Smart Home Simulator

SOEN 343

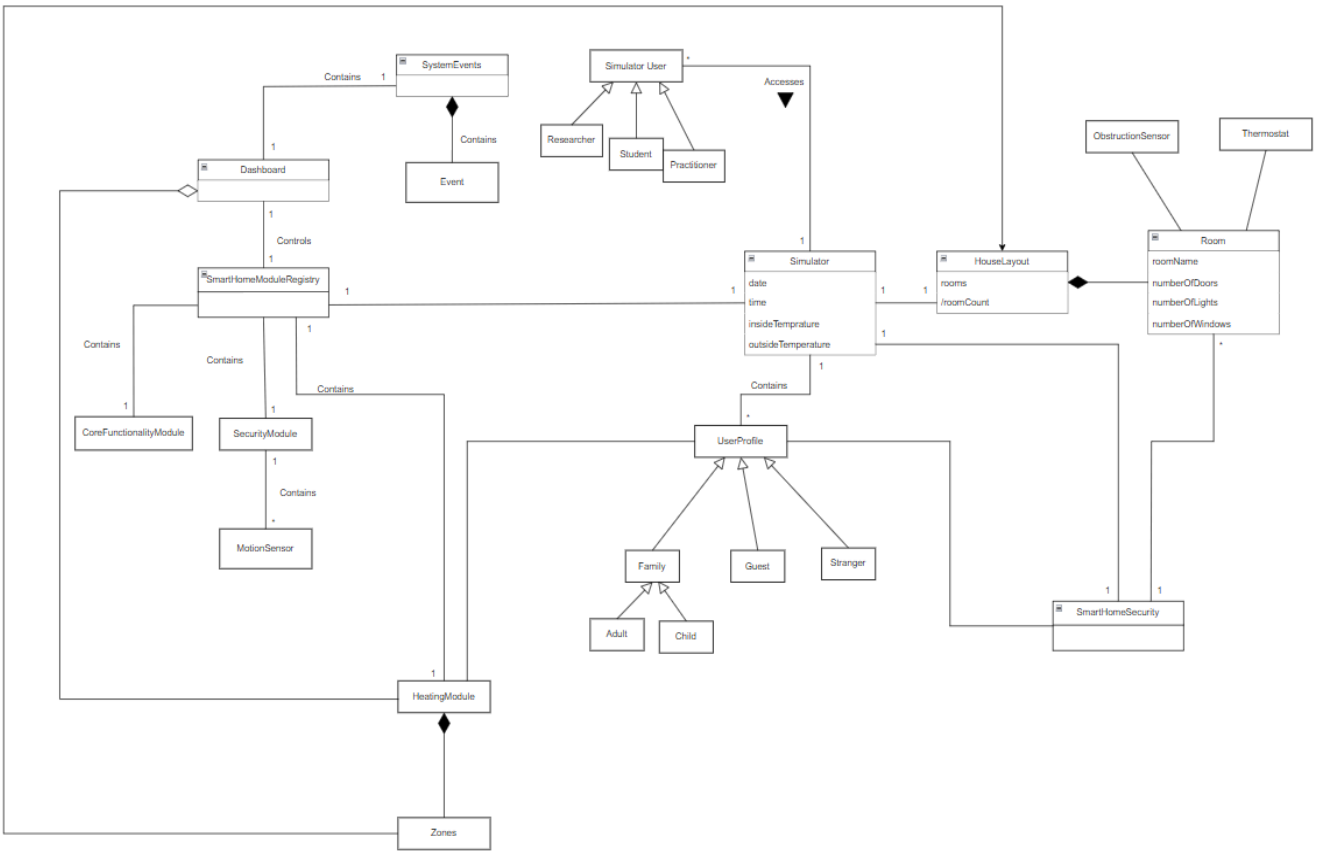
Lab Section WJ-X

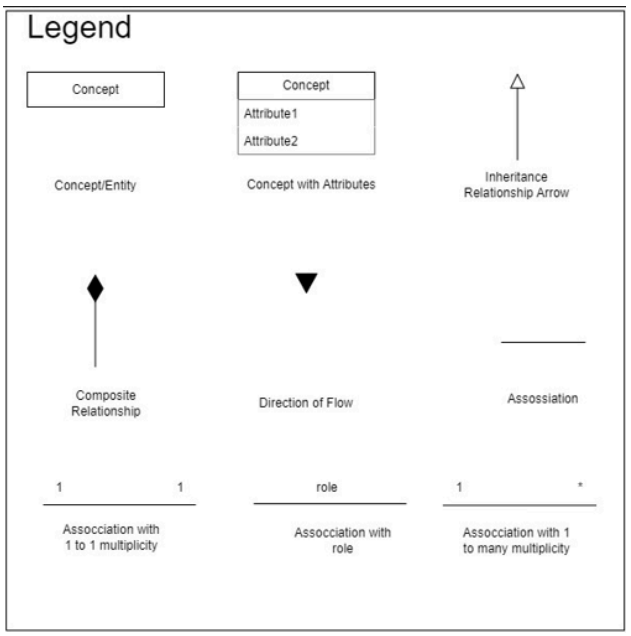
April 9th 2024

Clara Gagnon (40208598)(Team Leader) Role: Smart Home Security Implementation, Design Patterns
Ahmad Elmahallawy (40193418) Role: System Architecture, Smart Home Heater Implementation, General Implementation
Gulnoor Kaur (40114998) Role: SHP Implementation, Domain Diagram
Liam Daigle (40207583) Role: Smart Home Dashboard Implementation, SHP Implementation
Vanessa DiPietrantonio (40189938) Role: Simulation Parameters Implementation, Smart Home Implementation
Jessica Beauchemin (40188873) Role: Use Case Model: Use Case Tables, Use Case Diagram, Sequence Diagrams, State Machine Diagrams, Activity Diagrams

