**A black text on a white background

Description automatically generated**

**Zahid Iqbal 22P-9077**

**Zahid Aslam 22F-3394**

**Mubashir Rao 22P-9063**

**Arsal Hassan 22F-3384**

**Table of Contents**

1. **Introduction**
2. **Vision Document**

2.1. Problem Statement

2.2. Business Opportunity

2.3. Objectives

2.4. Scope

2.4.1. **System Features**

2.4.2. **User Capabilities**

2.4.3. **Integration**

2.4.4. **Data Management**

2.4.5. **Market Target**

2.4.6. **Geographical Reach**

2.4.7. **Regulatory Compliance**

2.4.8. **Support and Maintenance**

2.4.9. Exclusions

2.5. Constraints

#### 2.5.1. **Technical Constraints**

#### 2.5.2. **Budget Constraints**

2.5.3. **Regulatory Constraints**

#### 2.5.4. **User Adoption Constraints**

#### 2.5.5. **Market Constraints**

#### 2.5.6. **Environmental Constraints**

2.5.7. **Development Timeline**

#### 2.5.8. **Maintenance and Support Constraints**

2.6. Stakeholder and User Descriptions

2.6.1. Market Demographics

2.6.2. Stakeholder Summary

2.6.3. User Environment

2.6.4. Stakeholder Profiles

2.6.4.1. Development Team

2.6.4.2. End Users

2.6.4.3. Government Officials

2.6.4.4. Admin

**3. System Requirements Specification**

3.1. System Features

#### 3.1.1. **Adaptive Automation**

#### 3.1.2. **Energy Management**

#### 3.1.3. **Environmental Monitoring**

#### 3.1.4. **User Interface**

#### 3.1.5. **Security and Surveillance**

#### 3.1.6. **Customizable Alerts and Automation**

#### 3.1.7. **Data Analytics and Reporting**

#### 3.1.8. **Scalability and Integration**

3.2. Functional Requirements

3.2.1. User Setup and Profiles

3.2.1.1. User Registration and Login

3.2.1.2. Multiple User Profiles

3.2.1.3. User Permissions

3.2.2. Home Environment Monitoring

3.2.2.4. Real-time Environmental Monitoring

3.2.2.5. Room Occupancy Detection

3.2.3. Automation and Control

3.2.3.6. HVAC Control Based on Environment

3.2.3.7. Lighting Control

3.2.3.8. Automated Blind Control

3.2.3.9. Appliance Control

3.2.3.10. Energy-Efficient Appliance Scheduling

#### **3.2.4. Security and Safety**

**3.2.4.11. Smart Door Locks**

**3.2.4.12. Camera Monitoring**

**3.2.4.13. Intrusion Detection**

**3.2.4.14. Emergency Alerts**

#### **3.2.5. Energy Monitoring and Management**

**3.3.5.15. Energy Usage Data Collection**

**3.3.5.16. Energy Consumption Alerts**

**3.3.5.17. Energy Usage Forecasting**

**3.3.5.18. Renewable Energy Integration**

#### **3.2.6. User Interaction and Customization**

**3.2.6.19. Voice Control Integration**

**3.2.6.20. Mobile App Control**

**3.2.6.21. Customizable Automation Rules**

**3.2.6.22. Vacation Mode**

#### **3.2.7. External Integration and Data Management**

**3.2.7.23. Weather Forecast Integration**

**3.2.7.24. Third-party Device Integration**

**3.2.7.25. Automatic Updates and Maintenance Alerts**

3.3. Non-Functional Requirements

#### **3.3.1. Scalability and Flexibility**

**3.3.1.1. System Scalability**

**3.3.1.2. Device Compatibility**

**3.3.1.3. Modular Design**

#### **3.3.2. Performance and Reliability**

**3.3.2.4. Response Time**

**3.3.2.5. System Uptime**

**3.3.2.6. Low Latency**

**3.3.2.7. Data Refresh Rate**

**3.3.2.8. Offline Mode**

#### **3.3.3. Energy Efficiency**

**3.3.3.9. Energy Optimization**

**3.3.3.10. Energy Efficiency Standards**

#### **3.3.4. Security**

**3.3.4.11. Data Encryption**

**3.3.4.12. Access Control**

**3.3.4.13. Automatic Locking and Logout**

**3.3.4.14. Backup and Recovery**

#### **3.3.5. User Experience and Interface**

**3.3.5.15. User-Friendly Interface**

**3.3.5.16. Customizable Dashboard**

**3.3.5.17. Multi-Language Support**

#### **3.3.6. Compliance and Standards**

**3.3.6.18. Regulatory Compliance**

**3.3.6.19. Device Certification**

**3.3.6.20. Software Updates**

1. **Use case diagram**
2. **5. Use Case Specifications**
3. 5.1. User Case 1: User Registration and Login
4. 5.2. Use Case 2: Multiple User Profiles
5. 5.3. Use Case 3: Home Environment Monitoring
6. 5.4.Use Case 4: HVAC Control Based on Environment
7. 5.5.Use Case 5: Security Monitoring and Alerts
8. 5.6.Use Case 6: Energy Monitoring and Management
9. 5.7.Use Case 7: Customizable Automation Rules
10. **6. Architecture design**
11. **7. Domain Model**
13. **8. Behavioral Modeling**
14. 8.1. System Sequence Diagram
15. 8.2. Activity Diagram
16. 8.3. State Machines
17. **9. Logical Models**
18. 9.1. Class Diagram
19. 9.2. Sequence Diagram
20. **10. Operation Contracts**

### 10.1. **User Login Operation**

### 10.2. **Activate Vacation Mode**

### 10.3. **HVAC Control Based on Room Occupancy**

### 10.4. **Energy Consumption Alert**

### 10.5. **Add New Device to System**

### 10.6. **Automated Blind Control**

### 10.7. **Generate Energy Usage Report**

### 10.8. **Voice Command Integration**

**1. Introduction**

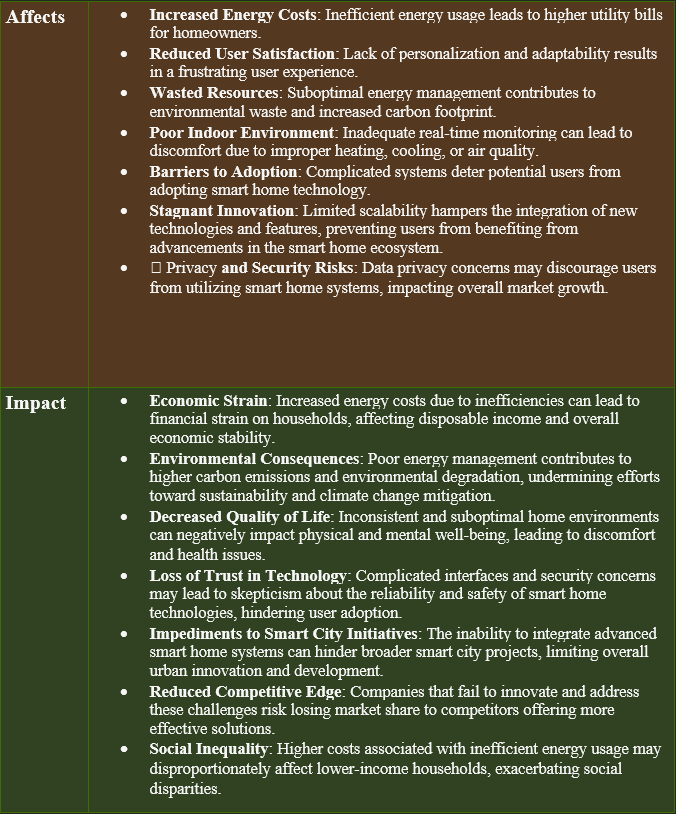
The development of a cutting-edge **Smart Home Automation and Energy Management System**. The system is designed to deliver an intelligent, energy-efficient, and user-friendly solution for automating daily household tasks, optimizing energy consumption, and providing real-time environmental monitoring. It aims to offer a scalable solution that enhances user comfort while promoting sustainability by reducing energy waste.

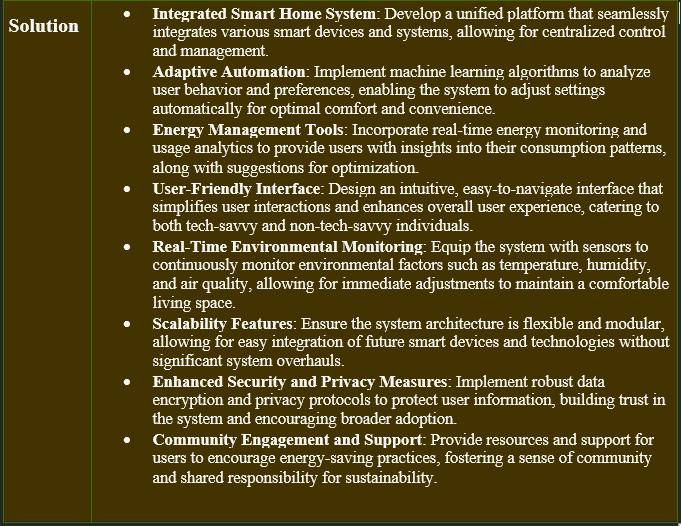
**2. Vision Document**

**2.1. Problem Statement:**

A screenshot of a computer

Description automatically generated





**2.2. Business Opportunities**

* **Growing Market Demand**: The increasing interest in smart home technologies presents a lucrative market opportunity as consumers seek greater convenience, energy efficiency, and enhanced living experiences.
* **Sustainability Trends**: As environmental concerns rise, the demand for energy-efficient solutions will drive growth in the smart home sector, allowing companies to market their products as eco-friendly and sustainable.
* **Partnerships with Utility Companies**: Collaborating with energy providers can create opportunities for demand-response programs, incentivizing users to reduce energy consumption during peak times and enhancing overall grid efficiency.
* **Integration with Smart City Initiatives**: Aligning the Smart Home system with broader smart city projects can open avenues for governmental contracts and partnerships, contributing to urban development and innovation.
* **Data Monetization**: By collecting and analyzing user data (with consent), companies can offer valuable insights to businesses and organizations seeking to understand consumer behavior and energy usage patterns.
* **Diverse Revenue Streams**: The potential to offer subscription-based services, maintenance packages, or premium features can create additional revenue streams beyond the initial hardware sales.
* **Customization and Personalization Services**: Providing tailored solutions that cater to specific user needs can differentiate the system in a competitive market, fostering customer loyalty and enhancing user satisfaction.
* **Expanding IoT Ecosystem**: As the Internet of Things continues to grow, there are ample opportunities for integration with other smart devices and systems, creating a comprehensive ecosystem that enhances user experience and functionality.
* **Government Incentives and Grants**: Opportunities may exist to leverage governmental support for energy-efficient technologies, including grants or tax incentives for both developers and users.
* **Targeting Emerging Markets**: There is significant potential for expansion into developing regions where demand for energy management solutions and smart home technologies is on the rise, creating a new customer base.

**2.3. Objectives**

* **Enhance Home Automation**: Develop a system that automates household tasks, such as lighting, heating, and security, to improve user convenience and comfort.
* **Optimize Energy Usage**: Implement real-time energy monitoring and management tools that help users track their consumption and identify opportunities for energy savings.
* **Provide Adaptive Solutions**: Utilize machine learning algorithms to create an adaptive system that modifies home settings based on user behavior and preferences, enhancing the personalization of the user experience.
* **Ensure Real-Time Environmental Monitoring**: Equip the system with sensors to continuously monitor indoor environmental conditions (e.g., temperature, humidity, air quality) and make automatic adjustments for optimal living conditions.
* **Design a User-Friendly Interface**: Create an intuitive and accessible interface that simplifies interaction with the system for users of all technical skill levels.
* **Support Scalability and Integration**: Ensure that the system is designed to be scalable and compatible with a wide range of existing and future smart home devices and technologies.
* **Enhance Security and Privacy**: Implement strong security measures to protect user data and ensure privacy, fostering trust and confidence in the system.
* **Promote User Engagement**: Encourage user involvement and feedback to continuously improve system functionalities and promote energy-saving practices within the community.
* **Facilitate Data Visualization**: Provide users with clear visual representations of their energy consumption patterns and environmental data to empower informed decision-making.
* **Contribute to Sustainability Goals**: Support users in reducing their carbon footprint and promoting environmentally sustainable practices through efficient energy management and conservation strategies.

**2.4. Scope**

The scope of the Smart Home Automation and Energy Management System encompasses the following key areas:

#### 2.4.1. **System Features**

* **Home Automation**: Automation of lighting, heating, cooling, security systems, and appliances to enhance convenience and comfort.
* **Energy Management**: Tools for real-time energy monitoring, usage analytics, and recommendations for optimizing consumption.
* **Environmental Monitoring**: Integration of sensors to continuously monitor indoor temperature, humidity, air quality, and other relevant environmental factors.
* **User Interface**: Development of a user-friendly mobile app and/or web portal for easy access and control of smart home features.

#### 2.4.2. **User Capabilities**

* **Customization**: Users can personalize settings based on their preferences and routines.
* **Remote Access**: Enable users to monitor and control their smart home system remotely via mobile devices.
* **Alerts and Notifications**: Provide users with alerts for unusual activities, energy usage spikes, or environmental concerns.

#### 2.4.3. **Integration**

* **Device Compatibility**: Ensure compatibility with a wide range of existing smart home devices, including third-party products.
* **IoT Connectivity**: Facilitate communication between devices using IoT protocols and standards for seamless integration.

#### 2.4.4. **Data Management**

* **Data Collection**: Capture and store data related to energy usage, environmental conditions, and user interactions for analysis and reporting.
* **Analytics**: Utilize data analytics to provide insights and recommendations for improving energy efficiency and comfort.

#### 2.4.5. **Market Target**

* **Residential Customers**: Focus on homeowners and renters seeking to improve energy efficiency and home automation.
* **Commercial Opportunities**: Potential to extend services to small businesses and office spaces looking to optimize energy management.

#### 2.4.6. **Geographical Reach**

* **Initial Launch**: Target urban and suburban markets with a higher concentration of smart home adopters.
* **Future Expansion**: Plan for scalability to reach additional markets and demographics as the product matures.

#### 2.4.7. **Regulatory Compliance**

* **Safety Standards**: Ensure the system complies with relevant safety and electrical standards.
* **Data Privacy Regulations**: Adhere to applicable laws regarding data protection and user privacy.

#### 2.4.8. **Support and Maintenance**

* **User Support**: Provide customer service and technical support to assist users with installation, troubleshooting, and system upgrades.
* **System Updates**: Regular updates to improve functionality, security, and compatibility with new devices and technologies.

### 2.4.9. Exclusions

* **Commercial-Grade Systems**: The initial focus will be on residential applications, with commercial-grade solutions considered for future iterations.
* **Legacy Device Integration**: The system may not support very old or proprietary devices lacking modern connectivity options.

**2.5. Constraints**

The development and implementation of the Smart Home Automation and Energy Management System will encounter several constraints, including:

#### 2.5.1. **Technical Constraints**

* **Device Compatibility**: Limitations in integrating with older or proprietary smart home devices may restrict system functionality.
* **Network Connectivity**: Dependence on stable internet and network connectivity for real-time monitoring and control may pose challenges, especially in areas with poor infrastructure.
* **Data Processing Capabilities**: The need for efficient data processing and storage solutions to handle real-time analytics and monitoring could limit the system's scalability.

#### 2.5.2. **Budget Constraints**

* **Development Costs**: Limited budget allocations may impact the extent of features and functionalities developed within the system.
* **Marketing and Promotion**: Constraints in marketing budgets may restrict outreach efforts, affecting user adoption rates.

#### 2.5.3. **Regulatory Constraints**

* **Compliance Requirements**: Adherence to various safety, energy efficiency, and data protection regulations may require additional resources and time for compliance checks.
* **Data Privacy Laws**: Navigating different privacy regulations across regions may complicate data collection and usage practices.

