



CS350 Software Engineering SafeHome Project

Design Model

Project Team #7 Members:

20230943	Thiwat
20230970	Tattep
20230988	Thommakoon
20231008	Ngo Thai Binh

I. Overview	4
1. Introduction	4
2. Goal	4
3. How the design work proceeded	4
4. Assumptions	5
II. Architectural structure	8
1. Overall Architecture Class Diagram	8
2. SystemHub Architecture Class Diagram CRC Card State Diagram	8
3. CloudServer Architecture Class Diagram CRC Card State Diagram	9
4. MobileApp Architecture Class Diagram CRC Card State Diagram	9
III. Class Diagram	10
1. Class Diagram – whole system overview Architecture	10
2. Class Diagram – SystemHub Architecture CRC Card State Diagram	11
a. Class Diagram – SecurityManager CRC Card State Diagram	11
b. Class Diagram – DeviceManager State Diagram	12
c. Class Diagram – CameraManager CRC Card State Diagram	13
3. Class Diagram – CloudServer Architecture CRC Card State Diagram	14
4. Class Diagram – MobileApp Architecture CRC Card State Diagram	15
IV. CRC cards	16
1. CRC Cards - SystemHub Architecture Class Diagram State Diagram	16
a. Around SecurityManager Class Diagram State Diagram	16
b. Around AlarmManager State Diagram	18
c. Around SensorManager State Diagram	19
d. Around SafetyZoneManager State Diagram	20
e. Around CameraManager Class Diagram State Diagram	21
f. Around ModeManager State Diagram	22
g. Around StorageRepository	23
i. Around LogManager	25
2. CRC Cards - MobileApp Architecture Class Diagram State Diagram	26
a. Around MobileApp	26
3. CRC Cards - CloudServer Architecture Class Diagram State Diagram	29
a. Around UserManager State Diagram	29
b. Around CloudServer	30
V. State Diagram	33
1. Classes around SystemHUB Architecture Class Diagram CRC Card	33
a. State Diagram for SecurityManager Class Diagram CRC Card	33
b. State Diagram for ModeManager CRC Card	34
c. State Diagram for SafetyZoneManager CRC Card	35
d. State Diagram for CameraManager Class Diagram CRC Card	35
e. State Diagram for DeviceManager Class Diagram	36
f. State Diagram for SensorManager CRC Card	37

g. State Diagram for AlarmManager CRC Card	37
2. Classes around CloudServer Architecture Class Diagram CRC Card	38
a. State Diagram for Video Streaming	38
b. State Diagram for User login CRC Card	39
3. Classes around MobileApp Class Diagram CRC Card Architecture	40
VI. Design evaluation	41
1. Architectural Design Metric	41
a. Design Structure Quality Index (DSQI)	41
b. Fenton's simple morphology metrics	41
2. CK Metrics	43
3. MOOD Metric	44
a. MIF (Method Inheritance Factor)	44
b. CF (Coupling Factor)	45
4. OO Metric Proposed by Lorenz and Kidd	46
VI. WHO DID WHAT	47
VII. MEETING LOGS	48
1st Meeting	48
2nd Meeting	49
3rd Meeting	49
APPENDIX A. GLOSSARY	51

I. Overview

1. Introduction

This document describes the design model of the SafeHome system proposed in the previous report. Since the design phase of the product is directly connected to the implementation phase, we focus on the well-formed and concrete design of the system. Architectural structure, class diagram, CRC cards, state diagrams, and sequence diagrams are presented to portray the design of the system.

2. Goal

- 1) Completely follow the requirement and the analysis model.
- 2) Achieve low coupling, high cohesion, and modularity.
- 3) Pursue testability, integrity, efficiency, maintainability, and reliability.
- 4) Minimize complexity and consider reusability and flexibility.

3. How the design work proceeded

- 1) Extraction of classes: To achieve correctness of the design model we used the method in chapter 8.7 of SEPA. We reviewed nouns and verbs from use case scenarios to extract classes.

Use Case ID	Type	Sentence	Nouns	Verbs	Candidate Class	Attribute	Operation	Notes	Collaborators	Who did this?	Reference
1. Intelligent Security											
1.1 Sensor Monitoring											
1.1.1 Physical Intrusion Detection											
1.1.1	Scenario	A 'Shock Sensor' on the living room window detects a breach event and transmits it to the SafeHome system hub.	Shock Sensor, window, breach event, SafeHome system hub	detects, transmits	ShockSensor	location, threshold	detectEvent(), sendEvent()	Window Impact Attempt	SystemHub	Thommakoon	References on SRS Team 1, Use Case: 1.1.1 Physical Intrusion Detection Page 31-32
1.1.1	Scenario	The system hub verifies the current security state is 'Away' and identifies the event as a 'physical intrusion attempt'.	System hub, security state, event, physical intrusion attempt	verifies, identifies	SystemHub	currentMode	verifySecurityState(), identifyIntrusion()	Trigger 3 scenario	ShockSensor	Thommakoon	References on SRS Team 1, Use Case: 1.1.1 Physical Intrusion Detection Page 31-32
1.1.1	Scenario	The system immediately sounds the internal siren (NFR: within 3 seconds).	System, internal siren	sounds	SystemHub	-	soundAlarm()	NFR: ≤3s	Siren	Thommakoon	References on SRS Team 1, Use Case: 1.1.1 Physical Intrusion Detection Page 31-32
1.1.1	Scenario	The hub sends the event to the cloud.	Hub, event, cloud	sends	SystemHub	-	sendToCloud()	Parallel step	CloudServer	Thommakoon	References on SRS Team 1, Use Case: 1.1.1 Physical Intrusion Detection Page 31-32
1.1.1	Scenario	The cloud service dispatches emergency push notifications and SMS messages to all registered Homeowners.	Cloud service, push notifications, SMS messages, Homeowners	dispatches	CloudServer	-	dispatchNotification(), dispatchSMS()	NFRs: ≤5s push, ≤10s SMS	MobileApp, Homeowner	Thommakoon	References on SRS Team 1, Use Case: 1.1.1 Physical Intrusion Detection Page 31-32
1.1.1	Scenario	The system automatically reports the intrusion to a pre-configured external security service.	System, intrusion, external security service	reports	SystemHub	-	reportToExternalService()	NFR: ≤15s	ExternalSecurityService	Thommakoon	References on SRS Team 1, Use Case: 1.1.1 Physical Intrusion Detection Page 31-32
1.1.1	Scenario	When the Homeowner opens the mobile app via the notification, it displays the alarm status and the live feed from the relevant camera.	Homeowner, mobile app, alarm status, live feed, camera	opens, displays	MobileApp	-	displayAlarmStatus(), showLiveFeed()	From notification	Camera, SystemHub	Thommakoon	References on SRS Team 1, Use Case: 1.1.1 Physical Intrusion Detection Page 31-32
1.1.1	Scenario	All outcomes (siren, notifications, dispatch results) and the initial detection event are permanent entries in the activity log.	outcomes, siren, notifications, dispatch results, detection event, activity log	recorded	ActivityLog	-	recordEvent()	Persistent logging	SystemHub, CloudServer, Siren	Thommakoon	References on SRS Team 1, Use Case: 1.1.1 Physical Intrusion Detection Page 31-32
1.1.1	Exception	The system logs the siren failure and proceeds with Homeowner notifications and external reporting.	system, siren, failure, Homeowner, notifications, reporting	logs, proceeds	SystemHub	failureRecord	logFailure()	Handle siren offline	CloudServer, ExternalSecurityService	Thommakoon	References on SRS Team 1, Use Case: 1.1.1 Physical Intrusion Detection Page 31-32
1.1.1	Exception	If a Homeowner's device is offline, the system logs the push notification failure.	Homeowner, device, system, push notification, failure	logs	CloudServer	failureRecord	logNotificationFailure()	Device offline handling	HomeownerApp	Thommakoon	References on SRS Team 1, Use Case: 1.1.1 Physical Intrusion Detection Page 31-32
1.1.1	Exception	SMS fallback is attempted while internal siren and external reporting proceed normally.	SMS, fallback, siren, reporting	attempted, proceed	CloudServer	-	sendSMSFallback(), continueReporting()	Notification fallback	SystemHub, ExternalSecurityService	Thommakoon	References on SRS Team 1, Use Case: 1.1.1 Physical Intrusion Detection Page 31-32
1.1.1	Exception	If a life-safety event occurs during an intrusion alarm, the system prioritizes the life-safety alarm.	system, life-safety event, intrusion alarm	occurs, prioritizes	SystemHub / AlarmManager	priorityLevel	prioritizeLifeSafetyAlarm()	Life-safety override: intrusion logged, push-only	AlarmManager, ActivityLog	Thommakoon	References on SRS Team 1, Use Case: 1.1.1 Physical Intrusion Detection Page 31-32
1.1.1	Exception	If conflicting commands arrive simultaneously, they are resolved in priority order.	commands, priority order, Control Panel, Mobile	arrive, resolved	CommandHandler	-	resolveConflict()	Command arbitration priority list	ControlPanel, MobileApp, Automation	Thommakoon	References on SRS Team 1, Use Case: 1.1.1 Physical Intrusion Detection Page 31-32
1.1.1	Exception	A 'Disarm' command cancels the dispatch only if it arrives before the dispatch is committed.	Disarm command, dispatch	cancels, arrives, committed	CommandHandler	timestamp	cancelDispatch()	Timing-based cancel	CloudServer, MobileApp	Thommakoon	References on SRS Team 1, Use Case: 1.1.1 Physical Intrusion Detection Page 31-32
1.1.1	Exception	Multiple intrusion triggers within 60 seconds are consolidated into a single incident.	intrusion triggers, incident, 60-second window	consolidated	EventManager	dedupWindow	consolidateEvents()	Rate-limit alerts (<3/min)	Sensor, CloudServer	Thommakoon	References on SRS Team 1, Use Case: 1.1.1 Physical Intrusion Detection Page 31-32

- 2) Creation of architectural structure: Based on the classes extracted and the use case scenario, we created the architectural structure of the SafeHome system.
- 3) Creation of class diagram: On the basis of extracted classes and architectural structure, we created the class diagram considering the implementation.
- 4) Creation of CRC card.
- 5) Refinement of class diagram: By testing the design using CRC card and reviewing the requirement document and the first report, we refined the class diagram. In addition we focused on ways to achieve low coupling and high cohesion. Actual implementation plan became more concrete and added some classes from java and for database access
- 6) Creation of state diagram.
- 7) Refinement of class diagram: Added some missing functions and attributes
- 8) Creation of sequence diagram: Based on the use case scenario we created the sequence diagram. It enabled us to check if the design followed the requirement specification and the first report.
- 9) Review of state and sequence diagram: Tested based on the first report.
- 10) Refinement of class diagram: Added some missing functions and attributes in the view point of implementation.

4. Assumptions

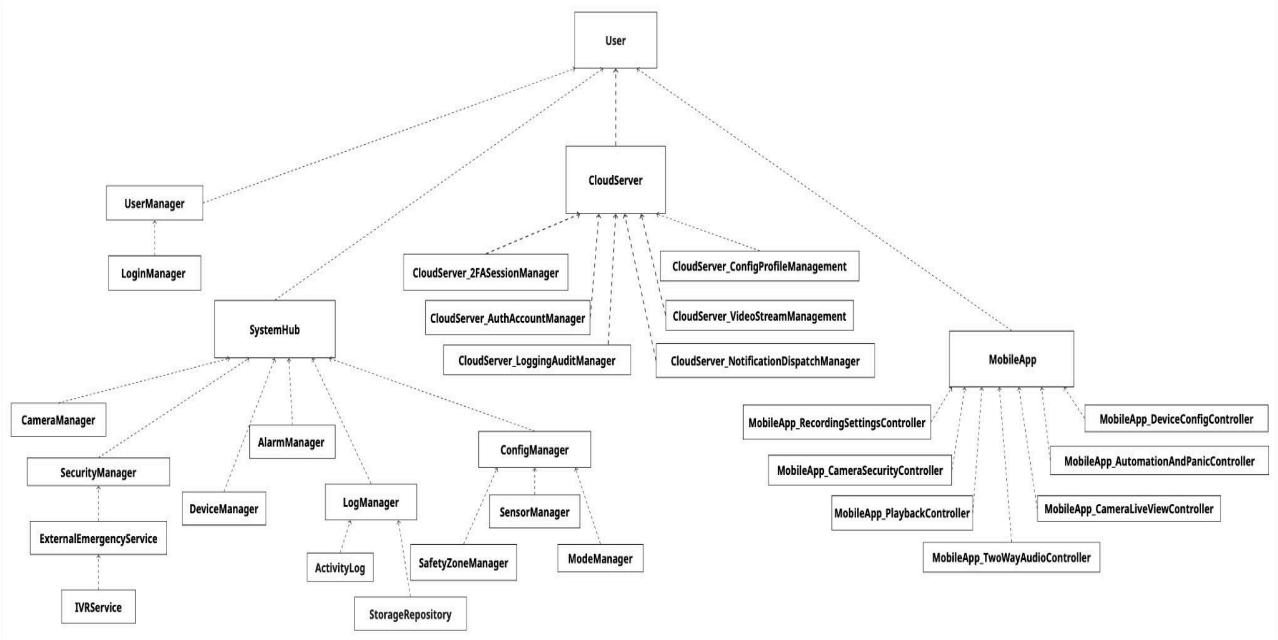
- 1) SystemHub as the Central Orchestrator:
 - SystemHub is the main coordinator for the SafeHome system, managing all local modules including security (SecurityManager), alarms (AlarmManager), sensors (SensorManager), cameras (CameraManager), safety zones (SafetyZoneManager), devices (DeviceManager), and operational modes (ModeManager).
 - It maintains the global system state and ensures synchronization of local actions with the cloud.
 - Assumption: SystemHub is always active when the system is running and acts as the single point of truth for all local operations.
- 2) CloudServer as the Central Backend:
 - CloudServer handles remote services such as authentication and authorization (CloudServer_AuthAccountManager), two-factor authentication (CloudServer_2FAStateManager), configuration management (CloudServer_ConfigProfileManagement), notifications and escalation (CloudServer_NotificationDispatchManagement), logging, and video stream management (CloudServer_VideoStreamManagement).
 - It ensures data persistence, consistency, and reliability for critical logs, device configurations, and user accounts.
 - Assumption: CloudServer is assumed reachable for remote synchronization and command propagation, with fallback mechanisms for temporary offline operation.
- 3) MobileApp as the Primary User Interface:
 - The MobileApp acts as the trusted client interface for users to interact with SafeHome, including live camera monitoring, playback, device control, automation routines, panic alerts, and account management.
 - It communicates with both SystemHub (local commands, sensor events) and CloudServer (authentication, configuration, notifications).
 - Assumption: MobileApp assumes secure communication channels, reliable