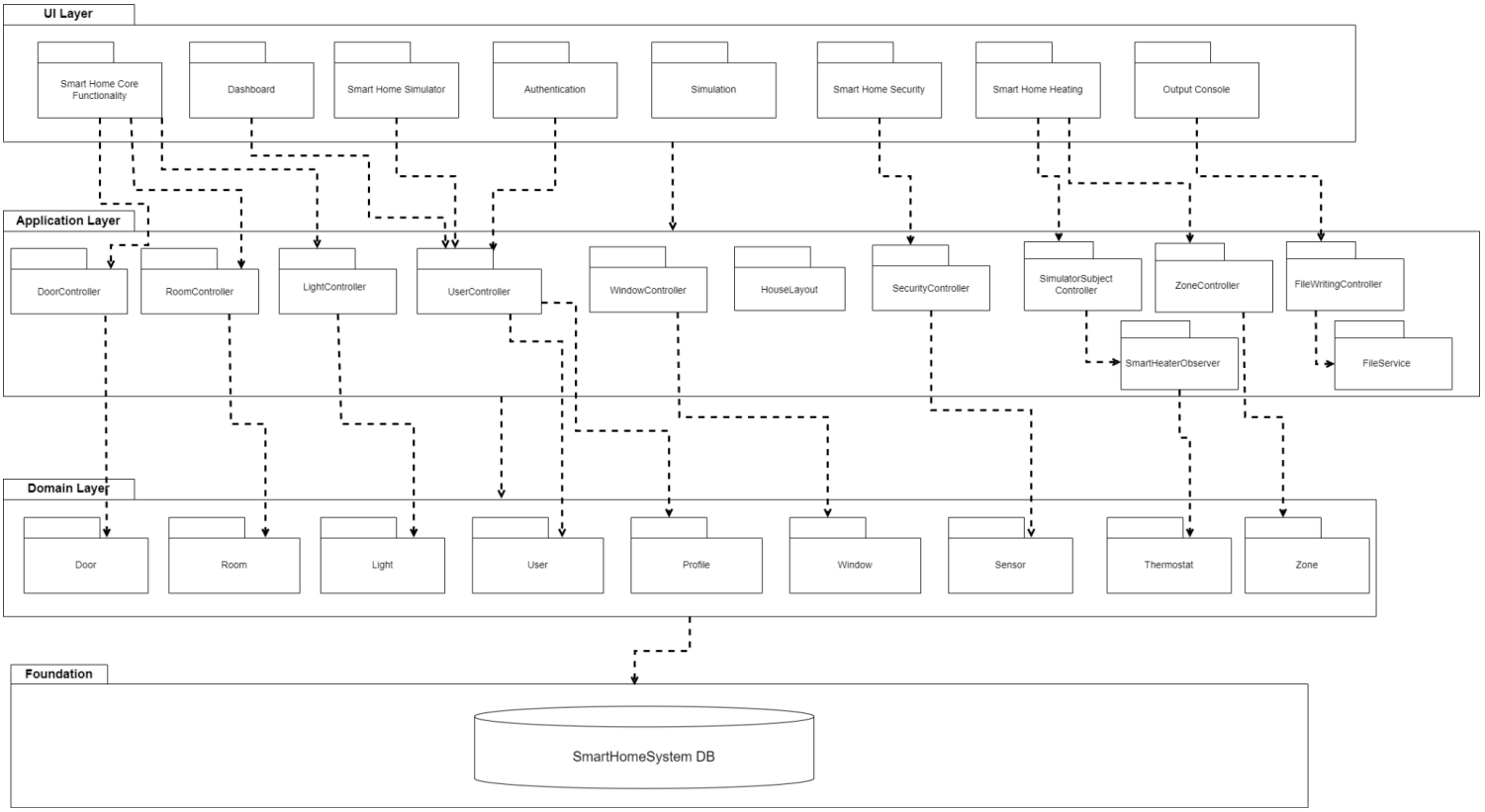
Table of Content

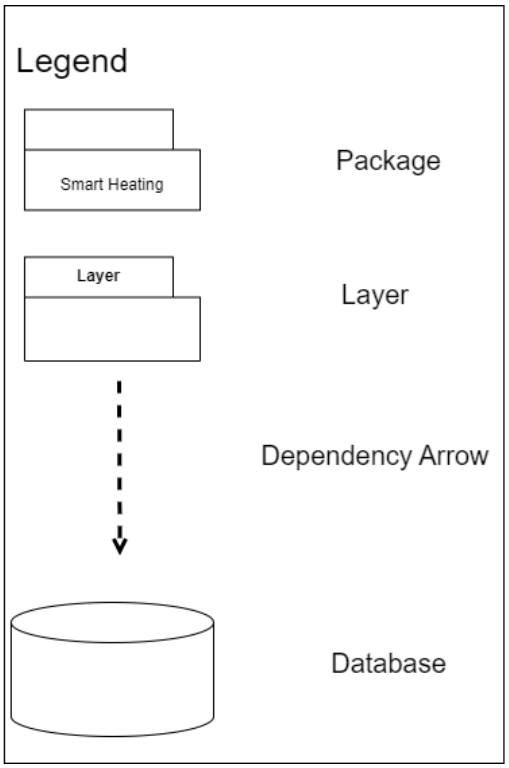
1. Domain Model Diagram





2. System Architecture





UI (View)

Description: The UI layer in our smart home simulation system, built with React frontend, encompasses the visual interface through which users interact with the simulated environment. It provides a user-friendly dashboard for controlling devices, monitoring the home's status, and adjusting simulation parameters. More specifically, the UI layer consists of frontend folders for the Smart Home Core, Dashboard, Smart Home Simulator, Smart Home Security and Smart Heating.

Application (Controller)

Description: The application layer, implemented with Spring Boot backend, serves as the backbone of the smart home simulation system. It manages user requests, coordinates interactions between the UI and domain layers, and executes business logic to simulate home behaviors and control devices. More specifically, the application layer comprises all the controllers for door, room, light, user, window, security and heating.

Domain (Model)

Description: The Domain layer serves as the backbone of our smart home system. It consists of the essential business logic that drives our operations. This includes managing data storage and retrieval for user preferences, device configurations, and simulation parameters in an efficient manner. Additionally, it enforces critical business rules and constraints, such as device usage limits and user permissions. Moreover, the Domain layer hosts the simulation engine, which governs the behavior of our simulated smart home environment over time and in response to user interactions.

Foundation

Description: The foundation layer represents the underlying infrastructure and core components upon which the rest of the system is built. This layer provides fundamental functionalities and services that support higher-level modules and applications. The main purpose of the foundation layer is to abstract away complexities, provide essential services, and ensure the stability, scalability, and maintainability of the entire system. It includes components responsible for managing data persistence, such as database management systems (DBMS).

3. Use Case Model

3.1 Use Case Tables - Smart Home Security

ID:	**UC – 7.1**
Title:	User can Activate/Deactivate Away Mode
Description:	The System shall allow users to set an *away* mode when they leave the property.
Primary Actor:	System, User
Preconditions:	Simulation is active. Away mode exists and is open for modification.
Postconditions:	Away mode is activated or deactivated.
Inputs:	Away mode toggle ON/OFF.
Outputs:	Away mode activation confirmation.
Main Success Scenario:	1. User opens the SHP tab on the Smart Home Simulator. 2. Systems displays the option to toggle the away mode. 3. User toggles the away mode (ON or OFF). 4. System activates or deactivates the away mode depending on the user's selection. 5. System initiates UC-7.2 – Activation of Away Mode

ID:	**UC – 7.2**
Title:	Away Mode: Activation
Description:	The System shall close all doors and windows when the away mode is activated.
Primary Actor:	System
Preconditions:	Away mode is activated by user. No windows are blocked by any arbitrary object.
Postconditions:	All doors and windows (of every room) are closed.
Inputs:	Activation of away mode.
Outputs:	Confirmation of closing of all doors and windows.
Main Success Scenario:	1. System notices the activation of the away mode. 2. System proceeds to close all doors and windows, verifying for each that there is no obstruction before closing. 3. System confirms successful closing of all doors and windows. 4. System confirms successful activation of away mode.

ID:	**UC – 7.3**
Title:	Away Mode: Motion Detection Notification
Description:	The System shall send notifications to users if any motion detectors are triggered while the user is away.
Primary Actor:	System
Preconditions:	Away mode is active. Motion detectors are active. A motion detector is triggered.
Postconditions:	Notification is sent to the user. Motion detector timer is initiated.
Inputs:	Motion detector is triggered.
Outputs:	Motion detection notification is sent to user.
Main Success Scenario:	1. A motion detector is triggered while the away mode is active. 2. System detects the motion and initiates notification process. 3. System sends a notification to the user indicating the activation of the motion detector. 4. System initiates a timer to track the time since the motion is detected.

ID:	**UC – 7.4**
Title:	Away Mode: Open Window and Door Notification
Description:	On away mode, the System shall send notifications to users if any doors or windows are opened.
Primary Actor:	System
Preconditions:	Away mode is active. A window or door is opened.
Postconditions:	User is notified of the open door or window.
Inputs:	System is notified of an open door or window.
Outputs:	A notification is sent to the user that there is an open door or window.
Main Success Scenario:	1. A window or door is opened while the away mode is active. 2. System detects the open door or window. 3. System initiates the notification process. 4. System sends a notification to the user indicating the open door or window.

ID:	**UC – 7.5**
Title:	Away Mode: Authority Notification Settings
Description:	The System shall allow users to set how much time should pass before alerting the authorities when motion detectors are triggered during set away mode.
Primary Actor:	System, User
Preconditions:	The user is a parent and has access to SHP settings. The system provides functionality for modifying the motion detector timer to alert the authorities. The motion detector timer element exists and is open for modification.
Postconditions:	The user's preferred time before alerting the authorities in case of motion detection is set and saved in the system.
Inputs:	User's specified time for authority notification.
Outputs:	System confirmation of successful update of authority notification settings.
Main Success Scenario:	1. User navigates to the away mode settings in the SHP tab. 2. Systems presents the option for setting the time duration before alerting the authorities. 3. User selects their preferred time. 4. System saves the user's input. 5. System confirms the successful update of the authority notification settings to the user.

ID:	**UC – 7.6**
Title:	Away Mode: 15˚C Temperature Rise
Description:	If the temperature inside a room goes up 15 degrees Celsius in 1 minute the *away* mode is turned off automatically and the users are notified.
Primary Actor:	System
Preconditions:	The away mode is active. System monitors the room temperatures and the time. System detects an increase in temperature of 15˚C in the span of 1 minute.
Postconditions:	Away mode is turned off. User receives a notification from the system.
Inputs:	Detection by system of a 15˚C increase in the span of 1 minute.
Outputs:	Away mode is turned off. System notifies user of significant rise of temperature and deactivation of away mode.
Main Success Scenario:	1. System continuously monitors room temperatures and elapsed time while away mode is active. 2. System detects a rapid increase in temperature of 15˚C within a minute. 3. System deactivates the away mode. 4. System sends a notification to the user informing them of deactivation of away mode due to the significant temperature rise.

ID:	**UC – 7.7**
Title:	Away Mode: Room Temperature Above 35˚C
Description:	If the temperature inside a room reaches 35 degrees Celsius, the *away* mode is turned off automatically and the users are notified.
Primary Actor:	System
Preconditions:	The away mode is active. System monitors the room temperatures. System detects that the temperature inside a room reaches 135˚C.
Postconditions:	Away mode is turned off. User receives a notification from the system.
Inputs:	Detection by system of room temperature reaching 135˚C.
Outputs:	Away mode is turned off. System notifies user of high room temperature and deactivation of away mode.
Main Success Scenario:	1. System continuously monitors room temperature while the away mode is active. 2. System detects that the temperature inside a room exceeds 135˚C. 3. System deactivates the away mode. 4. System sends a notification to the user informing them of deactivation of away mode due to high room temperature.

ID:	**UC – 7.8**
Title:	Away Mode: Authority Notification
Description:	The System shall alert the authorities after the set amount of time when motion detectors are triggered during set away mode.
Primary Actor:	System
Preconditions:	The away mode is active. The motion detectors are triggered. The motion detector timer is running and has reached the set amount of time by user.
Postconditions:	Authorities are notified. User is notified of motion detection and authority notification. Motion detection timer is stopped.
Inputs:	Motion detection timer has reached the specified amount of time.
Outputs:	A notification is sent to the user and authorities are contacted.
Main Success Scenario:	1. System detects that the motion detector timer has reached the specified time. 2. System initiates authority alerting procedure. 3. System notifies the authorities. 4. System sends notification to user.