#### 2.5.4. **User Adoption Constraints**

* **User Resistance**: Potential reluctance from users to adopt new technologies due to concerns about complexity, privacy, or data security.
* **Training and Support Needs**: Users may require extensive training or support to effectively utilize the system, impacting implementation timelines.

#### 2.5.5. **Market Constraints**

* **Competition**: Intense competition in the smart home market could limit market penetration and influence pricing strategies.
* **Economic Factors**: Fluctuations in the economy may impact consumer spending on smart home technologies, affecting sales and growth.

#### 2.5.6. **Environmental Constraints**

* **Energy Source Limitations**: Variability in local energy sources and costs may affect the overall effectiveness of energy management strategies.
* **Sustainability Considerations**: The need to design energy-efficient solutions that align with sustainability goals may impose design and operational constraints.

#### 2.5.7. **Development Timeline**

* **Time-to-Market**: Tight development timelines may limit the scope of features that can be included at launch, requiring phased rollouts for additional functionalities.

#### 2.5.8. **Maintenance and Support Constraints**

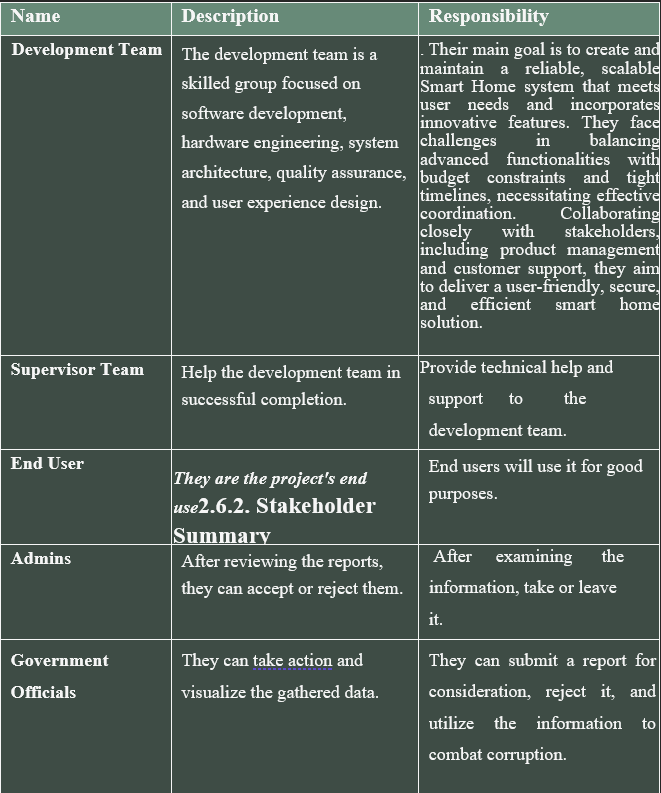
* **Ongoing Maintenance Costs**: Long-term costs associated with maintaining and updating the system could impact budget allocations for future enhancements.
* **User Support Limitations**: Limited resources for user support may lead to challenges in addressing customer inquiries and issues promptly.

**2.6.**  **Stakeholder and User Description**

#### 2.6.1. Market Demographics

* **Homeowners**: Primarily individuals aged 30-55, including families and professionals looking for convenience, energy savings, and enhanced security.
* **Renters**: Young adults and professionals aged 25-40 seeking affordable smart solutions to improve their living spaces without major renovations.
* **Tech Enthusiasts**: Individuals of various ages who are early adopters of technology, eager to incorporate the latest smart devices into their homes.
* **Environmental Advocates**: Homeowners and renters motivated by sustainability, looking for energy-efficient solutions to reduce their carbon footprint.
* **Small Businesses**: Small office spaces and business owners interested in energy management solutions to reduce operational costs.
* **Elderly Population**: Older adults looking for smart home technologies that enhance safety and independence in their living environments.

#### 2.6.2. Stakeholder Summary



#### 2.6.3. User Environment

* **Home Environment**: Users will interact with the Smart Home system primarily in residential settings, including single-family homes, apartments, and condominiums.
* **Mobile Access**: Users will access the system through smartphones, tablets, or computers, requiring responsive and user-friendly interfaces across all devices.
* **Diverse User Skills**: The user environment will include a range of technical proficiencies, from tech-savvy individuals to those less familiar with technology, necessitating an intuitive design.
* **Integration with Existing Devices**: The system should seamlessly integrate with various existing devices and technologies within the home, including smart appliances, security systems, and energy meters.

#### 2.6.4. Stakeholder Profiles

#### 2.6.4.1. Development Team

A blue screen with white text

Description automatically generated

**2.9.4.2. End Users**

A black screen with white text

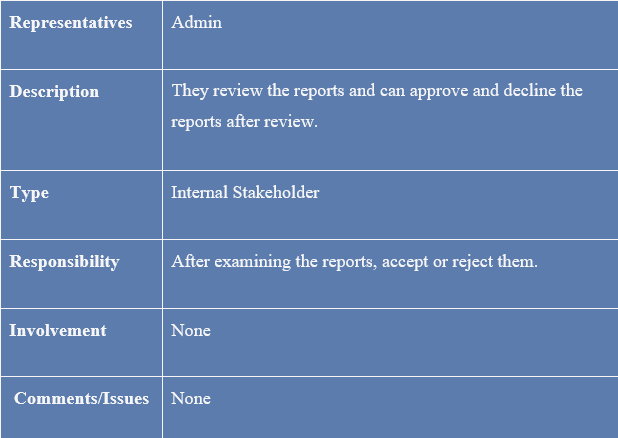
Description automatically generated

**2.6.4.3. Government Officials**

A screenshot of a computer

Description automatically generated

**2.6.4.4. Admin**



**3. System Requirements Specification**

**3.1. System Features**

The Smart Home Automation and Energy Management System is designed to offer a range of features that enhance user convenience, energy efficiency, and overall comfort. Key system features include:

#### 3.1.1. **Adaptive Automation**

* **Smart Scheduling**: Automate routines based on user preferences and habits, allowing for scheduled operations of lights, heating, cooling, and appliances.
* **Context-Aware Control**: Adjust settings automatically based on environmental conditions (e.g., turning off lights when no one is in the room).

#### 3.1.2. **Energy Management**

* **Real-Time Energy Monitoring**: Track energy consumption across devices in real-time, providing users with insights into usage patterns and costs.
* **Energy Efficiency Recommendations**: Generate personalized suggestions to optimize energy use, such as identifying high-consumption appliances or recommending usage changes.

#### 3.1.3. **Environmental Monitoring**

* **Sensor Integration**: Utilize sensors to monitor indoor air quality, temperature, humidity, and occupancy, providing data to inform system adjustments.
* **Alerts and Notifications**: Send alerts for abnormal conditions, such as high humidity or poor air quality, to help users maintain a healthy living environment.

#### 3.1.4. **User Interface**

* **Mobile and Web Applications**: Provide user-friendly interfaces for monitoring and controlling the system remotely from smartphones, tablets, or computers.
* **Voice Control Integration**: Enable compatibility with popular voice assistants (e.g., Amazon Alexa, Google Assistant) for hands-free control of smart home features.

#### 3.1.5. **Security and Surveillance**

* **Smart Security Systems**: Integrate smart locks, cameras, and alarms that can be monitored and controlled remotely for enhanced home security.
* **Access Control**: Manage user access to the home environment, including temporary access for guests or service personnel.

#### 3.1.6. **Customizable Alerts and Automation**

* **User-Customized Notifications**: Allow users to set preferences for alerts related to energy usage, environmental conditions, and system performance.
* **Automated Responses**: Configure the system to respond automatically to specific events, such as turning on lights when a door is opened at night.

#### 3.1.7. **Data Analytics and Reporting**

* **Usage Reports**: Provide periodic reports summarizing energy consumption, performance metrics, and recommendations for improvement.
* **Trend Analysis**: Analyze data over time to identify trends in energy usage and system performance, helping users make informed decisions.

#### 3.1.8. **Scalability and Integration**

* **Modular Design**: Allow users to easily add new devices and functionalities as their needs change, ensuring the system can grow with the user.
* **Compatibility with Third-Party Devices**: Support integration with a variety of third-party smart home devices and platforms for expanded functionality.

These features aim to create a comprehensive Smart Home system that enhances user convenience, promotes energy efficiency, and supports a sustainable lifestyle.

**3.2 Functional Requirements**

**3.2.1 User Registration and Login:**

The system shall allow homeowners to register and create an account.

The system shall provide login functionality with a username and password.

**3.2.2 Control Appliances Remotely:**

The system shall allow users to control home appliances (lights, fans, thermostats, etc.) remotely via a web or mobile app.

**3.2.3 Monitor Energy Usage:**

The system shall monitor the energy usage of connected devices in real-time.

The system shall provide energy consumption reports to users.

**3.2.4** **Set Schedules for Appliances**

The system shall allow users to schedule appliances (e.g., turning off lights at a certain time).

**3.2.5 Automated Appliance Control**

The system shall automatically turn off devices when not in use based on predefined rules.

**3.2.6** **Receive Alerts and Notifications**

The system shall notify users of important events (e.g., security breaches, energy overconsumption).

The system shall provide real-time notifications through mobile or email.

**3.2.7 Security System Integration**

The system shall integrate with security devices like cameras, motion sensors, and alarms.

The system shall trigger alarms and notify users in case of a breach.

**3.2.8 Voice Control Integration**

The system shall allow voice control through integration with platforms like Google Home, Alexa, etc.

**3.2.9 Manage Profiles**

The system shall allow the creation of multiple user profiles with individual preferences (e.g., different family members).

**3.2.10 Manage Environmental Conditions**

The system shall monitor temperature, humidity, and air quality using environmental sensors.

The system shall adjust appliances like HVAC or fans based on sensor data.

**3.2.11 Customizable Home Settings**

The system shall allow users to create custom automation rules (e.g., turn off lights when the room is unoccupied).

**3.2.12 Energy Usage Optimization**

The system shall optimize energy usage by switching to energy-efficient modes during peak hours or based on user preferences.

**3.2.13 Historical Data**

The system shall store historical data related to energy consumption, appliance usage, and environmental conditions for up to 12 months.

**3.2.14 Device Compatibility**

The system shall support the addition and removal of devices from the control system (e.g., new appliances, sensors).

**3.2.15 Control Multiple Homes**

The system shall allow users to manage multiple properties or homes under one account.

**3.2.16 Multi-Device Access**

The system shall be accessible from various devices, including desktops, smartphones, and tablets.

**3.2.17 User Authentication**

The system shall use secure login mechanisms like multi-factor authentication (MFA) to authenticate users.

**3.2.18 Automatic Software Updates**

The system shall automatically update to the latest version without requiring user intervention.

**3.2.19 Energy Consumption Alerts**

The system shall alert users when energy consumption exceeds a set threshold.

**3.2.20 Remote Monitoring of Home Environment**

The system shall allow users to monitor home conditions (e.g., temperature, humidity) from anywhere.

**3.2.21 System Customization**

The system shall allow users to customize the interface, including dashboard widgets and notification preferences.

**3.2.22 Data Export**

The system shall allow users to export energy usage reports and other system data for analysis.

**3.2.23 Local Weather Integration**

The system shall integrate with external weather services to adjust HVAC systems based on outdoor conditions.

**3.2.24 Appliance Malfunction Detection**

The system shall notify users if any connected appliance malfunctions (e.g., power surge, device failure).

**3.2.25 Offline Mode**

The system shall allow local control of home devices even when the internet connection is down.

**3.3 Non-Functional Requirements**

**3.3.1 Availability**

The system shall maintain an availability of 99.9%, excluding scheduled maintenance.

**3.3.2 Performance**

The system shall respond to user commands within 2 seconds, ensuring a seamless user experience.

**3.3.3 Scalability**

The system shall support up to 200 devices per home and accommodate 500 simultaneous users per server instance.

**3.3.4 Usability**

The user interface shall be intuitive and user-friendly, requiring minimal training for effective use.

**3.3.5 Energy Efficiency**

The system shall consume less than 1 watt of power in idle mode to promote energy sustainability.

**3.3.6 Reliability**

The system shall maintain reliable operation under high loads, such as controlling more than 50 appliances simultaneously.

**3.3.7 Interoperability**

The system shall support multiple communication protocols, including Wi-Fi, Zigbee, and Z-Wave, for compatibility with various devices.

**3.3.8 Security**

All user data and communications shall be encrypted with a minimum of 256-bit encryption to ensure data integrity and confidentiality.

**3.3.9 Maintainability**

The system architecture shall allow for easy maintenance and updates, minimizing downtime during upgrades.

**3.3.10 Flexibility**

The system shall accommodate changes in user requirements and support the integration of new device types.

**3.3.11 Testability**

The system shall be designed to facilitate testing, allowing for thorough validation of functionality and performance.

**3.3.12 Data Privacy Compliance**

The system shall comply with GDPR and other relevant data privacy regulations to protect user information.

**3.3.13 Audit Logging**

The system shall maintain audit logs for all user actions, including device control and configuration changes, with timestamps for security auditing.

**3.3.14 Fault Tolerance**

The system shall incorporate fault tolerance mechanisms to ensure continued operation in the event of device failures.

**3.3.15 Latency**

The system shall maintain a maximum latency of 500 milliseconds for sensor data updates to ensure timely responses.

**3.3.16 Customizability**

The user interface shall be fully customizable, allowing users to rearrange widgets, choose themes, and select colors.

**3.3.17 Responsiveness**

The system’s web and mobile interfaces shall maintain responsiveness across all device types and screen sizes.

**3.3.18 Survivability**

The system shall continue functioning during adverse conditions, such as power outages or network disruptions.

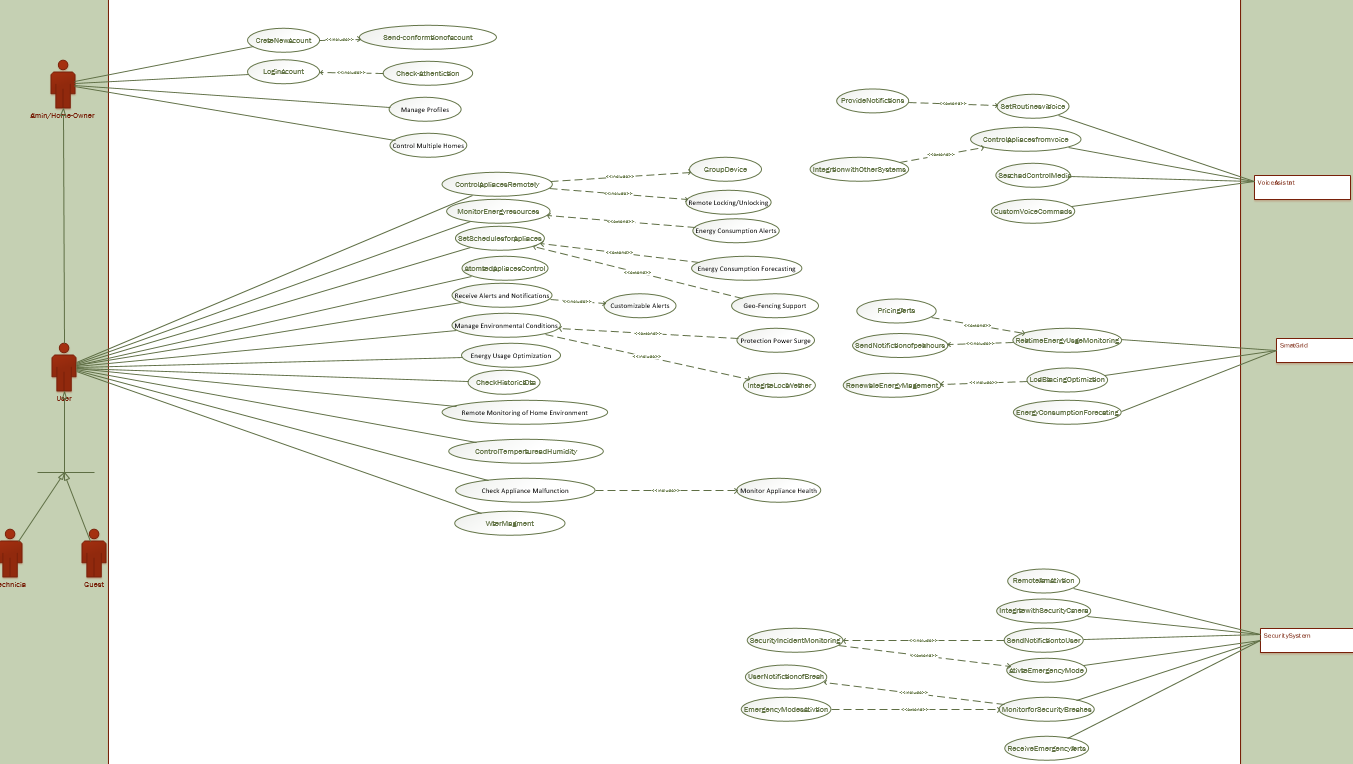
**3.3.19 Accessibility**

The system shall be designed to meet accessibility standards, ensuring usability for individuals with disabilities.

