Smart Home Simulator - Phase 1

Team Members

Christa Abou Arraje - ID: 40226631

(**role**: Domain Model, technologies used)

Gabriel Rodriguez - ID: 40208888

(**role**: Context Diagram)

Jose Semaan - ID: 40244141

(**role**: Domain Model)

Ashpreet Singh - ID: 40172137

(**role**: Problem Definition)

Mark Tadros - ID: 40250850

(**role**: Context Diagram, technologies used, team leader)

SOEN 343 - Software Architecture and Design

Concordia University

February 12th, 2024

Table of Contents

1. Problem definition

1.1 Problem Statement

The problem of	a lack of comprehensive tools for simulating, testing, and educating on smart home configurations in a virtual environment
Affects	developers, educators, and smart home enthusiasts who require deep insights beyond the capabilities of current smart home management systems like Google Home and Amazon Alexa
The impact of which is	restricted innovation in smart home technologies due to the inability to experiment with complex scenarios and system integrations without the cost or risk of real-world deployment
A successful solution would be	a platform that bridges this gap by offering advanced simulation capabilities, fostering innovation, and providing educational value beyond the operational control provided by existing market solutions

1.2 Product Position Statement

For	developers, researchers, and educators seeking an in-depth tool for smart home technology experimentation and learning
Who	need to explore and innovate within the smart home space without the limitations of physical device constraints
Beta	is a comprehensive simulation platform
That	offers unparalleled insights into smart home system behaviors, interactions, and potential innovations
Unlike	mainstream smart home solutions like Google Home and Amazon Alexa, which focus on device management and control
Our product	enables a deeper understanding and experimentation with smart home technologies, positioning SHS as a unique educational and developmental tool in the market

1.3 Product Overview

1.3.1 Product Perspective

Product	Similar Features	Differentiating Features	Competitive Advantage
SHS	- Virtual environment for smart home simulation - API interfaces for device behavior emulation	- In-depth educational and experimental platform - Customizable scenarios for advanced testing beyond real-time control	- SHS allows for extensive 'what-if' scenario testing, which is not typically offered by consumer-grade products
Google Home	- Voice-controlled home automation - Integration with various smart devices	- Primarily focused on real-time device management - Limited to Google's ecosystem	- SHS provides a broader educational focus, whereas Google Home is optimized for end-user convenience and control
Amazon Alexa	- User-friendly interface for smart device management - Wide range of compatible smart home products	- Closed system with limited scope for user customization - Focus on voice interaction	- SHS's simulation-based approach is unique and offers a deeper dive into smart home management compared to Alexa's more surface-level control
Apple HomeKit	- Secure and private system for managing smart home devices - Seamless integration with Apple products	- Requires Apple hardware and is limited to HomeKit-compatible devices - Lacks a virtual simulation environment	- SHS is platform-agnostic and does not require specific hardware, offering flexibility and a wider reach
Samsung SmartThings	- Integrates with a variety of smart devices - Offers some level of automation and control	- More hardware-centric, requiring a SmartThings Hub - Focused on device connectivity rather than simulation	- SHS stands out by providing a risk-free environment for testing and learning, which SmartThings does not directly address
OpenHAB	- Open-source platform for smart home integration - Highly customizable and flexible	- Steeper learning curve - Focuses on real-world integration over simulation	- SHS is specifically designed to be user-friendly and educational, potentially serving a different market segment than OpenHAB

1.3.2 Assumptions and Dependencies

Assumptions	**Dependencies**
Users are looking for a simulation platform to understand and innovate in smart home technology, not just a control interface like those offered by Google Home or Amazon Alexa.	The simulator's advanced features and usability must be clearly communicated to differentiate it from the convenience-oriented products in the market.
There is a market need for a tool that can simulate complex smart home scenarios for educational and developmental purposes, which is not currently met by existing consumer-grade products.	Ongoing updates and compatibility with various smart home protocols and devices to ensure SHS remains relevant against platforms like Apple HomeKit and Samsung SmartThings.
Educators and developers prefer a platform-agnostic tool that does not require specific hardware, unlike systems such as Apple HomeKit, which operates within the Apple ecosystem.	The success of SHS may depend on the availability of a robust online community or support system similar to that which supports open-source platforms like OpenHAB.
Potential users have the technical skill or willingness to engage with a more complex system that provides greater control and customization options than mainstream smart home systems.	Dependencies on external APIs and services must be managed to ensure SHS can simulate a range of devices and scenarios accurately.

