvm6onbDRmpkXoGUS 비밀번호: ngtsnk

Date	2025/10/30
Location	KAIST Academic Cultural Complex
Attendees	Attendee 1: Minjun Kim Attendee 2: Jehyung Kim Attendee 3: Onejune Lee Attendee 4: Yunje Lee
Summary	The process of consolidating the individually created content into a final report is underway

AI Note	there is no separate recording

Appendix A. Glossary

This glossary defines key terminology used throughout the SafeHome Software Requirements Specification. Terms are organized by category for easy reference.

System Components (Hardware)

SafeHome Hub

The central control unit that manages all connected devices, processes sensor data, executes automation rules, and communicates with the cloud server. Located within the home, it operates as the primary coordinator for all system functions.

- Related Use Cases: All system functions
- Reference: Introduction, Section II

Control Panel

A physical touchscreen device installed in the home that provides local access to system functions. Users can arm/disarm the system, view camera feeds, and manage basic settings without requiring internet connectivity.

- Related Use Cases: UC 1.3.1 (Set System Mode), UC 4.1.2 (Log In)
- Reference: Fig. 1, Page 6

Sensor

A wireless device that monitors specific conditions or events. SafeHome supports multiple sensor types:

- Contact Sensor: Detects opening/closing of doors and windows
- Motion Sensor: Detects movement in designated areas
- Environmental Sensor: Monitors fire, smoke, carbon monoxide, gas leaks, or water leaks
- Air Quality Sensor: Measures indoor pollutants (CO_2, VOCs)
- Sound Sensor: Detects specific audio patterns (e.g., dog barking, glass breaking)
- Related Use Cases: UC 1.1 (Sensor Monitoring)
- Reference: Section 1.1, Major Functionalities

Camera (IP Camera)

A network-connected video camera that provides live streaming, recording, and two-way audio capabilities. Cameras can be placed indoors or outdoors and support features such as pan, tilt, zoom (PTZ), night vision, and motion-triggered recording.

- Related Use Cases: UC 2.1 (Camera Viewing and Control)
- Reference: Section 2.1, Major Functionalities

Smart Home Device

IoT devices that can be controlled by the SafeHome system, including smart lights, smart thermostats, smart locks, and ventilation systems. Integration enables automation based on security modes or environmental conditions.

- Related Use Cases: UC 5.1.1 (Indoor Device Control), UC 5.2.1 (Air Quality Monitoring)
- Reference: Section 5, Major Functionalities (Second Increment)

System Components (Software)

SafeHome Cloud Server

The cloud-based backend infrastructure that enables remote access, stores user data, manages authentication, delivers push notifications, and synchronizes system state across multiple devices.

- Related Use Cases: All remote access functions, UC 4.1 (Account Management)
- Reference: Architecture overview in all Sequence Diagrams

Mobile Application

The primary user interface for SafeHome, available on iOS and Android platforms. Provides comprehensive access to all system functions including live camera viewing, security control, device management, and notifications.

- Related Use Cases: All user-facing use cases
- Reference: Section III (Prototype UI/UX)

Web Interface

A browser-based application that offers functionality equivalent to the mobile app, accessible from desktop or laptop computers. Useful for detailed configuration and system administration tasks.

- Related Use Cases: All user-facing use cases
- Reference: UC 4.1.2 (Log In)

Security Concepts

Security Mode

A predefined system configuration that determines which sensors are active and how the system responds to events. SafeHome supports the following modes:

- Home Mode: Interior motion sensors disabled, perimeter sensors active
- Away Mode: All sensors active, full security monitoring
- Sleep Mode: Perimeter sensors active, interior sensors partially active
- Overnight Travel / Extended Travel: Enhanced security with simulated occupancy features
- Related Use Cases: UC 1.3.1 (Set System Mode)
- Reference: Dialog slide 9, UC 1.3.1

Alarm Condition

An event detected by a sensor that requires immediate attention, such as unauthorized entry, environmental hazard, or system tampering. Alarm conditions trigger notifications, sirens, and potentially emergency service dispatch.

- Related Use Cases: UC 1.2.1 (Configure Alarm Conditions), UC 1.2.3 (Emergency Service Integration)
- Reference: UC 1.1, UC 1.2

Sensor Bypass

A temporary exclusion of a faulted sensor from monitoring to allow system arming when that sensor cannot be secured (e.g., an open window). Bypassed sensors are excluded only for the current armed session and automatically re-enabled upon disarming.

- Related Use Cases: UC 1.3.2 (Sensor Bypass)
- Reference: Dialog slide 10, UC 1.3.2

Panic Button

An emergency feature accessible from the mobile app that immediately activates the highest-priority alarm, bypassing all verification steps. Used by homeowners in perceived emergency situations.