**3.3.20 Supportability**

The system shall include comprehensive documentation and support mechanisms to assist users and developers in troubleshooting and maintenance.

**USE CASE MODEL DIAGRAM**

****

***USE CASE SPECIFICATION TABLE***

### Use Case Specification: UC-01

|  |  |
| --- | --- |
| **Use Case** | UC-01 |
| **Name** | Create New Account |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can create a new account to access the smart home system. |
| **Pre-Conditions** | The user must provide valid information for account creation. |
| **Post-Conditions** | A new account is created and the user is registered in the system. |
| **Main Scenario** | The user fills out the registration form and submits it. |
| **Alternative Scenario** | If the information is invalid, the system prompts the user to correct it. |
| **Frequency of Use** | Occasionally. |

### Use Case Specification: UC-02

|  |  |
| --- | --- |
| **Use Case** | UC-02 |
| **Name** | Login Account |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can log in to their account to access the system. |
| **Pre-Conditions** | The user must have an existing account. |
| **Post-Conditions** | The user gains access to their account and system features. |
| **Main Scenario** | The user enters credentials and successfully logs in. |
| **Alternative Scenario** | If credentials are incorrect, the user is informed and prompted to retry. |
| **Frequency of Use** | Frequently. |

### Use Case Specification: UC-03

|  |  |
| --- | --- |
| **Use Case** | UC-03 |
| **Name** | Manage Profile |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can manage their profile settings within the system. |
| **Pre-Conditions** | The user must be logged in. |
| **Post-Conditions** | The user’s profile information is updated in the system. |
| **Main Scenario** | The user edits their profile information and saves changes. |
| **Alternative Scenario** | If the user tries to save invalid data, the system prompts them to correct it. |
| **Frequency of Use** | Occasionally. |

### Use Case Specification: UC-04

|  |  |
| --- | --- |
| **Use Case** | UC-04 |
| **Name** | Control Multiple Homes |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can control and manage multiple smart homes from a single account. |
| **Pre-Conditions** | The user must have access to multiple homes set up in the system. |
| **Post-Conditions** | The user can switch between homes and manage their settings. |
| **Main Scenario** | The user selects a home and controls its appliances. |
| **Alternative Scenario** | If no homes are registered, the user is informed and prompted to add one. |
| **Frequency of Use** | Occasionally. |

### Use Case Specification: UC-05

|  |  |
| --- | --- |
| **Use Case** | UC-05 |
| **Name** | Control Appliances Remotely |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can control home appliances remotely via the application. |
| **Pre-Conditions** | Appliances must be connected to the network. |
| **Post-Conditions** | Appliances are controlled as per the user’s commands. |
| **Main Scenario** | The user selects an appliance and sends a command to turn it on/off. |
| **Alternative Scenario** | If the appliance is offline, the system notifies the user. |
| **Frequency of Use** | Frequently. |

### Use Case Specification: UC-06

|  |  |
| --- | --- |
| **Use Case** | UC-06 |
| **Name** | Monitor Energy Resource |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can monitor energy consumption from connected appliances. |
| **Pre-Conditions** | User must be logged in and connected to the energy monitoring system. |
| **Post-Conditions** | The user receives real-time energy consumption data. |
| **Main Scenario** | The user views energy usage statistics on the app. |
| **Alternative Scenario** | If the energy data is unavailable, the user is notified. |
| **Frequency of Use** | Frequently. |

### Use Case Specification: UC-07

|  |  |
| --- | --- |
| **Use Case** | UC-07 |
| **Name** | Set Schedules For Appliances |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can schedule appliances to operate at specific times. |
| **Pre-Conditions** | Appliances must be connected and operational. |
| **Post-Conditions** | Appliances operate as per the set schedule. |
| **Main Scenario** | The user sets a schedule for the washing machine to start at 7 PM. |
| **Alternative Scenario** | If the schedule conflicts with another, the user is alerted. |
| **Frequency of Use** | Occasionally. |

### Use Case Specification: UC-08

|  |  |
| --- | --- |
| **Use Case** | UC-08 |
| **Name** | Automated Appliances Control |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can enable automation for appliance control based on predefined conditions. |
| **Pre-Conditions** | The user must have set automation rules. |
| **Post-Conditions** | Appliances respond to automation triggers. |
| **Main Scenario** | The user sets the lights to turn on when the sun sets. |
| **Alternative Scenario** | If the automation rule cannot be executed, an error message is displayed. |
| **Frequency of Use** | Frequently. |

### Use Case Specification: UC-09

|  |  |
| --- | --- |
| **Use Case** | UC-09 |
| **Name** | Receive Alerts and Notification |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user receives notifications for system alerts and updates. |
| **Pre-Conditions** | Notifications must be enabled in user settings. |
| **Post-Conditions** | User is informed about relevant updates. |
| **Main Scenario** | The user receives a notification about appliance maintenance needs. |
| **Alternative Scenario** | If notifications are turned off, the user does not receive any alerts. |
| **Frequency of Use** | Frequently. |

### Use Case Specification: UC-10

|  |  |
| --- | --- |
| **Use Case** | UC-10 |
| **Name** | Manage Environmental Conditions |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can manage environmental conditions within their home. |
| **Pre-Conditions** | Sensors must be installed to monitor conditions. |
| **Post-Conditions** | Environmental settings are adjusted based on user input. |
| **Main Scenario** | The user adjusts the humidity levels through the app. |
| **Alternative Scenario** | If the user sets unrealistic levels, the system prompts for correction. |
| **Frequency of Use** | Frequently. |

### Use Case Specification: UC-11

|  |  |
| --- | --- |
| **Use Case** | UC-11 |
| **Name** | Energy Usage Optimization |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can optimize energy consumption for efficiency. |
| **Pre-Conditions** | The system must have energy consumption data. |
| **Post-Conditions** | Recommendations for energy savings are provided. |
| **Main Scenario** | The user receives tips on reducing energy usage based on patterns. |
| **Alternative Scenario** | If optimization fails, the user is informed of the issue. |
| **Frequency of Use** | Occasionally. |

### Use Case Specification: UC-12

|  |  |
| --- | --- |
| **Use Case** | UC-12 |
| **Name** | Check Historical Data |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can access historical data for energy usage. |
| **Pre-Conditions** | The system must have stored historical data. |
| **Post-Conditions** | The user views trends and usage over time. |
| **Main Scenario** | The user selects a date range to view historical energy data. |
| **Alternative Scenario** | If no data is available for the selected range, the user is notified. |
| **Frequency of Use** | Occasionally. |

### Use Case Specification: UC-13

|  |  |
| --- | --- |
| **Use Case** | UC-13 |
| **Name** | Remote Monitoring Of Home Environment |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can remotely monitor the home environment using sensors. |
| **Pre-Conditions** | Sensors must be installed and connected to the system. |
| **Post-Conditions** | The user receives real-time data about the home environment. |
| **Main Scenario** | The user views the environmental data from the sensors on the app. |
| **Alternative Scenario** | If sensors are offline, the user is notified of the issue. |
| **Frequency of Use** | Frequently. |

### Use Case Specification: UC-14

|  |  |
| --- | --- |
| **Use Case** | UC-14 |
| **Name** | Control Temperature and Humidity |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can adjust the temperature and humidity settings in their home. |
| **Pre-Conditions** | The HVAC system must be connected to the network. |
| **Post-Conditions** | The HVAC system adjusts to the user’s specified settings. |
| **Main Scenario** | The user sets desired temperature and humidity levels through the app. |
| **Alternative Scenario** | If the user inputs invalid values, the system prompts for correction. |
| **Frequency of Use** | Frequently. |

### Use Case Specification: UC-15

|  |  |
| --- | --- |
| **Use Case** | UC-15 |
| **Name** | Check Appliance Malfunction |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can check for any malfunctions in connected appliances. |
| **Pre-Conditions** | Appliances must have diagnostic features enabled. |
| **Post-Conditions** | The user is informed about the status of appliances. |
| **Main Scenario** | The user requests a status check for all appliances. |
| **Alternative Scenario** | If an appliance is malfunctioning, the user receives a detailed alert. |
| **Frequency of Use** | Occasionally. |

### Use Case Specification: UC-16

|  |  |
| --- | --- |
| **Use Case** | UC-16 |
| **Name** | Water Management |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can manage water usage and control irrigation systems. |
| **Pre-Conditions** | Irrigation systems must be connected and configured. |
| **Post-Conditions** | Water usage is optimized based on user settings. |
| **Main Scenario** | The user sets irrigation schedules based on weather conditions. |
| **Alternative Scenario** | If water usage exceeds limits, the system alerts the user. |
| **Frequency of Use** | Occasionally. |

### Use Case Specification: UC-17

|  |  |
| --- | --- |
| **Use Case** | UC-17 |
| **Name** | Set Routine via Voice |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can set routines for appliances using voice commands. |
| **Pre-Conditions** | Voice control must be enabled. |
| **Post-Conditions** | Routines are created and activated as per voice commands. |
| **Main Scenario** | The user instructs the voice assistant to set a routine. |
| **Alternative Scenario** | If the command is unclear, the system asks for clarification. |
| **Frequency of Use** | Occasionally. |

### Use Case Specification: UC-18

|  |  |
| --- | --- |
| **Use Case** | UC-18 |
| **Name** | Control Appliance from Voice |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can control appliances using voice commands. |
| **Pre-Conditions** | Voice recognition must be enabled. |
| **Post-Conditions** | The appliances respond to voice commands. |
| **Main Scenario** | The user says "turn on the living room lights" and the lights respond. |
| **Alternative Scenario** | If the command is not recognized, the system requests a repeat. |
| **Frequency of Use** | Frequently. |

### Use Case Specification: UC-19

|  |  |
| --- | --- |
| **Use Case** | UC-19 |
| **Name** | Search and Control Media |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can search for and control media devices (TV, speakers, etc.). |
| **Pre-Conditions** | Media devices must be connected. |
| **Post-Conditions** | Media playback is controlled as per user commands. |
| **Main Scenario** | The user searches for a movie and plays it on the TV. |
| **Alternative Scenario** | If the media is unavailable, the user is notified. |
| **Frequency of Use** | Occasionally. |

### Use Case Specification: UC-20

|  |  |
| --- | --- |
| **Use Case** | UC-20 |
| **Name** | Custom Voice Commands |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can create custom voice commands for various functions. |
| **Pre-Conditions** | Voice control must be configured. |
| **Post-Conditions** | Custom commands are stored and can be used. |
| **Main Scenario** | The user creates a command "Goodnight" to turn off all lights and appliances. |
| **Alternative Scenario** | If the command conflicts with existing commands, the user is alerted. |
| **Frequency of Use** | Occasionally. |

### Use Case Specification: UC-21

|  |  |
| --- | --- |
| **Use Case** | UC-21 |
| **Name** | Real Time Energy Usage Monitoring |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can monitor real-time energy usage of connected devices. |
| **Pre-Conditions** | Devices must be connected and capable of reporting usage. |
| **Post-Conditions** | The user receives up-to-date energy consumption data. |
| **Main Scenario** | The user views real-time energy usage on the app dashboard. |
| **Alternative Scenario** | If there is a connectivity issue, the user is informed. |
| **Frequency of Use** | Frequently. |

### Use Case Specification: UC-22

|  |  |
| --- | --- |
| **Use Case** | UC-22 |
| **Name** | Load Balancing Optimization |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can optimize load balancing across connected appliances. |
| **Pre-Conditions** | The system must analyze current load data. |
| **Post-Conditions** | Appliances operate efficiently to balance load. |
| **Main Scenario** | The user sets preferences for optimal appliance usage. |
| **Alternative Scenario** | If load balancing cannot be achieved, the user is notified. |
| **Frequency of Use** | Occasionally. |

### Use Case Specification: UC-23

|  |  |
| --- | --- |
| **Use Case** | UC-23 |
| **Name** | Energy Consumption Forecasting |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can forecast future energy consumption based on historical data. |
| **Pre-Conditions** | Historical data must be available. |
| **Post-Conditions** | The user receives forecasts for future energy usage. |
| **Main Scenario** | The user views predicted energy consumption for the next month. |
| **Alternative Scenario** | If data is insufficient for forecasting, the user is informed. |
| **Frequency of Use** | Occasionally. |

### Use Case Specification: UC-24

|  |  |
| --- | --- |
| **Use Case** | UC-24 |
| **Name** | Remote Alarm Activation |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can activate alarms remotely through the app. |
| **Pre-Conditions** | Alarm systems must be connected. |
| **Post-Conditions** | Alarms are activated as per user command. |
| **Main Scenario** | The user activates the home security alarm when leaving. |
| **Alternative Scenario** | If the alarm cannot be activated, an error message is displayed. |
| **Frequency of Use** | Occasionally. |

### Use Case Specification: UC-25

|  |  |
| --- | --- |
| **Use Case** | UC-25 |
| **Name** | Integrate with Security Cameras |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can integrate and monitor security camera feeds. |
| **Pre-Conditions** | Security cameras must be connected to the network. |
| **Post-Conditions** | The user views camera feeds in real-time. |
| **Main Scenario** | The user checks the camera feed from the app. |
| **Alternative Scenario** | If the feed is unavailable, the user is notified. |
| **Frequency of Use** | Frequently. |

### Use Case Specification: UC-26

|  |  |
| --- | --- |
| **Use Case** | UC-26 |
| **Name** | Send Notification to User |
| **Actor** | System |
| **Type** | Primary |
| **Description** | The system can send notifications to the user based on events. |
| **Pre-Conditions** | The user must have notifications enabled. |
| **Post-Conditions** | User receives timely notifications. |
| **Main Scenario** | The system sends a notification when a security breach is detected. |
| **Alternative Scenario** | If notifications are turned off, the system cannot send alerts. |
| **Frequency of Use** | Frequently. |

### Use Case Specification: UC-27

|  |  |
| --- | --- |
| **Use Case** | UC-27 |
| **Name** | Activate Emergency Mode |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user can activate emergency protocols for safety. |
| **Pre-Conditions** | The system must support emergency features. |
| **Post-Conditions** | Emergency protocols are initiated. |
| **Main Scenario** | The user activates emergency mode in response to a threat. |
| **Alternative Scenario** | If emergency mode fails to activate, the user is notified. |
| **Frequency of Use** | Occasionally. |