ID:	**UC – 7.9**
Title:	Motion Detector Location Settings
Description:	The System shall allow the user to set the location of the motion detectors.
Primary Actor:	System, User
Preconditions:	The user is a parent. The system allows user to access the SHP settings. The room exists. The room does not already have a motion detector associated with it.
Postconditions:	Motion detector element(s) is created. Room – motion detector association is created.
Inputs:	User selection of room in which to add the motion detector.
Outputs:	Motion detector is assigned to a room.
Main Success Scenario:	1. User accesses the SHP tab. 2. User selects the motion detector management settings. 3. System presents a list of available rooms in which to add a motion detector. 4. User selects the desired room. 5. System creates a new motion detector element. 6. System associates the motion detect with the chosen room. 7. Confirmation of successful addition is sent to the user.

3.2 Use Case Diagram for the Whole System

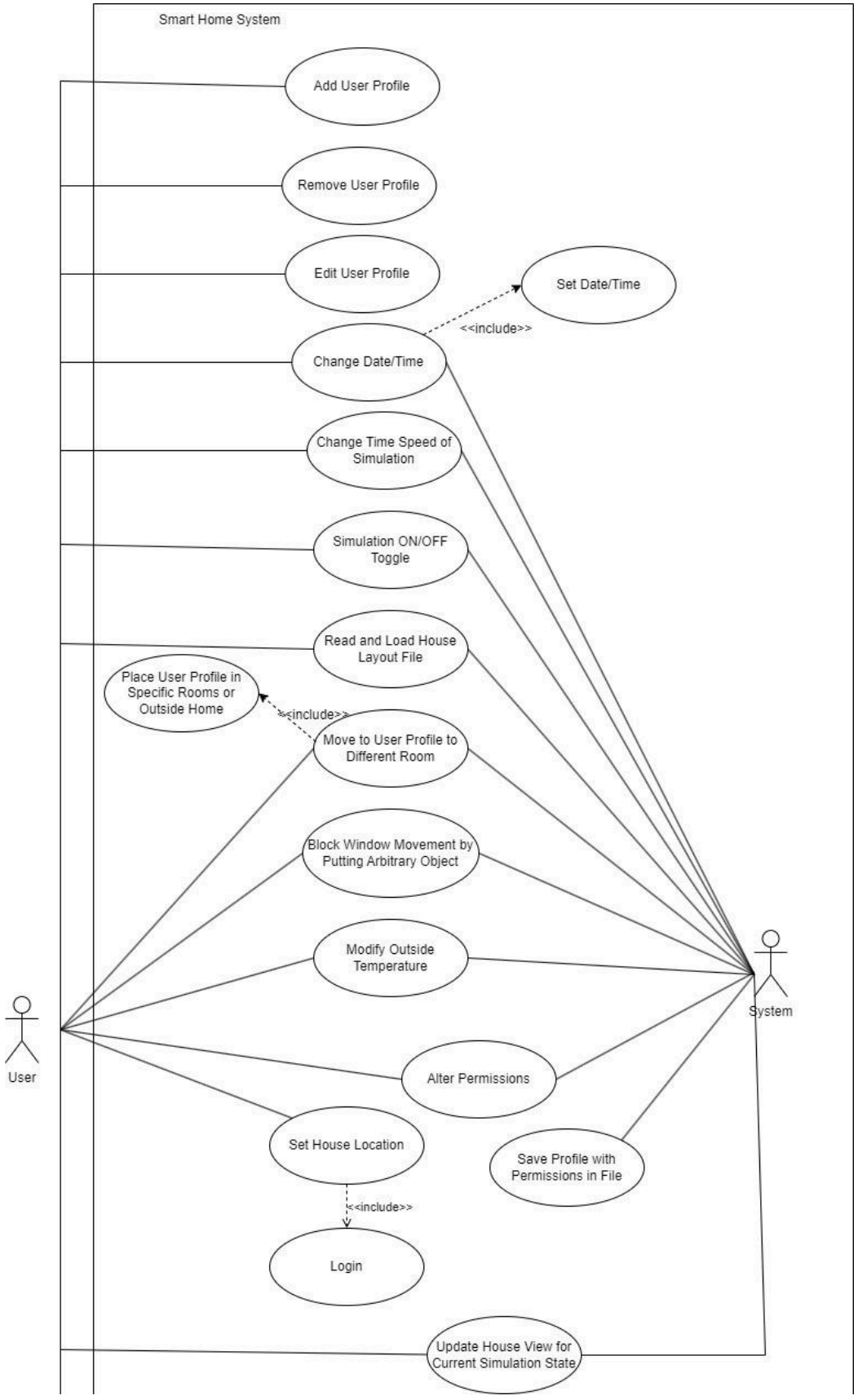


Figure 1: Use Case Diagram (figure continued on next page)

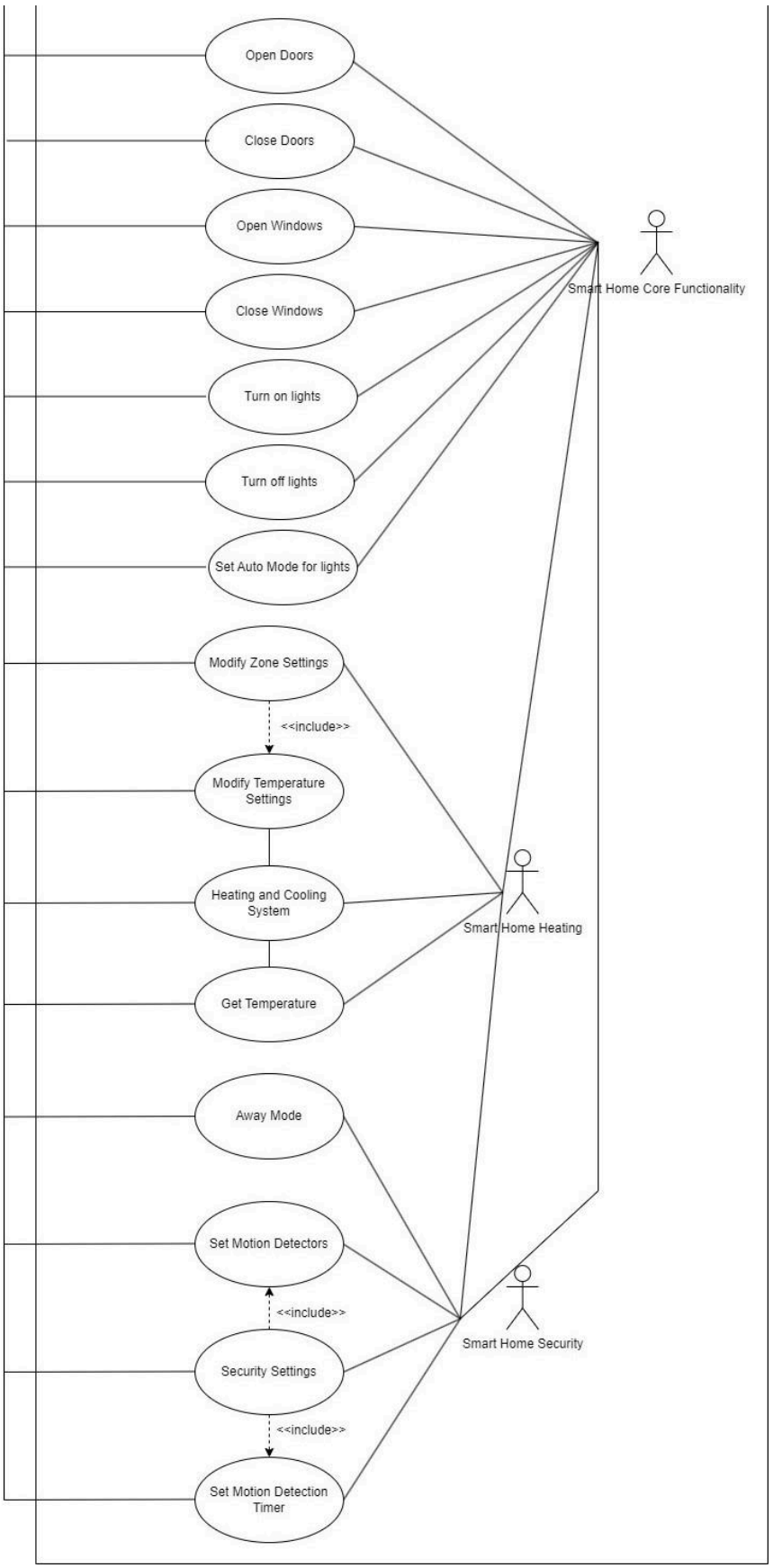
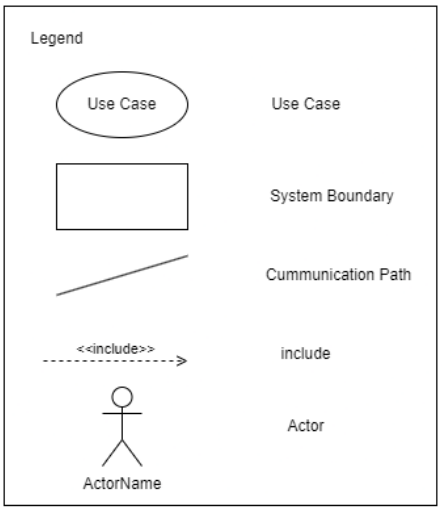


Figure 1: Use Case Diagram (continued)



