SOEN 343:
SOFTWARE ARCHITECTURE AND DESIGN

Phase I

Tarek Elalfi 40197527
Khaled Saleh 40210125
Jay Patel 40203705
Omar Shehata 40164194 Team Leader
Samy Arab 40209595

Instructor: Dr. Rodrigo Morales Alvarado

Winter 2024

Contents

1. Problem definition

1.1 Problem Statement

In this section, we'll explore challenges in the smart homes simulator for the stakeholders, propose improvements and a successful solution.

The problem of	The lack of a comprehensive and user-friendly simulator platform for managing and optimizing simulated smart home systems with connected devices, resulting in user frustration.
Affects	Homeowners, manufacturers and developers.
The impact of which is	That the user would have to spend more time manually adjusting and configuring the devices in their home environment.
A successful solution would be	A user-friendly platform that simplifies the management and optimization of simulated smart home systems. This platform should provide intuitive tools for addressing complexities by separating the automated features into modules such as security, heating, lighting, and core functionalities within the simulated environment. Additionally, it should enable efficient experimentation and refinement processes, fostering homeowners, and innovation in the field of smart home technology.

1.2 Product Position Statement

In this section, we'll examine the product position statement for the smart home simulation platform. We'll discuss its target audience, unique value proposition, and differentiation from other solutions like Google Home and Amazon Alexa.

For	Homeowners
Who	Require management and optimization of a smart home simulation system.
The smart home simulation	Is a platform designed with comprehensive user-friendliness, specifically tailored for managing and optimizing simulated smart home environments with great functionalities and UI.
That	Offers easy-to-use features, and modules for improving and experimenting with smart home simulation technology and devices.
Unlike	Google home and Amazon alexa
Our product	Provides a comprehensive platform application specifically designed for managing and optimizing simulated smart home environments.

1.3 Product Overview

1.3.1 Product Perspective

The Smart Home Simulator is an independent platform designed specifically for managing and optimizing simulated smart home automation systems. It provides a user-friendly interface and intuitive tools for experimentation, refinement, and innovation in smart home technology. Although the simulator is self-contained, it interacts with various modules, devices and technologies commonly used in smart home environments.

1.3.2 Assumptions and Dependencies

In this section, we'll outline the assumptions and dependencies crucial for developing the smart home simulation platform. We'll cover user knowledge, software requirements, and data access for realistic simulations.

Assumptions	Dependencies
Users have basic knowledge of using smart home technology, devices and concepts.	Availability of necessary software libraries and frameworks for the simulator smart home development.
The simulator runs on standard computers with access to necessary resources.	Adequate hardware resources (CPU, memory, storage) to ensure smooth operation and responsiveness of the simulator.
Simulated smart home modules and devices accurately reflect real-world behavior and interactions.	Ensure that the simulated application platform can accurately interact with the external files for the house layout necessary for realistic simulation scenarios.
Users have access to the necessary data and resources for configuring and running simulations.	The devices that are controlled by the modules send/receive signals that are needed by the smart home simulator

2. Technology Used

In this section, we discuss what tools/languages are team will use for team collaboration, control version system, monitoring and verification, coding and development framework.

2.1 Team Collaboration

In our project ecosystem, various tools play distinct yet interconnected roles, harmonizing our development efforts seamlessly. Discord serves as our virtual meeting room, facilitating scrums and general discussions among team members. It doubles as a platform for file sharing and collaboration on design endeavors, fostering efficient communication and idea exchange.

2.2 Control Version System and Monitoring/Verification

Jira takes charge of our project's organizational backbone, meticulously tracking the status of technical artifacts and user stories. This meticulous oversight ensures swift progress aligned with scrum methodologies, keeping our development cycles agile and effective. Meanwhile, GitHub Wiki acts as our shared repository of technical documentation, ensuring all team members have unfettered access to vital resources.

