Technical Report - **Product specification**

HomeMaid

Course: IES - Introdução à Engenharia de Software

Date: Aveiro, 7 de outubro de 2024

Students: 109061 - Ângela Ribeiro
114220 - Rita Silva
114129 - Regina Tavares
113889 - Hugo Castro

Project abstract: HomeMaid is a system for monitoring and managing devices and conditions in smart houses. Using consumption sensors, the platform collects real-time data and identifies anomalous conditions. The system allows users to remotely monitor and control the environment through a web portal.

Table of contents:

1 Introduction

HomeMaid is a cutting-edge web application designed to provide homeowners with an intuitive, seamless way to monitor and control their homes. As modern technology continues to advance, the demand for convenience and automation in daily life has grown significantly. People today expect their homes to integrate with their lifestyles, offering solutions that save time and simplify routine tasks. HomeMaid rises to meet this expectation by delivering an easy-to-use, highly interactive platform that empowers users to manage their household environments with minimal effort.

Our app transforms the concept of smart homes into a reality, giving users the ability to control various aspects of their living spaces—from lighting and temperature to security and appliances—all from a single interface. With just a few clicks, HomeMaid turns the complexities of home management into simple, streamlined actions.

The vision behind HomeMaid is to alleviate the burden of repetitive household chores and enhance everyday comfort through automation. We aim to redefine how people interact with their homes, offering features that not only make life easier but also give users more time to focus on what truly matters. Whether it's scheduling the perfect ambiance for a relaxing evening or ensuring that security systems are active while you’re away, HomeMaid simplifies these tasks, delivering a smarter, more efficient living experience.

By harnessing the power of modern technology, HomeMaid opens the door to a future where your home is more than just a place to live—it's a dynamic, responsive space that works with you, for you.

2 Product concept

Vision statement

HomeMaid is a web-based application designed to give homeowners an intuitive platform to monitor, control, and automate various aspects of their household environment. At its core, the system will be used to enhance the convenience of home management, allowing users to control devices like lights, temperature settings, security systems, and even household appliances. Through the HomeMaid interface, users will interact with smart home devices without needing to understand the technical underpinnings, making the system user-friendly and accessible to all.

The high-level business problem that HomeMaid addresses is the increasing demand for automated home management solutions that simplify everyday tasks. In today’s fast-paced world, people are seeking ways to streamline mundane chores, reduce the amount of manual intervention required in household management, and improve the overall quality of life by optimizing how they interact with their living spaces. HomeMaid

solves this problem by offering a comprehensive home automation platform that makes controlling a smart home easier, more efficient, and accessible to a broader audience.

Initially, our vision for HomeMaid included the integration of a voice-control feature, allowing users to manage their homes via voice commands. However, due to technical challenges and the desire to focus on delivering a stable and intuitive visual interface first, we opted to delay this feature for future iterations of the product. Instead, we prioritized a responsive web interface that can be accessed from both desktop and mobile devices, ensuring that users can control their home systems remotely with ease. By shifting focus to perfecting the UI/UX experience and integration with existing smart home devices, we have laid a stronger foundation for potential future enhancements, such as voice control.

HomeMaid is similar to other smart home management systems like Google Home or Amazon Alexa in its goal of providing a centralized platform for controlling a variety of home devices. However, HomeMaid differentiates itself by offering a more customizable, user-centric interface that emphasizes simplicity without sacrificing control. Unlike some well-known platforms that require users to invest heavily in proprietary devices, HomeMaid aims to be compatible with a broader range of existing smart home devices, making it a more versatile option for users who may have a variety of smart home products from different manufacturers.

Additionally, while some platforms focus heavily on voice control or ecosystem lock-ins, HomeMaid emphasizes device independence, offering users the freedom to integrate and control devices across various brands and technologies with ease.

Personas and Scenarios

Persona 1:

Name: Maria
Age: 38
Location: Espinho
Job Title: Marketing Manager
Income: Upper Middle Class

As a single mother of two children, Maria balances the demands of her job with the responsibility of caring for her family. Safety and peace of mind are priorities for her, especially when her children are home alone after school. Maria wants to ensure her children's safety by seeking solutions that help her monitor the house from a distance.

Goal: Ensure the safety and security of her children when they are home alone. She wants to receive notifications about any unusual activities or potential dangers in the house, like open doors, strangers, or alarms going off.

Pain Points: Her frustrations stem from anxiety about not being able to track activities at home, fearing potential dangers like open doors or the presence of strangers. She feels vulnerable due to her difficulty in managing multiple devices simultaneously, which prevents her from having a clear view of what is happening.