- Related Use Cases: UC 1.2.4 (Panic Button)
- Reference: UC 1.2.4

Two-Factor Authentication (2FA)

An additional security layer requiring users to provide two forms of verification during login: their password and a time-limited code from an authenticator app or SMS. Significantly reduces the risk of unauthorized account access.

- Related Use Cases: UC 4.1.7 (Two-Factor Authentication Management), UC 4.1.2 (Log In)
- Reference: UC 4.1.7

Exit Delay / Entry Delay

A configurable time window that allows users to leave or enter the home without triggering an alarm after arming the system. Typically 30-90 seconds.

- Related Use Cases: UC 1.3.1 (Set System Mode)
- Reference: UC 1.3.1, Scenario step 3a

User Roles and Access

Homeowner

The primary user of the SafeHome system with full access to all functions. Homeowners can arm/disarm the system, view all cameras, manage devices, configure settings, and invite other users.

- Related Use Cases: All use cases
- Reference: All use case Primary Actor fields

Admin Homeowner

A homeowner with elevated privileges who can modify system settings, manage other user accounts, add or remove devices, and configure security policies. At least one admin account must exist per system.

- Related Use Cases: UC 3.1.1 (Add and Configure New Devices), UC 3.3.1 (User Role and Access Control), UC 4.1.6 (Change Password)
- Reference: UC 3.3.1, UC 4.1.7

Guest

A limited-privilege user granted temporary or restricted access to specific system functions. Guests may be able to arm/disarm the system or view certain cameras but cannot modify settings or manage devices.

- Related Use Cases: UC 1.3.1 (Set System Mode), UC 4.1.2 (Log In)
- Reference: UC 3.3.1 (permission templates)

System Administrator (Software)

Not a human role, but the automated SafeHome system software that manages device communication, executes automation rules, and maintains system integrity.

- Related Use Cases: Referenced as secondary actor in multiple use cases
- Reference: Assumptions section

Operational Modes and States

Armed / Disarmed

The security state of the system:

- Armed: Sensors are actively monitoring for intrusion events
- Disarmed: Sensors continue operating but do not trigger alarms
- Related Use Cases: UC 1.3.1 (Set System Mode)
- Reference: UC 1.3.1

Alarm Verification

A workflow that requires homeowner confirmation before full alarm escalation. When a non-critical sensor event occurs, the system captures visual evidence and prompts the homeowner to confirm whether it is a genuine threat or false alarm.

- Related Use Cases: UC 1.2.2 (Alarm Verification Step)
- Reference: UC 1.2.2

Cooldown Period

A time interval after an event notification during which additional notifications of the same type are suppressed to prevent alert fatigue. For example, a motion-detected camera may only send one notification every 5 minutes.

- Related Use Cases: UC 1.1.3 (Outdoor Motion Detection), UC 2.3.2 (Notification Policy and Cooldown)
- Reference: UC 2.3.2

Activity Log / Timeline

A chronological record of all system events, including sensor activations, user actions, device state changes, and system alerts. Provides an audit trail for security review and incident investigation.

- Related Use Cases: UC 3.2.2 (Activity Logs and Timeline)
- Reference: Dialog slide 39, UC 3.2.2

Recording and Media

Live View

Real-time video streaming from a camera to the user's device with minimal latency (typically < 5 seconds). Enables users to observe their home remotely.

- Related Use Cases: UC 2.1.1 (Single Camera Live View)
- Reference: Dialog slides 29-31, UC 2.1.1

Recording

Captured video footage stored locally on the hub or in cloud storage. SafeHome supports:

- Continuous Recording: 24/7 video capture
- Event Recording: Motion-triggered or alarm-triggered capture with pre-roll and post-roll buffers
- Related Use Cases: UC 2.3.1 (Recording Settings), UC 2.2.1 (Search and Playback Recordings)
- Reference: UC 2.3.1

Camera Lock / Password Protection

A security feature that requires a specific password to access a camera's live view or recordings, providing additional privacy for sensitive areas.

- Related Use Cases: UC 2.1.3 (Protect Sensitive Camera Feed with a Password)

- Reference: Dialog slides 19-31, UC 2.1.3

Two-Way Audio

Bidirectional voice communication through a camera's built-in speaker and microphone, allowing the homeowner to speak to people at the camera's location in real-time.

- Related Use Cases: UC 2.1.2 (Two-Way Audio)
- Reference: Dialog slide 16, UC 2.1.2

Evidence Export

The ability to download or securely share recorded video clips or snapshots for purposes such as reporting to authorities, insurance claims, or personal backup.