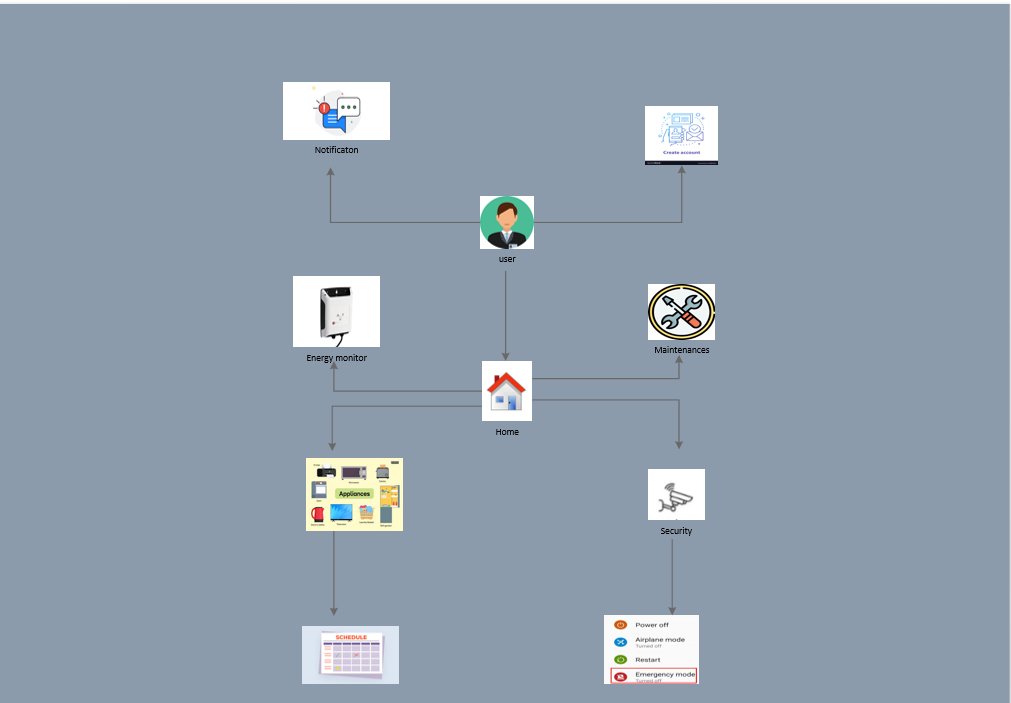
### Use Case Specification: UC-28

|  |  |
| --- | --- |
| **Use Case** | UC-28 |
| **Name** | Monitor for Security Breaches |
| **Actor** | System |
| **Type** | Primary |
| **Description** | The system continuously monitors for security breaches. |
| **Pre-Conditions** | Security systems must be active. |
| **Post-Conditions** | The system alerts the user if a breach is detected. |
| **Main Scenario** | The system monitors all entry points for unauthorized access. |
| **Alternative Scenario** | If monitoring fails, the system alerts the administrator. |
| **Frequency of Use** | Continuously. |

### Use Case Specification: UC-29

|  |  |
| --- | --- |
| **Use Case** | UC-29 |
| **Name** | Receive Emergency Alerts |
| **Actor** | User |
| **Type** | Primary |
| **Description** | The user receives alerts in case of emergencies. |
| **Pre-Conditions** | The user must have notifications enabled. |
| **Post-Conditions** | User is informed of the emergency. |
| **Main Scenario** | The user receives a push notification about a fire alarm. |
| **Alternative Scenario** | If alerts are disabled, the user does not receive notifications. |
| **Frequency of Use** | Occasionally. |
|  |  |

***ARCHITECTURE DIAGRAM***

******

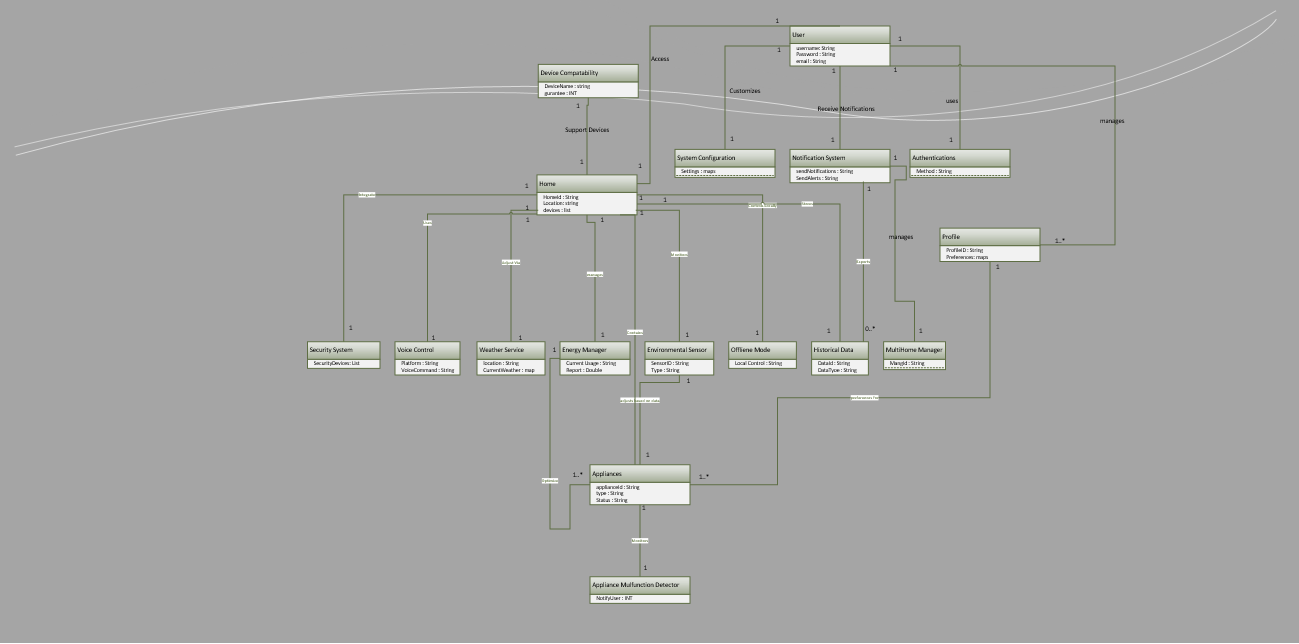
***Project Phase 2***

***Domain Model***

 **Purpose**: A domain model, also known as a conceptual model, represents the main concepts or objects within a problem domain and their relationships. It's used to understand and define the structure and terminology of the domain before moving on to more detailed design or coding.

 **How It Works**: The domain model consists of classes, attributes, and relationships that describe the entities and their interactions within the domain. It does not include any details about system behavior.

 **Significance**: It helps clarify and define the vocabulary of the domain for all stakeholders, ensuring everyone has a shared understanding. This is especially helpful in the initial stages of system design.

******

***Behavioral Modeling***

***1.1System Sequence Diagram (SSD)***

 **Purpose**: A system sequence diagram illustrates the sequence of interactions between an external actor (user or another system) and the system. It shows the order of events required to complete a particular use case or system operation.

 **How It Works**: SSDs show the system as a “black box” with external actors interacting with it. Each interaction is represented as a sequence of messages (method calls) sent to the system, displayed in the order they occur.

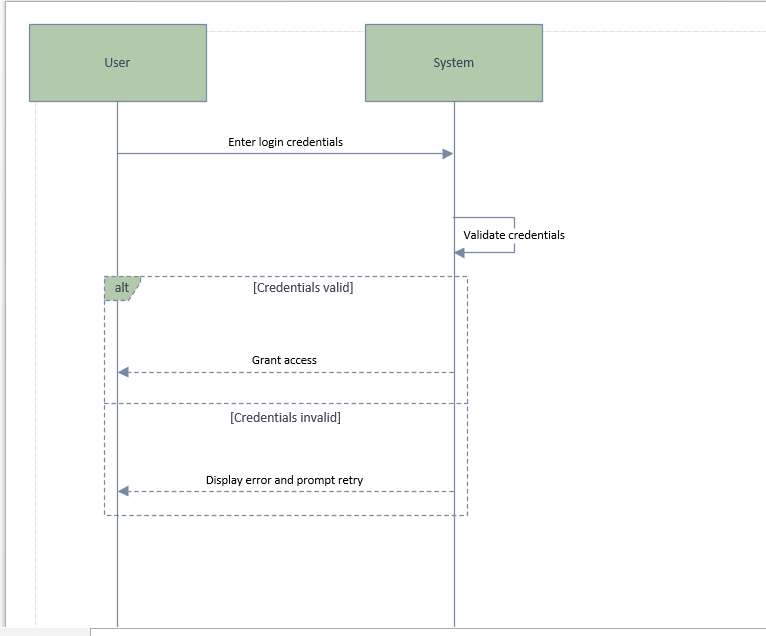
 **Significance**: SSDs are essential for visualizing system events in sequence, helping to define the input-output behavior without going into internal processing. They also aid in identifying system operations and defining the boundaries of the system.

UC-01: **Create New Account**

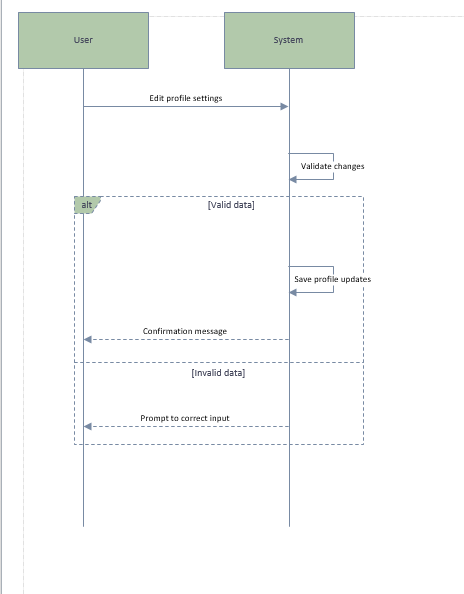
***A screenshot of a computer

Description automatically generated***

***UC-02: Login Account***

******

***UC-03: Manage Profile***

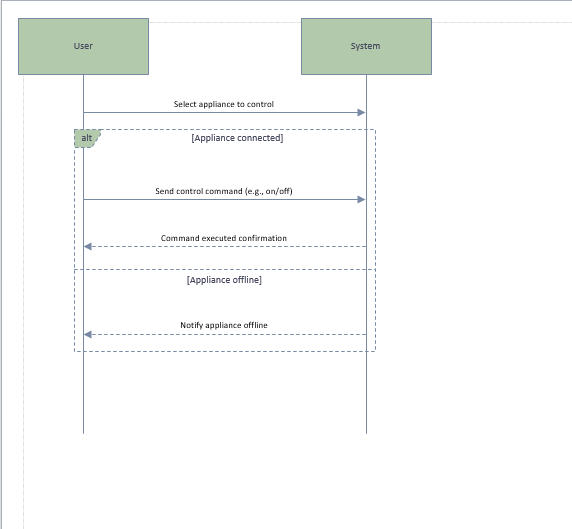
******

***UC-04: Control Multiple Homes***

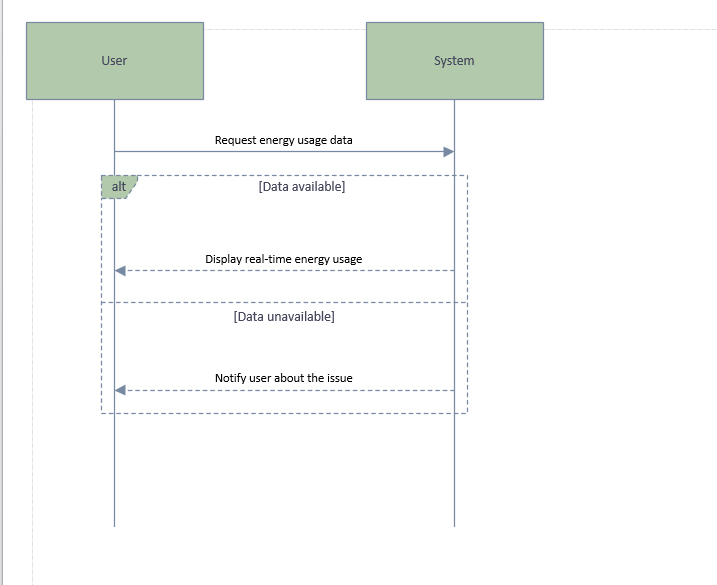
A screenshot of a computer

Description automatically generated

***UC-05: Control Appliances Remotely***

******

***UC-06: Monitor Energy Usage***

******

***UC-07: Set Schedules for Appliances***

***A screenshot of a computer screen

Description automatically generated***

***UC-08: Automated Appliance Control***

***A diagram of a system

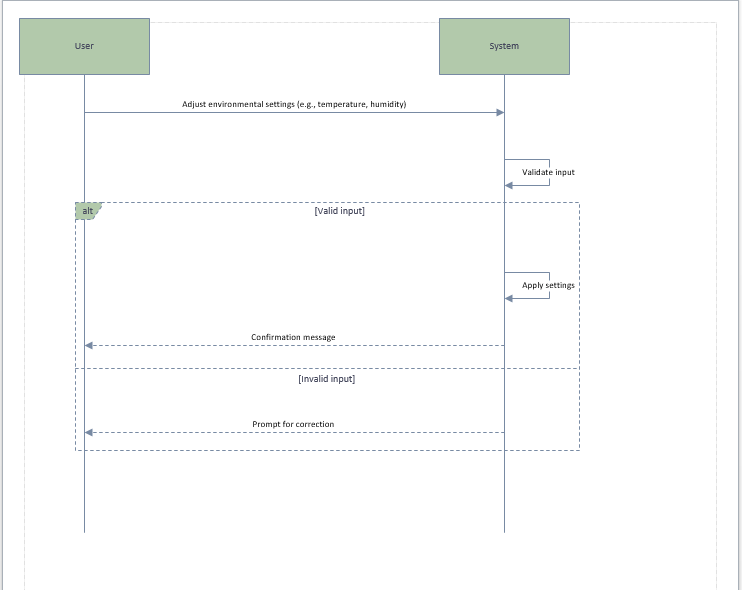
Description automatically generated***

***UC-09: Receive Alerts and Notifications***

***A diagram of a system

Description automatically generated***

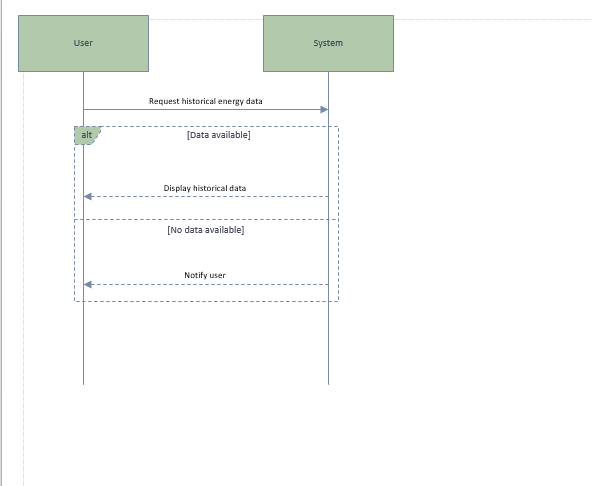
***UC-10: Manage Environmental Conditions***