Persona 2:

Name: Catarina
Age: 32
Location: Penafiel
Job Title: CEO
Income Level: Upper Class

With a busy schedule filled with meetings, work trips, and social commitments, Catarina values efficiency and looks for solutions that save her time and effort. For Catarina, automating all kinds of repetitive tasks is crucial to simplifying her daily life.

Goal: Automate repetitive and tedious tasks, such as adjusting the temperature, turning off lights, and playing music at specific times, to simplify her daily life.

Frustrations: She experiences significant frustration when she loses time on tasks that could easily be automated. It is common for her to forget to adjust the home settings before leaving for work, leaving her dissatisfied with the manual processes that consume her precious time.

Persona 3:

Name: João
Age: 35
Location: São Miguel
Job Title: IT Professional
Income Level: Upper Class

Proud of maintaining order in every aspect of his life, João values full control over his environment and dislikes unpredictability. João is detail-oriented and loves having complete visibility of everything in his life while also being eco thoughtful.

Goal: Monitor his house at any given moment and be able to correct something that is wrong (e.g., an appliance left on, a door unlocked). He also wants to be able to improve his house to be as eco-friendly as possible.

Frustrations: His frustrations arise from a lack of real-time updates or delays in receiving notifications. This uncertainty and the occurrence of unexpected issues when he is away from home heighten his anxiety and dissatisfaction, as he wishes to monitor his house at any moment and act swiftly if something goes wrong.

Scenario 1:
Maria just changed jobs and is now working 1 hour away from home instead of working 10 minutes away as she used to. Due to this change Maria won't be home when her children arrive from school. Maria now needs to know when they get home and she wants to automatically block all the entries so that they are safe until she gets home.

Scenario 2:
Catarina has been very occupied lately and has been sleeping less, so in the morning she wants her coffee to be made automatically at 5:30am. She also wants her walk-in closet to be warmed up by the morning (5:00am) so her clothes aren't cold when she needs to dress fast. At night she wants the app to set her water running so her bath is ready when she gets there (9:00pm).

Scenario 3:
João's vacuum cleaner has been malfunctioning for some days, the device turns on itself and this has been worrying him because it's wasting energy. Due to this, he wants to receive an alert as soon as this happens so he can turn it off.

Product requirements (User stories)

Epic 1: Monitoring
User Story 1: As João, I want to check the status of all devices and ambience in my home instantly, ensuring that nothing is left on unintentionally and the atmosphere is perfect. (High)

Epic 2: User Automation

User Story 2: As Catarina, I want to set up an automation to my coffee machine a certain hour to ensure that I save time. (Medium)
User Story 3: As João, I want to set up limits for how long devices can stay on to prevent wasteful consumption. (Medium)

Epic 3: Alerts
User Story 4: As João, I want to receive real-time alerts when something in my home isn't functioning properly so I can immediately take action. (Medium)

User Story 5: As Maria, I want to have alerts if someone enters my house or if my children left the front door open, so I can better protect my kids. (High)

Epic 4: Notifications

User Story 6: As Catarina, I want to be notified whenever my automations are triggered, so I

know that everything is running as planned. (Low)

User Story 7: As João, I want to receive notifications if my energy usage spikes beyond a set limit, so I can take steps to prevent excessive consumption. (High)

Epic 4: User Account Management

User Story 8: As Maria, I want to log in to my account quickly and securely and be able to add and remove devices to my account, so that I can customize my smart home system as my family's needs change. (High)

Epic 5: Device Control

User Story 9: As Catarina, I want to control all the devices in my home from one app, so I don't waste time switching between different apps. (High)

Epic 6: Integration with external Resources

User Story 10: As Maria, I want to integrate my smart home system with Alexa, so that I can control my home through voice commands when I'm busy with my children. (Low)

Epic 7: Data Backlog

User Story 11: As João, I want to see a detailed breakdown of my home's usage trends, so I can identify any spikes or areas where energy is being wasted. (Medium)

Epic 8: Location-Based Automation

User Story 12: As Catarina, I want the water to get running when I leave my job, saving me the trouble of waiting more for my bath to be ready. (Low)

3 Architecture notebook

Key requirements and constraints

The HomeMaid system is designed to provide users with an efficient and secure platform for managing smart home devices. To ensure the system meets user expectations and functions effectively in real-world scenarios, several technical and functional requirements have been identified:

- The HomeMaid system must support a wide range of devices from different manufacturers.
- Since HomeMaid handles sensitive information (such as the status of doors and security cameras), it is crucial to ensure that all communications between users and devices are secure.

