SOEN 343:
SOFTWARE ARCHITECTURE AND DESIGN

Phase I of Smart Home Simulator

Yasmine Hilout ID: 40214158

Asmae Loulidi ID: 40210936

Naika Jean-Baptiste ID: 40227367 - LEADER

Megan Coscia ID: 40214158

Denis Oh ID: 40208580

February 12, 2024

Winter 2024

Contents

1 Smart Home Simulator

1.1 Problem definition

1.1.1 Problem Statement

The problem of	*The challenge of managing a smart home system with connected devices and features like heating, security, and lighting, while also keeping an eye on changes affected by factors like outside temperature, people in rooms, and objects blocking pathways. This complexity makes it hard to control and improve the system's performance, leading to problems in understanding how everything works together and fixing any issues that come up.*
Affects	*People like researchers, students, professionals, and regular users who are interested in trying out virtual smart home features and seeing how they interact.*
The impact of which is	*Difficulty in trying out, fixing problems with, and understanding the ins and outs of smart home systems, making it tough to make them work better and give users a good experience.*
A successful solution would be	*A smart home simulator that provides a graphical representation of interactions between household elements and IoT systems, allowing for experimentation, problem-solving, and optimization in a controlled virtual environment. This would make testing easier, help people understand smart home features better, and improve the overall user experience.*

1.1.2 Product Position Statement

For	*Researchers, students, and practitioners interested in experimenting with virtual smart home modules*
Who	*Seek to simulate and understand the complexities of smart home systems*
The Smart Home Simulator	*is a software desktop app*

That	*Offers a graphical interface to model and interact with smart home environments, allowing users to experiment with different configurations and scenarios*
Unlike	*Traditional programming-based approaches or basic simulation tools*
Our product	*Provides a user-friendly platform with detailed control over simulation parameters, customizable family profiles, and modular smart home modules, enabling in-depth analysis and solution exploration for smart home challenges.*

1.2 Product Overview

1.2.1 Product Perspective

Our product is independent and totally self-contained, meaning that it does not rely on any external system and is not part of a larger system. It offers all-in one functionality, such as the Smart Home Dashboard, Smart Home Simulator, and multiple Smart Home Modules. It is designed to provide users a way to manage their home environment.

1.2.2 Assumptions and Dependencies

Assumptions	Dependencies
The device will be able to execute the simulator.	*Availability of hardware operating systems capable of running our simulator. The system is developed using the Java language and therefore requires the Java Virtual Machine (JVM).*
The features of the smart home simulator will affect the design of its interface	*Implementation of features related to controlling and monitoring simulated devices will affect ease of use, intuitiveness and accessibility of the system interface*
Smart home modules are independent and can communicated through an API	*Implementation of an API for communication between SH modules*
The simulation accurately models the behavior of sensors, window/door movers, lights, and other smart devices	*Integration of sensors, window movers, light activation, and to capture and control environmental variables.*
SHC module behaves as a central controller for managing home devices & permissions	*Proper integration and functionality of SHC module to coordinate or prevent actions*

SHH module monitors indoor and outdoor temperatures and controls heating and cooling	*Availability of sensors (thermostat) for temperature monitoring*
SHP module detects intrusions & triggers appropriate responses	*Integration of sensors for motion and for window/door status.*

2 Technology used

2.1 Control version system

We are using Git as our version control system. GitHub is the hosting platform for our repository. Together, Git and GitHub allow our team to track and manage changes made to the project codebase efficiently. We are able to maintain a detailed history of all code modifications for future reference. Team members can develop in parallel with their individual branches and merge their changes to the main branch when ready. GitHub is the central hub for our repository. It provides us with additional tools for us to organize our project with assigned milestones, issues, and tasks.

2.2 Team collaboration

Discord is our team’s primary platform for communication. We have created a Discord server dedicated solely to discussions relating to this project. Discord’s real-time messaging capabilities are ideal for quick questions or announcements to other team members. The ability to “mention” specific members in chats is used when there are concerns regarding specific members that are not as important to the rest of the team. We are also able to audio call each other for deeper discussion while sharing screens for others to watch.

The Moodle Forum is also used for more structured and formal communication needs. Mainly for detailed discussions and decisions which need to be documented for future reference. Since the forum will be updated less frequently, discussions will be more focused on delivering information in a concise, easily understood manner.

2.3 Monitoring and verification

In order to complete everything that needs to be done by sprint deadlines we monitor all tasks required diligently. Tasks which can be easily divided up into relatively equal parts are distributed in a manner which takes advantage of team members' strengths. For tasks which cannot be easily divided and require discussion, the team sets a time to meet and work through it together. This can be done over discord group calls in our project server or during lab time discussions. Tasks are monitored in our GitHub and updated as needed when the task is changed or completed. Tasks which are finished are then verified for quality assurance by the group and marked as complete.