- synchronization with the cloud, and real-time updates for critical events like alarms or security state changes.
- 4) Modular and Cohesive Design:
 - Each functional area is implemented via dedicated manager classes (e.g., SecurityManager, AlarmManager, SensorManager, CameraManager, LogManager, ModeManager, DeviceManager, ConfigManager) to ensure high cohesion and single responsibility.
 - Assumption: Each manager handles its domain independently but coordinates through defined interfaces or the SystemHub.
 - 5) Traceable Responsibilities:
 - Every class responsibility is linked to one or more use cases, ensuring traceability, accountability, and comprehensive coverage.
 - Assumption: All functionality in CRC cards is assumed implemented according to linked use cases.
 - 6) Expandable and Extensible Architecture:
 - The system allows addition of new classes, modules, or devices without breaking existing functionality. Examples include emergency services (ExternalEmergencyService), IVR workflows (IVRService), new sensors, or new smart devices (SmartDevices).
 - Assumption: The architecture is modular and can accommodate future system expansions.
 - 7) Sensor and Device Abstraction:
 - All sensors (Sensor, MovementSensor, ShockSensor, COSensor, SoundAnalyzer) and smart devices (Device, LightDevice, VentilationSystem) are abstracted and managed via central managers (SensorManager, SmartDevices).
 - Assumption: This abstraction standardizes detection, event processing, automation commands, and status reporting across heterogeneous devices.
 - 8) Safety Zone and Mode Management:
 - Safety zones (SafetyZone, SafetyZoneManager) and system modes (ModeManager) coordinate with sensors, devices, and other modules to control system behavior.
 - Assumption: Zones and modes are configurable, independently managed, and always reflect the current system state.
 - 9) Camera and Media Management:
 - Cameras (Camera) are managed through CameraManager and cloud services (CloudServer_VideoStreamManagement) for live streaming, recording, snapshots, audio, and event notifications.
 - Assumption: Camera operations are reliable, with real-time streaming and fault handling for offline or error states.
 - 10) Alarm and Security Integration:
 - Security (SecurityManager) and alarm (AlarmManager) managers handle intrusion detection, verification, escalation, notifications, and user alerts, in coordination with sensors, users, and other modules.
 - Assumption: Critical events are processed with priority, and alarms escalate automatically when required.
 - 11) Authentication, Accounts, and 2FA:
 - User accounts (User), login sessions (LoginManager), password management, and 2FA (CloudServer_2FASessionManager) ensure secure authentication and access control.
 - Assumption: Users are validated through secure protocols, and session integrity is

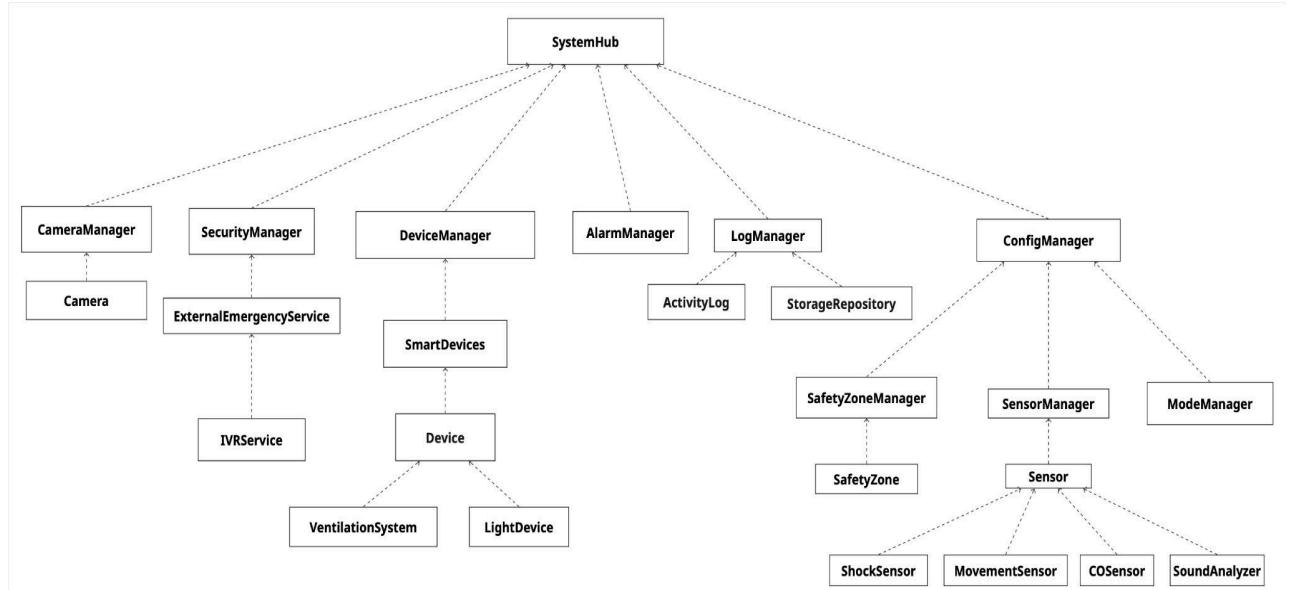
- maintained across cloud and mobile interfaces.
- 12) Logging and Storage:
 - Logs and events are stored and managed by LogManager and StorageRepository, ensuring fault tolerance, chronological integrity, and export capabilities.
 - Assumption: System can operate locally if cloud storage is temporarily unavailable.
 - 13) Cloud and Notification Services:
 - Cloud modules handle device configuration distribution, emergency notifications, alerts, and escalation sequences.
 - Assumption: Cloud services are reliable and maintain consistency across mobile and local modules.
 - 14) Fail-Safe and Redundancy:
 - Local storage and local system operation provide temporary redundancy in case of cloud outages.
 - Assumption: Critical functions (e.g., alarms, arming/disarming, sensor monitoring) continue locally even if cloud services are unavailable.
 - 15) Event-Driven Communication:
 - System modules communicate primarily via events, interfaces, or centralized coordination (SystemHub), rather than direct coupling.
 - Assumption: Event-driven architecture ensures scalability, decoupling, and maintainability.
 - 16) Real-Time and Low-Latency Operations:
 - The system assumes real-time or near-real-time response for critical operations such as alarms, security mode changes, panic actions, and camera streaming.
 - Assumption: Network latency between SystemHub, CloudServer, and MobileApp is low enough not to affect critical security operations.
 - 17) Class Cohesion and Single Responsibility:
 - Managers encapsulate closely related functionality to reduce coupling.
 - Assumption: Each class performs a clearly defined role and interacts with collaborators only via defined responsibilities.

II. Architectural structure

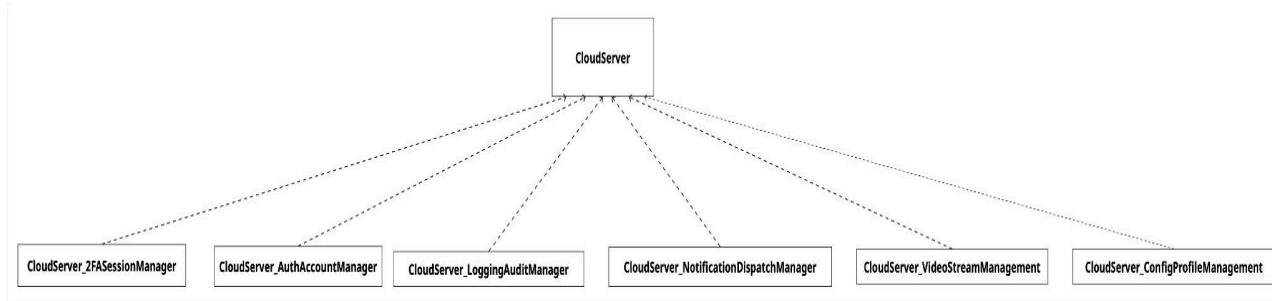
1. Overall Architecture [Class Diagram](#)



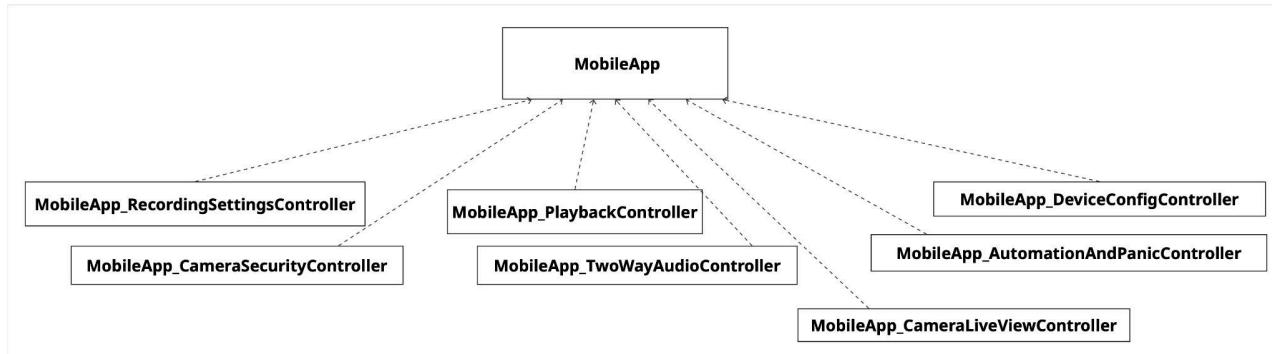
2. SystemHub Architecture [Class Diagram CRC Card State Diagram](#)