2.3 Coding

For the backend infrastructure, we leverage the power of Express.js, a sleek framework built atop Node.js. This combination empowers us to craft robust web applications and APIs with ease, thanks to its streamlined routing and middleware capabilities. On the frontend, React.js steps into the limelight, orchestrating our user interface with reusable components and modern design principles, all while maintaining impeccable performance standards.

Git stands as our vigilant guardian of version control, facilitating seamless collaboration, code merging, and historical tracking across our distributed team. MongoDB, our database of choice, embraces the NoSQL paradigm, storing data in nimble JSON documents. This flexibility, coupled with its seamless integration with Next.js, ensures a smooth flow of data from server to client, in perfect harmony with the MVC architectural pattern.

2.4 Development Framework

To ensure a consistent and efficient development experience, we've rallied behind VS Code as our IDE of choice. Its robust feature set, bolstered by a vibrant ecosystem of extensions and community support, fosters a cohesive environment for coding excellence. And when it comes to design and modeling, Draw.io emerges as our trusted ally, empowering us to visualize and iterate upon system architectures with ease.

Finally, Figma takes center stage in our quest for stunning user interfaces, offering a canvas where creativity knows no bounds. With its suite of tools for interface design, we sculpt elegant, responsive UIs that elevate the user experience to new heights. Together, these tools form the bedrock of our development journey, enabling us to transform ideas into reality with precision and finesse.

3. Context diagram

A context diagram serves to illustrate the scope of a system alongside its interactions with external entities such as actors, physical objects or small-scale systems. Each external entity interacts with the system, while the system may also interact with entities. In the context of our project, the context diagram shown below helps to better understand the design of our smart home simulator system.