3.3 Sequence Diagram for the Smart Home Core Functionality (SHC) Module

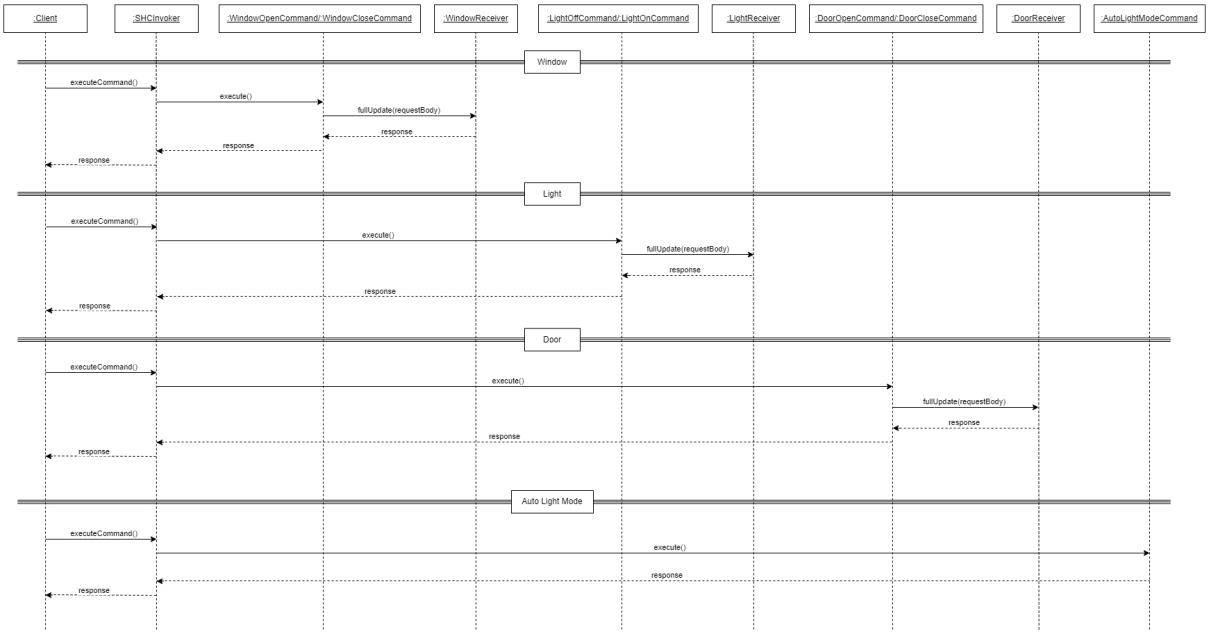


Figure 2: Sequence Diagram

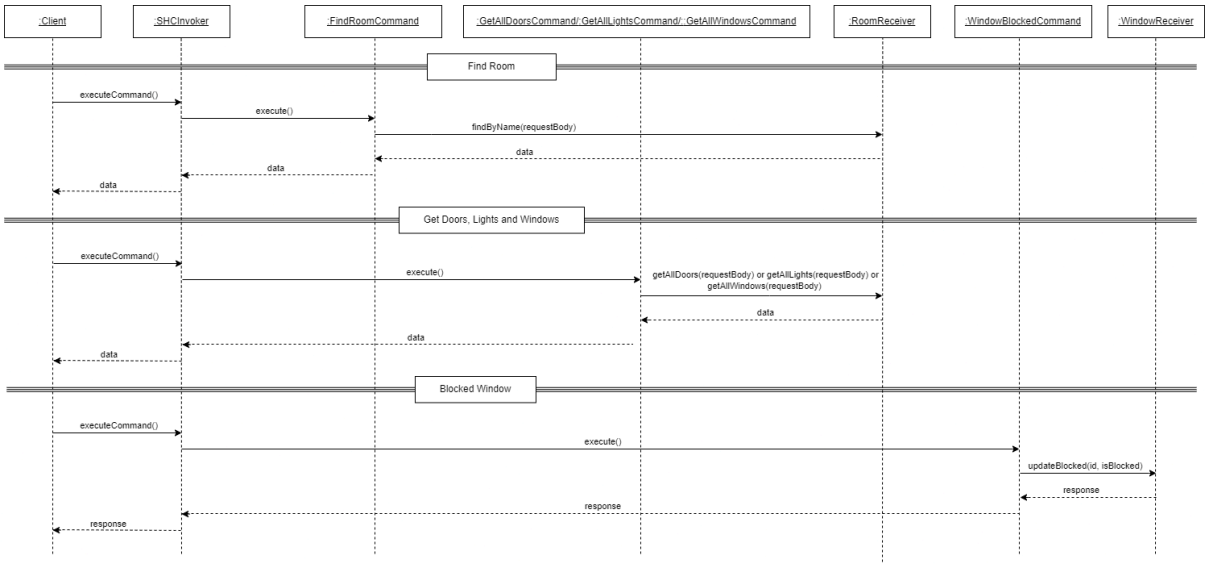


Figure 3: Note that this is an extension of Figure 2

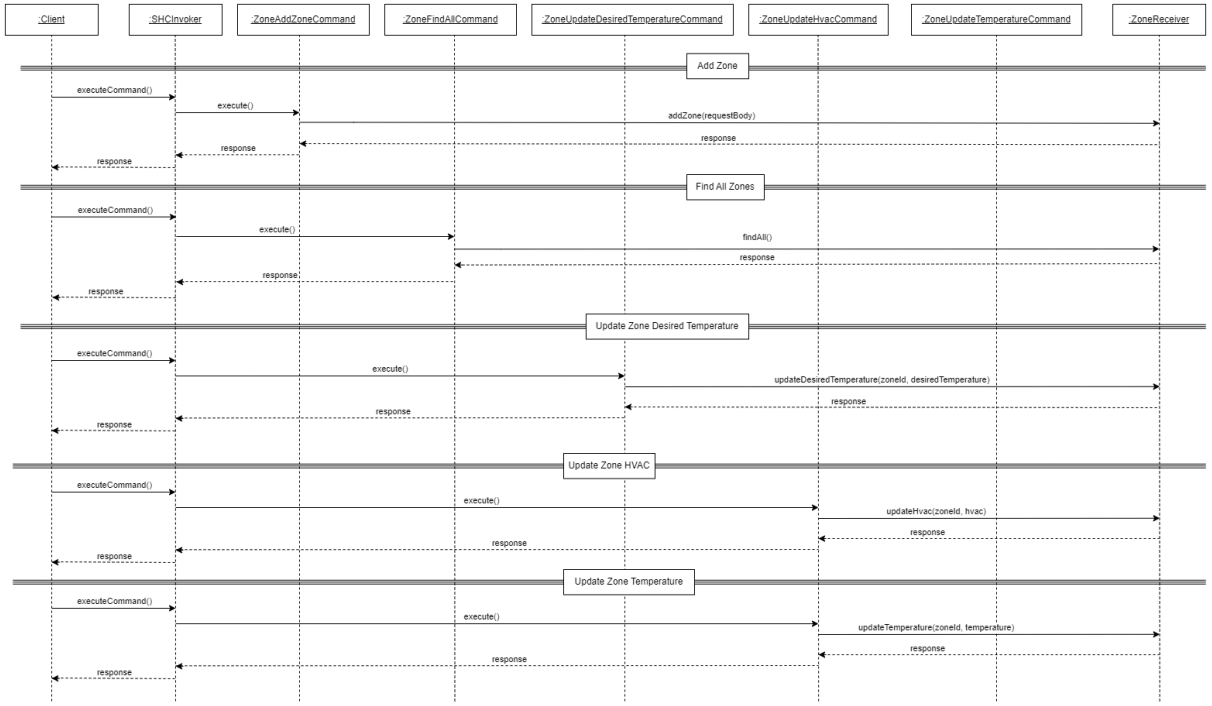
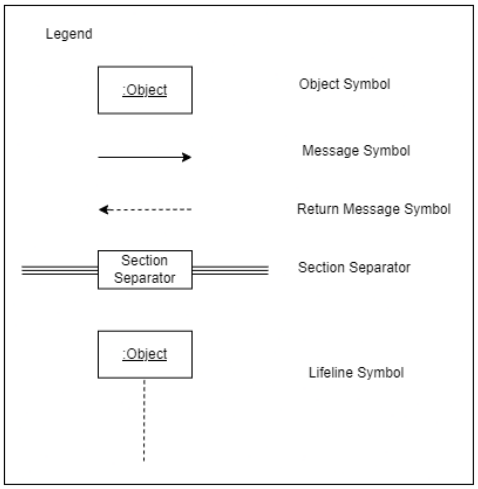


Figure 4: Note that this includes SHH in Sequence Diagram (extension of Figure 2 and 3)