3. CloudServer Architecture [Class Diagram](#) [CRC Card](#) [State Diagram](#)

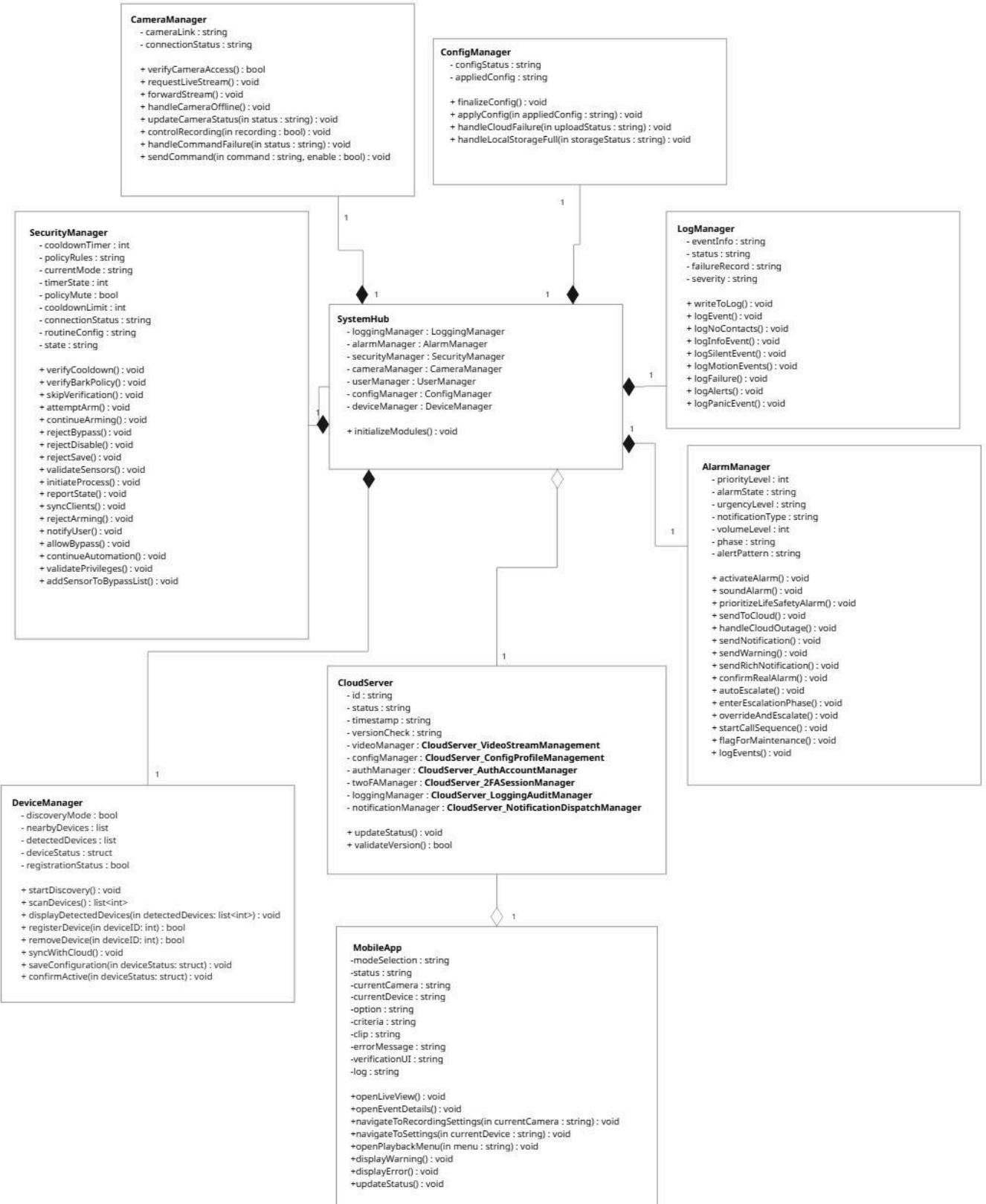


4. MobileApp Architecture [Class Diagram](#) [CRC Card](#) [State Diagram](#)



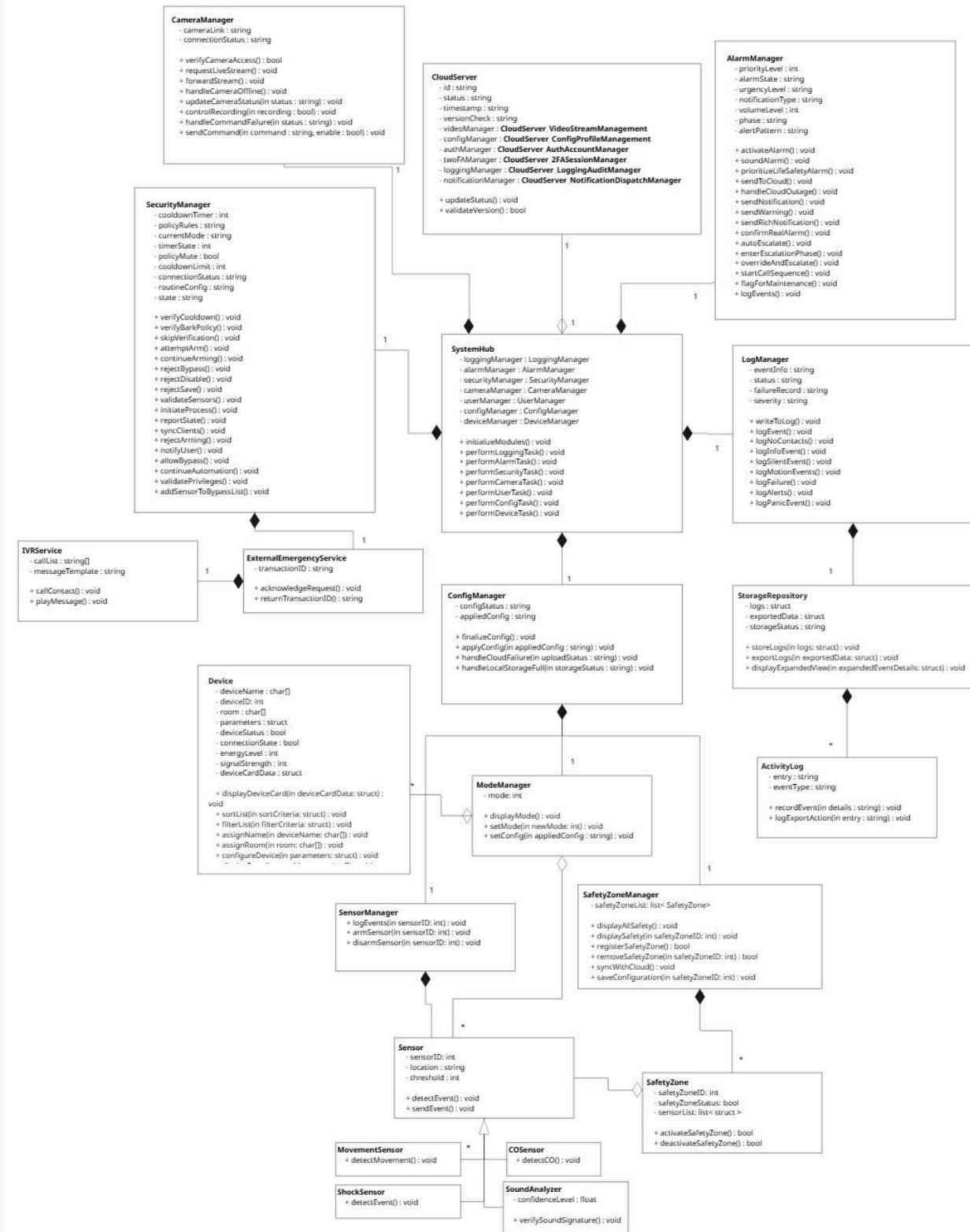
III. Class Diagram

1. Class Diagram – whole system overview [Architecture](#)

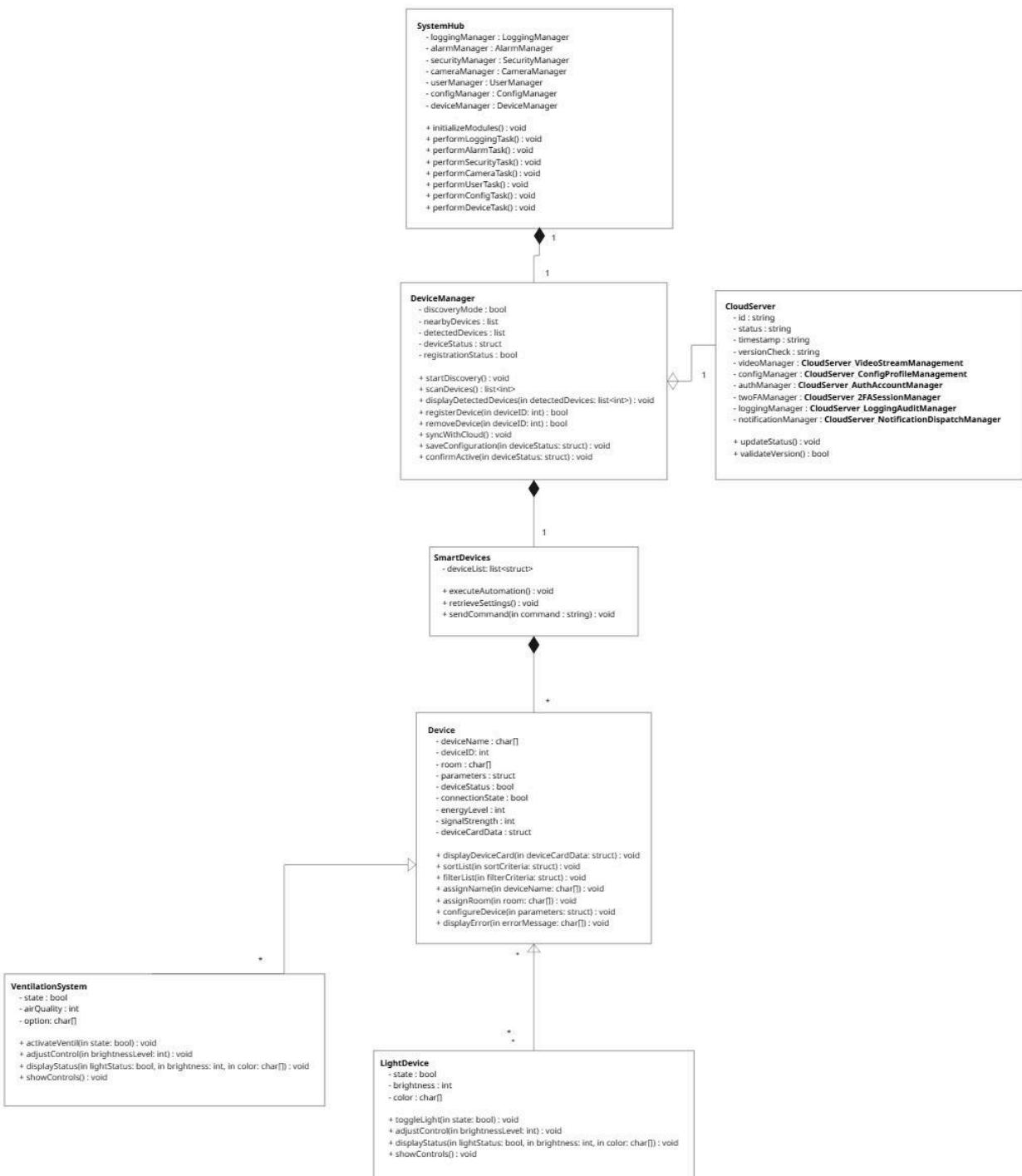


2. Class Diagram – SystemHub [Architecture](#) [CRC Card State Diagram](#)

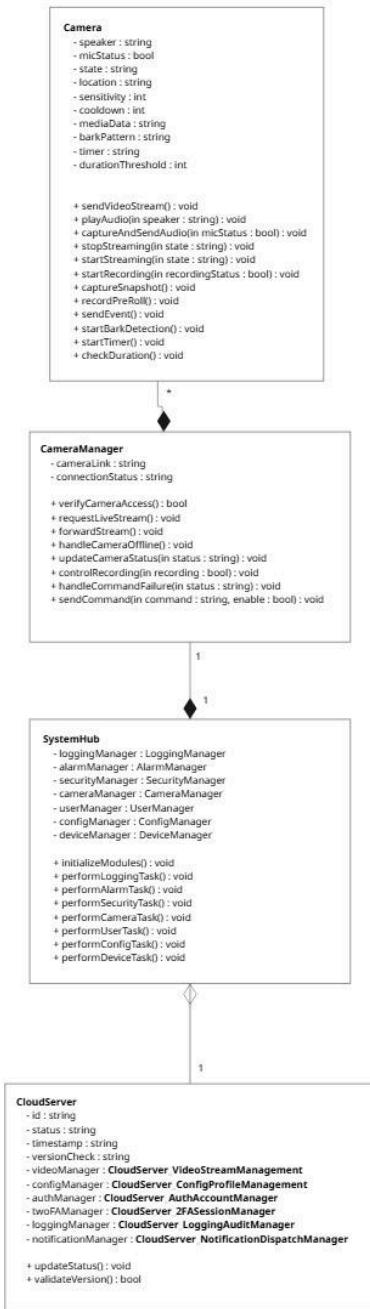
a. Class Diagram – SecurityManager [CRC Card State Diagram](#)



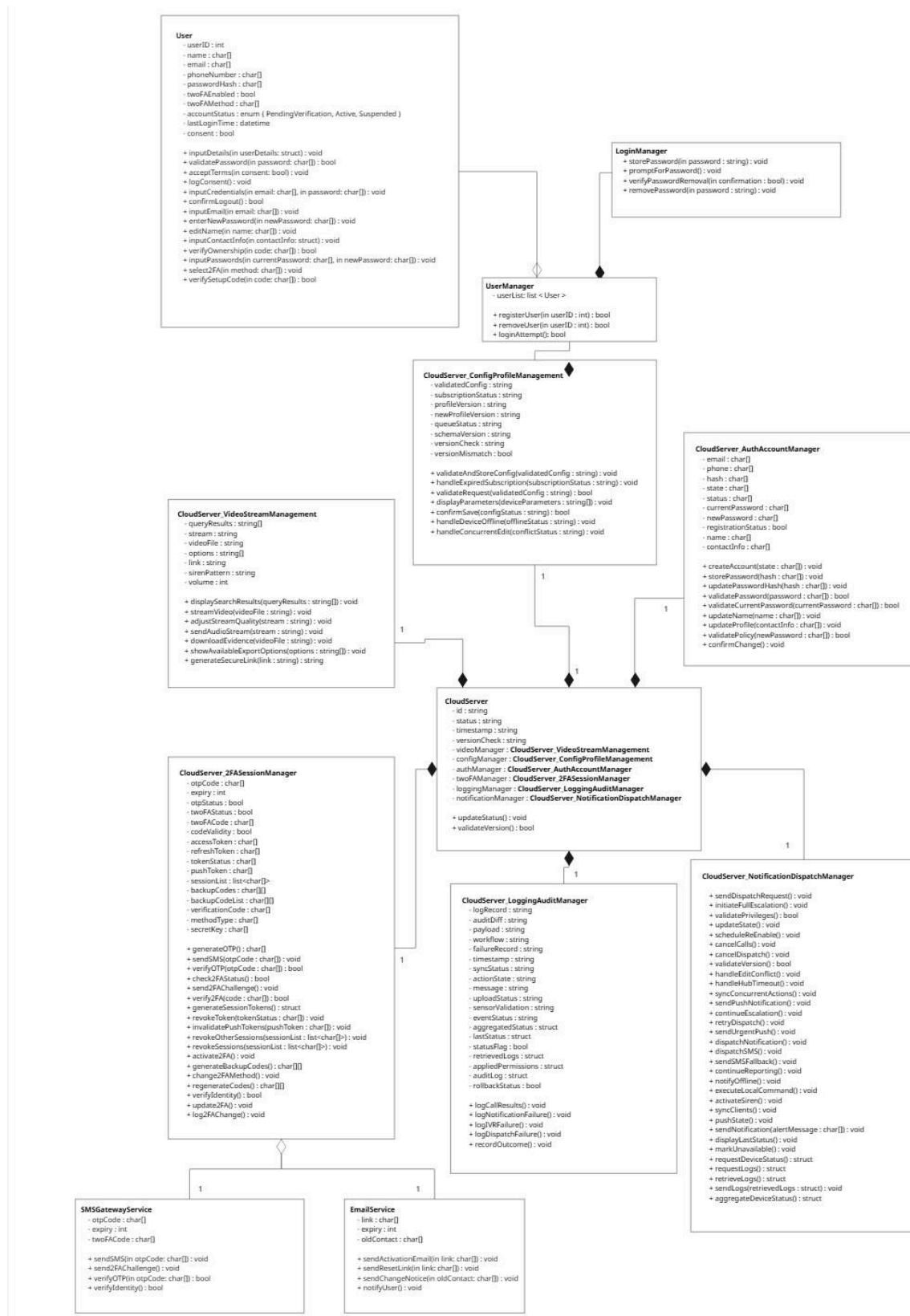
b. Class Diagram – DeviceManager State Diagram



c. Class Diagram – CameraManager [CRC Card State Diagram](#)



3. Class Diagram – CloudServer Architecture CRC Card State Diagram



4. Class Diagram – MobileApp Architecture CRC Card State Diagram



IV. CRC cards

1. CRC Cards - SystemHub [Architecture Class Diagram State Diagram](#)

Class: SystemHub	
Central coordinator of the entire intelligent home security ecosystem	
Responsibilities	Collaborators
Initialize all core modules ()	
Performs system logging tasks (UC 3.2.2)	LoggingManager
Performs alarm-related tasks (UC 1.1.1–1.1.4, 1.2.1–1.2.4)	AlarmManager
Performs security-related tasks (UC 1.3.1–1.3.3)	SecurityManager
Performs camera-related tasks (UC 2.1.1–2.1.4)	CameraManager
Performs user-management tasks (UC 3.3.1, 4.1.1–4.1.7)	UserManager
Performs configuration-related tasks (UC 1.2.1, 1.3.1–1.3.3, 2.3.1–2.3.2)	ConfigManager
Performs device-management tasks (UC 3.1.1, 5.1.1, 5.2.1–5.2.2)	DeviceManager

a. Around SecurityManager [Class Diagram State Diagram](#)

Class: SecurityManager	
Manages intrusion detection, arming processes, verification rules, and overall security mode behavior	
Responsibilities	Collaborators
Verifies cooldown timers before arming (UC 1.2.1, 2.3.2)	ConfigManager, SystemHub
Checks bark/alert policy rules (UC 2.3.2, 1.1.4)	ConfigManager, UserManager
Validates user privileges before security actions (UC 3.3.1)	UserManager, SystemHub
Validates all active sensors (UC 1.1.1–1.1.4, 1.3.3)	SensorManager, AlarmManager, SystemHub
Initiates the security arming/disarming process (UC 1.3.1, 1.3.3)	SensorManager, SystemHub
Allows or rejects bypass requests (UC 1.3.2)	UserManager, MobileApp, SystemHub
Allows or rejects automation continuation (UC 1.2.2–1.2.3)	UserManager, AlarmManager, SystemHub
Notifies user of security state changes (UC 1.1.1–1.1.4, 1.2.2)	ConfigManager, MobileApp