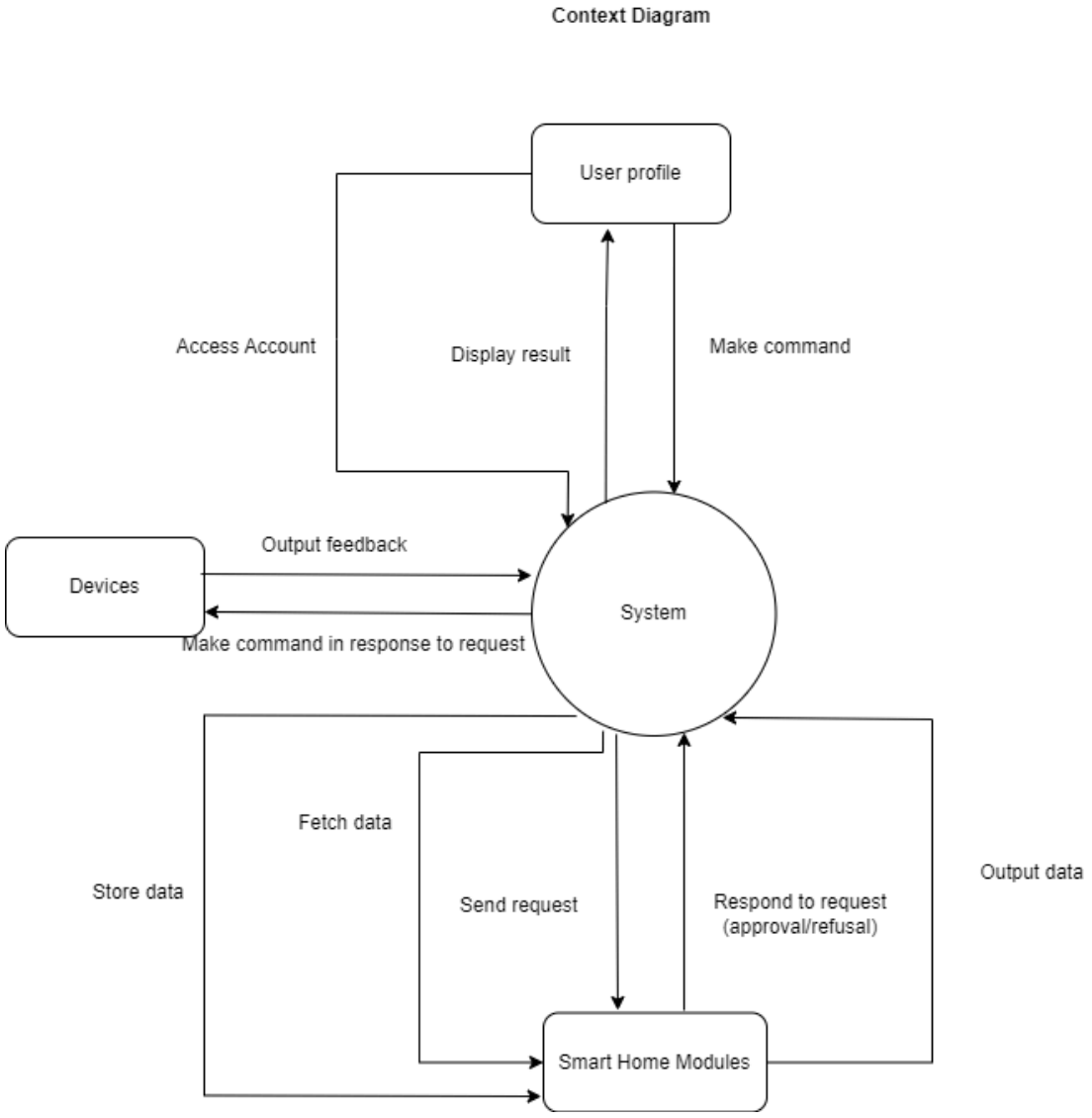

Figure 1: Context Diagram

In the context diagram of our smart home system, the system itself is positioned at the center and acts as the core intelligence and control hub, charged among other things with dispatching messages to the correct modules. The system is surrounded by three components: the user profile which contains all the users of every type (Family members, Guests, and strangers), the devices and the smart home modules. The User profile component encompasses the structure itself along with its inhabitants, indicating the environment in which

the smart home system operates. Furthermore, regarding the Devices segment, which include sensors, thermometers, cameras, lights, blinds, along with all the other possible appliances fall under this umbrella term that symbolizes the various endpoints within the home that interact with the system. Lastly, the Smart Home Modules component englobes the software responsible for managing and coordinating the interconnected device, facilitating communication, automation and user interaction. This context diagram illustrates the broader context in which the smart home system operates, highlighting its integration with both the physical environment and the technological infrastructure within the household.
A user will send a command to the system (through a user interface), the system will read and redirect a request to the relevant module, which will check the validity of the request and return an answer to the system. Based on this answer, the system will send a command to the desired device (if approved) and consequently let the user know of any changes (if any). In the context of fetching information like at what time a given door was opened or light closed, the process would entail a similar path, going from a user command, a system request to the appropriate module, and an output from that module.

4. Domain model

A domain model is a structured representation of the core elements and relationships within a problem domain. It defines entities and relationships, aiding in system design by abstracting away unnecessary details. The following domain model illustrates each component and the interaction between them.

