UNIVERSIDADE da MADEIRA

SOFTWARE REQUIREMENT SPECIFICATION

for

Masters Thesis Application

Version alpha 0.8

Written by Nelson Vieira

March 21, 2023

Contents

1 Introduction

This report aims to define the details and principles of the privacy assisting IoT application with the ultimate goal of assisting in the development of the application. Defining the scope of the project helps to understand how to better implement the application using this document as its base. Stakeholders tell us who communicates with the system directly or indirectly. Business requirements will have to be defined to know in detail the possibilities of value for both sides. The swimlane gives a general understanding of the main case and what cause has each action, the contextual diagram helps to give an understanding of all actions between the system and the active parties. The data flow diagram, lets you know in detail what results from the particular activities. Finally, the technology requirements allow to know how the system will look like and to define the budget of the work.

1.1 The scope and vision of the project

This project is carried out in the context of the Master's thesis in Informatics Engineering, which aims to create a mobile application that would provide information about IoT devices in the users' surroundings like the type of information these devices collect and what privacy options are available. The main objective of this application is to give users another option in order to protect their private data. The application will show the geolocation of the IoT devices, what type of device it is, what type of data is being collect by the device. The application will not detect the devices by itself, this will be done by the users themselves. As for competition there are other similar online systems with the same scope as this project. The application offers an easier way to search for information about the IoT devices that are around users' location.

1.2 Stakeholders

A stakeholder can be a person, group, or organisation that is involved in the project, is affected by its process and outcome or can influence its process and outcome. Stakeholders can be internal or external to the project team and the organisation. [1]
It is important to identify the stakeholders to make sure to get all the right requirements for the project and to develop a system that can match the proposed problem well.
The following stakeholders are identified in this project:

- **Programmer/designer of the application**: The programmers are the ones who will create the application and even if they do not use it they are directly related to it.

- **IoT Device Owners**: These device owners will be priority stakeholders being that the application in good part will be directed to them, device owners have an indirect influence.
- **Application Users**: The users will be the main focus of this project, they are the ones that provide the information that will be inserted in the application, since they can change the course of the project they have a direct influence.
- **Thesis advisor**: Because this application is implemented in the context of a master's thesis, the advisor has an indirect influence during the implementation and the final product.
- **University of Madeira**: Because this application is implemented in the context of a master's thesis, the university as an organization has an indirect influence on the final product.
- **Legislation**: The legislation in relation to the privacy of the collected data allows to impose rules on the use of the data. It has an indirect influence.

2 System Relations with Stakeholders

2.1 Contextual diagram

The contextual diagram aims to establish links between the system and the other actors that interact with the application.
Identifies the identities external to the application that interact with the system with data and control between the external entities and the application.

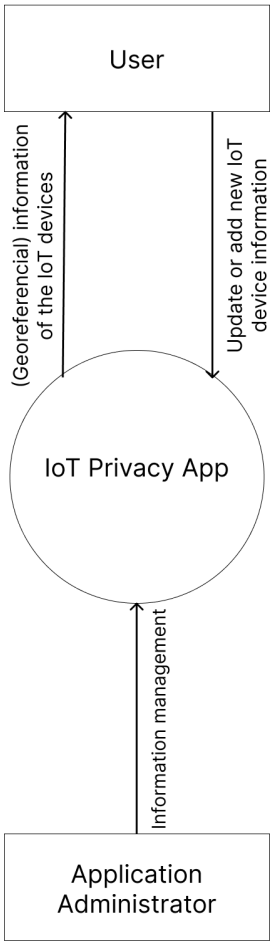

Figure 2.1: IoT Privacy Contextual Diagram

The contextual diagram is based on showing the actions performed with the application.

User:

→ Receives:
- (Georeferencial) information of the IoT devices
→ Sends:
- Update or add new IoT device information

Application Administrator (Programmer):

→ Receives:
- All information related to the application
→ Sends:
- Information management

2.2 Data flow diagram

A data flow diagram shows how information flows between the various entities in the system and their relationships.