2.4 Design and modeling work

For our design and modeling work, we used Drawio to create the domain model and context diagram. Drawio made it easy for our team to work together and build these important parts. We used it to draw the domain model, showing what classes compose our system and how everything connects. We also used Drawio to make a context diagram, showing how our system interacts with external entities. This helped us understand our project better and work more efficiently as a team.

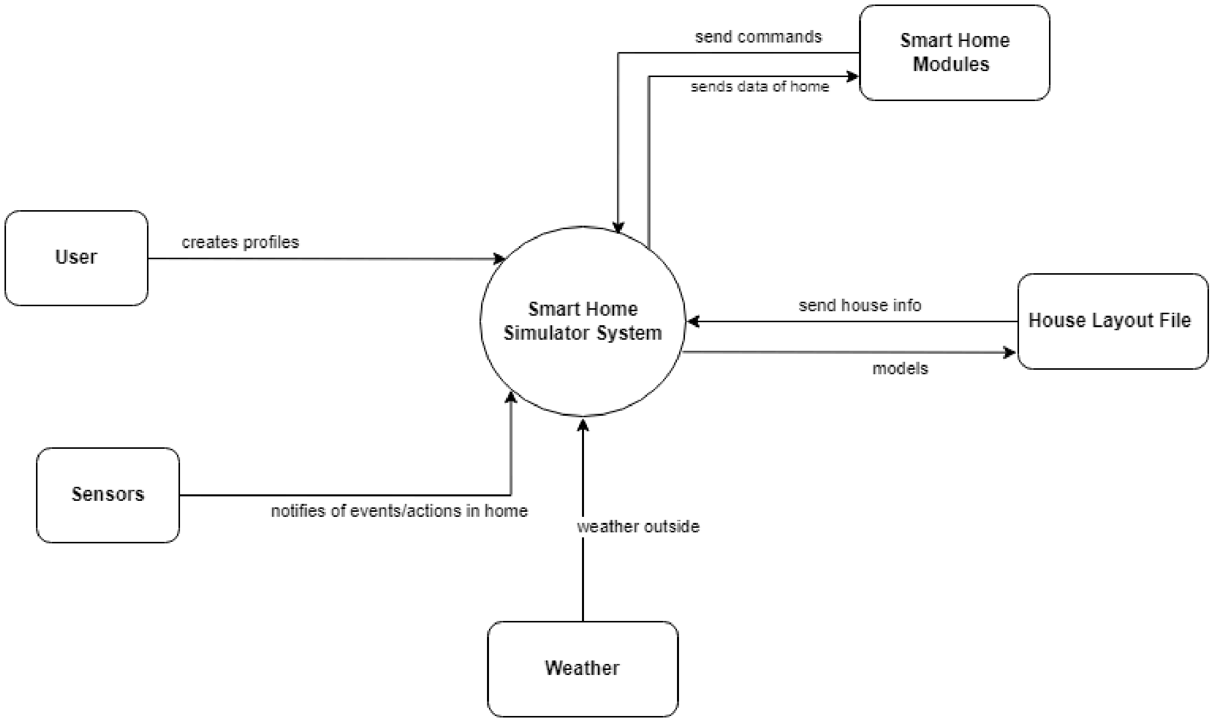
2.5 Development framework

We are considering using Java Swing as our development framework to create a user-friendly graphical interface for our application. It has a cross-platform compatibility, extensive component library, customizable features, and is able to adapt to various operating systems. By using Swing's event-driven architecture, we expect to handle user interactions efficiently while keeping things organized. In summary, we think that Java Swing could be a useful framework to create a strong and visually pleasing software solution that meets both project needs and user preferences.