2. Technology Used

2.1 Control version System

For the control version system, we will be using GitHub.
Here is the link to our repository: christa-ux/Beta (github.com) .
Since we're still in sprint 1, our GitHub is mainly empty.

2.2 Team Collaboration

Concerning team collaboration and communication, Discord is our platform of choice. It allows us to have different channels, which means that our conversations can be divided into types like "general", "sprint1", "documents", etc. This helps with the organization, and it allows for easy access. For example, in the "documents" channel, we only share documents related to our work. For example if someone finished their part, then can send it there, or if we're working on a specific sprint, the instructions would also be found there.

2.3 Monitoring and Verification

Starting Sprint 2, we'll be using commits on Github to track each person's finished tasks. We also have our main branch protected, so that any thing that wants to be merged will have to be reviewed by 2 people first. In addition, to stay on track and on the same page, we do regular meetings, mainly on Discord or Zoom to make sure everyone is okay with their part.
For testing the code, we will be conducting unit testing with hopefully at least 80% coverage, using JUnit.

2.4 Design and Modeling Work

The design of the Context Diagram and Domain Model were done using *draw.io*, a simple software that provides us with built-in tools to draw our models and diagrams. PowerPoint was also used since it provides us with shapes and some teammates are more familiar with it.

2.5 Development Framework

As a first decision, we opted for React as the framework for JavaScript for the front-end. Concerning the back-end, we'll be using Java, and we're considering Spring boot as it offers rich functionality for web-applications.

2.6 Coding

For the development of the “Smart Home” simulator, we discussed, as a team, that the most appropriate and direct programming languages to work with are: HTML, CSS, JavaScript and Java. The frontend, mainly the looks and the visual design of the simulator, will be implemented using the first 4 languages. As for the backend, which has to do with how the system functions, the storing of information and the interaction between entities, it will mainly be developed using Java which is an Object-oriented Programming language.

For simplicity, you can look at the following table to see which technology will be used for which activity:

Activities	**Used Technology**
Control Version System	GitHub and Moodle
Team Collaboration	Discord
Monitoring and Verification	GitHub, Discord and Zoom
Design and Modeling Work	Draw.io and PowerPoint
Development Framework	ReactJS/React Native, Spring boot
Coding	HTML, CSS, JavaScript and Java