Class: SecurityManager	
Manages intrusion detection, arming processes, verification rules, and overall security mode behavior	
Responsibilities	Collaborators
Verifies cooldown timers before arming (UC 1.2.1, 2.3.2)	ConfigManager, SystemHub
Checks bark/alert policy rules (UC 2.3.2, 1.1.4)	ConfigManager, UserManager
Validates user privileges before security actions (UC 3.3.1)	UserManager, SystemHub
Validates all active sensors (UC 1.1.1–1.1.4, 1.3.3)	SensorManager, AlarmManager, SystemHub
Initiates the security arming/disarming process (UC 1.3.1, 1.3.3)	SensorManager, SystemHub
Allows or rejects bypass requests (UC 1.3.2)	UserManager, MobileApp, SystemHub
Reports current security state (UC 3.2.1)	SensorManager, AlarmManager, CameraManager
Synchronizes security state to connected clients (UC 4.1.2, 4.1.7)	MobileApp, CloudServer
Handles arming attempts and continuation logic (UC 1.3.1-1.3.3, 1.2.2)	SensorManager, AlarmManager, UserManager, ConfigManager, SystemHub, MobileApp
Rejects invalid disable/arm/disarm operations (UC 1.3.1-1.3.3 ,1.2.2)	UserManager, ConfigManager, AlarmManager, SystemHub, MobileApp
Skips verification when allowed (UC 1.2.2)	UserManager, ConfigManager, SystemHub, MobileApp
Adds sensors to the bypass list (UC 1.3.2)	SensorManager, UserManager, SystemHub

Class: ExternalEmergencyService	
Handles communication with external emergency authorities or third-party responders	
Responsibilities	Collaborators
Acknowledges emergency requests sent by the system (UC 1.2.3)	SystemHub, AlarmManager
Generates and returns a transaction ID for tracking (UC 1.2.3)	SystemHub
Initiates external contact workflows using the IVR system (UC 1.2.3)	SystemHub, IVRService
Ensures emergency notification procedures complete successfully (UC 1.2.3)	IVRService, CloudServer_NotificationDispatchManager, SystemHub

Class: IVRService

Automated Interactive Voice Response component used for calling contacts and playing emergency messages	
Responsibilities	Collaborators
Calls external emergency contacts through automated dialing(UC 1.2.3)	ExternalEmergencyService, UserManager
Plays predefined emergency alert messages (UC 1.2.3)	ExternalEmergencyService, AlarmManager
Loads and uses message templates during emergencies (UC 1.2.3)	CloudServer_NotificationDispatchManager, AlarmManager
Executes automated voice flows as triggered by an emergency request (UC 1.2.3)	ExternalEmergencyService, SystemHub, AlarmManager

b. Around AlarmManager [State Diagram](#)

Class: AlarmManager	
Handles all alarm activation, notification, escalation, and maintenance processes within the SafeHome system	
Responsibilities	Collaborators
Activates the alarm when a valid triggering event is detected (UC 1.1.1–1.1.4, 1.2.1)	SensorManager, SystemHub, SafetyZone
Produces the audible alarm sound through the system's alarm devices. (UC 1.1.1–1.1.4, 1.2.1)	Device, SystemHub
Prioritizes life-safety alarms (fire, smoke, CO) above all other alarm types (UC 1.1.2, 1.2.1)	SensorManager, SystemHub, CloudServer_NotificationDispatchManager
Sends alarm event data to the cloud for remote monitoring and backup (UC 1.1.1–1.1.4, 1.2.1–1.2.3)	CloudServer, SystemHub
Handles cloud outages by switching to local storage and fallback notification mechanisms (UC 1.2.3)	CloudServer, StorageRepository, CloudServer_NotificationDispatchManager
Sends standard alarm notifications to registered users (UC 1.1.1–1.1.4, 1.2.2–1.2.3)	CloudServer_NotificationDispatchManager, UserManager, SystemHub
Sends warning-level alerts for non-critical alarm conditions (UC 1.2.2)	CloudServer_NotificationDispatchManager, UserManager
Sends rich, enhanced alarm notifications containing images, audio, or video (UC 1.2.3)	CameraManager, CloudServer_NotificationDispatchManager, SystemHub
Confirms whether an alarm event is a real emergency based on user input and sensor validation (UC 1.2.2)	UserManager, SystemHub, SensorManager
Automatically escalates alarm events when no confirmation is received or the situation becomes critical (UC 1.2.3)	ExternalEmergencyService, IVRService, CloudServer_NotificationDispatchManager

Enters the escalation phase and updates the system state to reflect an ongoing emergency escalation (UC 1.2.3)	ExternalEmergencyService, IVRService, SystemHub
Overrides normal verification rules and escalates the alarm immediately at the user's request (UC 1.2.4)	UserManager, SystemHub, ExternalEmergencyService
Starts the emergency call sequence to notify all contacts in the escalation list (UC 1.2.3, 1.2.4)	IVRService, UserManager, ExternalEmergencyService
Flags the alarm system for maintenance when operational issues or faults are detected (UC 3.2.1)	AlarmManager, SystemHub, UserManager
Logs all alarm-related events and status changes for auditing and troubleshooting (UC 3.2.2)	LogManager, CloudServer, SystemHub

c. Around SensorManager [State Diagram](#)

Class: SensorManager	
Handles all sensor-related operations, including arming, validation, event handling, and logging within the SafeHome security system	
Responsibilities	Collaborators
Logs all sensor-related status changes for auditing and troubleshooting (UC 3.2.2)	LogManager, SystemHub
Arms sensors according to the current system mode and configuration rules (UC 1.3.1, 1.3.3)	ModeManager, ConfigManager, ShockSensor, MovementSensor, COSensor, SoundAnalyzer
Disarms sensors according to the current system mode and configuration rules (UC 1.3.1, 1.3.3)	ModeManager, ConfigManager, ShockSensor, MovementSensor, COSensor, SoundAnalyzer

Class: Sensor	
Represents a generic sensor in the SafeHome system, responsible for detecting events and communicating them to the system	
Responsibilities	Collaborators
Detects environmental events based on sensor type and threshold (UC 1.1.1–1.1.4)	SensorManager, SecurityManager
Sends detected events to the SensorManager for further processing (UC 1.1.1–1.1.4)	SensorManager, SystemHub

Class: MovementSensor

Detects motion events and reports them through the Sensor system	
Responsibilities	Collaborators
Detects motion within its assigned area and triggers a motion event. (UC 1.1.3)	SensorManager, SecurityManager

Class: ShockSensor	
Detects vibration or physical impact events	
Responsibilities	Collaborators
Detects vibration or shock that may indicate forced entry. (UC 1.1.1)	SensorManager, SecurityManager

Class: COSensor	
Detects carbon monoxide concentration levels	
Responsibilities	Collaborators
Monitors CO levels and detects abnormal or hazardous concentrations (UC 1.1.2)	SensorManager, SecurityManager

Class: SoundAnalyzer	
Analyzes acoustic patterns to verify specific sound signatures (e.g., breaking glass, baby crying, abnormal noise)	
Responsibilities	Collaborators
Analyzes detected sound patterns and determines whether they match known risk signatures (UC 1.1.4)	SensorManager, SecurityManager

d. Around SafetyZoneManager State Diagram

Class: SafetyZoneManager	
Manages all safety zones in the SafeHome system, including creation, display, modification, and cloud synchronization	
Responsibilities	Collaborators
Displays all safety zones and their current status for user or system review (UC 1.3.1, 3.2.1)	SafetyZone, ConfigManager

Displays detailed information of a specific SafetyZone when requested (UC 3.2.1)	SafetyZone, Sensor
Registers a new SafetyZone into the system and initializes its configuration (UC 3.1.1)	SafetyZone, SystemHub, ConfigManager
Removes an existing SafetyZone from the system safely and updates linked components (UC 3.1.1)	SafetyZone, SensorManager, ConfigManager
Synchronizes safety zone information with the cloud to ensure consistent remote access (UC 4.1.2, 4.1.7)	CloudServer_ConfigProfileManagement, SystemHub
Saves updated SafetyZone configuration after modifications or mode changes (UC 1.3.1, 1.3.3)	SafetyZone, ConfigManager

Class: SafetyZone	
Represents a logical area or grouping of sensors in the SafeHome system, controlling activation status and managing zone behavior	
Responsibilities	Collaborators
Activates the SafetyZone and enables all sensors within it. (UC 1.3.3)	Sensor, SensorManager, SecurityManager
Deactivates the SafetyZone and disables all associated sensors (UC 1.3.3)	Sensor, SensorManager, SecurityManager

e. Around CameraManager [Class Diagram](#) [State Diagram](#)

Class: CameraManager	
Handles all camera operations and communication within the SafeHome system	
Responsibilities	Collaborators
Verify camera access (UC 2.1.1, 2.1.3)	Camera
Request live stream (UC 2.1.1)	Camera
Forward video stream to system (UC 2.1.1)	Camera
Handle camera going offline (UC 2.1.1)	Camera
Update camera status (active, idle, cooldown, recording) (UC 2.1.4, 2.3.1)	Camera
Control recording start/stop (UC 2.3.1, 2.2.1)	Camera
Handle command failure from camera (UC 2.1.1–2.1.4)	Camera
Send command (enable/disable mic, speaker, stream, record) (UC 2.1.2, 2.1.4, 2.3.1)	Camera

Class: Camera	
Represents a physical camera device that handles streaming, recording, audio, and event detection	
Responsibilities	Collaborators
Send video stream (UC 2.1.1)	CameraManager
Play audio through speaker (UC 2.1.2)	CameraManager
Capture and send audio (mic) (UC 2.1.2)	CameraManager
Start video streaming (UC 2.1.1)	CameraManager
Stop video streaming (UC 2.1.1)	CameraManager
Start recording (based on recordingStatus) (UC 2.3.1, 2.2.1)	CameraManager
Capture snapshot (UC 2.2.2)	CameraManager
Record pre-roll footage (UC 2.2.1)	CameraManager
Send event (motion, bark, offline, error) (UC 1.1.3, 1.1.4, 2.1.1)	CameraManager
Start bark detection (UC 1.1.4)	CameraManager
Start internal timer (UC 2.3.2)	CameraManager
Check duration threshold for operations (UC 2.3.2)	CameraManager

f. Around ModeManager [State Diagram](#)

Class: ModeManager	
Controls the SafeHome system's global operation mode (Home, Away, Night, etc.) and applies configuration changes linked to these modes	
Responsibilities	Collaborators
Display the current mode (UC 1.3.1, 3.2.1)	
Set a new SafeHome mode (UC 1.3.1)	SafetyZoneManager, SensorManager, ConfigManager, SafetyZone
Apply configuration for the selected mode (UC 1.2.1, 1.3.1, 1.3.3)	SafetyZoneManager, SensorManager, ConfigManager, StorageRepository
Identify SafetyZone's armed state (UC 1.3.1, 1.3.2, 1.3.3)	SafetyZone
Retrieve/modify SafetyZone name (UC 1.2.1)	SafetyZone
Retrieve/define SafetyZone ID (UC 1.2.1)	SafetyZone
Retrieve sensor list of SafetyZone (UC 1.2.1, 1.3.2, 1.3.3)	SafetyZoneManager
Update/remove sensors (UC 3.1.1, 1.3.2, 1.3.3)	SafetyZoneManager
Store entire SafetyZone configuration (UC 1.2.1,	SystemHub

3.1.1)	
--------	--

g. Around StorageRepository

Class: StorageRepository	
The place to store and manage logs and data	
Responsibilities	Collaborators
Stores a log (UC 2.2.1, 3.2.2)	LogManager
Exports a log (UC 2.2.2)	LogManager
Displays the expanded view of a media file (UC 2.2.1)	LogManager, CameraManager, Camera
Handles when local storage is full (UC 2.3.1)	LogManager, ConfigManager

Class: CloudServer VideoStreamManagement	
Manage the video streams on the Cloud Server	
Responsibilities	Collaborators
Display all search results on Cloud Server (UC 2.2.1)	CloudServer
Shows video live stream on a camera (UC 2.1.1)	CameraManager, Camera, CloudServer
Adjust stream quality (UC 2.1.1, 2.3.1)	CameraManager, Camera, CloudServer
Saves Audio stream (UC 2.2.1, 2.3.1)	StorageRepository, CloudServer
Download video stream (UC 2.2.2)	CloudServer
Show all available export options (UC 2.2.2)	CloudServer
Generate a secure link for video stream (UC 2.2.2)	CloudServer

h. Around ConfigManager

Class: ConfigManager	
Stores and manages the configuration data for Safety Zones, system modes, and system-wide settings.	
Responsibilities	Collaborators
Retrieve device & zone configuration (UC 3.1.1)	SystemHub, SafetyZoneManager
Store updated configuration (UC 3.1.1)	SystemHub
Validate device configuration changes (UC 3.1.1)	SmartDevices, SystemHub
Manage SafetyZone metadata (UC 1.2.1)	SafetyZoneManager
Maintain system configuration cache	SystemHub

Class: SmartDevices	
Represents abstractions of all smart devices in the home, allowing unified access to device capabilities for the system.	
Responsibilities	Collaborators
Store device event information (UC 3.2.2)	Device
Provide event logs to system (UC 3.2.2)	StorageRepository
Execute automation commands (UC 1.3.1, 5.1.1)	SystemHub, MobileApp_AutomationAndPanicController
Manage on/off and mode states (UC 5.1.1)	SystemHub
Report device sensor status (UC 3.2.1, 5.2.1, 5.2.2)	SystemHub

Class: Device	
A base class representing a single hardware device with standardized behaviors and attributes.	
Responsibilities	Collaborators
Save individual event logs (UC 3.2.2)	StorageRepository
Retrieve stored event logs (UC 3.2.2)	StorageRepository
Update device health status (UC 3.2.1)	SystemHub
Provide device metadata (UC 3.1.1)	ConfigurationManager