******

***UC-11: Energy Usage Optimization***

******

***UC-12: Check Historical Data***

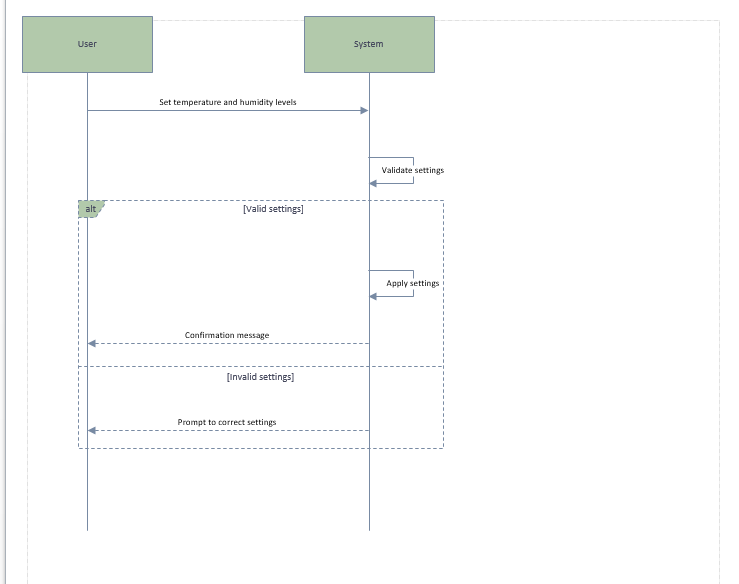
******

***UC-13: Remote Monitoring of Home Environment***

***A diagram of a system

Description automatically generated***

***UC-14: Control Temperature and Humidity***

******

***UC-15: Check Appliance Malfunction***

***A screenshot of a computer

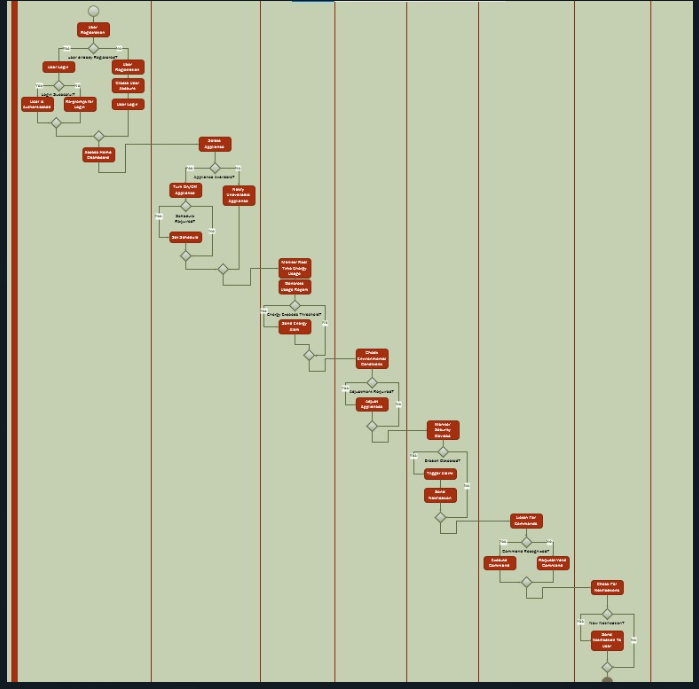
Description automatically generated***

***1.2 Activity Diagram OR SWIM LANE***

 **Purpose**: Activity diagrams illustrate the flow of activities or actions in a system, often used to describe workflows or business processes. They can represent both sequential and parallel flows and are beneficial for mapping out complex processes.

 **How It Works**: This diagram consists of actions (tasks), control flows (sequence of actions), decisions (branches), and parallel nodes (forks and joins for concurrent actions).

 **Significance**: Activity diagrams are useful for analyzing system functionality in terms of workflows. They provide a high-level view of the process, helping to identify bottlenecks and improve efficiency.

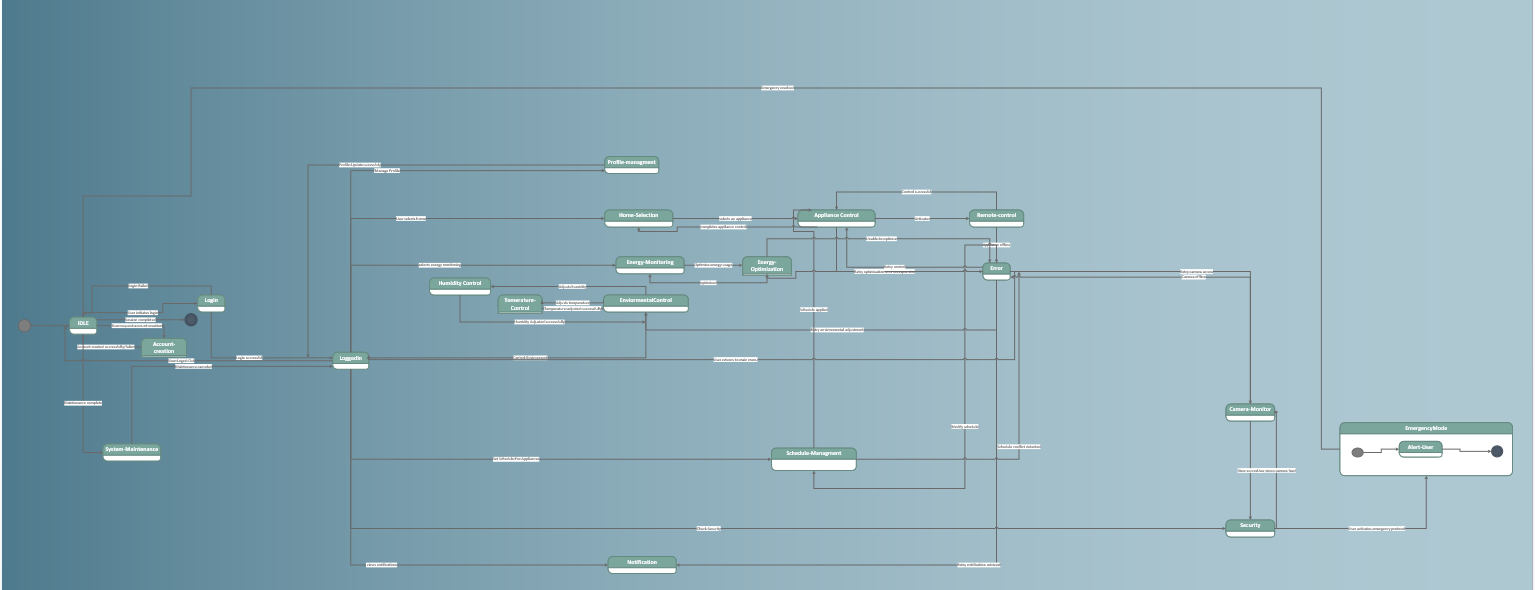


***1.3 State Machine Diagram***

 **Purpose**: State machine diagrams (or state diagrams) show the states of an object and how it transitions between states in response to events. They’re often used for objects with complex life cycles.

 **How It Works**: The diagram consists of states, transitions, events, and actions. Each state represents a status of the object, while transitions depict changes from one state to another based on events.

 **Significance**: State machine diagrams are essential for understanding and managing the life cycle of objects within a system, especially those with complex behaviors. They help ensure the object behaves correctly in response to events, particularly useful in embedded systems and real-time applications.

******

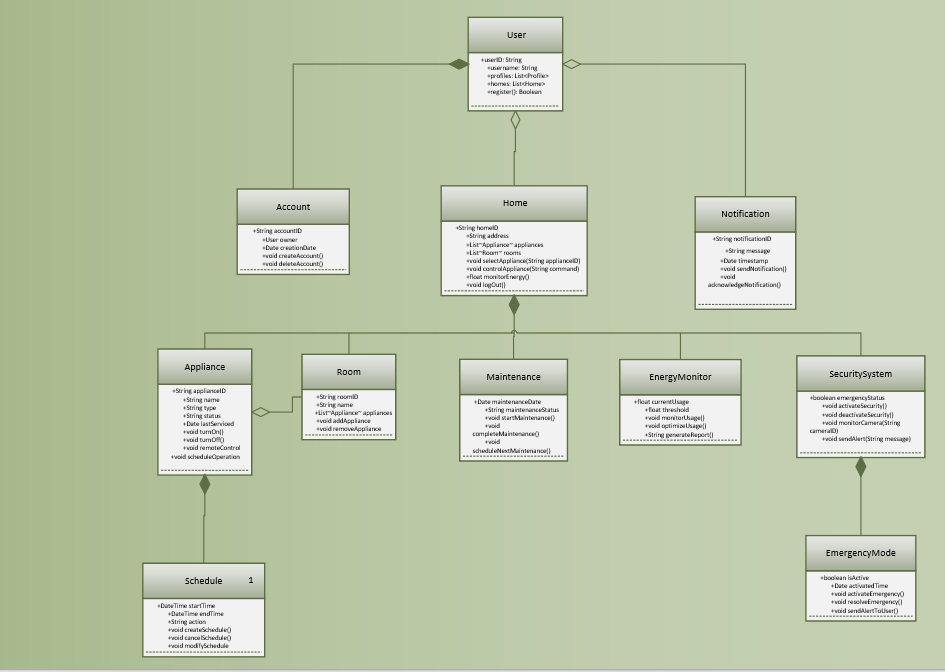
***Logical Models***

* 1. ***Class Diagram***

 **Purpose**: A class diagram is a static structure diagram that represents the classes in a system and their relationships, such as inheritance, associations, and dependencies.

 **How It Works**: Class diagrams show classes with attributes, methods, and visibility (e.g., public, private), along with lines connecting classes to indicate relationships. The classes represent the primary components and their relationships within the system.

 **Significance**: Class diagrams are critical for object-oriented design, providing a blueprint for coding and a clear structure for managing data. They are foundational for system architecture and serve as a reference for developers during implementation.

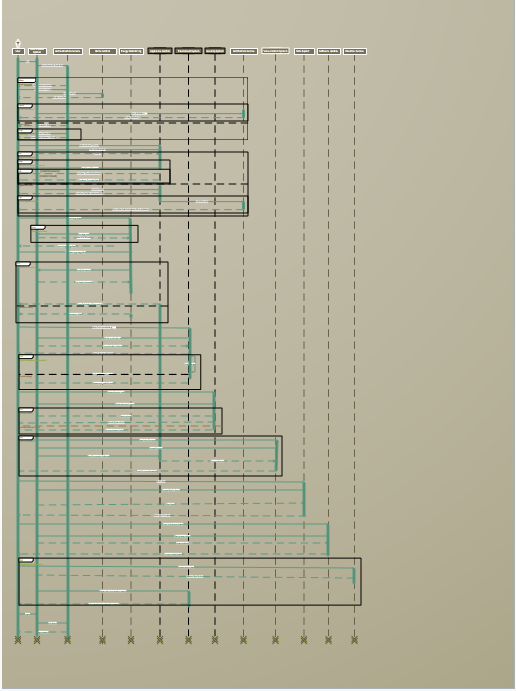


* 1. ***Sequence Diagram***

 **Purpose**: Sequence diagrams depict the order of messages exchanged between objects to carry out a specific process or interaction. They show how and in what order different parts of the system interact.

 **How It Works**: This diagram shows objects as columns and messages as arrows from one object to another in chronological order, from top to bottom. The focus is on the message flow rather than the internal structure of the objects.

 **Significance**: Sequence diagrams help clarify the flow of control and data between different system parts, making it easier to understand the system’s operation at a granular level. They are useful for identifying issues with interactions early in the design process.



***Operation Contracts***

 **Purpose**: An operation contract defines the effects of a system operation in terms of changes to the state of the system. It specifies what must be true before the operation (preconditions) and what is guaranteed after the operation (postconditions).

 **How It Works**: Operation contracts consist of preconditions, postconditions, and sometimes invariants (conditions that must always be true). They focus on the system’s state changes due to a particular operation.

 **Significance**: Operation contracts are valuable for defining precise, clear specifications for system operations, ensuring that system changes are controlled and predictable. They play a crucial role in documenting requirements and guiding implementation, particularly in complex systems where state changes are significant.

***OC1:***

|  |  |
| --- | --- |
| **Operation Name** | loginUser(username: String, password: String) |
| **Cross Reference** | User Login and Authentication |
| **Pre-Condition** | The username and password fields are non-empty.The user is not already logged in. |
| **Post Condition** | **Instance created:** User session is created if authentication is Successful.  **Association formed**: User is associated with the Dashboard**.**  **Attribute modification:** Displays dashboard or error message if authentication fails. |

***OC2:***

|  |  |
| --- | --- |
| **Operation Name** | controlAppliance(applianceID: ApplianceID, action: String) |
| **Cross Reference** | Control Appliances Remotely |
| **Pre-Condition** | The user is logged in.The appliance exists and is connected.Action is either "on" or "off." |
| **Post Condition** | **Instance created:** Control action on appliance is logged.  **Association formed**: Appliance status updated.  **Attribute modification:** Confirmation message sent to user. |

***OC3:***

|  |  |
| --- | --- |
| **Operation Name** | scheduleAppliance(applianceID: ApplianceID, time: DateTime, action: String) |
| **Cross Reference** | Set Schedules for Appliances |
| **Pre-Condition** | The user is logged in.The appliance is available and connected.Time is a future date. |
| **Post Condition** | **Instance created:** Schedule set for appliance.  **Association formed:** Schedule added to active schedules.  **Attribute modification:** Confirmation message displayed. |

***OC4:***

|  |  |
| --- | --- |
| **Operation Name** | monitorEnergyUsage() |
| **Cross Reference** | Monitor Energy Usage |
| **Pre-Condition** | The user is logged in.System has real-time access to energy data. |
| **Post Condition** | **Instance created:** Energy consumption report generated.  **Association formed:** Data saved for historical reference.  **Attribute modification:** Displays energy usage data. |

***OC5:***

|  |  |
| --- | --- |
| **Operation Name** | createAlert(alertType: String, threshold: Integer) |
| **Cross Reference** | Receive Alerts and Notifications |
| **Pre-Condition** | The user is logged in.Alert type is valid. |
| **Post Condition** | **Instance created:** Alert configuration created.  **Association formed:** System notifies user when criteria met.  **Attribute modification:** Alert stored for reference. |

***OC6:***

|  |  |
| --- | --- |
| **Operation Name** | configureSecurityIntegration(deviceID: DeviceID) |
| **Cross Reference** | Security System Integration |
| **Pre-Condition** | The user is logged in.Security device is available. |
| **Post Condition** | **Instance created:** Security device activated.  **Association formed:** Device linked to security monitoring.  **Attribute modification:** Continuous monitoring enabled. |

***OC7:***

|  |  |
| --- | --- |
| **Operation Name** | setEnvironmentalControl(targetTemp: Integer, targetHumidity: Integer) |
| **Cross Reference** | Manage Environmental Conditions |
| **Pre-Condition** | The user is logged in.Target levels are within valid ranges. |
| **Post Condition** | **Instance created:** HVAC adjusted.  **Association formed**: Environmental settings logged.  **Attribute modification:** Confirmation sent to user. |

***OC8:***

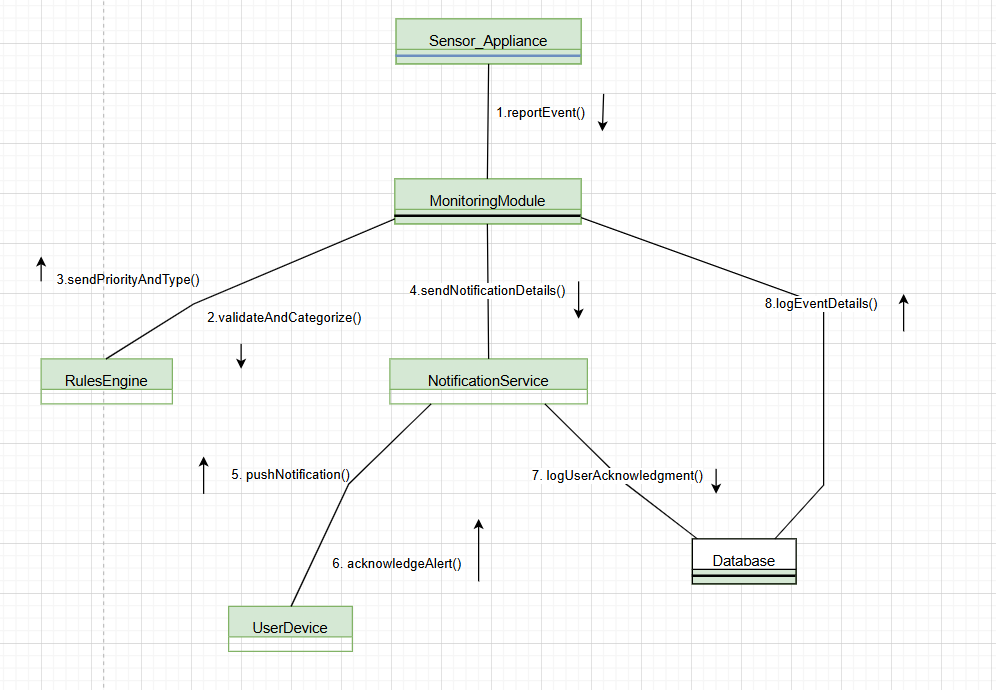
|  |  |
| --- | --- |
| **Operation Name** | exportData(reportType: String) |
| **Cross Reference** | Data Export |
| **Pre-Condition** | The user is logged in.Report type is valid. |
| **Post Condition** | **Instance created:** Report exported.  **Association formed:** Report saved for access.  **Attribute modification:** Confirmation message displayed. |
|  |  |
|  |  |