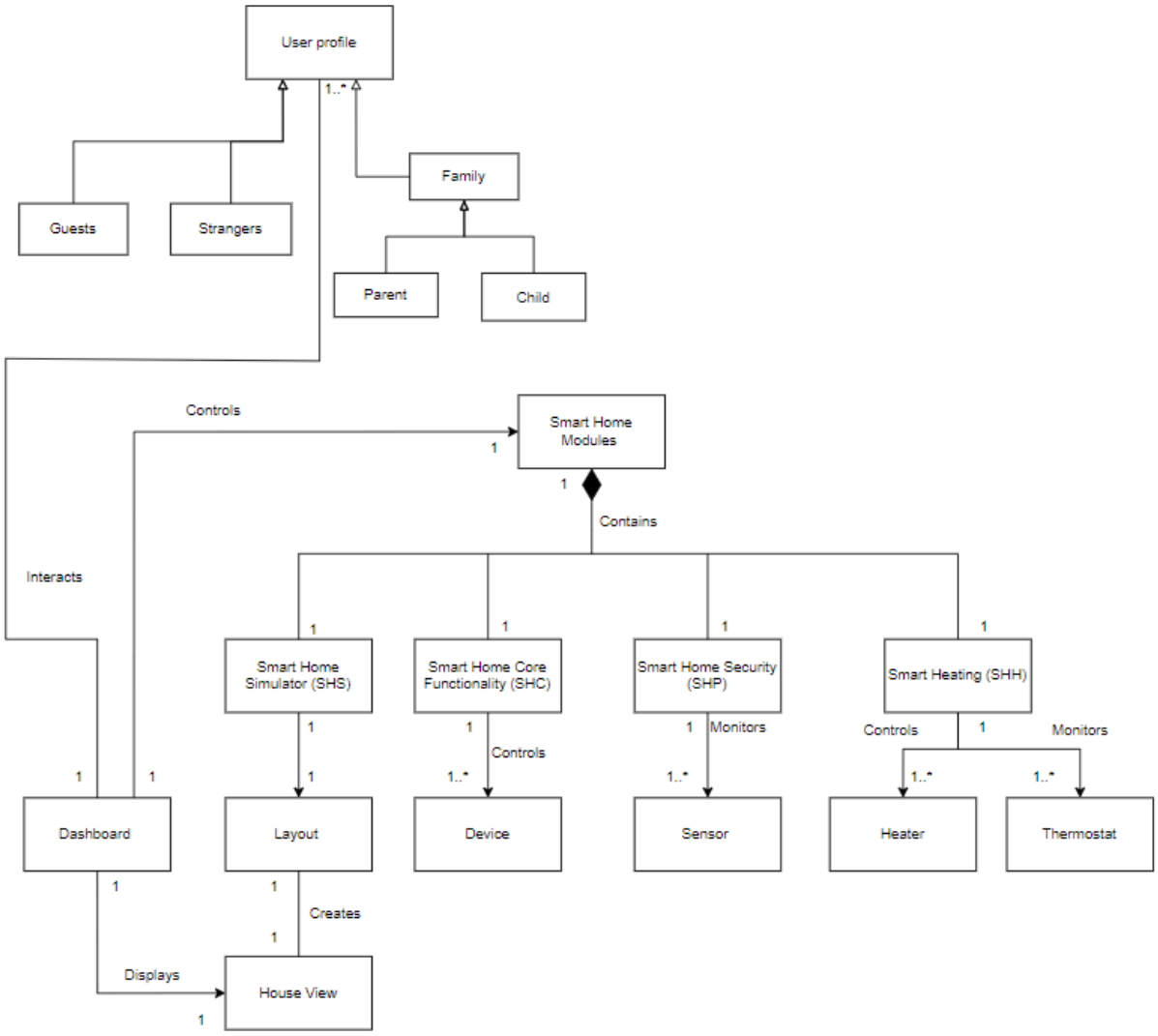

Figure 2: Domain Model

In the domain model of our smart home system, the user profile component encompasses the users inside the house, including Family members, guests, and strangers. The Dashboard component displays the main window of the application along with the modules and the House View. The House View component will be the 2D representation of the interior of the house. Furthermore, the Layout component will be a purely backend/logical component that reads the information about the interior organization of the house and translates this data in a

way that the House View component can render. The Sensor, Heater and Thermostat components will abstract the behaviour of its real life counterparts. The Smart Home Modules component contains all the modules (Smart Home Simulator, Smart Home Core Functionality, Smart Home Security, Smart Heating) of this application and purely acts as a container. Finally, the Device component encompasses all the hardware that might be needed for the SHC component to function properly, for example, garage doors, front/back doors, light bulbs, etc.