Class: LightDevice	
A concrete device class responsible for controlling lighting, storing light-related parameters (brightness, color, on/off state).	
Responsibilities	Collaborators
Turn light on/off (UC 5.1.1)	SystemHub, MobileApp
Adjust brightness (UC 5.1.1)	MobileApp
Update & report color values (UC 5.1.1)	MobileApp
Report device state (UC 3.2.1, 5.1.1)	DeviceManager, SystemHub
Apply automation commands (UC 5.1.1, 1.3.1)	SmartDevices

Class: VentilationSystem	
Represents the ventilation and indoor air-flow system with login-protected configuration rules.	
Responsibilities	Collaborators
Manage ventilation on/off behavior (UC 5.2.1)	LoginManager

Manage user login/out for access	LoginManager
Control password change process	LoginManager, StorageRepository
Report ventilation state (UC 5.2.1)	SystemHub

i. Around LogManager

Class: LogManager	
Centralized manager for system activity logs, error logs, and audit logs.	
Responsibilities	Collaborators
Store and retrieve system logs (UC 3.2.2)	ActivityClass
Provide log filtering to modules (UC 3.2.2)	SystemHub
Maintain chronological event history (UC 3.2.2)	SystemHub
Export logs to cloud backup (UC 3.2.2)	CloudServer

Class: ActivityClass	
Represents a single activity or log entry (timestamp, type, details).	
Responsibilities	Collaborators
Record timestamp of event (UC 3.2.2)	LogManager
Store event type & details (UC 3.2.2)	LogManager
Provide event detail to system modules (UC 3.2.2)	SystemHub

2. CRC Cards - MobileApp [Architecture Class Diagram State Diagram](#)

a. Around MobileApp

Class: MobileApp_RecordingSettingsController	
Manage the Recording Settings options on the Mobile App	
Responsibilities	Collaborators
Navigates to Recording Settings (UC 2.3.1)	CameraManager, MobileApp
Selects a Recording method (UC 2.3.1)	CameraManager, MobileApp
Selects a Storage destination (UC 2.3.1)	StorageRepository, MobileApp
Configures Recording Settings Options (UC 2.3.1)	CameraManager, MobileApp
Saves all changes to Recording Settings (UC 2.3.1)	CameraManager, MobileApp

Class: MobileApp_PlaybackController	
Manage the Playback menu on the Mobile App	
Responsibilities	Collaborators
Navigates to the Playback menu (UC 2.2.1)	CameraManager, MobileApp
Searches for video clip (UC 2.2.1)	CameraManager, StorageRepository, MobileApp
Selects a video clip (UC 2.2.1)	CameraManager, StorageRepository, MobileApp
Uses playback controls (UC 2.2.1)	CameraManager, MobileApp
Choose an export option (UC 2.2.2)	CameraManager, MobileApp

Class: MobileApp_TwoWayAudioController	
Manage the Two way Audio settings on the Mobile App	
Responsibilities	Collaborators
Initialize two way audio (UC 2.1.2)	
Retrieves audio recording (UC 2.1.2)	CameraManager
Saves audio recording (UC 2.1.2)	StorageRepository
End two way audio (UC 2.1.2)	
Handle audio issues (UC 2.1.2)	SecurityManager

Class: MobileApp CameraLiveViewController	
Manage the Camera Live views menu on the Mobile App	
Responsibilities	Collaborators
Selects a camera (UC 2.1.1)	CameraManager, Camera, MobileApp
Asks for the camera password if exists (UC 2.1.3)	CameraManager, Camera, MobileApp
Enters camera password if exists (UC 2.1.3)	CameraManager, Camera, MobileApp, Security Manager
Shows live camera recording (UC 2.1.1)	Camera, MobileApp
Adjust recording's qualities (UC 2.1.1, 2.3.1)	Camera, MobileApp
Toggle selected camera (UC 2.1.4)	CameraManager, Camera, MobileApp
Handle offline cameras (UC 2.1.1)	CameraManager, Camera
Display the alarm status of the camera (UC 1.1.1–1.1.4, 2.1.1)	CameraManager, Camera, AlarmManager
Opens the event details of the alarm status (UC 1.1.1–1.1.4, 1.2.2, 3.2.2)	CameraManager, Camera, AlarmManager, StorageRepository
Opens the live recording of the alarm status (UC 1.1.1–1.1.4, 2.2.1)	CameraManager, Camera, AlarmManager, StorageRepository

Class: MobileApp CameraSecurityController	
Manage the Camera Security settings on the Mobile App	
Responsibilities	Collaborators
Enable a password lock to a chosen camera (UC 2.1.3)	CameraManager, Camera, MobileApp, SecurityManager
Sets a camera password (UC 2.1.3)	CameraManager, Camera, MobileApp, SecurityManager
Saves all changes to the password menu (UC 2.1.3)	CameraManager, Camera, MobileApp, SecurityManager
Disable a password lock to a chosen camera (UC 2.1.3)	CameraManager, Camera, MobileApp, SecurityManager
Confirms the password removal from camera (UC 2.1.3)	MobileApp, SecurityManager

Class: MobileApp	
Acts as the primary user interface for SafeHome. Allows users to access live view, playback, device settings, automation, panic actions, and account-related operations.	
Responsibilities	Collaborators
Display login, account, and security UI (UC	User, CloudServer

4.1.1-4.1.7)	
Send login/logout requests (UC 4.1.2,4.1.3)	CloudServer
Send password change and profile update requests (UC 4.1.4-4.1.6)	CloudServer
Manage and update device configuration UI (UC 3.1.1)	MobileApp_DeviceConfigController
Handle automation and panic UI actions (UC 1.2.4, 1.3.1)	MobileApp_AutomationAndPanicController
Display warnings, errors, and notifications	CloudServer
Sync with cloud for account/session updates (UC 4.1.2,4.1.3)	CloudServer
Trigger actions (arming, bypass, disable) (UC 1.3.1-1.3.3)	CloudServer, SystemHub

Class: MobileApp_DeviceConfigController	
Handles device configuration flows in the mobile app, including navigation to device settings, editing parameters, and saving configuration to the system.	
Responsibilities	Collaborators
Display device configuration parameters (UC 3.1.1, 2.3.1, 2.3.2)	CloudServer_ConfigProfileManagement
Validate configuration inputs (UC 3.1.1, 2.3.1, 2.3.2)	MobileApp, CloudServer_ConfigProfileManagement
Send updated configuration to cloud (UC 3.1.1, 2.3.1, 2.3.2)	CloudServer_ConfigProfileManagement
Show confirmation or error messages (UC 3.1.1, 2.3.1, 2.3.2)	MobileApp
Request device parameter list (UC 3.1.1)	CloudServer_ConfigProfileManagement

Class: MobileApp_AutomationAndPanicController	
Controls the automation routines, panic actions, quick modes, and emergency functions accessible from the mobile app.	
Responsibilities	Collaborators
Trigger panic alert actions(UC 1.2.4)	CloudServer_NotificationDispatchManagement, SystemHub
Manage automation mode selection (UC 1.3.1)	SmartDevices, SystemHub
Control automation routines from UI (UC 1.3.1, 5.1.1)	SmartDevices
Display warnings/errors for automation	MobileApp
Forward automation commands to system (UC	CloudServer_NotificationDispatchManagement

1.3.1, 5.1.1)	ment, SystemHub
---------------	-----------------

3. CRC Cards - CloudServer [Architecture Class Diagram State Diagram](#)

a. Around UserManager [State Diagram](#)

Class: UserManager	
Manages user profile, user privileges, and provides user-related data to other modules.	
Responsibilities	Collaborators
Manage user account profiles (UC 4.1.1–4.1.7)	User
Validate user permissions for system actions (UC 4.1.2, 4.1.3)	LoginManager, SystemHub
Retrieve user information when needed (UC 4.1.2)	CloudServer_AuthAccountManager
Update user account attributes (UC 4.1.4, 4.1.5)	CloudServer_AuthAccountManager
Provide user role information to subsystems (UC 4.1.2)	SystemHub

Class: User	
Represents an individual SafeHome user, storing credentials, contact info, and access rights.	
Responsibilities	Collaborators
Store user credentials (UC 4.1.2)	LoginManager
Store user contact information (email, phone) (UC 4.1.4, 4.1.7)	CloudServer_AuthAccountManager
Store account status (active, locked, etc.)	LoginManager
Provide identity for authentication (UC 4.1.2)	CloudServer_AuthAccountManager
Provide contact endpoints for 2FA (UC 4.1.7)	SMSGatewayService, EmailService

Class: LoginManager	
Handles user login, logout, authentication validation, and session state.	
Responsibilities	Collaborators
Authenticate user credentials (UC 4.1.2)	User, CloudServer_AuthAccountManager
Manage login/logout flow (UC 4.1.2, 4.1.3)	MobileApp, CloudServer
Handle incorrect password attempts (UC 4.1.2)	User
Manage session timeout (UC 4.1.3)	CloudServer
Validate password change requests (UC 4.1.4)	User, CloudServer_AuthAccountManager

b. Around CloudServer

Class: CloudServer AuthAccountManager	
Handles account authentication and authorization at the cloud level.	
Responsibilities	Collaborators
Validate login requests from LoginManager (UC 4.1.2)	LoginManager, User
Process logout from sessions (UC 4.1.3)	LoginManager, MobileApp
Manage password reset / change (UC 4.1.4, 4.1.7)	User, EmailService, SMSGatewayService
Update user profile information (UC 4.1.5)	User, MobileApp
Provide user identity mapping to other cloud modules (UC 4.1.2)	CloudServer

Class: CloudServer LoginAuditManager	
Logs all login attempts and security-related account events.	
Responsibilities	Collaborators
Log successful/failed login attempts (UC 4.1.2)	LoginManager
Maintain audit trail for account activity (UC 4.1.2)	CloudServer
Flag suspicious login behaviour (UC 4.1.2)	CloudServer_AuthAccountManager
Generate security alerts for risky events (UC 4.1.2)	CloudServer_NotificationDispatchManagement

Class: CloudServer 2FASessionManager	
Manages two-factor authentication sessions and verification codes.	
Responsibilities	Collaborators
Generate verification codes (UC 4.1.7)	SMSGatewayService, EmailService
Validate submitted 2FA codes (UC 4.1.7)	User, LoginManager
Track active 2FA sessions (UC 4.1.7)	CloudServer
Re-send verification codes when requested (UC 4.1.7)	SMSGatewayService, EmailService

Class: SMSGatewayService	
Handles delivery of SMS messages, especially for authentication.	
Responsibilities	Collaborators
Deliver 2FA verification codes (UC 4.1.7)	CloudServer_2FASessionManager
Notify user of important account events (UC 4.1.7)	CloudServer_NotificationDispatchManagement
Confirm delivery status (UC 4.1.7)	CloudServer

Class: EmailService	
Handles delivery of email-based notifications and verification messages.	
Responsibilities	Collaborators
Send password reset emails (UC 4.1.4)	CloudServer_AuthAccountManager
Send 2FA verification emails (UC 4.1.7)	CloudServer_2FASessionManager
Notify user of account changes (UC 4.1.7)	CloudServer

Class: CloudServer	
The central backend system that processes requests from MobileApp, manages authentication, configurations, notifications, and integrates with SystemHub.	
Responsibilities	Collaborators
Authenticate users (login, logout) (UC 4.1.1, 4.1.2, 4.1.3)	User, MobileApp
Manage password reset, verification codes (UC 4.1.4, 4.1.7)	User, EmailService, SMSGatewayService
Manage session tokens (UC 4.1.2, 4.1.3)	MobileApp
Synchronize device/account data (UC 3.1.1, 3.2.1, 3.2.2)	SystemHub, MobileApp
Provide configuration data to app (UC 3.1.1, 5.1.1, 5.2.1, 5.2.2)	CloudServer_ConfigProfileManagement
Dispatch notifications (UC 1.1.1, 1.1.2, 1.2.3, 1.2.4)	CloudServer_NotificationDispatchManagement
Validate 2FA and Identity (UC 4.1.7)	User, SMSGatewayService

Class: CloudServer_ConfigProfileManagement	
Processes device configurations, validates parameters, saves device profiles, and forwards updates to the SystemHub.	
Responsibilities	Collaborators
Validate incoming configuration updates (UC	MobileApp_DeviceConfigController

3.1.1, 2.3.1, 2.3.2)	
Store validated configuration (UC 3.1.1)	SystemHub
Push configuration to SystemHub (UC 3.1.1, 2.3.1, 2.3.2)	SystemHub
Return device parameters to MobileApp (UC 3.1.1)	MobileApp_DeviceConfigController
Handle version control and conflicts	SystemHub

Class: CloudServer NotificationDispatchManagement

Handles push notifications, SMS, emergency messages, IVR flows, and escalation sequences for urgent events.