3. Context Diagram

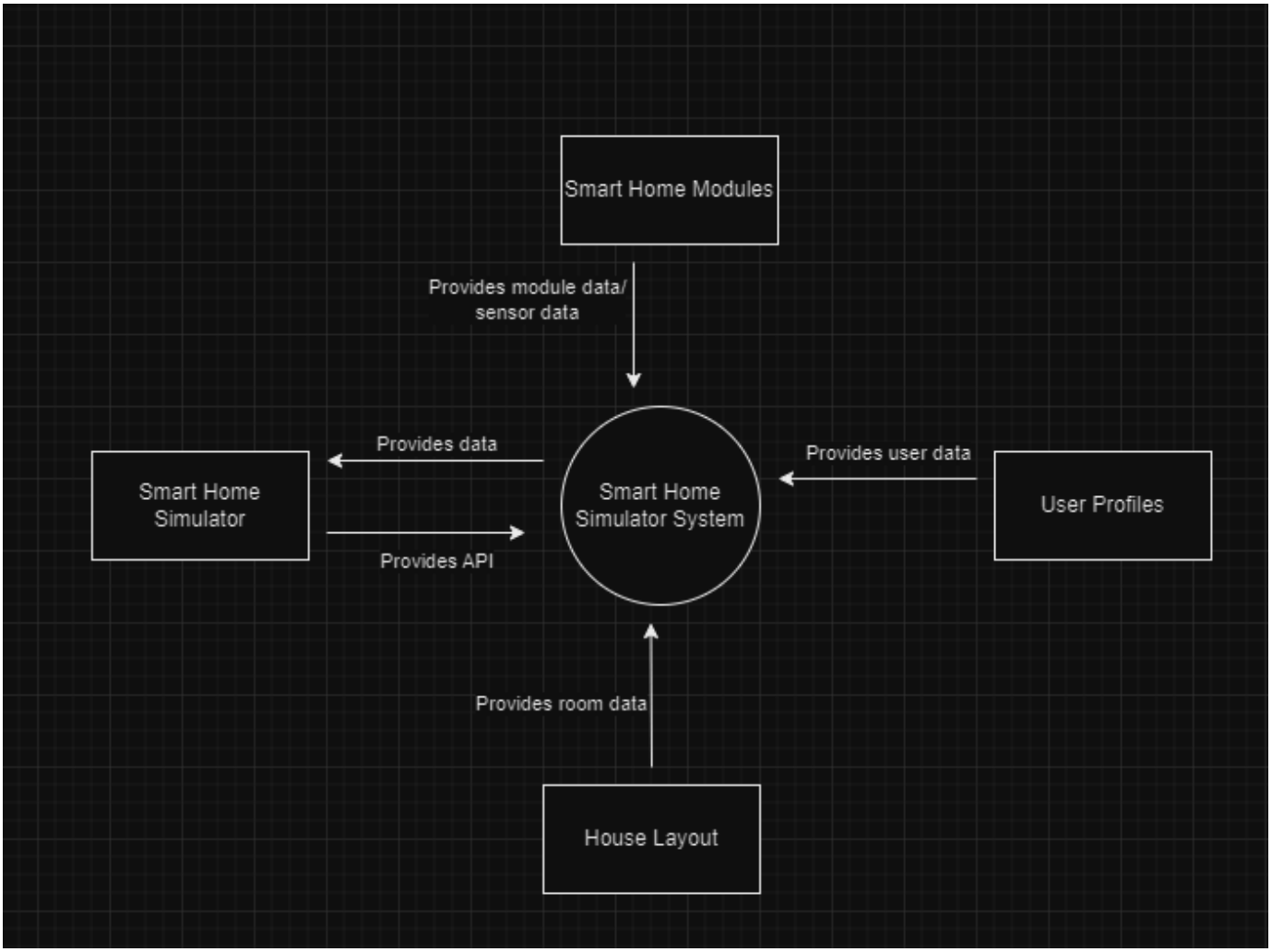

Figure 1: The context diagram displays how the external factors affect the system. The house layout is taken from a text file with the rooms and its configurations. User profiles are created to interact with the system as family members, guests or strangers. The modules are the elements the systems work with such as heating and security. Finally, there is the simulator with the dashboard and API that makes it all work together.