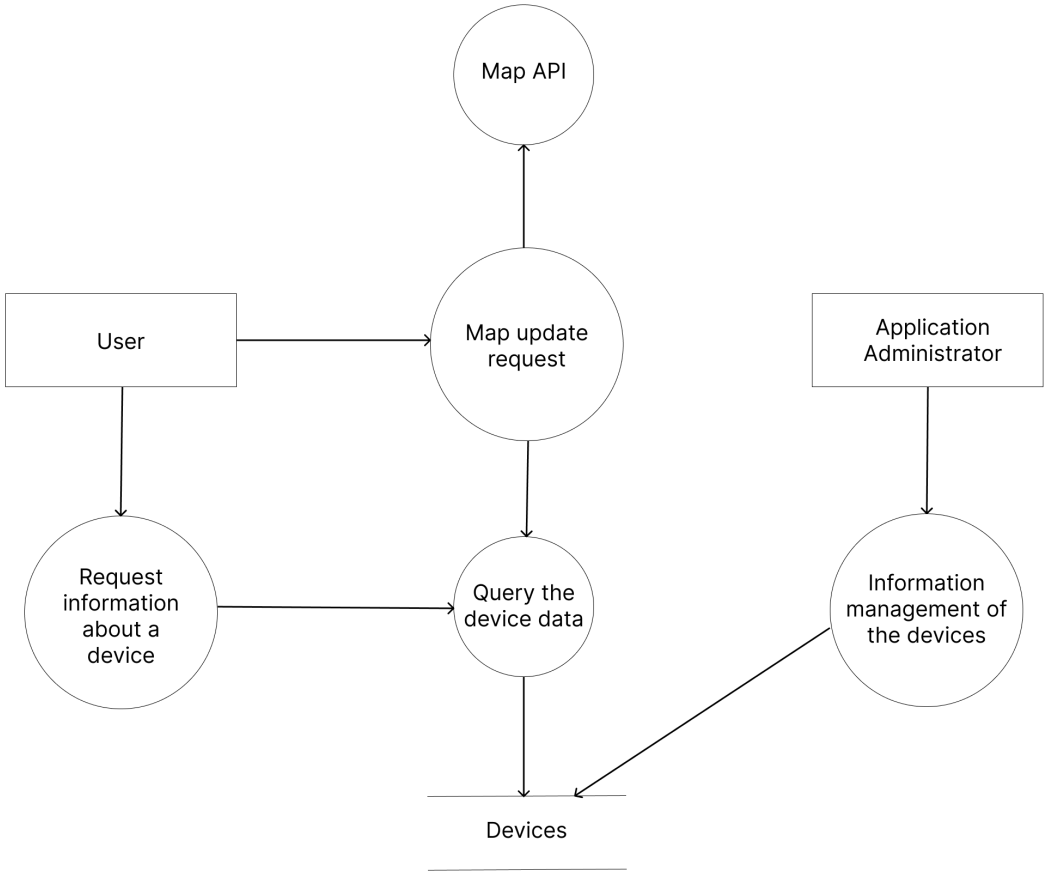

Figure 2.2: Data flow diagram

The user can browse the application map, which will be created with an API, and see the locations of the IoT devices, the user can also look up information about the devices by clicking on a device on the map or by searching for the device in the application. The administrator of the application is responsible for its maintenance (correcting or deleting incorrect data), security and for the veracity of the information.

2.3 Swimlane diagram

A swimlane diagram is a type of flowchart in which processes and decisions are grouped into lanes. Parallel lines divide the diagram into lanes, each lane being assigned to people/groups and application.