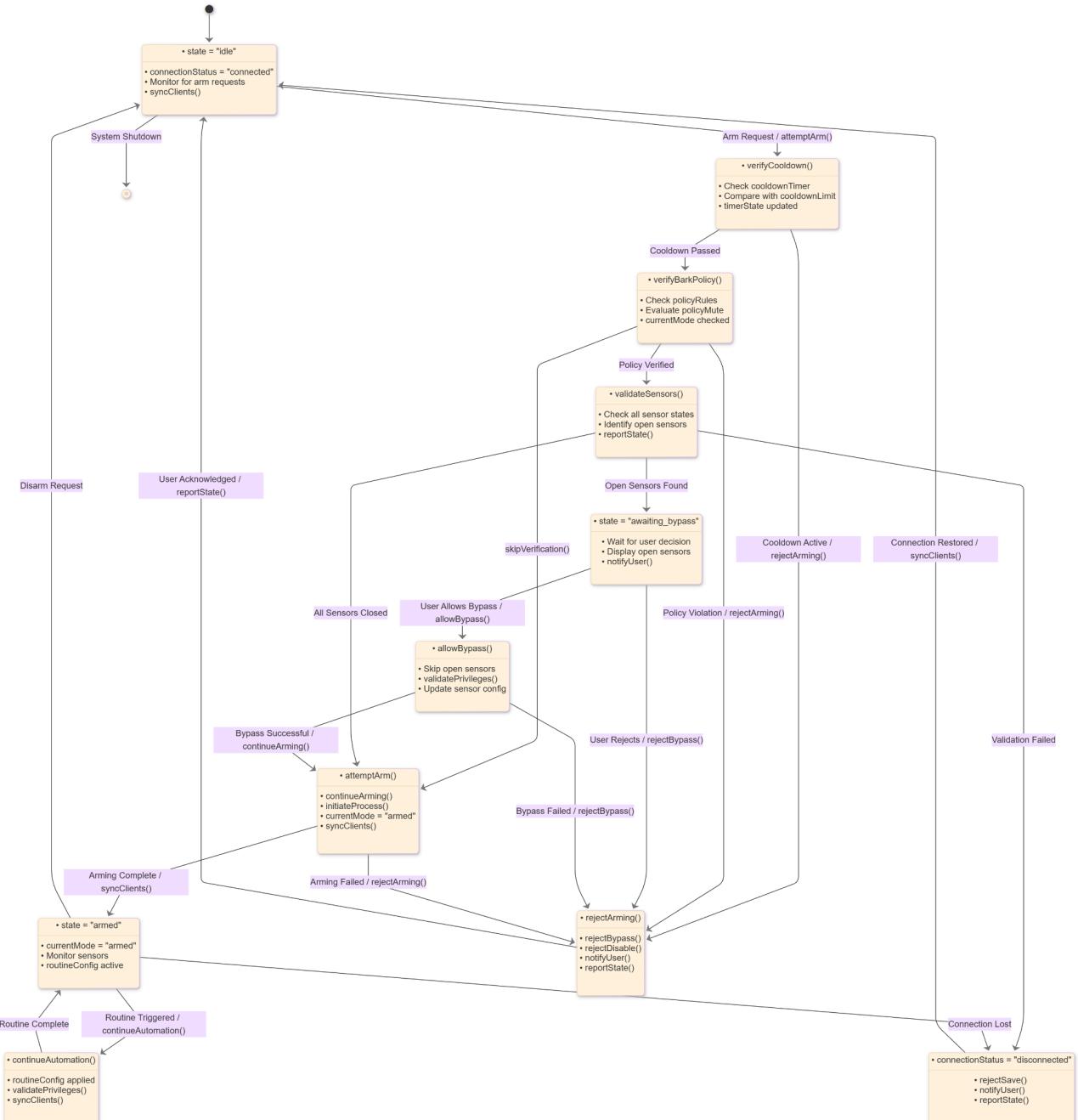
Responsibilities	Collaborators
Send push notifications to users (UC 1.1.1, 1.1.2, 1.2.3, 1.2.4)	
Initiate dispatch workflows (UC 1.2.3)	ExternalEmergencyService, SystemHub
Generate secure notification payloads (UC 1.2.3)	SystemHub
Log dispatch failures or retries (UC 1.2.3)	SystemHub
Manage emergency escalation (UC 1.2.3, 1.2.4)	ExternalEmergencyService

V. State Diagram

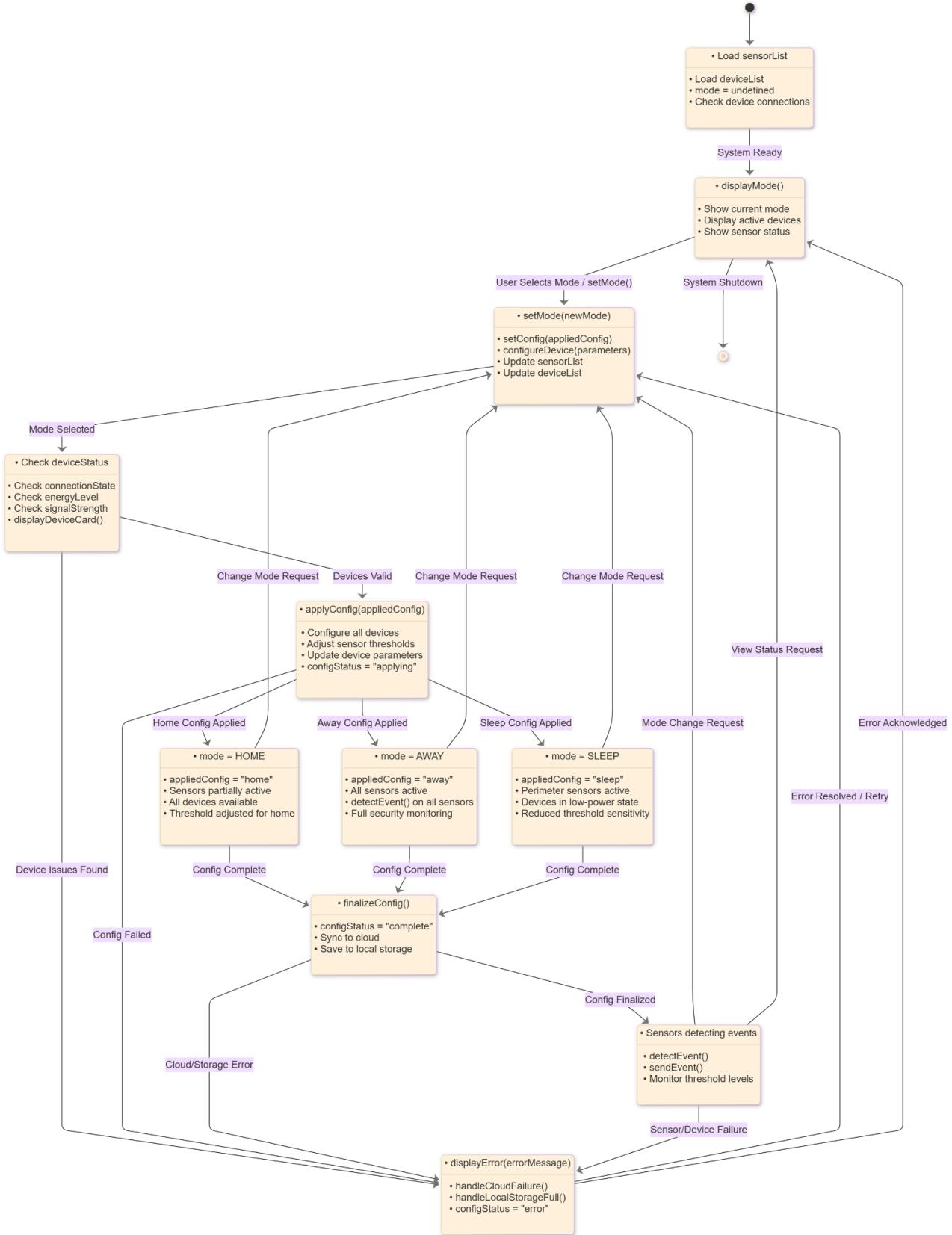
1. Classes around SystemHUB [Architecture](#) [Class Diagram](#) [CRC](#)

Card

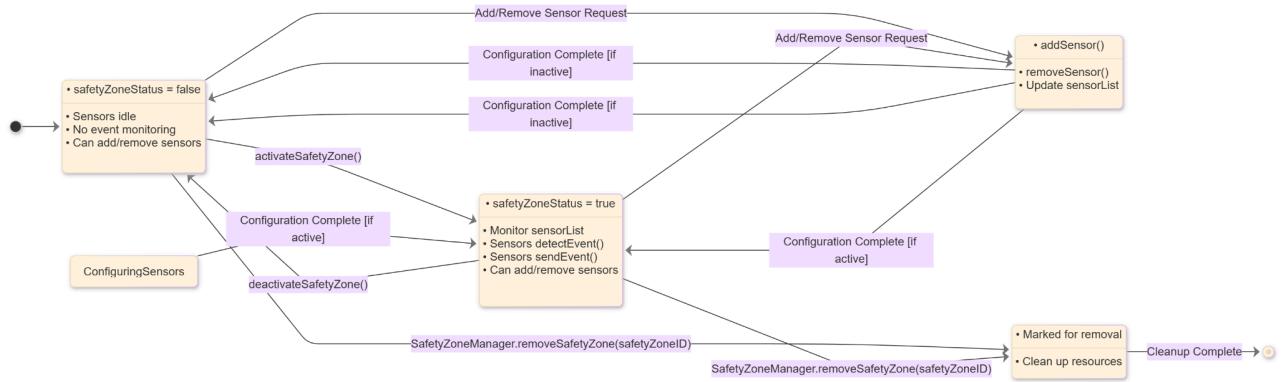
a. State Diagram for SecurityManager [Class Diagram](#) [CRC Card](#)



b. State Diagram for ModeManager [CRC Card](#)

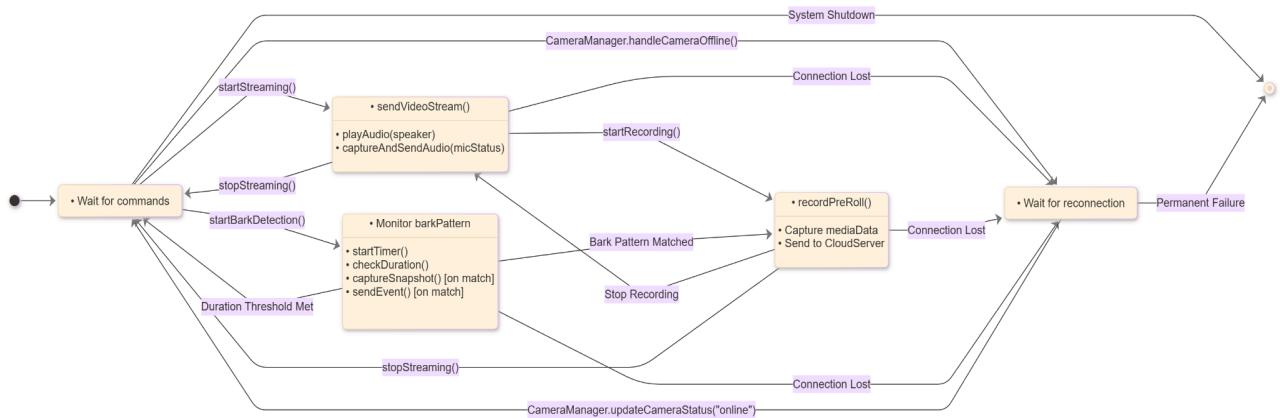


c. State Diagram for SafetyZoneManager [CRC Card](#)

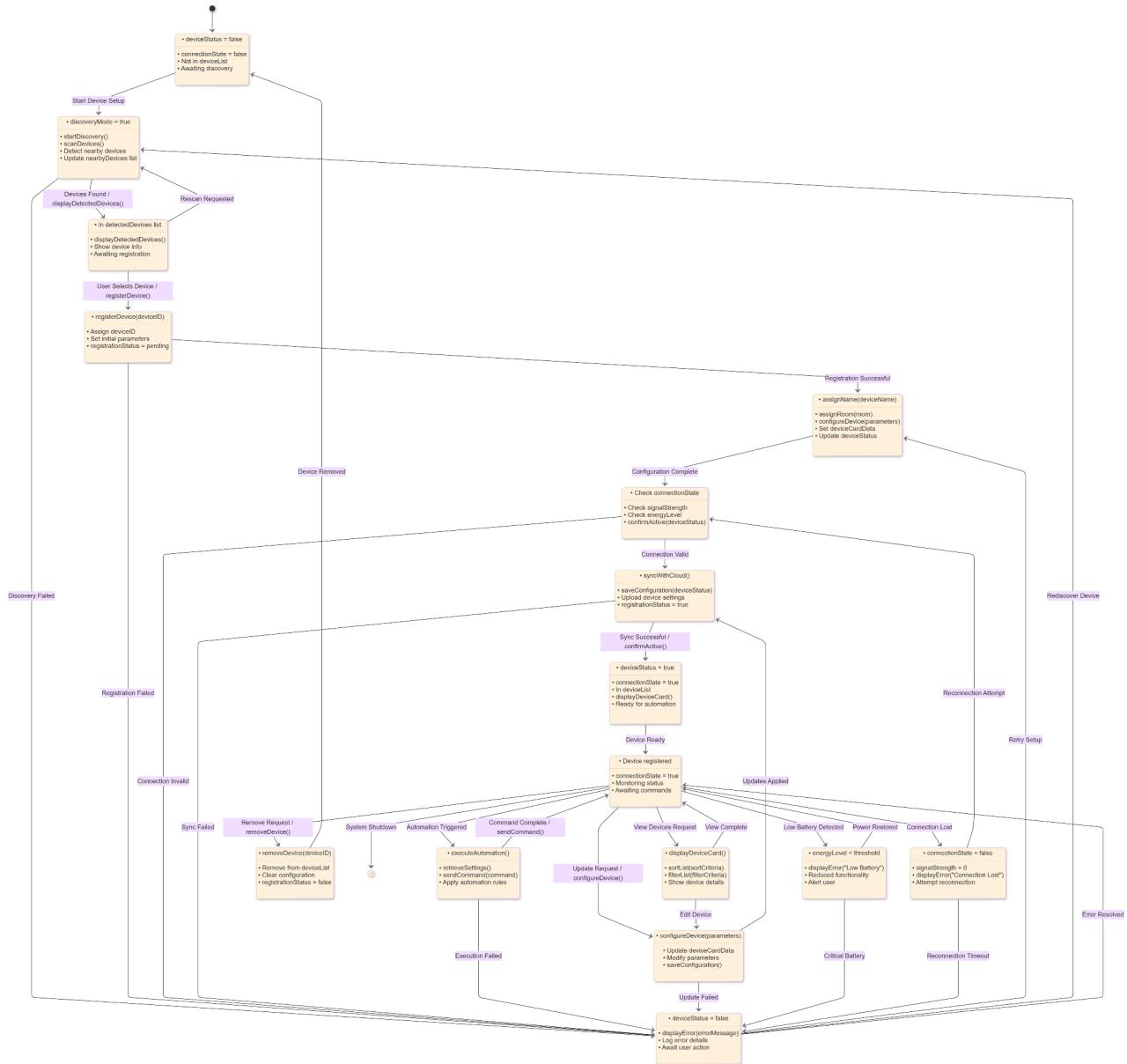


d. State Diagram for CameraManager [Class Diagram](#) [CRC](#)