- The system must be capable of sending real-time notifications and alerts to users when anomalous conditions are detected, such as excessive energy consumption or unexpected door openings.
- Users should be able to remotely access the HomeMaid system via internet-connected devices, such as smartphones or computers.
- The system must be designed to handle multiple users and devices simultaneously, ensuring it can scale as more devices are added
- The system must integrate with third-party services, such as voice assistants (e.g., Alexa or Google Assistant), weather APIs for automation, or security systems.
- HomeMaid should allow users to create custom automation rules (e.g., turning off lights after 10 minutes of inactivity).

Architetural view

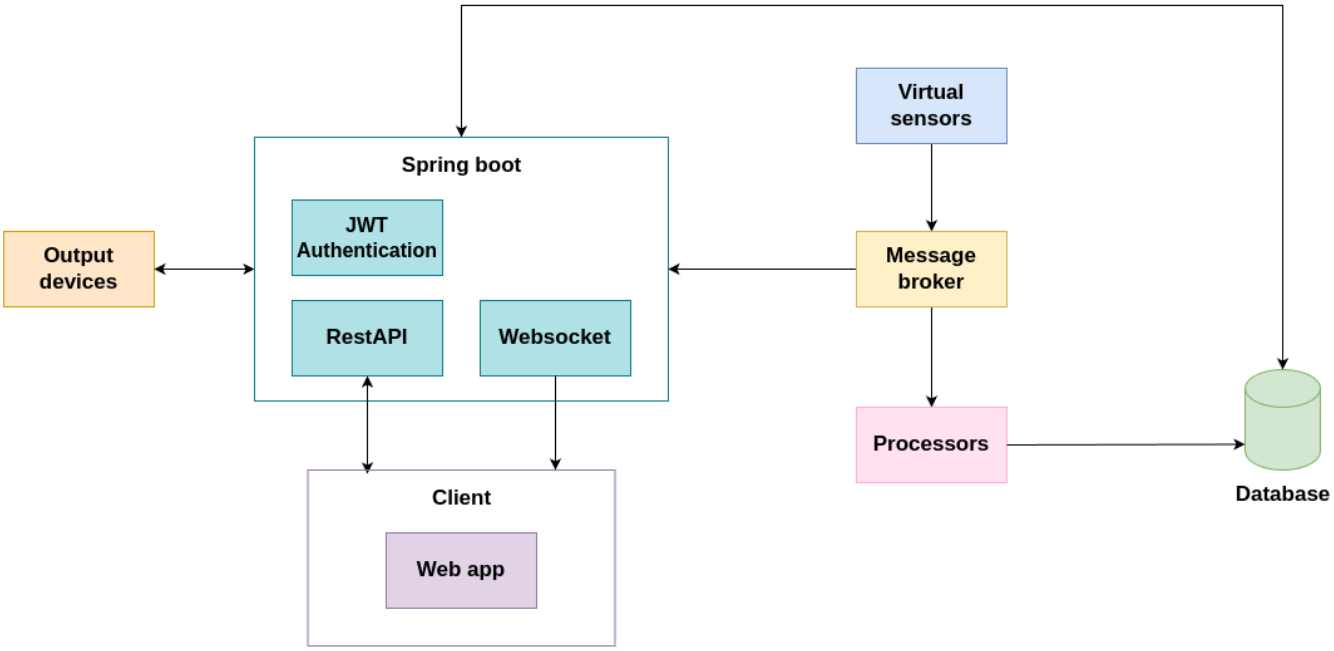
The HomeMaid system will utilize **virtual sensors** simulated by agents in **Java** to represent devices like temperature and humidity sensors. These sensors generate data in real time and send it to **RabbitMQ**, a message broker that routes the data through queues to dedicated processors. The processed data is then stored in a **MongoDB** database through a middleware built with **Spring Boot**, which also handles requests for historical and aggregated data.

In addition to sensors, **output devices** like smart lights and thermostats allow users to control home automation in real time. These devices execute commands received from users via the **Spring Boot** middleware. For instance, users can turn on lights or adjust the thermostat, and the output devices will respond to these commands. Real-time updates are sent back to the front end via **RabbitMQ-WebSocket**, ensuring users receive immediate feedback when device states change.

Security is managed through **Spring Boot** using **JWT authentication**, with tokens stored in the browser's local storage. Access to sensitive endpoints is restricted to admin users, and passwords are securely encrypted using **Bcrypt**.