3.4 State-Machine Diagram for the Context of the Simulation

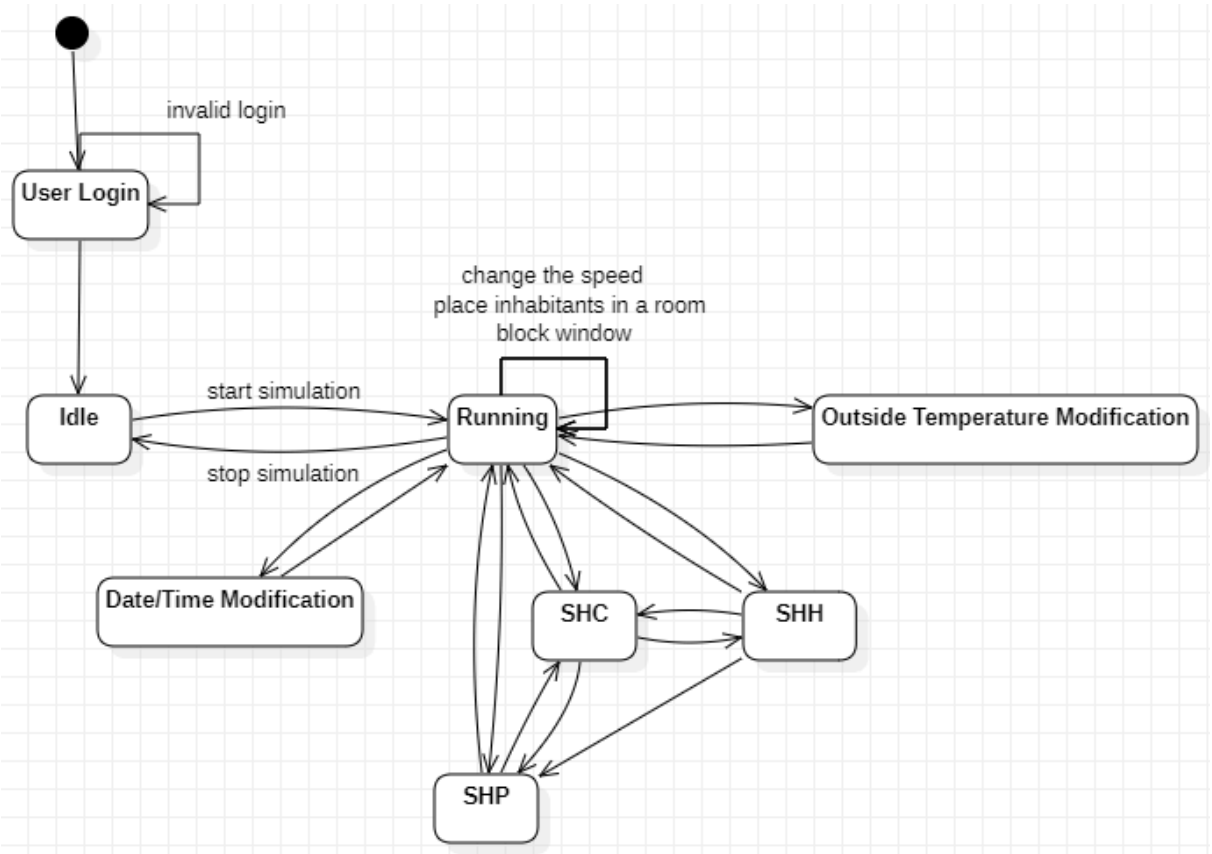
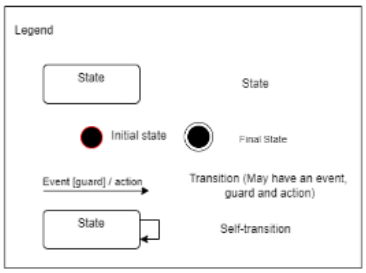