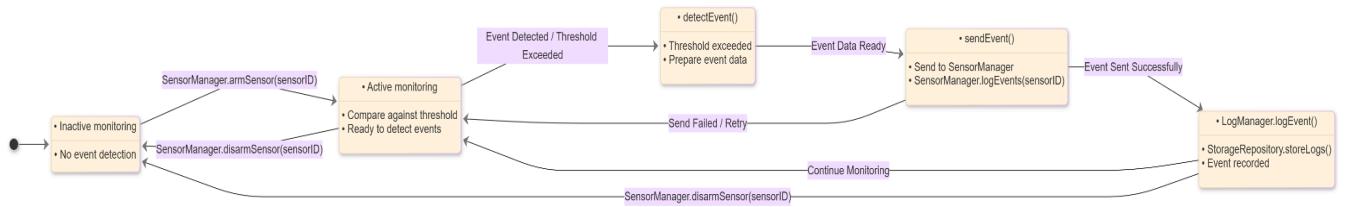
Card



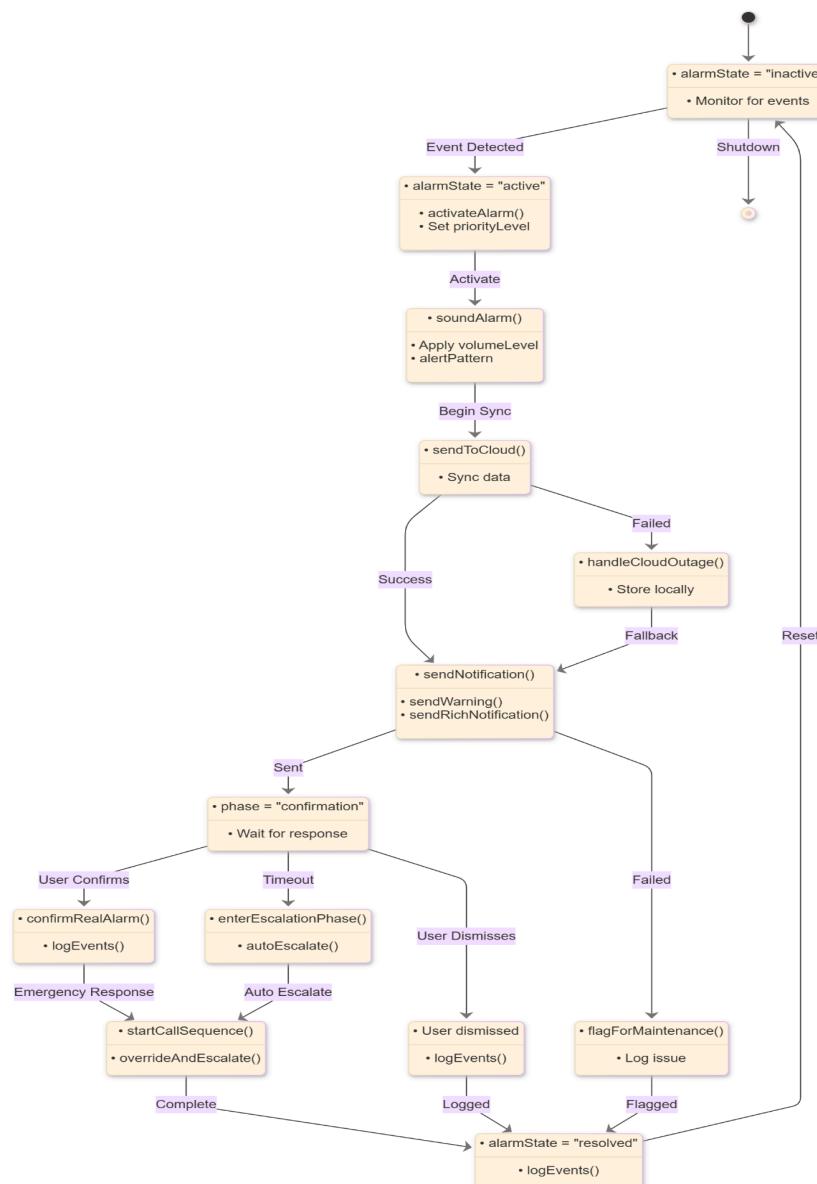
e. State Diagram for DeviceManager Class Diagram



f. State Diagram for SensorManager [CRC Card](#)

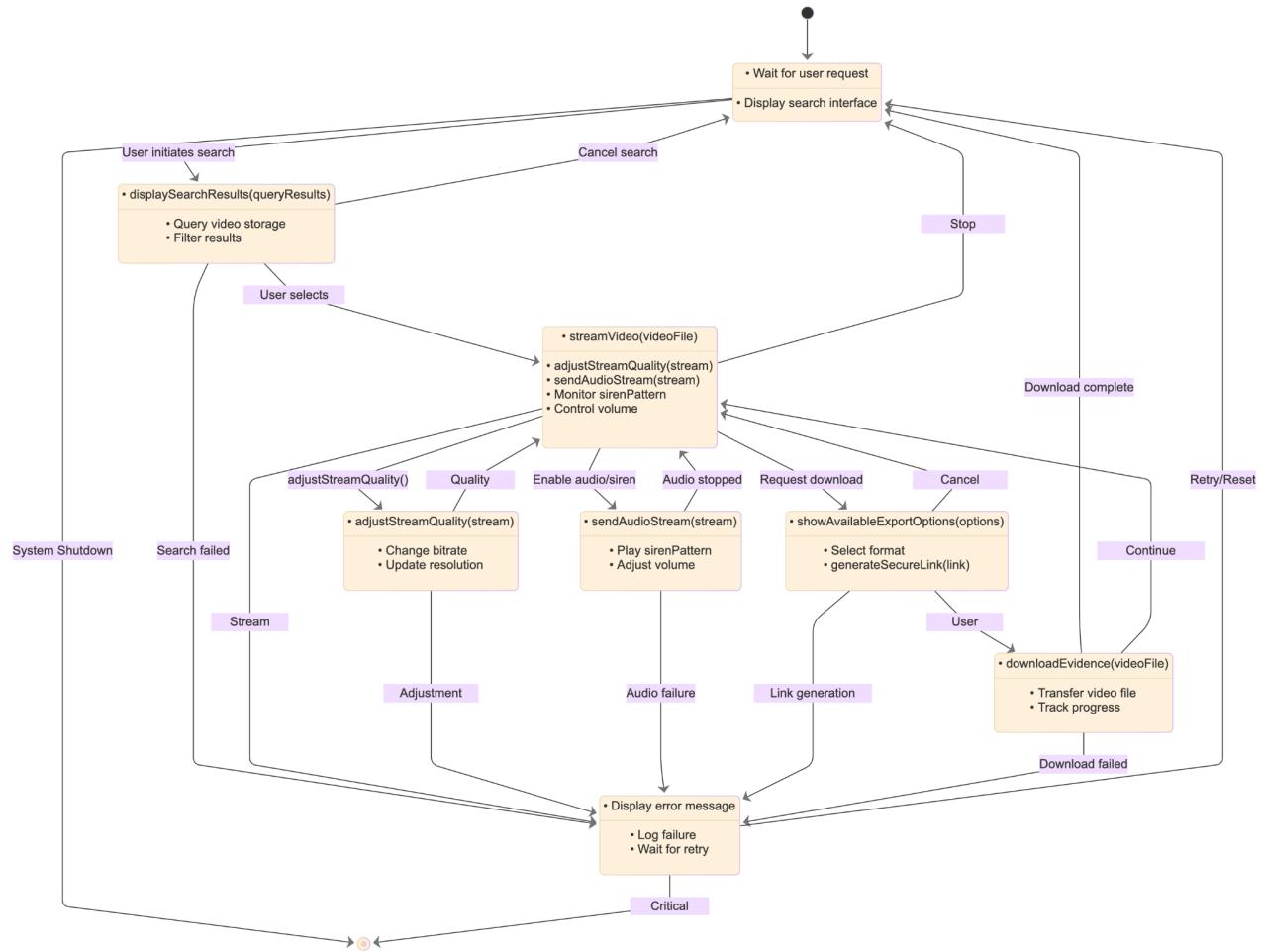


g. State Diagram for AlarmManager [CRC Card](#)

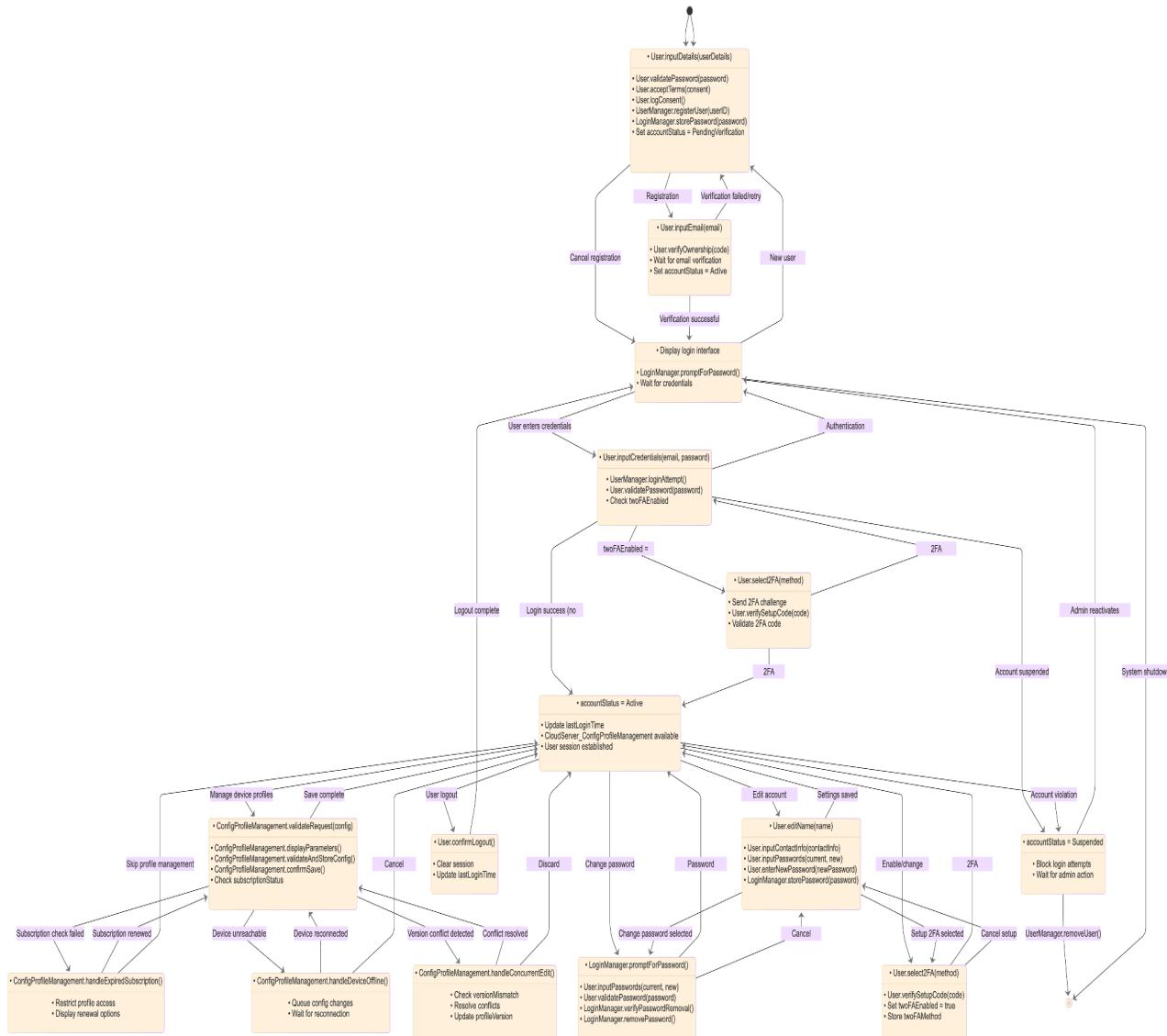


2. Classes around CloudServer [Architecture](#) [Class Diagram](#) [CRC Card](#)

a. State Diagram for Video Streaming

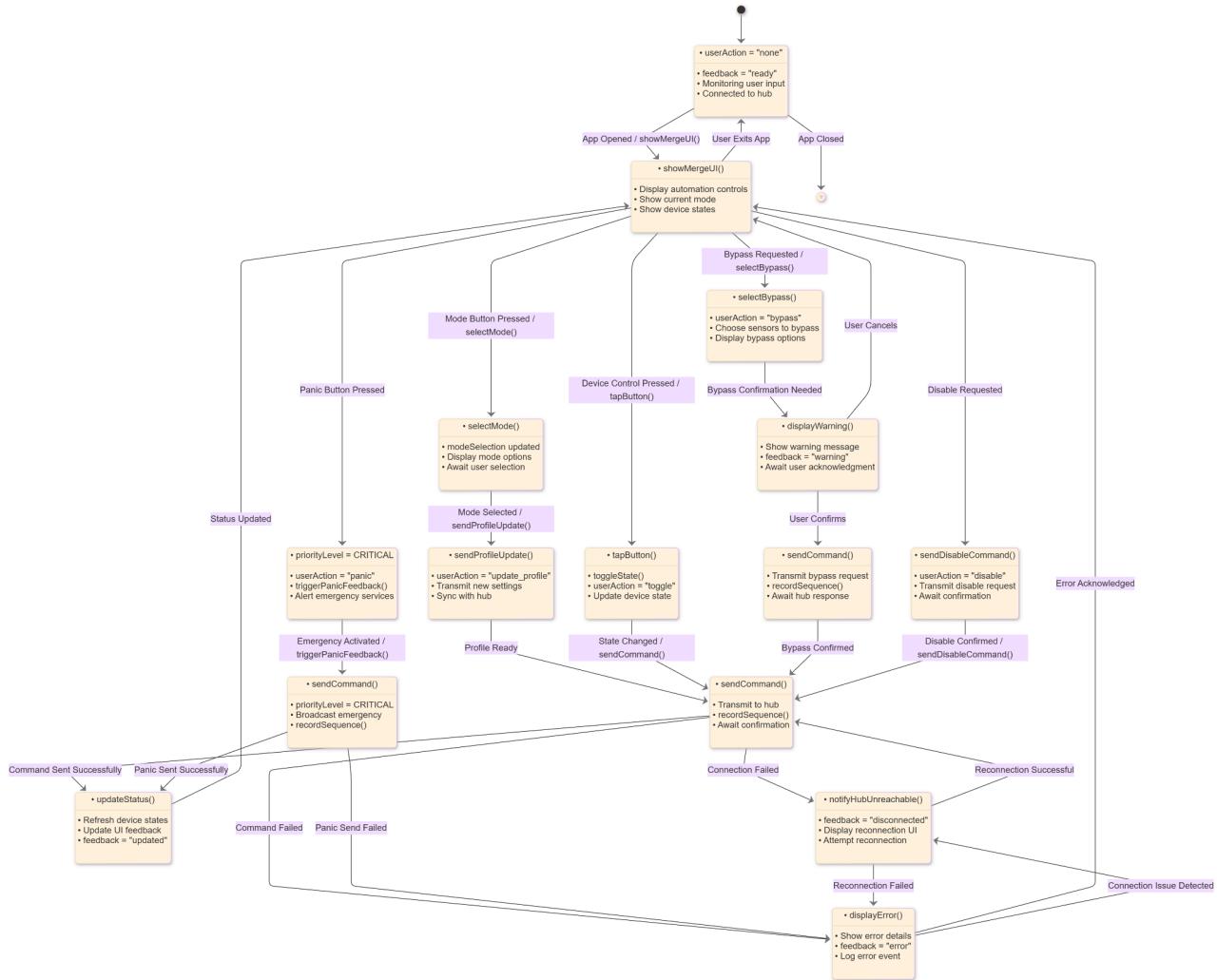


b. State Diagram for User login [CRC Card](#)