**PHASE 3**

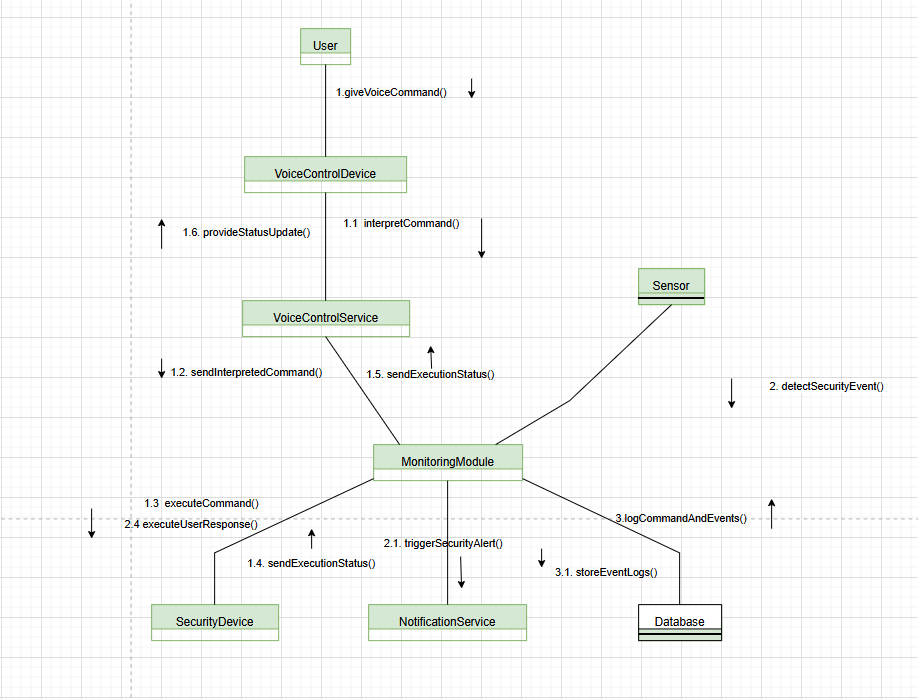
**1. Communication (Collaboration) Diagram**

* **Definition**: Shows how objects or components interact with each other through message passing.
* **Purpose**: Focuses on the relationships and interactions between objects in a system.
* **Key Elements**:
  + Objects or components
  + Messages exchanged between them
* **Example**: In an online shopping system, a communication diagram may show how the user interacts with a shopping cart and payment system.
  1. User Authentication

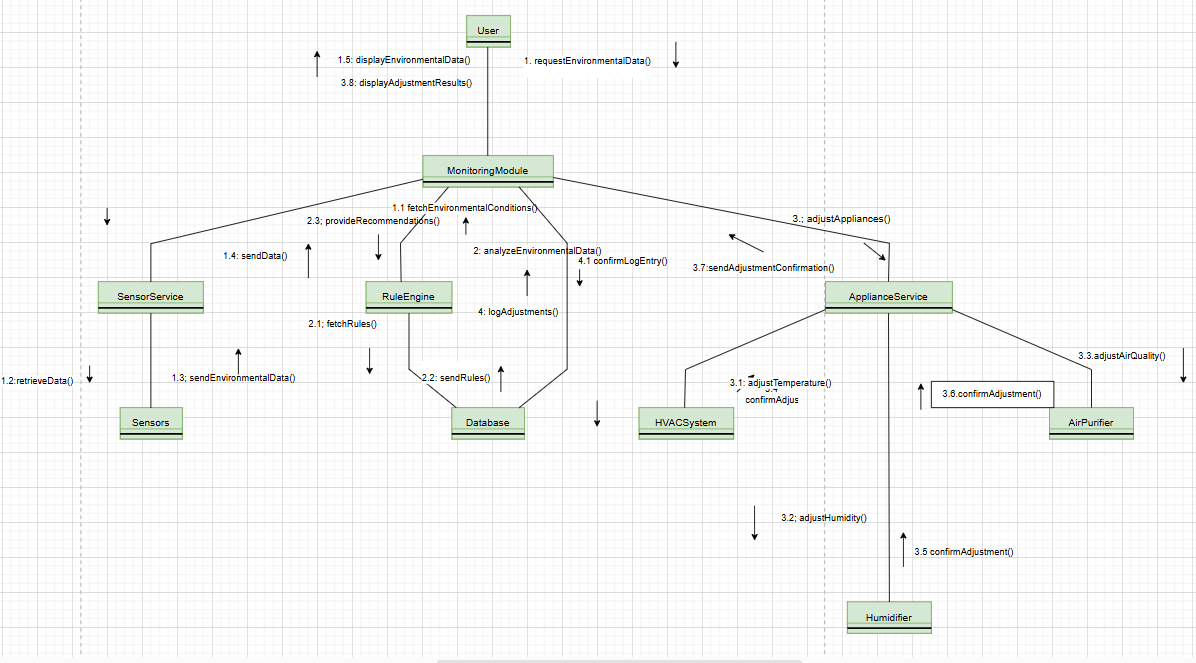
**1.2 Notification Alert**



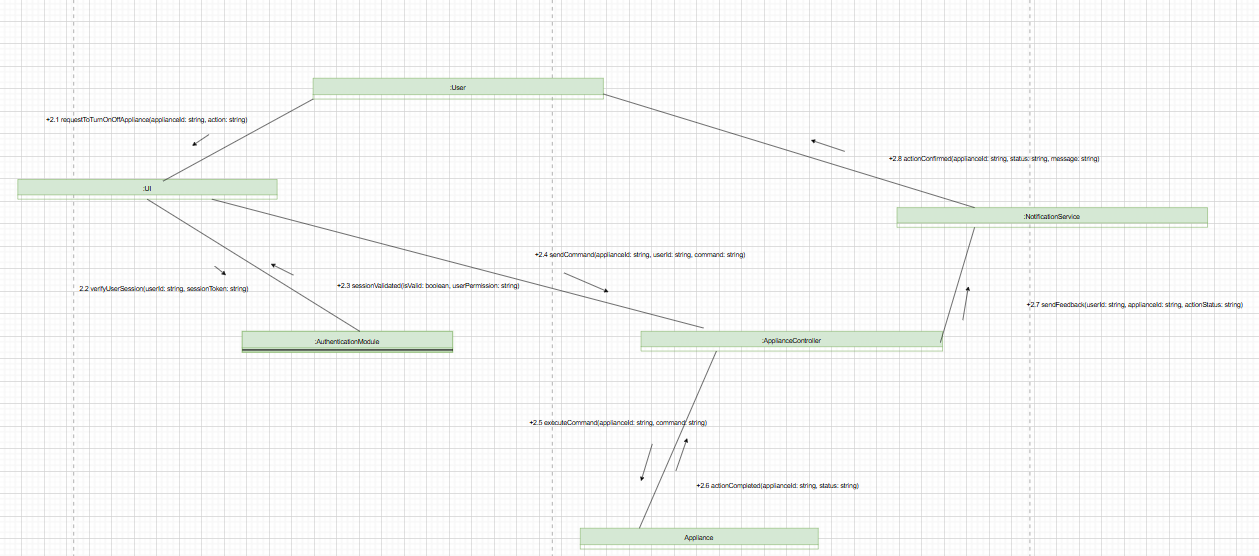
* 1. **Voice Integration**



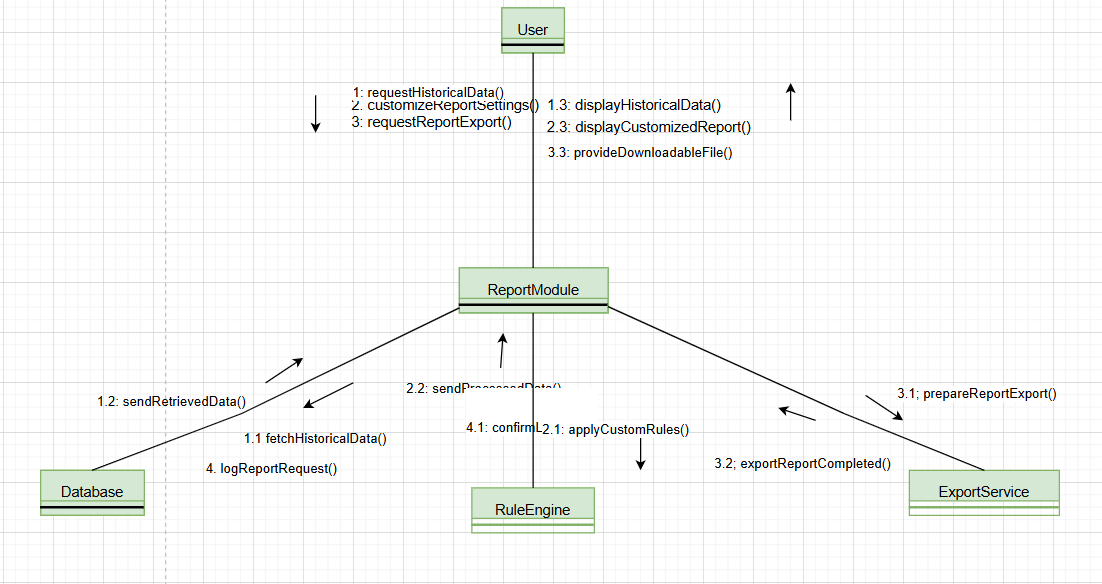
* 1. **Environmental Adjustment**



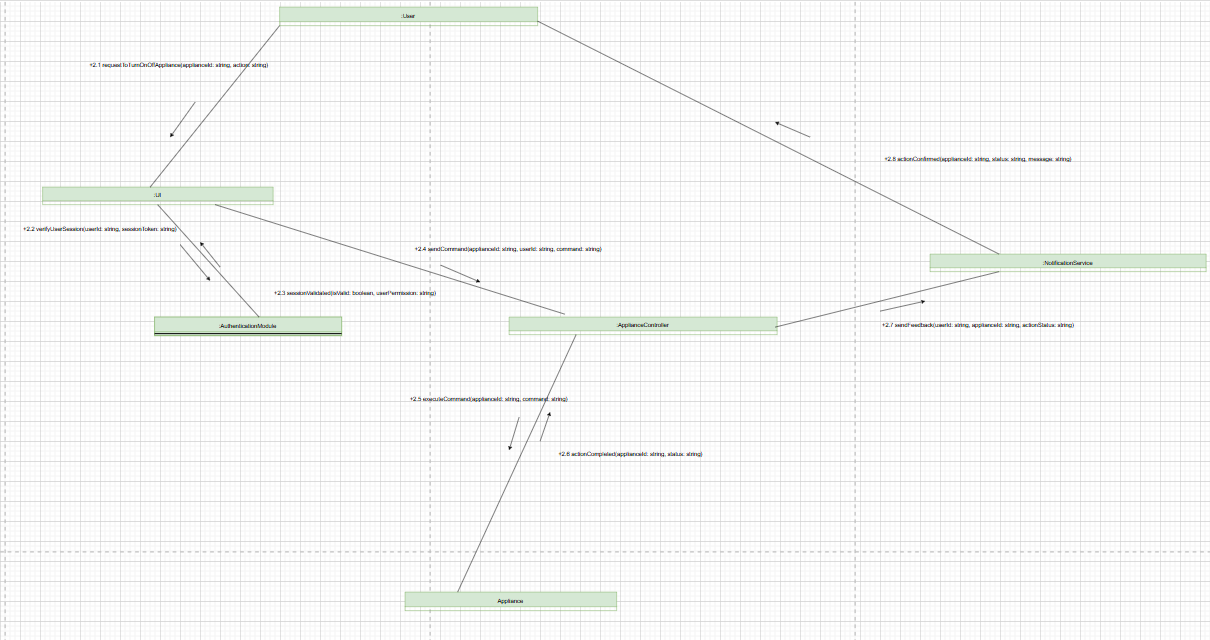
**1.5 Appliance Control**



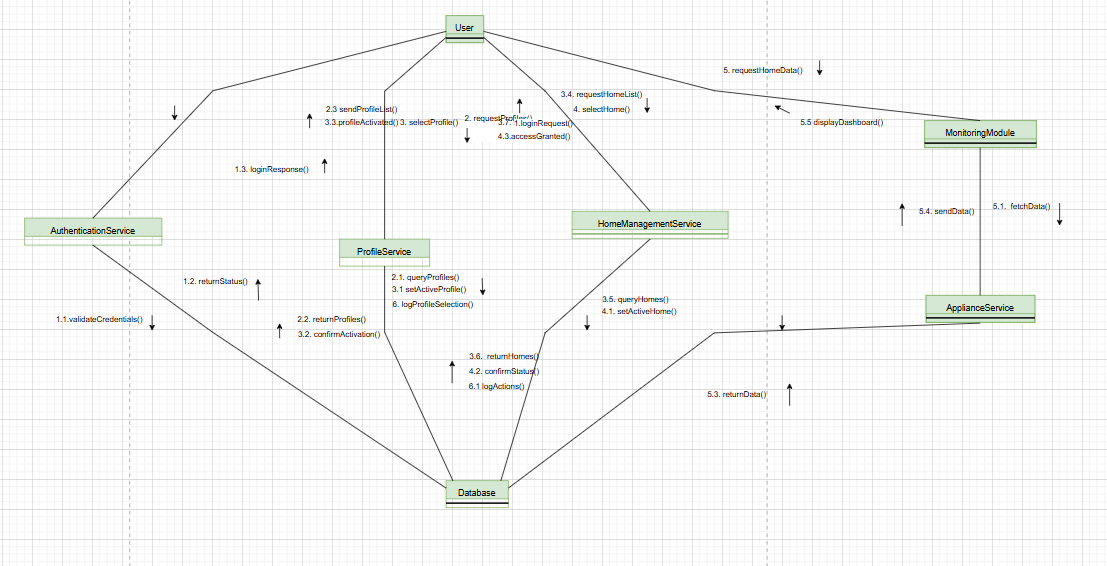
**1.6 Data Handling**



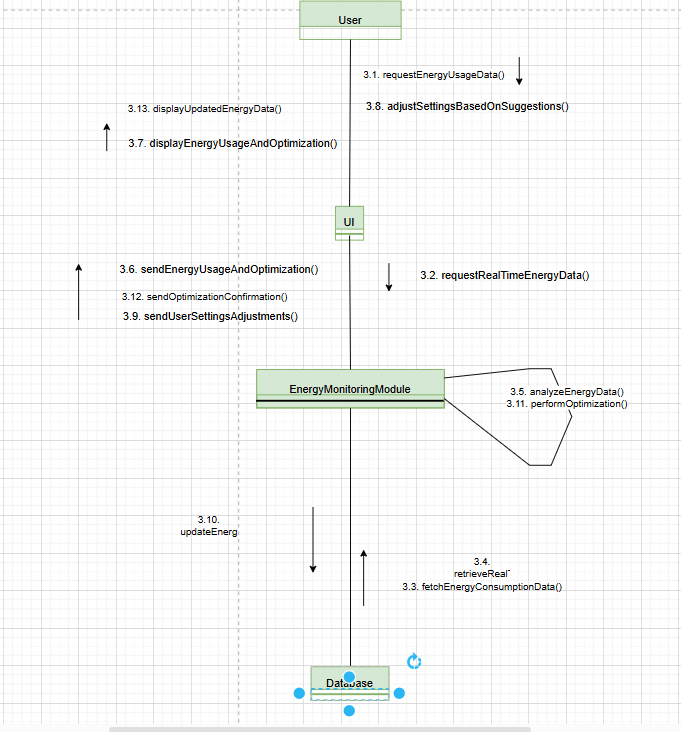
**1.7 Communication Appliance**



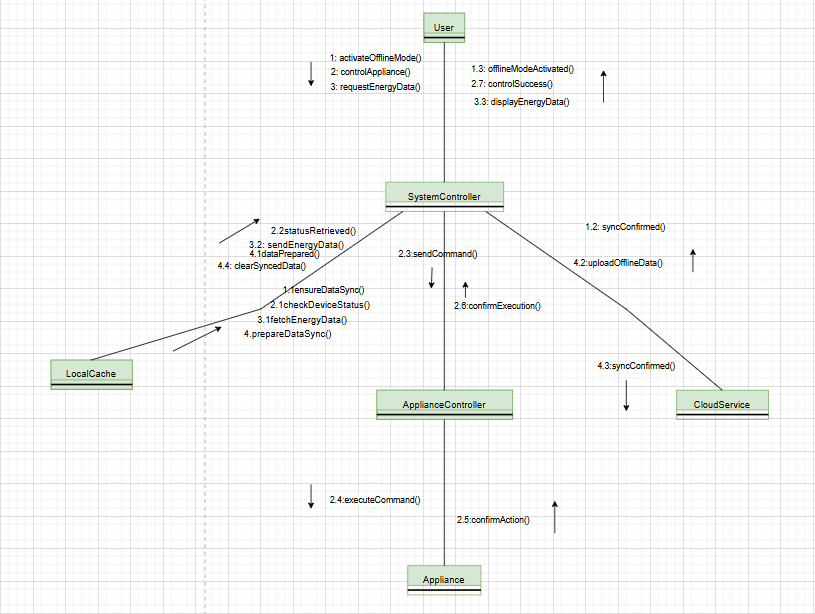
**1.8 Multi-profile\_ and\_multi-home\_managment**