Figure 5: State Machine Diagram - Context of Simulation



3.5 Activity Diagram for the Context of the Simulation

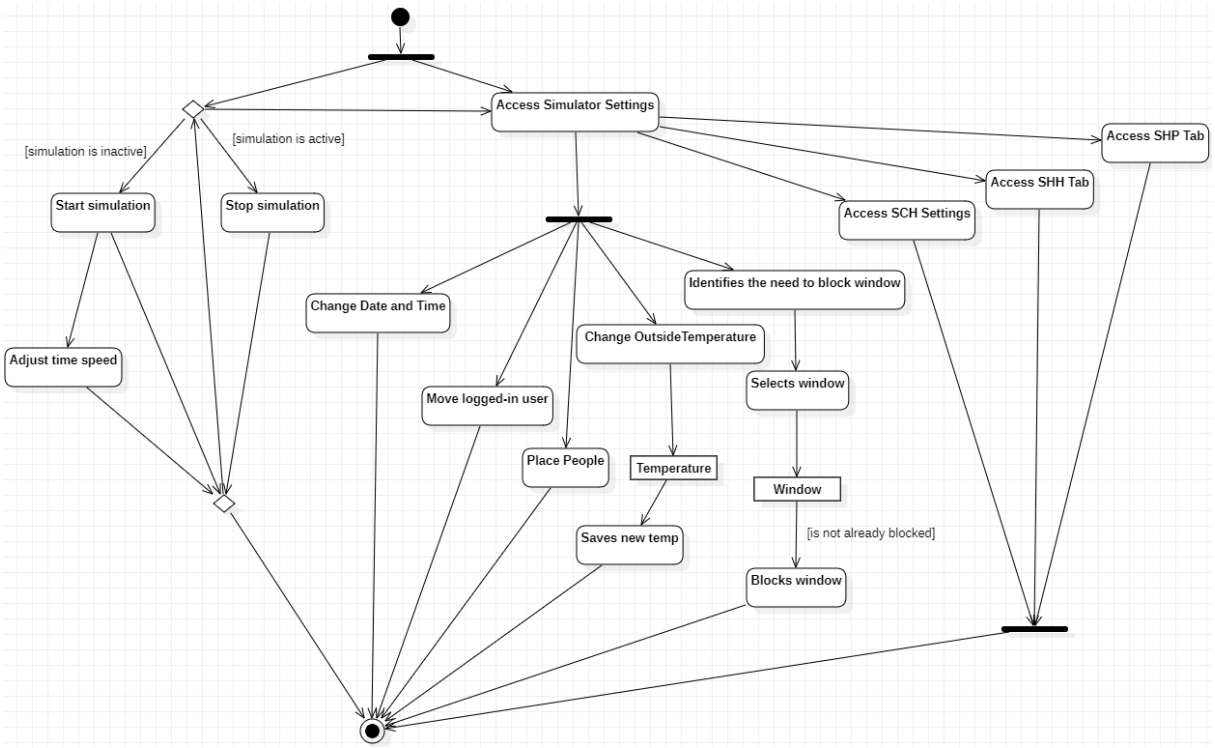


Figure 6: Activity Diagram for Context of the Simulation

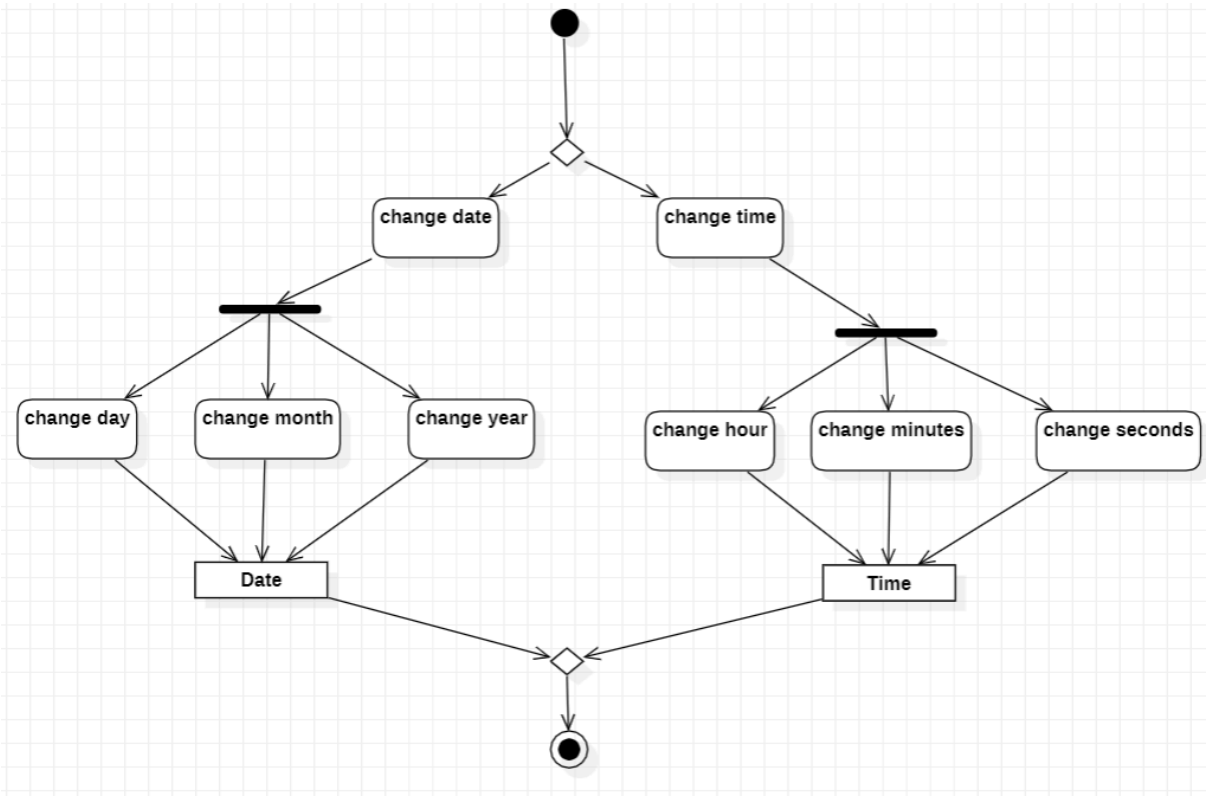


Figure 7: Activity Diagram for the Sub-Activity Change Date and Time

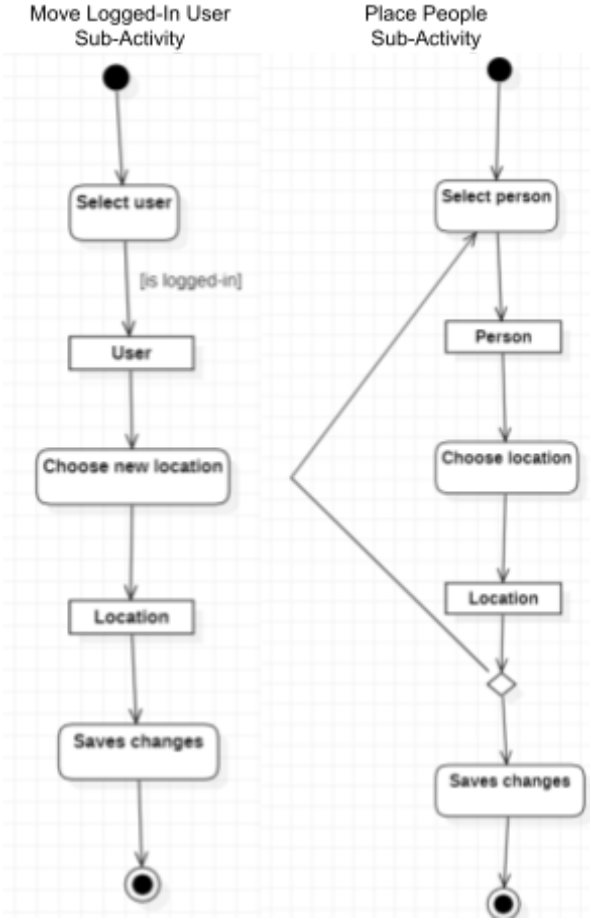


Figure 8: Move and Place People Sub-Activity

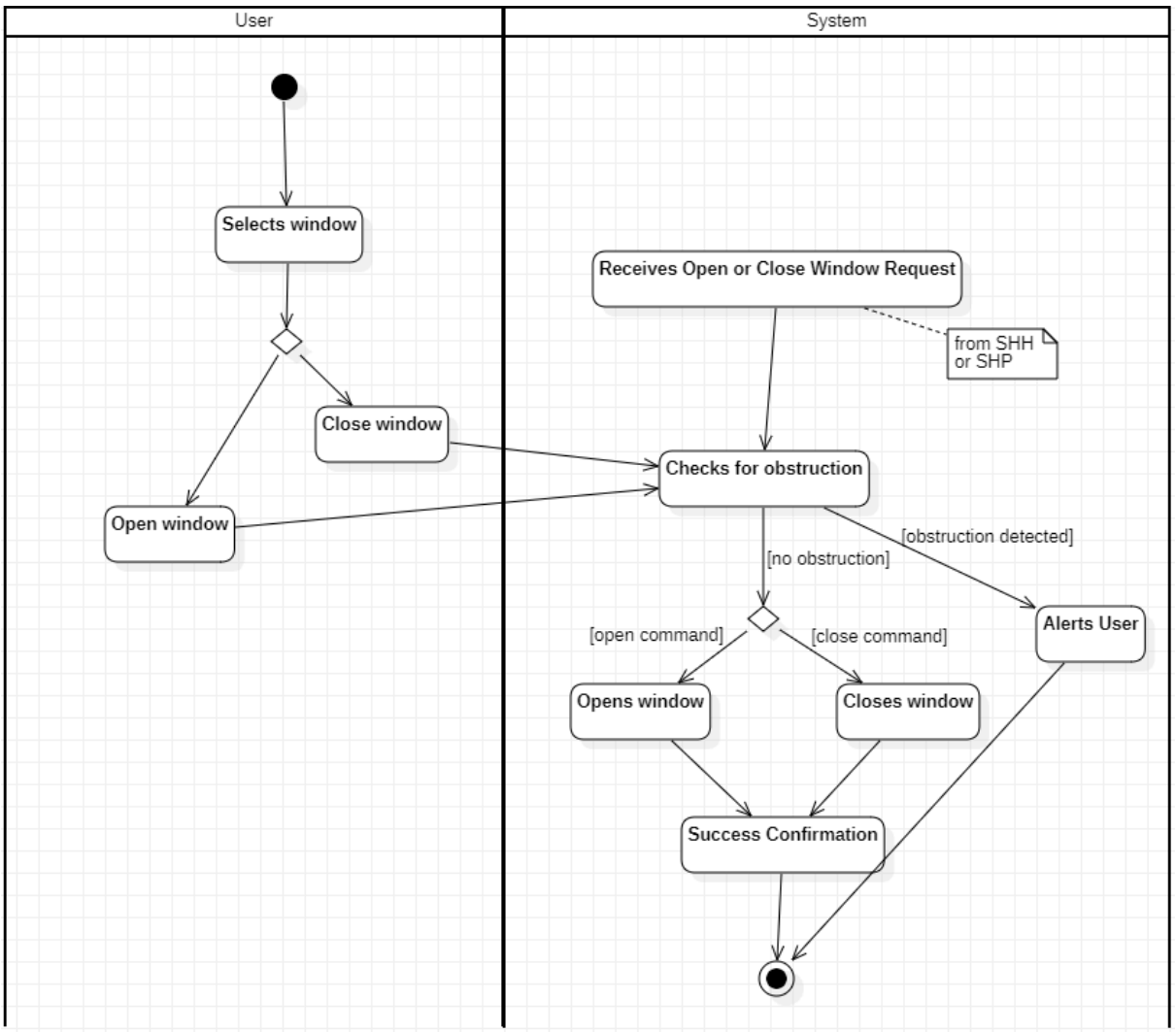
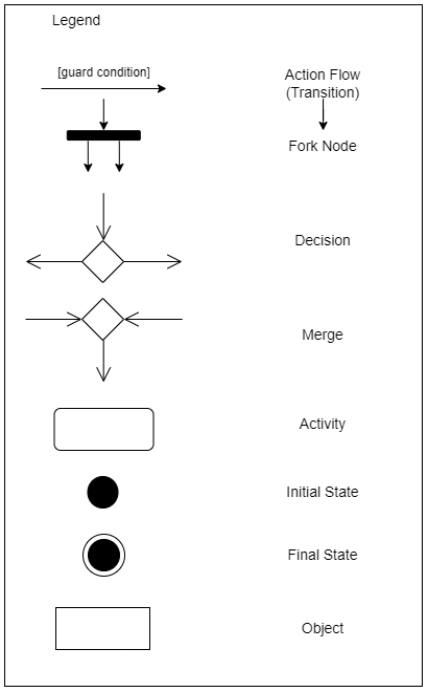


Figure 9: Window Obstruction SHC Module Sub-Activity (updated)



