Group

PL1

- Ahmed Selim Hammami (2023239425)
- Elyot Hamon (2023255505)
- João Manuel Monteiro Penacho (2006124245)
- Thiago Lütz Dias (2023171576)

The system could be used within a domestic / business environment. The following scenarios were thought as the necessities of a retail store. They aim to exemplify what reasons may make a store to buy such a system.

Scenarios

The main problem is employee interaction with the customers. The first scenario is related to how the work and workplace may affect the employees mood, and as a consequence the interaction.

The second one is associated with how our employees use their working hours. We want to maximize the time they have for interacting with the client and diminish the burden of managing products / resources.

Self opening / closing

Description

The necessity of having employees turn everything on in the shop sometimes leaves them in a bad mood (it's tedious work). As a result of that, they may be rude to customers. On the other hand, the last shift employees are thinking about how much they will still need to work after the store closes, making them not want to interact with the customers, due to how tired they are. We want to improve their working conditions.

Types of Devices

- cleaning robots

- smart lights
- smart plugs (automatically turn on machines)
- smart AC / Heater

Inventory Management

Description

Keeping track of products and inventory is a necessity, however, it takes a lot of time from our employees. This time is time they are not interacting and convincing customers to buy from us. If we could lower the time our employees spend on inventory, they would have more time for dealing with customers. Besides that, we could use data about how fast / slow our products are sold and be more efficient in our re-supllying.

Types of Devices

- Inventory system (software but thought it was important enough to be here)
- RFID tags / beacons (location, humidity or temperature (niche))
- Shelf sensors (detect when a product has been picked up)
- Digital price tags (with storage availability)

Stakeholders

- *SH1* - Store owner
- *SH2* - Store employee
- *SH3* - Store manager
- *SH4* - Development team
- *SH5* - Hosting platform

Architectural Drivers (Requirements)

Functional Requirements

- *FR1* - Device Integration: allow integration with any IoT device of the main brands (e.g

Google, Alexa, LG, Apple)

- *FR2* - Extendability: allow developers to extend their smart home/office through access to an API
- *FR3* - Interface: the system must be available as a website and as a mobile app
- *FR4* - Software Integration: allow integration with major inventory management software
- *FR5* - Control: allow the user to schedule when the shop should "self open" and "self close", and which machines should be turned on/off
- *FR6* - Metrics: the system must collect inventory data and display metrics, such as the item most sold, the item that gave most profit, etc
- *FR7* - Customizability: the users must be able to add, delete, and edit devices; give custom names to them, and reorder them
- *FR8* - Inventory: automatically update number of available items

Quality Attributes

- *QA1* - Availability: ensure that the system will be available for atleast 95% of working hours
- *QA2* - Responsiveness: guarantee that the system will take at most 1s of computation time, disregarding network
- *QA3* - Privacy: ensure that the data is well protected and is not available for unauthorized people
- *QA4* - Security: guarantee that the system is secure against malicious users and attempts of unwanted actions by both local and external network users
- *QA5* - Compatibility: assure that the mobile application works on both android and iOS devices, and the website works in all Chromium based browsers
- *QA6* - Host: the system must have the option of being self hosted
- *QA7* - Load: when not self hosted, the system must comfortably deal with 5 thousand requests/minute

Constraints

- *C1* - Data: the system must abide to data protection laws
- *C2* - Cost: the upkeep of the system must not surpass $5.000/month
- *C3* - Deploy: when not self hosted, the system must be deployed to AWS
- *C4* - Language: the API must be implemented in Go

- *C5* - Process: the development must follow the SCRUM principles, with one of the stores employees as domain expert