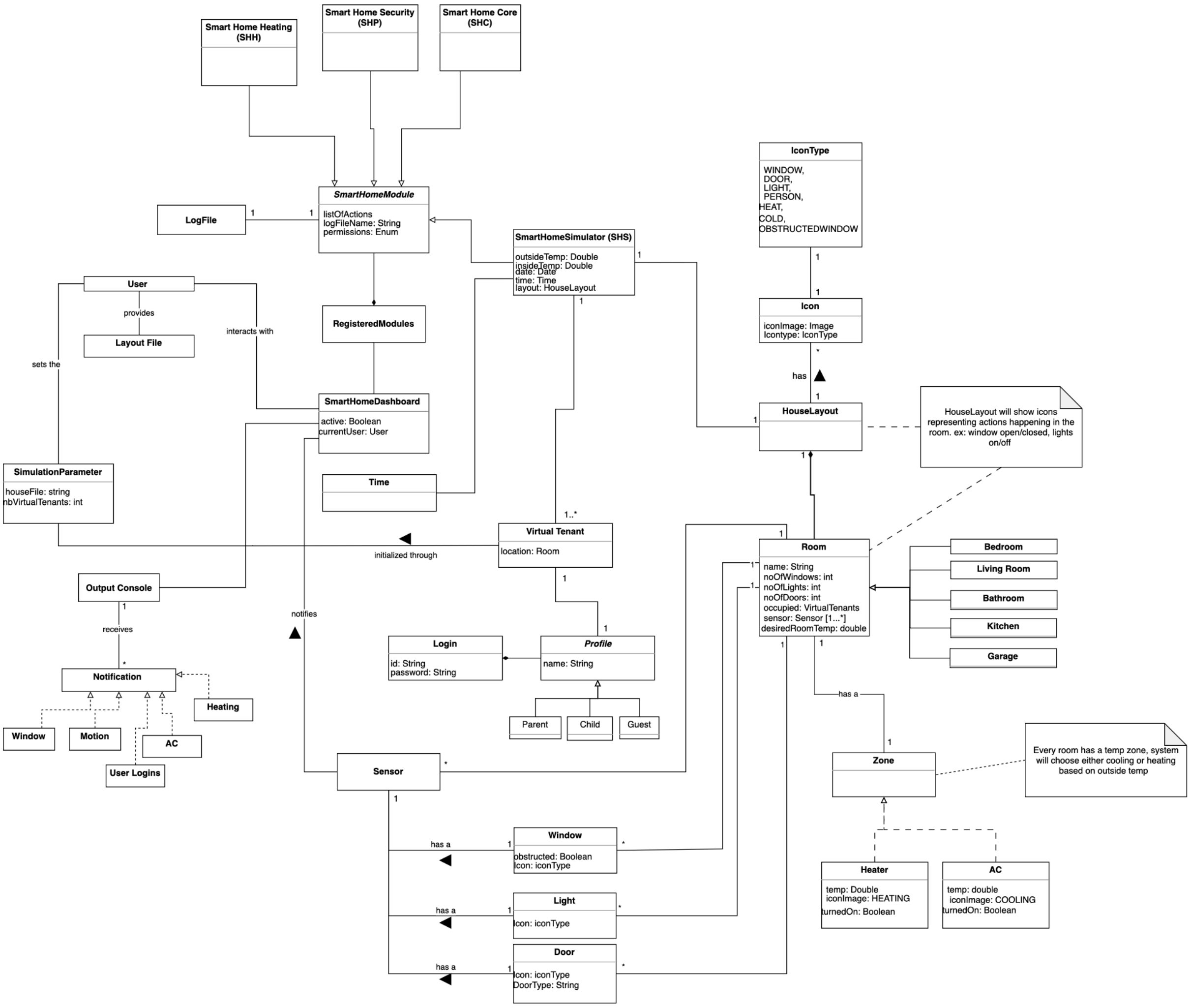
2.6 Coding

Opting for Java as our development language presents a multitude of advantages. Its platform independence ensures our application's compatibility across diverse systems. Java facilitates code reuse through its extensive library of classes and support for object-oriented programming principles like inheritance and polymorphism. It allows the creation of maintainable and secure software.

3 Context diagram



4 Domain model


Smart Home Heating (SHH)
Smart Home Security (SHP)
Smart Home Core (SHC)
SmartHomeModule
listOfActions
logFileName: String
permissions: Enum
LogFile
1
1
SmartHomeSimulator (SHS)
outsideTemp: Double
insideTemp: Double
date: Date
time: Time
layout: HouseLayout
1
1
IconType
WINDOW,
DOOR,
LIGHT,
PERSON,
HEAT,
COLD,
OBSTRUCTEDWINDOW
1
1
Icon
iconImage: Image
Icontype: IconType
*
has
1
HouseLayout
1
1
HouseLayout will show icons representing actions happening in the room. ex: window open/closed, lights on/off
User
provides
Layout File
interacts with
sets the
RegisteredModules
SmartHomeDashboard
active: Boolean
currentUser: User
SimulationParameter
houseFile: string
nbVirtualTenants: int
Time
1..*
Virtual Tenant
location: Room
initialized through
Output Console
1
receives
*
Notification
Window
Motion
AC
User Logins
Heating
notifies
Login
id: String
password: String
1
1
Profile
name: String
Parent
Child
Guest
Room
name: String
noOfWindows: int
noOfLights: int
noOfDoors: int
occupied: VirtualTenants
sensor: Sensor [1...*]
desiredRoomTemp: double
1
1
1
1
1
Bedroom
Living Room
Bathroom
Kitchen
Garage
has a
1
Zone
Every room has a temp zone, system will choose either cooling or heating based on outside temp
Heater
temp: Double
iconImage: HEATING
turnedOn: Boolean
AC
temp: double
iconImage: COOLING
turnedOn: Boolean
Sensor
*
1
has a
1
Window
obstructed: Boolean
Icon: iconType
*
has a
1
Light
Icon: iconType
*
has a
1
Door
Icon: iconType
DoorType: String
*