4. Domain Model

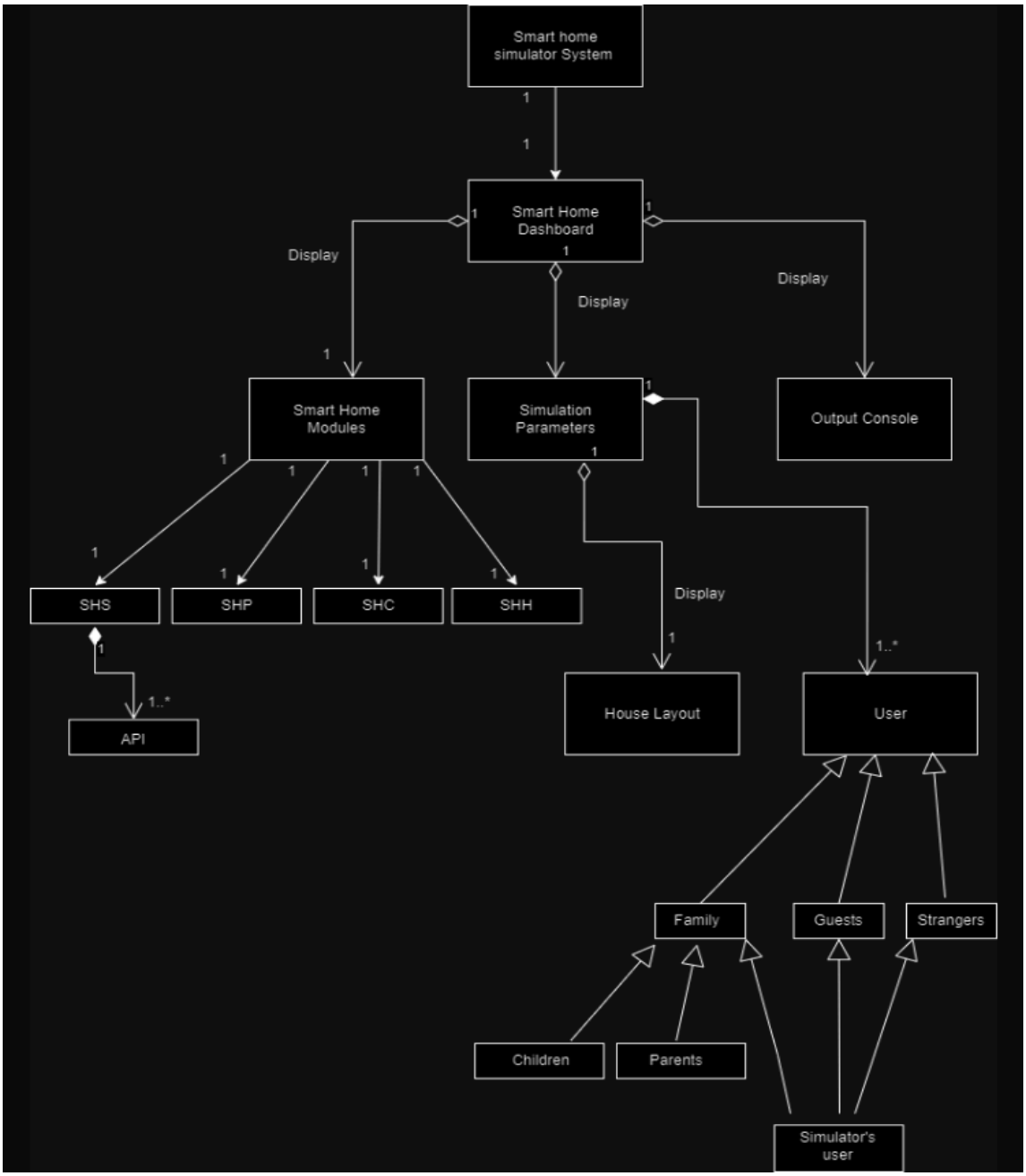

Figure 2: Our smart home system simulator is centered on the dashboard which is the main interface. This dashboard is divided into 4 sub parts: System Parameters, Output Console, House View, and Smart Home Modules. Moreover, users can be of different types, and the simulator user has access to every type, in addition to using the system simulator as a whole.

5. Reference

Context diagram. IST Project Management Office. (2023, October 5). https://uwaterloo.ca/ist-project-management-office/tools-and-templates/tools/context-diagram#:~:text=Context%20diagrams%20show%20the%20interactions,system%20will%20be%20part%20of.

Top 51 software development frameworks. Top Software Outsourcing Company in Vietnam - Orient Software. (n.d.). https://www.orientsoftware.com/blog/software-development-frameworks/

Johns, R. (n.d.). Want to level up in Java? Use these Java frameworks! *Hackr.io*. https://hackr.io/blog/java-frameworks