The front end, developed with **ReactJS**, **Vite**, **TailwindCSS**, and **DaisyUI**, focuses on a sleek and user-friendly interface. Real-time updates are achieved via **WebSockets**, while historical data can be accessed via the **REST API**.

In summary, HomeMaid integrates **virtual sensors for monitoring**, **output devices for control**, a message broker, secure authentication, and a responsive front end to provide a scalable smart home management solution.

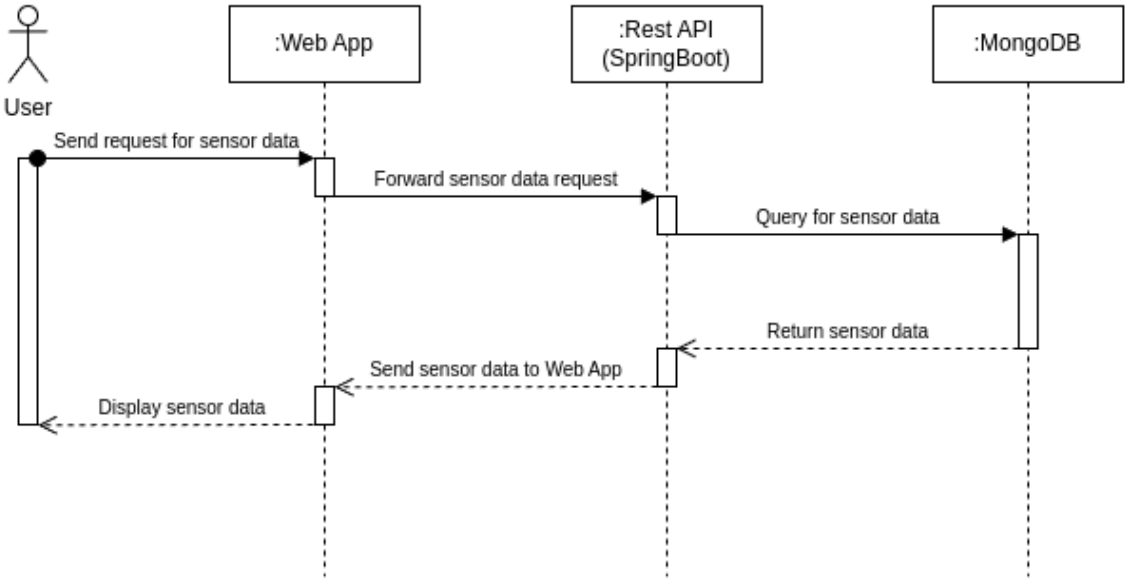


Module interactions

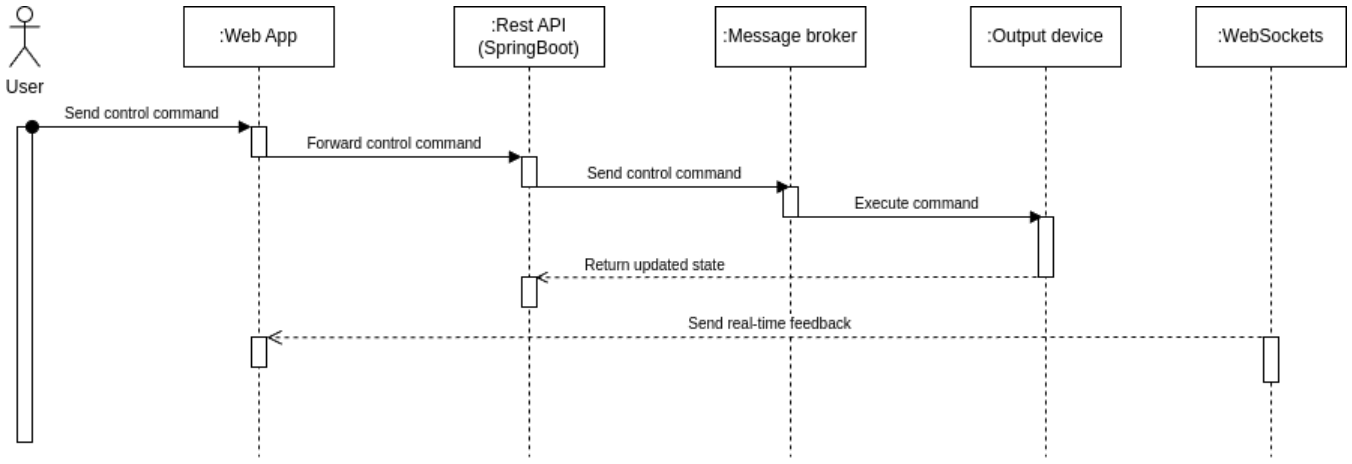
The virtual sensors send monitoring data, such as temperature and humidity, to RabbitMQ, which acts as a broker, distributing this data to the processors. The processors handle and analyze the data, storing the results in MongoDB. Users interact with the system through the Web App, sending commands via the REST API to Spring Boot, which authenticates and processes the requests. Spring Boot sends commands to the output devices, which execute the actions and return their updated state. This state is sent back to the client via WebSockets for real-time feedback. Additionally, users can request historical data, which Spring Boot retrieves from MongoDB and displays on the interface.