3.6 State-Machine Diagram for the Smart Home Security Module

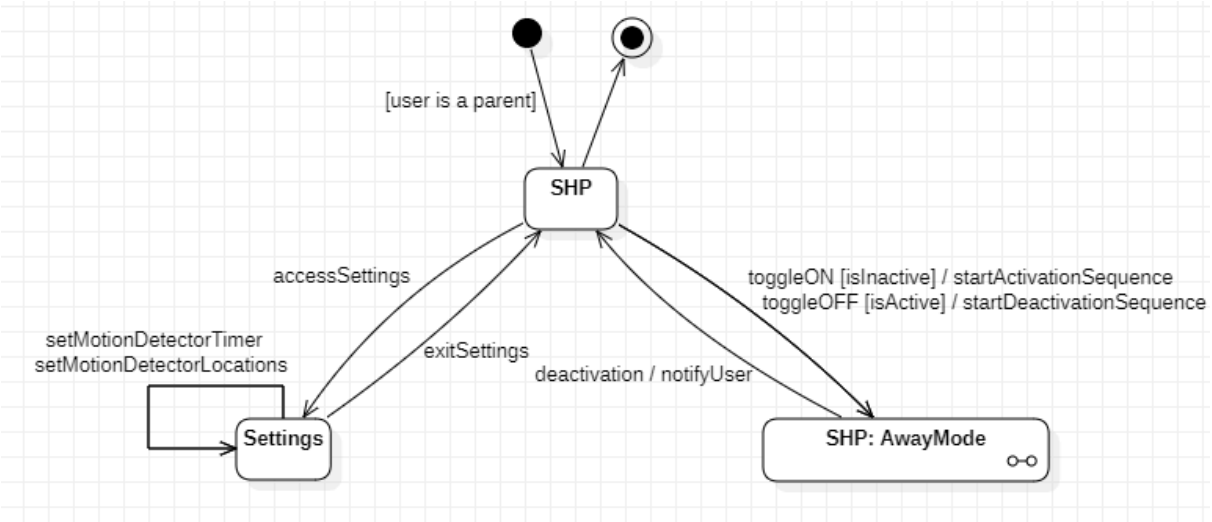


Figure 10: Smart Home Security Module State Machine

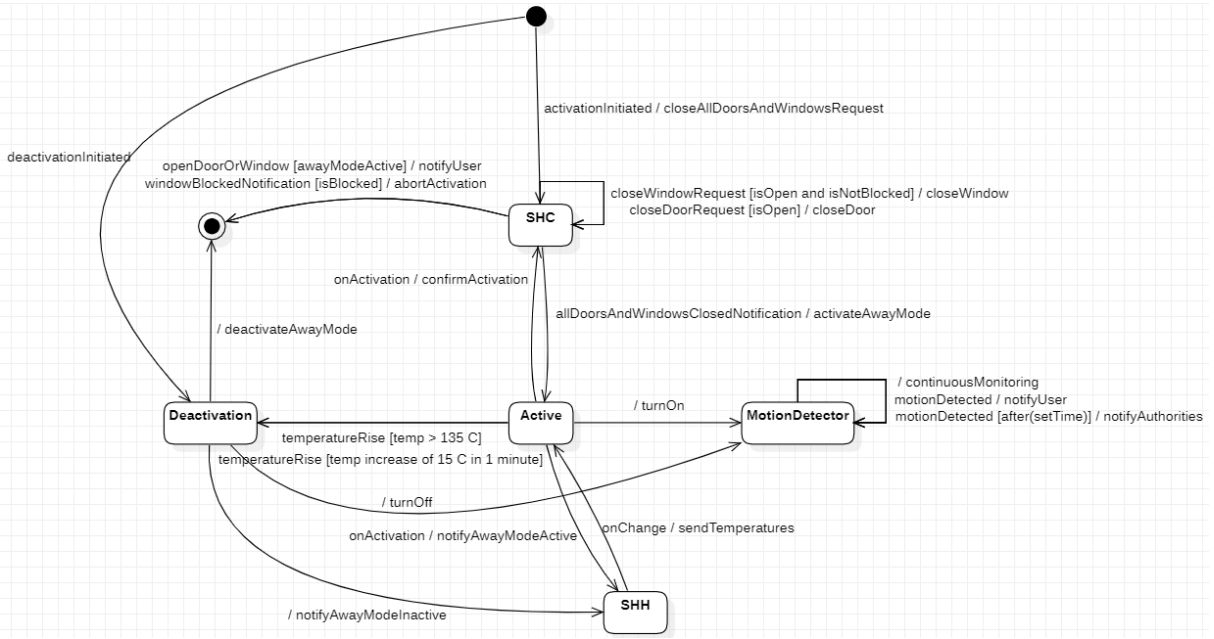
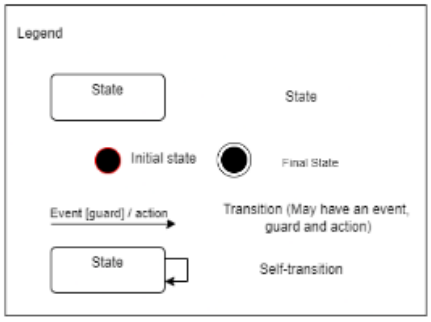


Figure 11: SHP Module State Machine Away Mode Sub-Machine



3.7 Activity Diagram for the Smart Home Security

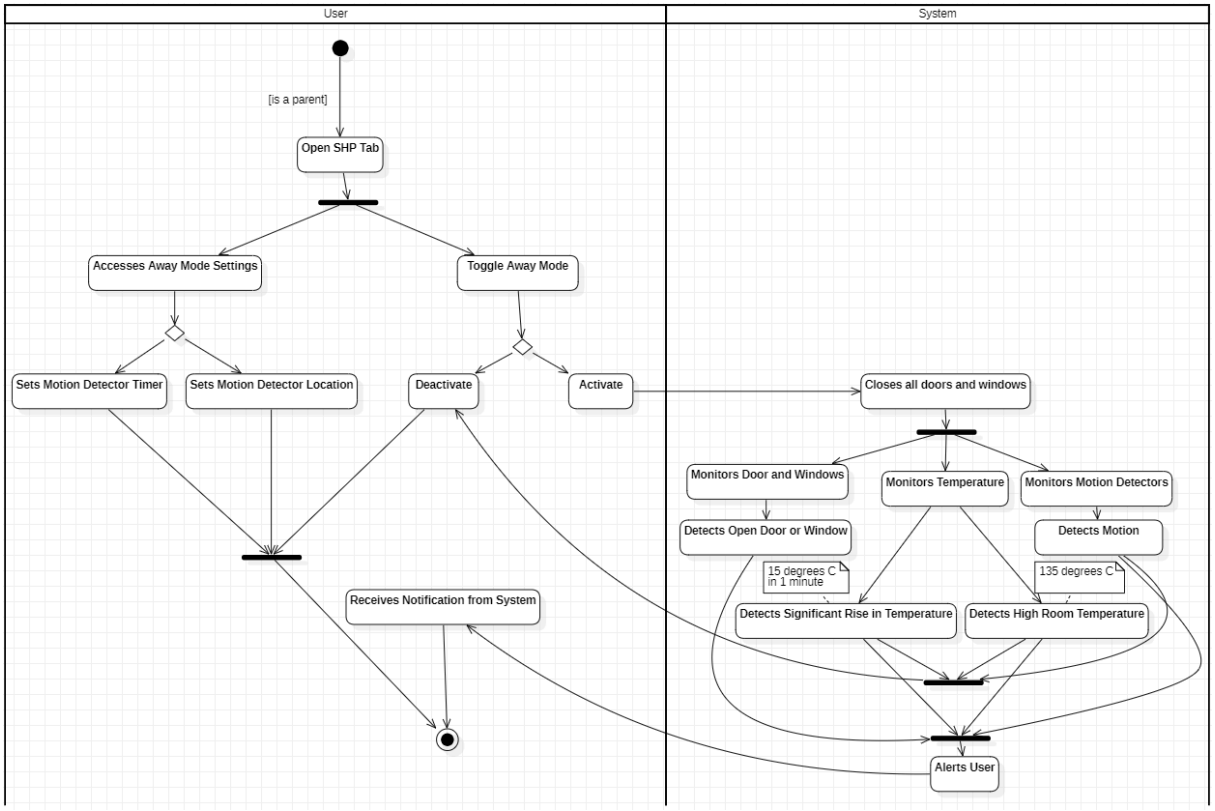
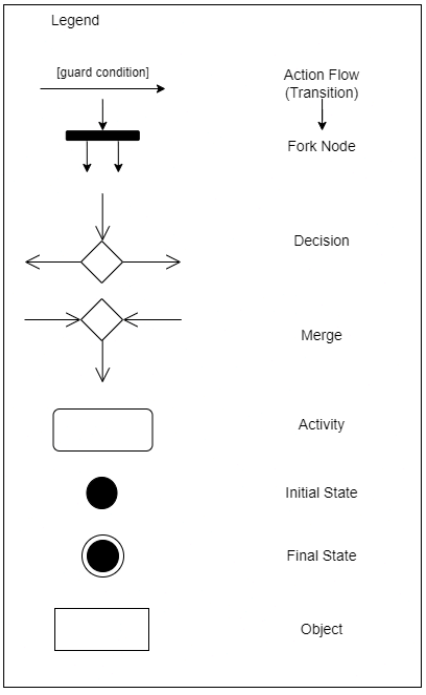


Figure 12: Activity Diagram - SHP



4. Design Patterns

All three of the patterns below have been implemented in our code.

4.1 Command Pattern

Adopting the command pattern in the smart home simulator project facilitates the encapsulation of requests as objects, allowing for parameterization of actions and decoupling of the requester from the executor. This pattern proves advantageous in scenarios where users or other modules issue commands to control various devices and functionalities within the smart home environment. Each command encapsulates a specific action along with its parameters, enabling easy queuing, logging, and undo/redo functionalities. For instance, commands such as "open window," "turn on lights," "open door" or “turn on auto mode” can be encapsulated as command objects, which are then executed by the Smart Home Core Functionality (SHC) module or other relevant modules.