- Related Use Cases: UC 2.2.2 (Evidence Sharing and Export)
- Reference: UC 2.2.2

Device Management

Device Registration / Pairing

The process of adding a new sensor or camera to the SafeHome system. Involves discovery, secure authentication, and initial configuration.

- Related Use Cases: UC 3.1.1 (Add and Configure New Devices)
- Reference: Dialog slide 58, UC 3.1.1

Device Status

Real-time information about a device's operational health, including:

- Online / Offline: Network connectivity state
- Battery Level: Remaining power for battery-operated devices
- Signal Strength: Quality of wireless connection to the hub
- Related Use Cases: UC 3.2.1 (System Status Dashboard)
- Reference: UC 3.2.1

Sensor Activation / Deactivation

The ability to enable or disable a sensor's monitoring function for a specified duration or indefinitely. Deactivated sensors remain registered but do not trigger events.

- Related Use Cases: UC 1.3.3 (Sensor Activation and Deactivation), UC 2.1.4 (Camera Activation and Deactivation)
- Reference: Meeting 2025.10.26/29, UC 1.3.3, UC 2.1.4

Technical Terms

Push Notification

An instant alert message sent to the user's mobile device by the SafeHome cloud server. Used for security events, system status updates, and reminders.

- Related Use Cases: UC 1.2.1 (Configure Alarm Conditions), UC 1.1.3 (Outdoor Motion Detection)
- Reference: Multiple use cases in Section 1 and 2

One-Time Password (OTP)

A time-limited verification code sent via SMS or email, used for account verification, password reset, and two-factor authentication.

- Related Use Cases: UC 4.1.1 (Sign Up), UC 4.1.4 (Reset Password), UC 4.1.7 (2FA Management)
- Reference: UC 4.1.1, UC 4.1.7

Session Token

A secure credential issued by the cloud server after successful authentication that grants the user access to the system for a limited time without requiring re-login.

- Related Use Cases: UC 4.1.2 (Log In), UC 4.1.3 (Log Out)
- Reference: UC 4.1.2

End-to-End Encryption (E2EE)

A security protocol that encrypts data at the source (e.g., camera) and decrypts it only at the destination (e.g., user's app), preventing intermediate parties from accessing the content.

- Related Use Cases: UC 2.1.1 (Single Camera Live View)

- Reference: Goal section, Non-functional requirements

Audit Log

A detailed, tamper-evident record of all administrative actions and security-sensitive operations, including who performed the action, when, and from which device. Used for security compliance and forensic analysis.

- Related Use Cases: UC 3.2.2 (Activity Logs and Timeline), UC 4.1.1 (Sign Up)
- Reference: Multiple use cases across all sections

Idempotent Request

A request that can be safely repeated multiple times without causing duplicate or unintended effects. SafeHome uses idempotent requests with unique IDs to prevent double-processing during network retries.

- Related Use Cases: UC 4.1.1 (Sign Up), UC 1.2.1 (Configure Alarm Conditions)
- Reference: UC 4.1.1, UC 1.2.1 Exception handling

Network and Connectivity

LAN (Local Area Network)

The home's private network that connects the hub, sensors, cameras, and smart devices. SafeHome uses Wi-Fi (802.11n) for wireless connectivity.

- Related Use Cases: All device communication
- Reference: Assumptions section

Cloud Connectivity

The internet connection between the SafeHome hub and cloud server that enables remote access, push notifications, firmware updates, and data synchronization.

- Related Use Cases: All remote access features, UC 4.1.2 (Log In)
- Reference: Major Functionalities section

Offline Mode

A degraded operational state when the hub loses internet connectivity. Core security functions (sensor monitoring, local alarms) continue to operate, but remote access and cloud notifications are unavailable until connection is restored.

- Related Use Cases: Exception scenarios in UC 1.1.2, UC 1.2.3
- Reference: Multiple use case exception sections

Environmental Monitoring (Second Increment)

Indoor Air Quality (IAQ)

A measure of air cleanliness based on pollutant levels (CO_2, VOCs, particulates). SafeHome can monitor IAQ and automatically activate ventilation when thresholds are exceeded.

- Related Use Cases: UC 5.2.1 (Indoor Air Quality Monitoring)
- Reference: Dialog slide 27, UC 5.2.1 (Second Increment)

Smart Meter

A device that monitors real-time electrical power consumption, enabling energy usage tracking and analysis.

- Related Use Cases: UC 5.2.2 (Real-Time Power Consumption Monitoring)
- Reference: Dialog slide 29, UC 5.2.2 (Second Increment)