Examples of interactions between the modules:

1. **Accessing Sensor Data (Temperature, Humidity, etc.):**
    - In this interaction, the authenticated user requests current sensor data through the Web App. The request is forwarded to the Spring Boot REST API, which retrieves the relevant sensor data from the MongoDB database. Once the data is retrieved, it is returned to the Web App and displayed to the user. This interaction focuses on simple data retrieval without requiring complex processing.

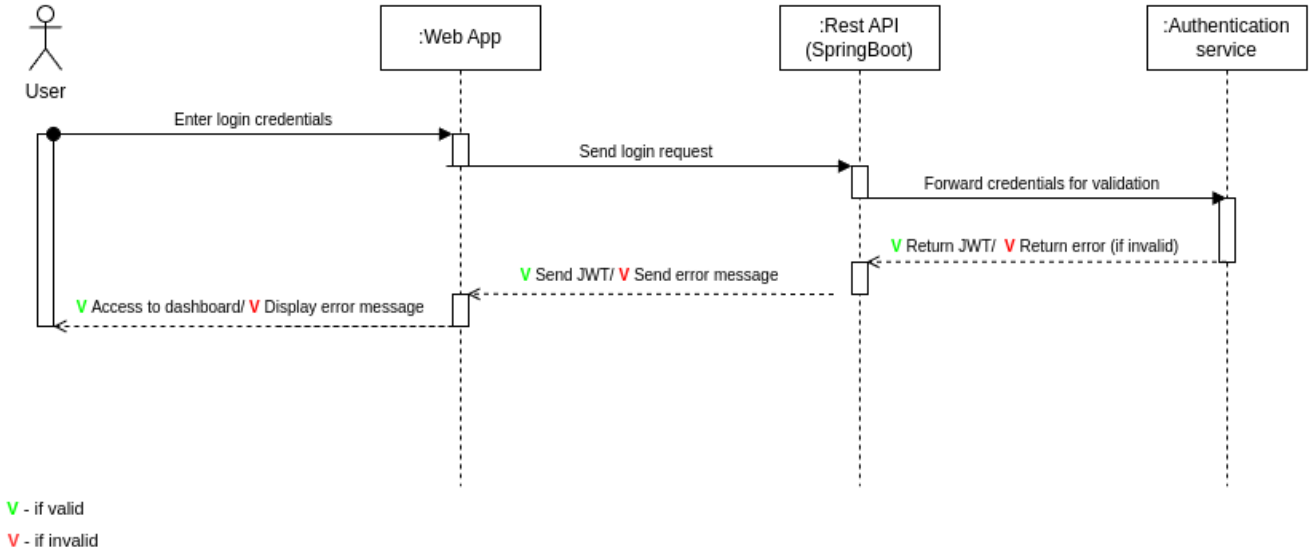


2. **Controlling Output Devices (Turning on Lights, Adjusting Thermostat, etc.):**
    - In this interaction, the authenticated user sends a command through the Web App to control an output device (e.g., turning on a light or adjusting the thermostat). The command is forwarded to the Spring Boot REST API, which processes and authenticates the request. Spring Boot then sends the command to the corresponding output device via a message broker. Once the device executes the action, it sends its updated state back to Spring Boot, which forwards the response to the Web App using WebSockets, providing real-time feedback to the user.



3. **User Login Interaction (Success/Failure):**
    - In this interaction, the user enters their login credentials in the Web App. The Web App forwards the credentials to the Spring Boot REST API, which in turn sends them to the Authentication Service for validation. If the credentials are valid, the Authentication Service returns a JWT token,

granting the user access to the dashboard. If the credentials are invalid, an error message is returned to the Web App, which informs the user that the login attempt failed.



4 Information perspetive

<which concepts will be managed in this domain? How are they related?>

<use a logical model (UML classes) to explain the concepts of the domain and their attributes>

5 References and resources

<document the key components (e.g.: libraries, web services) or key references (e.g.: blog post) used that were really helpful and certainly would help other students pursuing a similar work>