**1.9 energy monitoring**

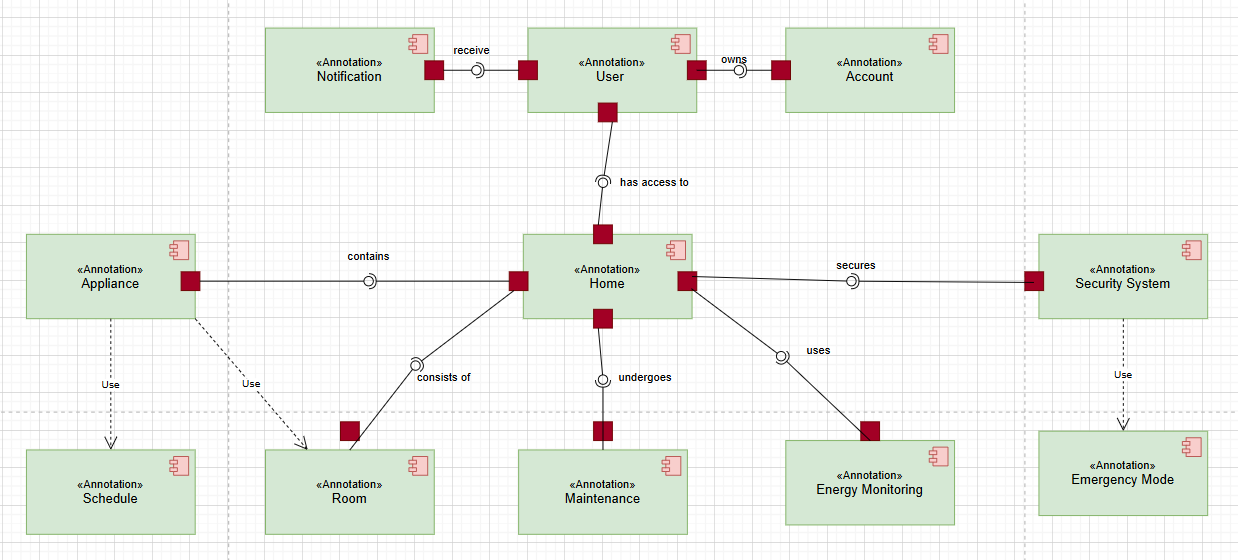


**1.10 Offline Functionality**



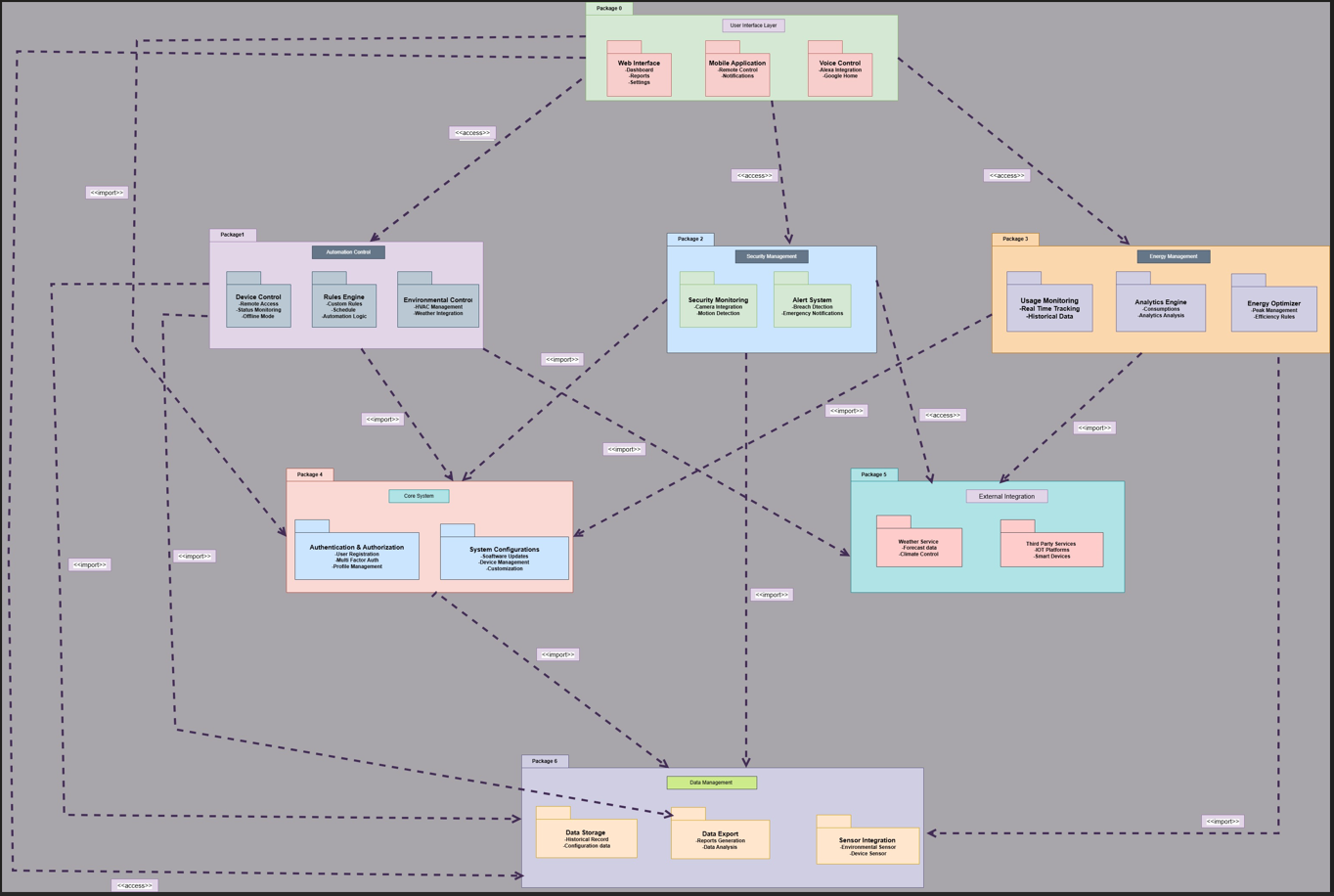
**2. Component Diagram**

* **Definition**: Represents the physical components (software, hardware, or files) and their interdependencies.
* **Purpose**: Used in system implementation to describe the structure of software components and their interactions.
* **Key Elements**:
  + Components (e.g., modules, libraries)
  + Interfaces
  + Dependencies
* **Example**: A component diagram for a web app may include components like the user interface, database, and API.



**3. Package Diagram**

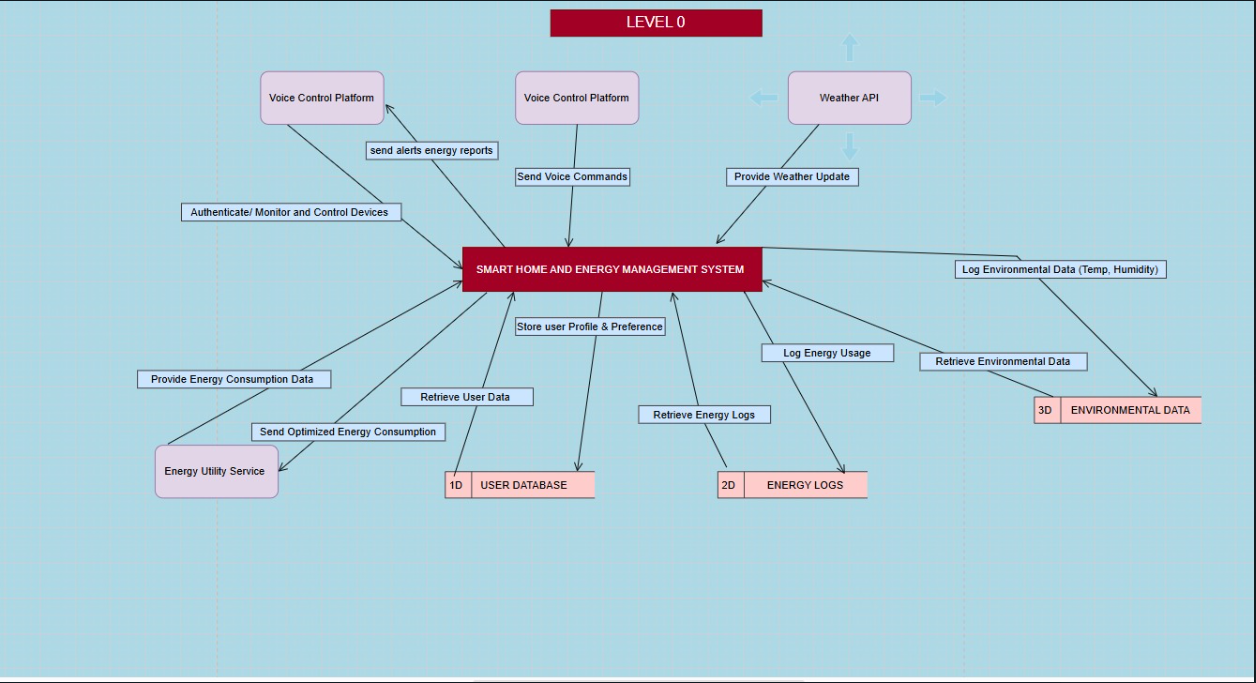
* **Definition**: Shows the organization and grouping of packages (logical containers) in a system.
* **Purpose**: Helps visualize the dependencies and modularization of a system.
* **Key Elements**:
  + Packages (containers for classes, components, etc.)
  + Dependencies between packages
* **Example**: A package diagram for a library system may include packages for "User Management," "Books," and "Borrowing History."



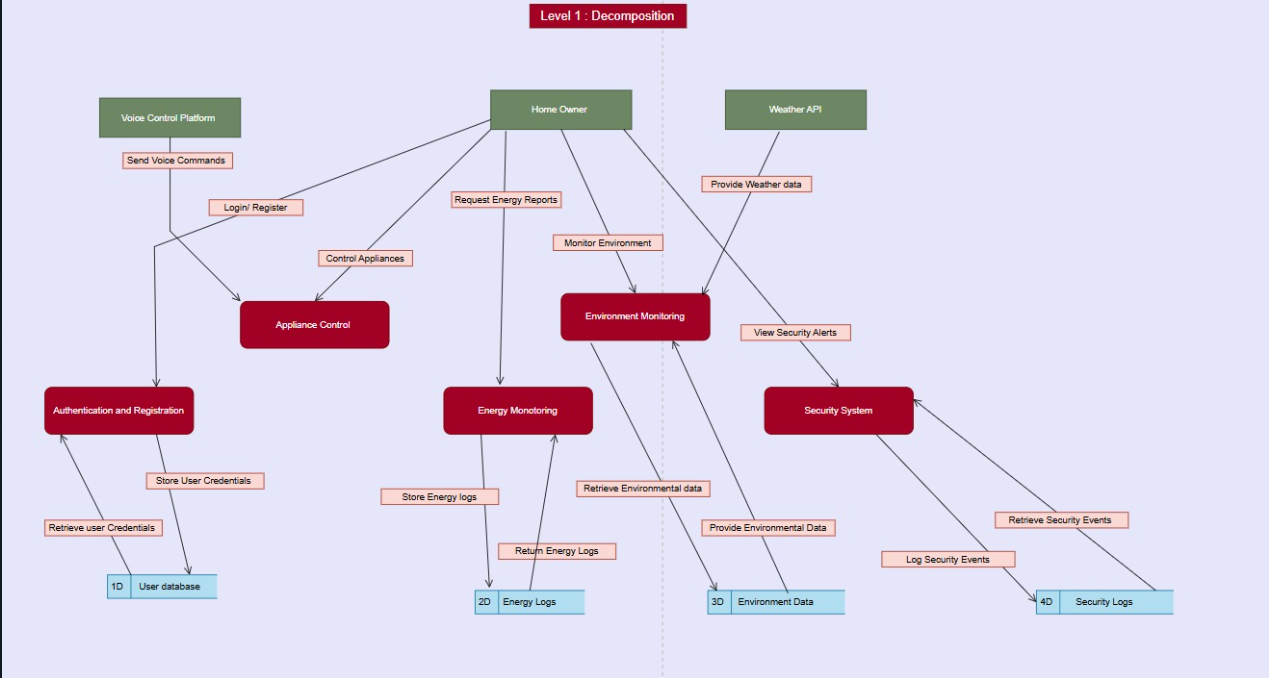
**4. Data Flow Diagram (DFD)**

* **Definition**: Illustrates the flow of data within a system, showing inputs, processes, and outputs.
* **Purpose**: Used to understand how data moves and is processed in a system.
* **Key Elements**:
  + Processes (circles)
  + Data stores (rectangles)
  + External entities (squares)
  + Data flows (arrows)
* **Example**: A DFD for a banking system may show how customer data flows from the user to the bank’s database through various processes.