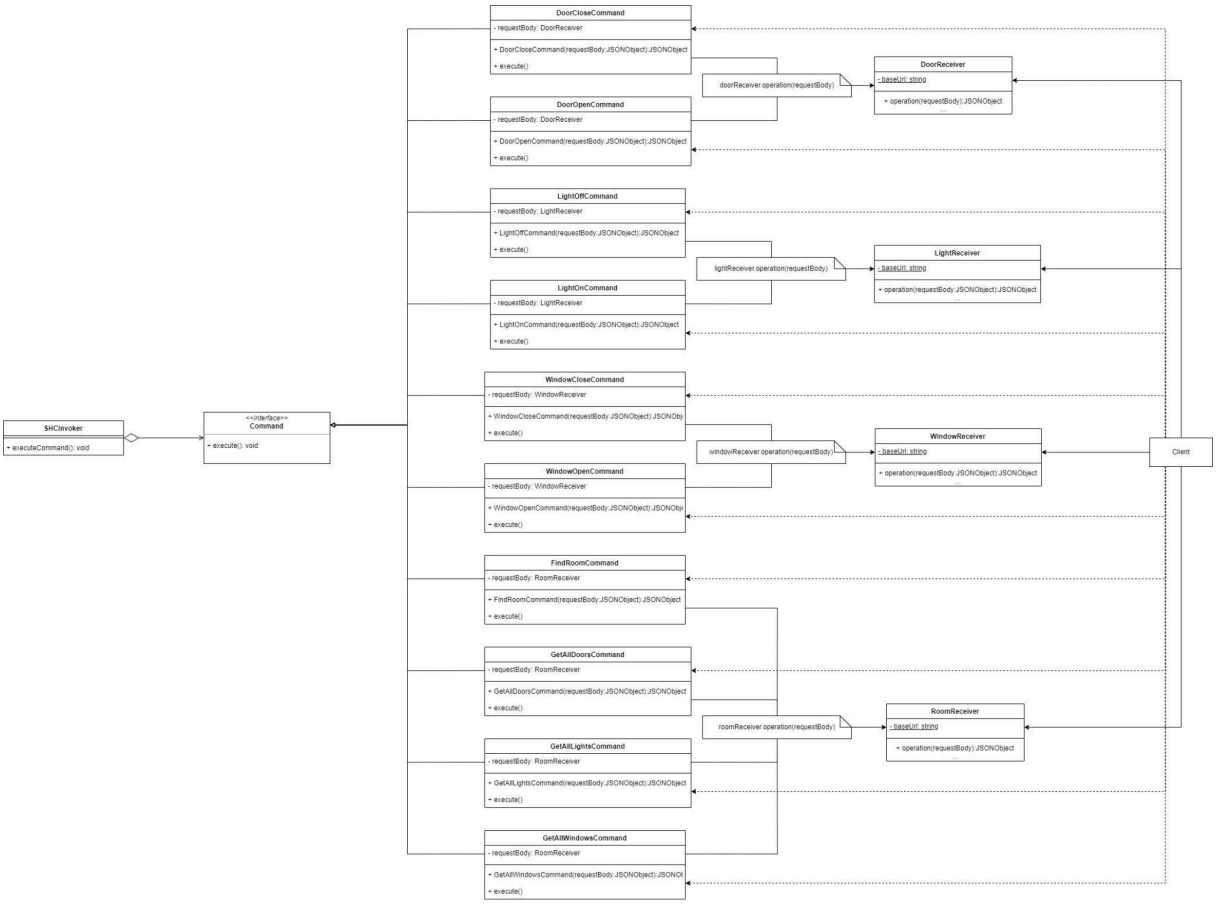


Figure 13: Command Pattern

4.2 Observer Pattern

Implementing the observer pattern for the Smart Heating (SHH) module within the smart home simulator is a good design decision for several reasons. Firstly, the observer pattern facilitates a decoupled and modular architecture, enhancing the maintainability and scalability of the system. By employing this pattern, the SHH module can seamlessly integrate with the simulator and the Smart Home Security module without being tightly coupled to its implementation details. Secondly, the SHH needs to constantly monitor various environmental factors like indoor and outdoor temperatures, occupancy status (Away status), and system events, the observer pattern allows it to efficiently subscribe to relevant updates from the simulator and the Smart Home Security Module. Whenever there's a change in temperature or occupancy status (Away Mode), the SHH is promptly notified, enabling it to make real-time adjustments to heating and cooling systems accordingly. Lastly, the observer pattern promotes flexibility since we can easily accommodate multiple observers if needed for other modules.

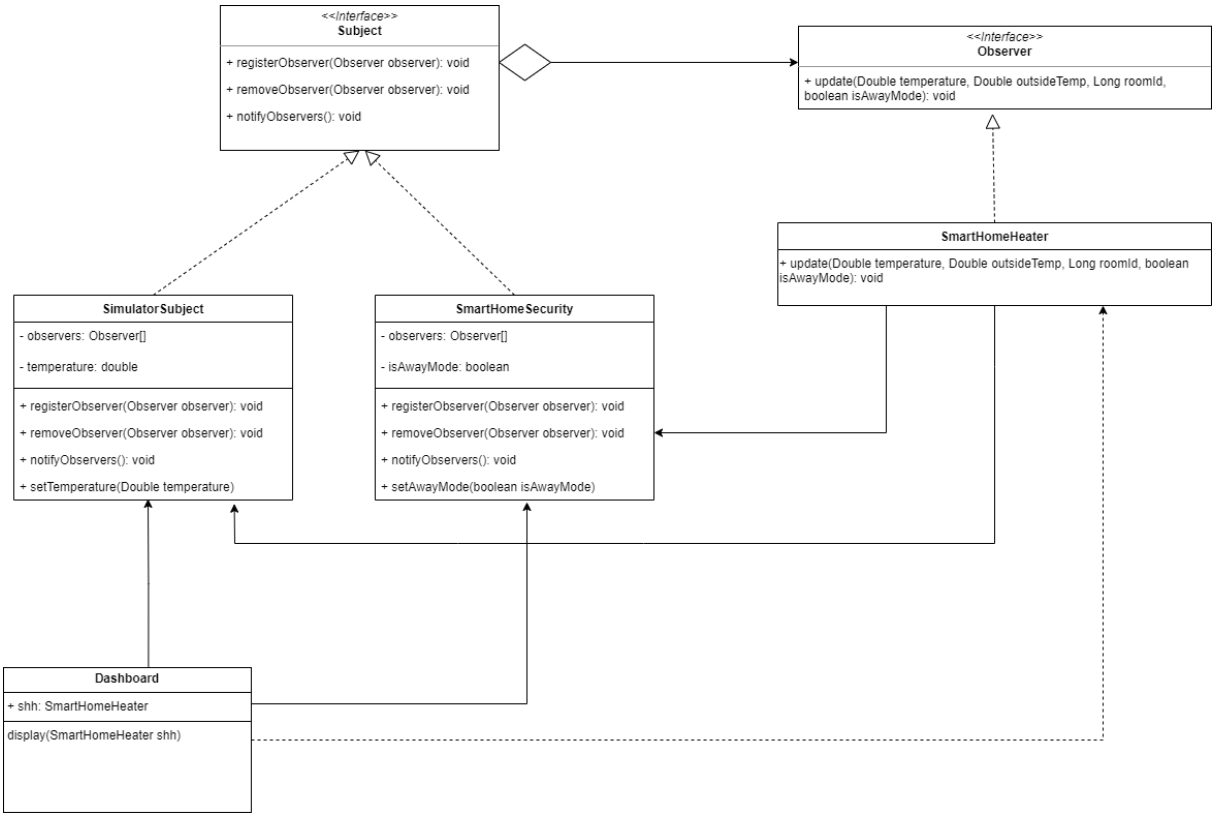


Figure 14: Observer Pattern

4.3 Singleton Pattern

We decided to use the Singleton design pattern for the outdoor temperature since it encapsulates the overall temperature outside. It does not make logical sense to have more than one temperature for outside of the home, so using the Singleton pattern ensures that there are no errors when getting instances of OutdoorTemperature.

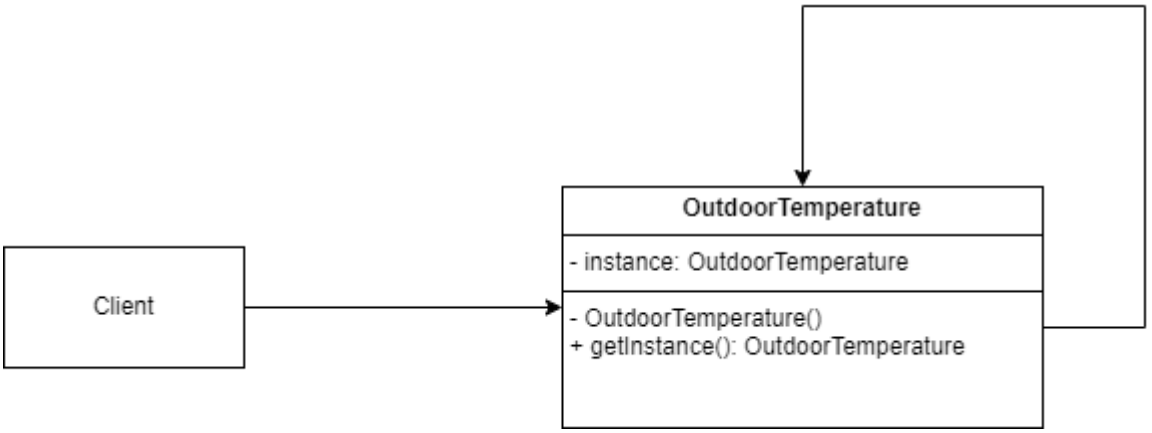
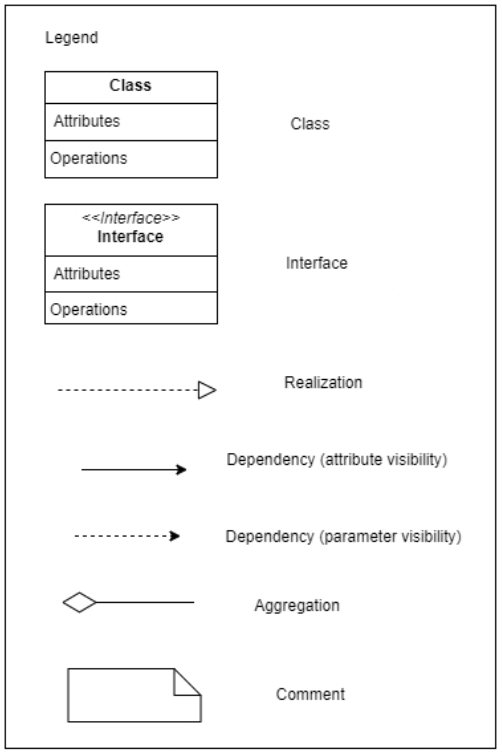


Figure 15: Singleton Pattern

Below is the legend for the Design Patterns.



References

[1] https://www.conceptdraw.com/How-To-Guide/picture/Design-elements-UML-class-diagrams.png
[2] https://cs.uwlax.edu/~mzheng/CS743Fall19/UseCaseDiagrams.html
[3] https://conceptdraw.com/a3156c3/preview