3. Classes around MobileApp [Class Diagram](#) [CRC Card Architecture](#)

a. State Diagram of MobileApp_AutomationAndPanicController



VI. Design evaluation

1. Architectural Design Metric

a. Design Structure Quality Index (DSQI)

- $S_1 = 45$

- $S_2 = 29$

- $S_3 = 32$

- $S_4 = 40$

- $S_5 = 25$

- $S_6 = 6$

- $S_7 = 42$

▪ Program Structure:	$D_1 = 1$	
▪ Module independence:	$D_2 = 1 - 29 / 45 =$	$16 / 45$
▪ Modules not depend on prior processing:	$D_3 = 1 - 32 / 45 =$	$13 / 45$
▪ Database size:	$D_4 = 1 - 25 / 40 =$	$15 / 40$
▪ Database compartmentalization:	$D_5 = 1 - 6 / 40 =$	$34 / 40$
▪ Module entrance/exit characteristic:	$D_6 = 1 - 42 / 45 =$	$3 / 45$
▪ $W_1 = 0.25$		
▪ $W_2 = 0.3$		
▪ $W_3 = 0.25$		
▪ $W_4 = 0.05$		
▪ $W_5 = 0.05$		
▪ $W_6 = 0.1$		
▪ $DSQI = \sum W_i D_i = 0.496$		

b. Fenton's simple morphology metrics

- Node = 45

- Arc = 106
- Size = 151
- Depth = 5
- Width = 9
- arc-to-node ratio = 2.36

2. CK Metrics

Depth of the inheritance tree	5
Maximum Number of Children	9
Average Number of Children	1.31
Maximum Coupling Between Object classes	9
Average Coupling Between Object classes	2.36

3. MOOD Metric

a. MIF (Method Inheritance Factor)

		$M_d(C_i)$	$M_i(C_i)$	$M_a(C)$
1	ActivityLog	2	0	2
2	AlarmManager	15	0	15
3	Camera	12	0	12
4	CameraManager	20	12	32
5	CloudServer	109	107	216
6	CloudServer_2FASessionManager	26	8	34
7	CloudServer_AuthAccountManager	9	0	9
8	CloudServer_ConfigProfileManagement	28	21	49
9	CloudServer_LoginAuditManager	5	0	5
10	CloudServer_NotificationDispatchManagement	32	0	32
11	CloudServer_VideoStreamManagement	7	0	7
12	ConfigManager	4	0	4
13	COSensor	1	0	1
14	Device	15	8	23
15	DeviceManager	28	20	48
16	EmailService	4	0	4
17	ExternalEmergencyService	4	2	6
18	IVRService	2	0	2
19	LightDevice	4	0	4
20	LoginManager	4	0	4
21	LogManager	15	6	21
22	ModeManager	24	21	45
23	MobileApp	171	163	334
24	MobileApp_AutomationAndPanicController	14	0	14
25	MobileApp_CameraLiveViewController	11	0	11
26	MobileApp_CameraSecurityController	5	0	5
27	MobileApp_DeviceConfigController	9	0	9
28	MobileApp_PlaybackController	5	0	5
29	MobileApp_RecordingSettingsController	5	0	5
30	MobileApp_TwoWayAudioController	5	0	5
31	MovementSensor	1	0	1
32	SafetyZone	8	6	14
33	SafetyZoneManager	14	8	22
34	SecurityManager	22	4	26
35	Sensor	6	4	10
36	SensorManager	9	6	15
37	ShockSensor	1	0	1
38	SmartDevices	18	15	33
39	SMSGatewayService	4	0	4
40	SoundAnalyzer	1	0	1
41	StorageRepository	6	2	8
42	SystemHub	210	209	419
43	User	14	0	14
44	UserManager	21	18	39
45	VentilationSystem	4	0	4

$$\text{MIF} = 640 / 1574 = \text{Approximately } 0.407$$

b. CF (Coupling Factor)

* Class numbering follows the table in VII. 3. a.

$$CF = 106 / (45 * 45 - 45) = 0.054$$

4. OO Metric Proposed by Lorenz and Kidd

	# of operation	# of attribute
ActivityLog	2	2
AlarmManager	15	7
Camera	12	10
CameraManager	8	2
CloudServer	2	10
CloudServer_2FASessionManager	18	16
CloudServer_AuthAccountManager	9	10
CloudServer_ConfigProfileManagement	7	8
CloudServer_LoginAuditManager	5	19
CloudServer_NotificationDispatchManagement	32	0
CloudServer_VideoStreamManagement	7	7
ConfigManager	4	2
COSensor	1	0
Device	7	9
DeviceManager	8	5
EmailService	4	3
ExternalEmergencyService	2	1
IVRService	2	2
LightDevice	4	3
LoginManager	4	0
LogManager	9	4
ModeManager	3	1
MobileApp	8	10
MobileApp_AutomationAndPanicController	14	4
MobileApp_CameraLiveViewController	11	9
MobileApp_CameraSecurityController	5	4
MobileApp_DeviceConfigController	9	7
MobileApp_PlaybackController	5	4
MobileApp_RecordingSettingsController	5	5
MobileApp_TwoWayAudioController	5	4
MovementSensor	1	0
SafetyZone	2	3
SafetyZoneManager	6	1
SecurityManager	18	9
Sensor	2	3
SensorManager	3	0
ShockSensor	1	0
SmartDevices	3	1
SMSGatewayService	4	3
SoundAnalyzer	1	1
StorageRepository	4	3
SystemHub	8	7
User	14	10
UserManager	3	1
VentilationSystem	4	3

VI. WHO DID WHAT

Team Member 20230943
<ol style="list-style-type: none">1. Extracted Class: extract verb and noun from use case of 1. Intelligent Security2. Do hyperlink and Document format3. Draw Class Diagram4. Draw State Diagram5. Extracting words for glossary
Team member 20230970
<ol style="list-style-type: none">1. Extracted Class: extract verb and noun from use case of 4. Remote Access and Account2. Draw Class Diagram of MobileApp3. Do UML for 4. Remote Access and Account4. Write a CRC card for 3. System and User management and 5. Indoor Monitoring and Device Control and 4. Remote Access and Account5. Reference all class in CRC card6. Do a Evaluational Design Metric
Team member 20230988
<ol style="list-style-type: none">1. Extracted Class: extract verb and noun from use case of 3. System and User management and 5. Indoor Monitoring and Device Control2. Summarize class from all candidate class based on their functionality3. Draw and Derive Class Diagram for Overall System and SystemHub a-c4. Draw State Diagram for Classes around SystemHub a-g and MobileApp5. Add Glossary for document
Team member 20231008
<ol style="list-style-type: none">1. Extracted Class: extract verb and noun from use case of 2. Live Surveillance2. Draw Class Diagram of UserManager and MobileApp3. Do UML for 2. Live Surveillance4. Write a CRC card for 1. Intelligent Security and 2. Live Surveillance

VII. MEETING LOGS

1st Meeting

- Date: November 10th(Mon), 2025
- Duration: 19:00 - 20:00 (60 minutes)
- Location: Online(Google Meet)
- Participants: Tattep Lakmuang, Nitheekulawatn Thommakoon, Thiwat Chatkham, Ngo Thai Binh
- Recorder: Ngo Thai Binh
- Meeting Agenda:
 - Workflow outline and work assignment
 - Review extraction of nouns and verbs from use cases
 - Review summarization of classes from extracted verb and noun
- Main discussion Topics:
 - Workflow outline and work assignment for 1st step:
 - Workflow outline: extract nouns and verbs and fill google sheet, write CRC card, draw Class diagram, draw Architecture diagram, calculate Metric, draw State Diagram
 - Work assignment for extracting nouns and verbs and fill google sheet: Thiwat (Intelligent Security), Binh (Live Surveillance), Thommakoon (System and User Management, Indoor Monitoring and Device Control), Tattep (Remote Access and Account).
 - When god classes and tightly coupled classes exist:
 - Solution: group operation for tightly coupled classes and create subclasses for god classes
 - Members in charge: Thiwat, Thommakoon
 - The original SRS document doesn't contain more detail on safety zone and floor plan
 - Solution: consult TA and add more use cases from the new specification from lecture 11/9/2025 and from GUI in SRS.
 - When classes from add-on use case conflict with existed class
 - Solution: extract classes/create subclasses from add-on use case

with Safety Zone from sensor and predetermined floor plan and sensors

- Members in charge: Thiwat, Thommakoon

2nd Meeting

- Date: November 12th(Wed), 2025
- Duration: 20:00 - 21:30 (90 minutes)
- Location: N11 Study Room 2A
- Participants: Tattep Lakmuang, Nitheekulawatn Thommakoon, Thiwat Chatkham, Ngo Thai Binh
- Recorder: Ngo Thai Binh
- Meeting Agenda:
 - Review and polish overall class definitions
 - Review CRC Card criterias and work assignment
 - Review class diagram criterias and work division
- Main discussion Topics:
 - Work assignment for 2nd step and 3rd step:
 - Work assignment for creating CRC Cards: Tattep, Binh
 - Work assignment for drawing Class Diagrams: Thiwat, Binh, Thommakoon, Tattep (Class creations); Thiwat, Thommakoon (Class diagram layout)
 - When overall structure conflict exists:
 - Solution: draw overall UML class and find missing sub-class/parent-class
 - Members in charge: Thiwat, Binh, Thommakoon, Tattep
 - Missing sensor management in sensor class
 - Solution: create SensorManagement class
 - Members in charge: Thommakoon
 - Mode settings and configuration needs management
 - Solution: create ModeManager class
 - Members in charge: Thiwat

3rd Meeting

- Date: November 13th(Thu), 2025
- Duration: 19:00 - 21:00 (90 minutes)
- Location: N11 Study Room 1A
- Participants: Tattep Lakmuang, Nitheekulawatn Thommakoon, Thiwat Chatkham, Ngo Thai Binh
- Recorder: Ngo Thai Binh
- Meeting Agenda:
 - SRS structure review and redesigned
 - Review architectural diagram criterias and work division
 - Review design evaluation criteria and work division
 - Review state diagram criterias and work division
- Main discussion Topics:
 - Work assignment for the last 3 steps:
 - Work assignment for calculating metrics: Tattep
 - Work assignment for drawing Architectural Diagrams: Tattep, Thommakoon
 - Work assignment for drawing State Diagrams: Tattep, Thommakoon
 - Work assignment for writing meeting logs: Binh
 - Work assignment for writing glossaries: Thommakoon
 - SRS structure is no longer suitable for main class definitions:
 - Solution: change from 5 part SRS to 3 main class: SystemHub, CloudServer, and MobileApp and reorder CRC Cards/ Class Diagrams accordingly
 - Members in charge: Thommakoon, Tattep
 - Responsibilities in CRC Cards need references
 - Solution: referencing each responsibility to its respective use cases
 - Members in charge: Tattep

APPENDIX A. GLOSSARY

Term	Definition
2FA	Two-factor authentication process for user verification.
Account Status	Current state of a user account (active, locked, suspended).
AlarmManager	Manages alarms, notifications, escalations, and event logging.
Automation Mode	Predefined system configuration controlling multiple devices.
Automation Routines	Scheduled or triggered sequences of automated actions.
AWAY Mode	Fully armed mode; indoor and perimeter sensors active.
Camera	Device capturing video/audio, supports alarms, passwords, and streaming.
CameraManager	Handles camera control, streaming, recording, and alarm events.
CloudServer	Backend managing authentication, data sync, configurations, and notifications.
CloudServer_2FASessionManager	Manages 2FA sessions and verification codes.
CloudServer_AuthAccountManager	Handles login, logout, authentication, and password management.
CloudServer_ConfigProfileManagement	Manages device configurations, version control, and updates to SystemHub.
CloudServer_NotificationDispatchManager	Sends notifications, SMS, emergency alerts, and panic/automation commands.
ConfigManager	Maintains system/device/SafetyZone settings and automation rules.

COSensor	Detects carbon monoxide levels.
Device	Base class for hardware devices handling events and status.
DeviceManager	Manages smart device registration, discovery, and control.
HOME Mode	Partially armed mode; indoor sensors limited, perimeter active.
IVRService	Automated voice system for emergency notifications and calls.
LightDevice	Controls lighting, brightness, color, and automation commands.
LogManager	Manages system and audit logs; stores, filters, and exports events.
MobileApp	User interface for control, playback, notifications, and automation.
ModeManager	Controls global system mode (HOME, AWAY, SLEEP) and applies configurations.
MovementSensor	Detects motion events.
Panic Alert	User-triggered emergency notification.
SLEEP Mode	Night mode; perimeter active, indoor sensors mostly disabled.
SecurityManager	Handles intrusion detection, arming/disarming, and security rules.
SensorManager	Monitors, arms, and validates sensor events.
ShockSensor	Detects vibrations or impacts.
SmartDevices	Connected devices controllable via automation and modes.
SoundAnalyzer	Detects specific sound patterns (glass break, barking,

	abnormal noise).
SystemHub	Central coordinator integrating modules, processing commands, and syncing data.
User	Account holder storing credentials, contact info, and access rights.
UserManager	Manages user profiles, permissions, and roles.
Video Clip	Recorded segment of camera footage.
VentilationSystem	Controls indoor airflow and reports system state.
StorageRepository	Stores logs, media files, and configuration data.
Live View	Real-time camera video feed.
Two-Way Audio	Bidirectional audio communication through cameras.
Quick Modes	Preset automation configurations for fast system control.
Recording Quality	Video resolution and clarity settings.
Password Lock	Restricts camera access with authentication.