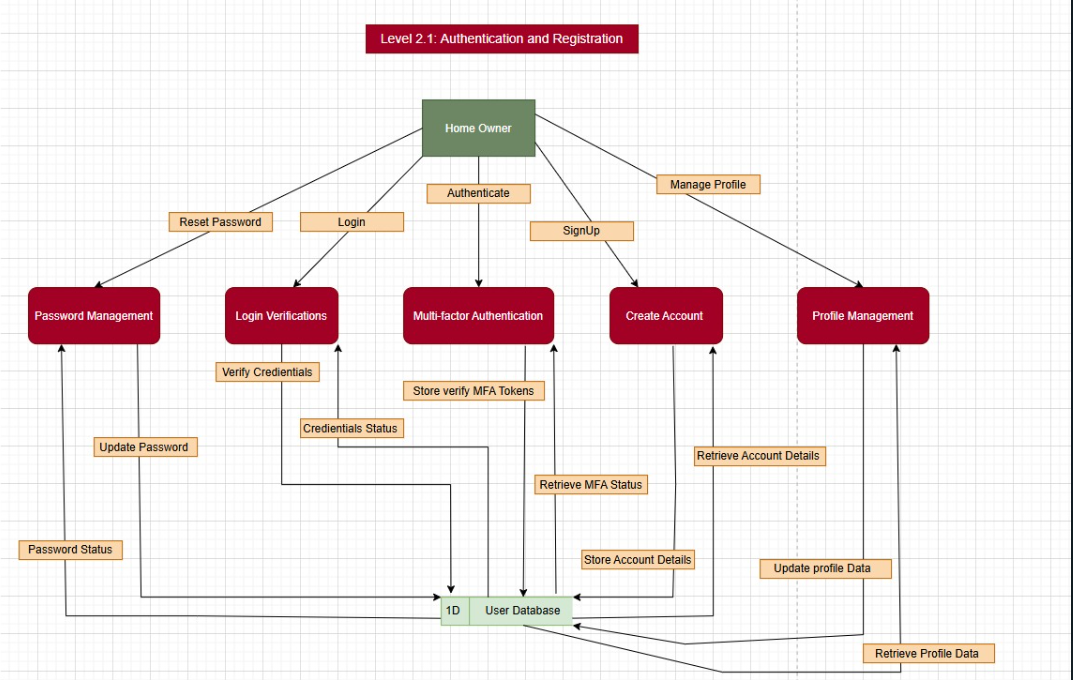
**Level 0:**

****

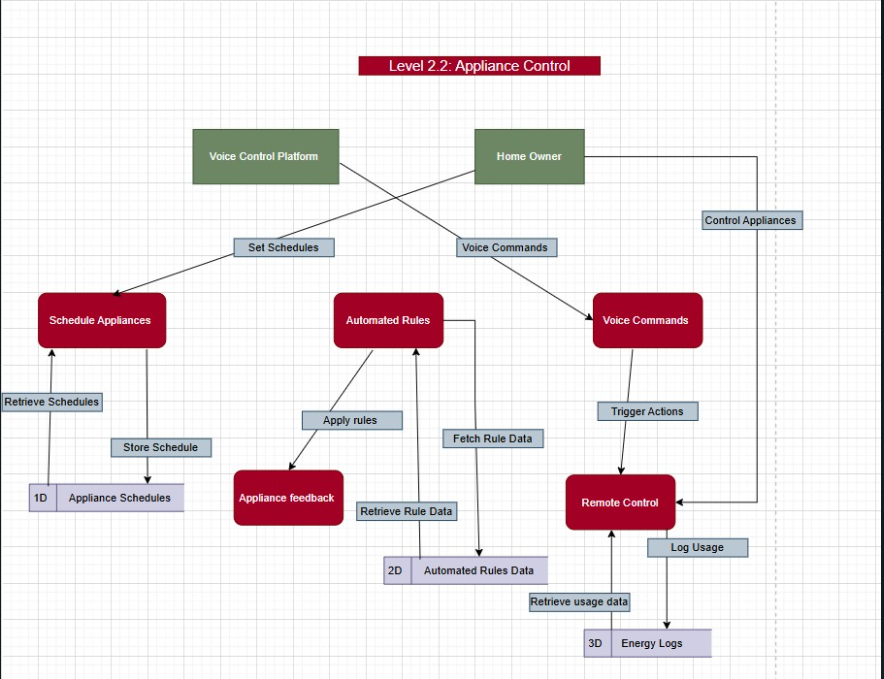
**Level 1:**

****

**LEVEL 2.1**

****

**Level 2.2**

****

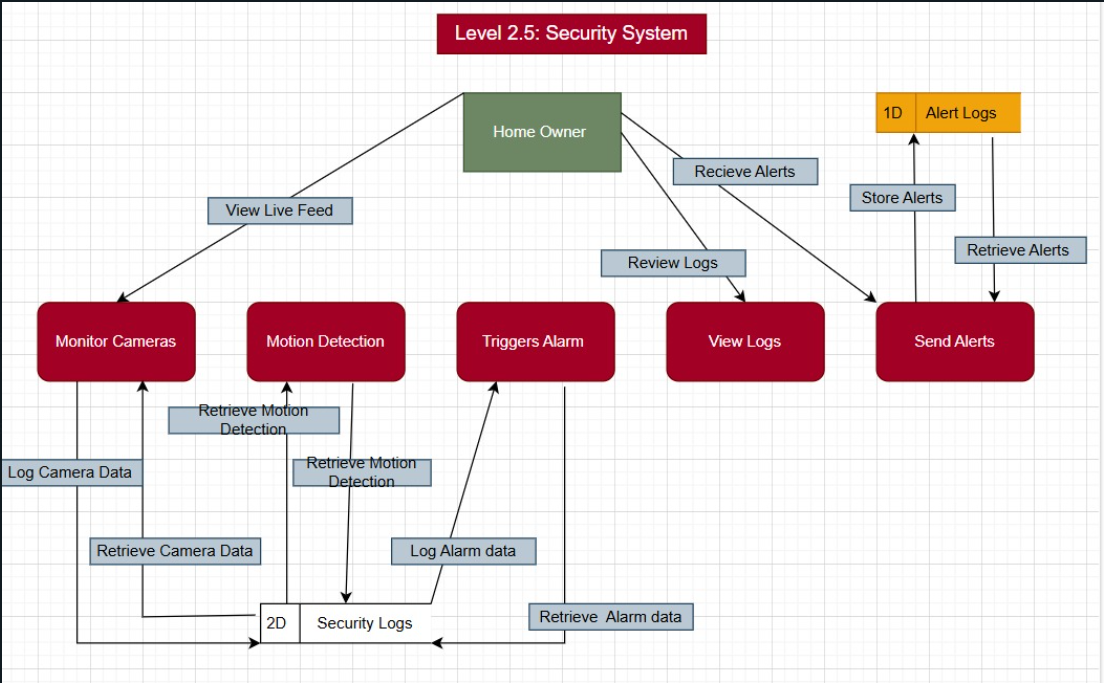
**Level 2.3**

****

**Level 2.4**

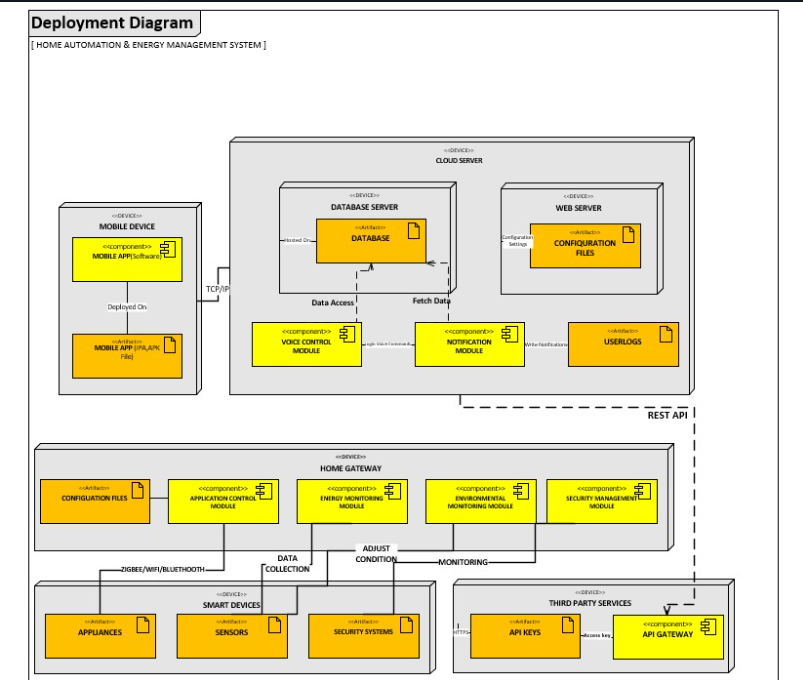
****

**Level 2.5**

****

**5. Deployment Diagram**

* **Definition**: Represents the physical deployment of artifacts (software components) on hardware nodes.
* **Purpose**: Describes the system's architecture in terms of hardware and software deployment.
* **Key Elements**:
  + Nodes (hardware)
  + Artifacts (software)
  + Connections (network links)
* **Example**: A deployment diagram for a cloud-based system may show the web server, application server, and database server deployed on different physical or virtual machines.



***QUESTION 7:***

1: Creational Pattern:

Singleton Pattern

o Used for the Energy Management Controller to ensure a single instance

manages energy optimization across the system.



A screen shot of a computer

Description automatically generated

2. Structural Patterns:

Adapter Pattern:

o Integrates external Smart Device APIs (e.g. third-party IOT devices) into the

system seamlessly, allowing flexibility in device compatibility.

### 

A screen shot of a computer

Description automatically generated

3. Behavioral Patterns:

Observer Pattern:

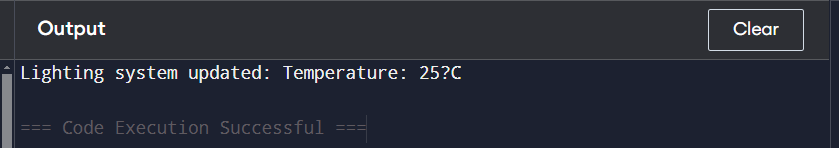
o Used for Environmental Sensors, notifying dependent components (e.g.

Lighting and Thermostats) about changes in temperature, humidity, or

occupancy.

A screenshot of a computer screen

Description automatically generated



**8. GRASP Patterns**

 Instructions:

 Analyze your existing class and sequence diagrams.

 Answer the following exercises by identifying design improvements or

explaining the application of GRASP principles.

 Write clear and concise responses for each exercise.

1. Creator  
**Task: Which component in your design creates smart device instances such as lights, thermostats, or cameras? Should this responsibility be assigned to the Home Automation Controller? If not, justify why and suggest a better alternative.**

**Analysis:**

The Home Automation Controller (HAC) is responsible for controlling appliances and devices, but creating instances of smart devices should not be its primary responsibility. According to the Creator pattern, it is better to assign this responsibility to a component that has knowledge of the devices and how they are instantiated.

**Recommendation:**

The Smart Device Factory (a dedicated factory class) should be responsible for creating instances of smart devices like lights, thermostats, and cameras. This approach adheres to SRP (Single Responsibility Principle) and helps in managing the creation of devices more efficiently.

**2. Information Expert**  
**Task: Does the Environmental Sensor Hub provide data like temperature, humidity, and occupancy? If not, identify which component currently provides this information and discuss whether it aligns with the Information Expert principle.**

**Analysis:**

The Environmental Sensor Hub is a suitable candidate to provide environmental data like temperature, humidity, and occupancy, as it interacts directly with the sensors. If another component, such as EMS (Energy Management System), is currently handling this, then it is not adhering fully to the Information Expert principle.

Recommendation:

The Environmental Sensor Hub should be the Information Expert for providing environmental data. It should be the primary source of data, making the system more cohesive and minimizing unnecessary dependencies between components.

**3. Controller**  
**Task: Evaluate your sequence diagram for the Energy Management Controller. Does it handle all energy-related events such as adjusting power usage and scheduling appliances? If not, suggest how its role can be refined.**

**Analysis:**

The Energy Management Controller in the sequence diagram might be handling energy-related events, but it should also manage the scheduling and optimization of appliance use based on both user preferences and real-time environmental data.  
**Recommendation:**

The Energy Management Controller should refine its role to include:  
Adjusting power usage dynamically. Scheduling appliances based on user input and environmental conditions. Communicating with HAC to ensure appliances are turned on/off as needed.

**4. Low Coupling**  
**Task: Does your design use a Smart Device Adapter to abstract interactions with external smart device APIs? If not, suggest how you would implement an adapter to reduce coupling between the system and external devices.**

**Analysis:**

The design may not yet include a Smart Device Adapter. Without such an adapter, your system is tightly coupled to the specific smart device APIs, making it difficult to add new devices or change APIs.

**Recommendation**:

Implement a Smart Device Adapter that abstracts interactions with external smart device APIs (e.g., for lighting, thermostats). This adapter will allow your system to interact with any device without modifying other components. The Adapter pattern reduces coupling by isolating external APIs from the core system.

**5. High Cohesion**  
Task: In your class diagram, is the Lighting Controller focused solely on managing lighting tasks? If it has additional unrelated responsibilities, propose how these could be reassigned to other components.

**Analysis:**

The Lighting Controller should focus only on managing lighting-related tasks (e.g., turning lights on/off, adjusting brightness). If it has other responsibilities, such as managing temperature or security tasks, these should be moved to more appropriate components.

**Recommendation:**

Separate concerns by introducing dedicated classes like a TemperatureController or SecurityController. The LightingController should only manage lighting, ensuring high cohesion by making the class focused on a single responsibility.

**6. Polymorphism**  
**Task: Are common actions for devices (e.g., turning on/off, adjusting settings) implemented polymorphically? If not, suggest how a common interface or abstract class could simplify the design.**

Analysis:

**If devices (lights, thermostats, cameras) implement common actions like turning on/off or adjusting settings, they might currently be implemented separately for each device type.**

Recommendation:

Introduce an abstract class or interface (e.g., SmartDevice) that defines common methods like turnOn(), turnOff(), and adjustSettings(). Each device (light, thermostat) should then implement these actions polymorphically, allowing your system to handle any device in a unified way.

**7. Pure Fabrication**  
**Task: Is there a standalone Notification Manager in your design to handle alerts and notifications? If not, propose adding one and explain how it improves modularity and reduces the burden on other components.**

Analysis:

If the Notification Manager is not present, the task of handling alerts (e.g., notifications for security breaches, energy usage alerts) might be distributed across other components like Security and EMS.

**Recommendation:**

Introduce a Notification Manager component that handles all alerts and notifications. This will improve modularity, reduce the burden on other components (such as Security or EMS), and centralize the logic for sending notifications (e.g., through email, SMS, or push notifications).

**8. Indirection**  
**Task: Does your design include an Event Mediator to handle communication between sensors and controllers? If not, suggest how introducing an Event Mediator can decouple these components and make the system more scalable.**

**Analysis:**

Currently, sensors might directly trigger actions in controllers, creating tight coupling between them. The Event Mediator pattern could be useful for decoupling these components.

**Recommendation:**

Introduce an Event Mediator component to handle communication between sensors (e.g., temperature sensor) and controllers (e.g., HAC, EMS). This mediator would listen for sensor data and notify the appropriate controllers, improving scalability and modularity.  
Combining Adaptive Automation and Energy Conservation  
Adaptive automation and energy conservation will work together by utilizing both user preferences and environmental factors (from sensors and weather data) to create an effective system. Here’s how they will integrate:  
  
**User Preferences:**  
  
Users will set schedules for devices (e.g., lights, HVAC) through the UI and Scheduler.  
The ProfileManager will store personalized settings for different family members, allowing customization of device behavior.  
Environmental Factors:  
  
The EMS will receive real-time data from the Environmental Sensors (temperature, humidity, occupancy) and adjust energy consumption accordingly.  
WeatherService will provide external weather data to adjust HVAC settings based on outdoor conditions.

**Energy Optimization:**  
  
The Energy Management System (EMS) will use both user schedules and environmental data to optimize energy usage, ensuring energy-efficient operation.  
During peak energy consumption times or when the house is empty (detected by occupancy sensors), devices will be turned off or adjusted to save energy.  
**Automation:**  
  
HAC will ensure that appliances like lights, thermostats, and fans are turned off or adjusted based on environmental data and schedules.  
VoiceControl integration will allow users to control devices using voice commands for even more convenient automation.  
By combining these factors, the system can provide an intelligent, responsive home environment that adapts to both user preferences and real-time environmental conditions while optimizing energy consumption.