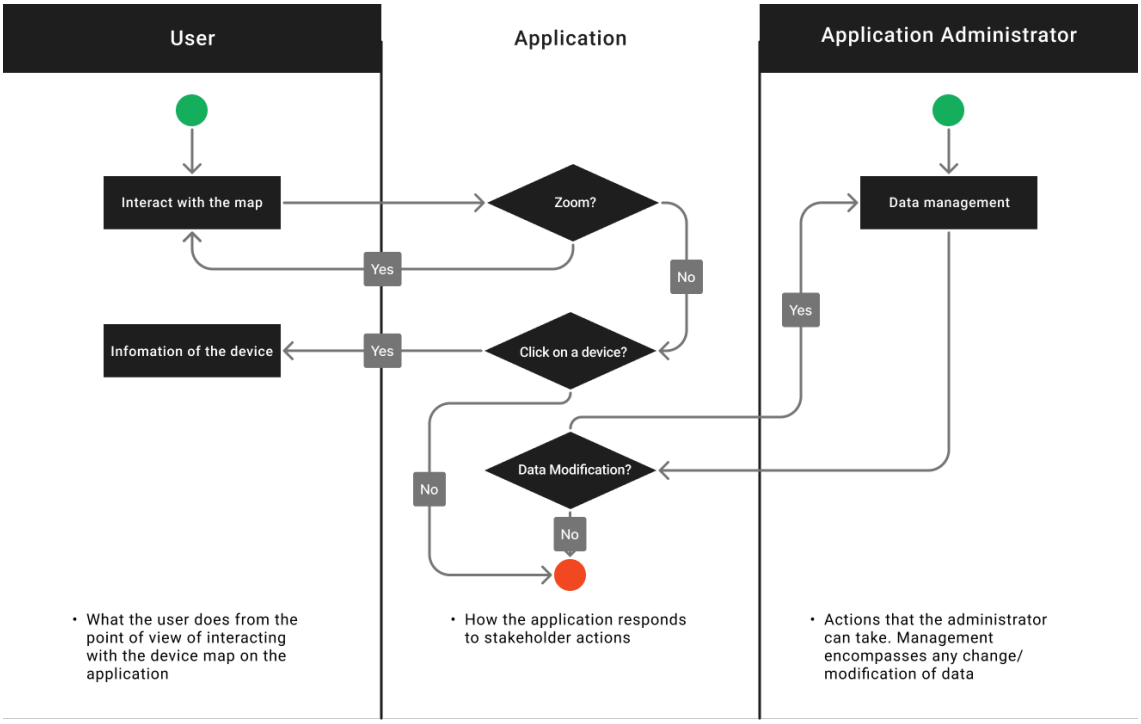

Figure 2.3: Swimlane diagram

This swimlane diagram represents a high level view of a possible user interaction from the application's map, in this map the user can view the location of IoT devices and can see more information about a particular device by selecting it on the map. The application administrator, as mentioned above, can modify the devices' data.

3 System Features and Requirements

3.1 Business requirements

Business requirements describe in business terms what must be delivered or achieved to deliver value. It is what defines the way of doing business, reflecting the internal policy, the defined process and/or the basic rules of conduct. In other words, it is a set of instructions that users already follow and that the system to be developed must contemplate. Restrictions, validations, conditions and process exceptions are classic examples of business rules. A business rule will not necessarily be reflected in the system as a functionality, but it will certainly determine the behaviour of one or more functionalities of the system.
No business requirements have been identified for this project.

3.2 Technology requirements

Technology requirements describe what both hardware and software must be used in order for a system to be realizable. In terms of hardware, it describes what kind of physical components are needed for the software to work. The software to be chosen must take into account the hardware that has been chosen and what is intended by the stakeholders. This has implications for how the system is implemented.
The technology requirements that have been identified for this project are as follows:

- Firestore or similar database server
- Flutter framework with Dart being the main programming language
- Accessible on any smartphone (iOS or Android)
  Note: No web based version it to be available.

These requirements have been chosen so that the system is available to as many users as possible regardless of the hardware they use. The database will allow to store the information that the users provide about the IoT devices. The application will be developed with Flutter since it uses ahead of time and just in time compilation with Dart as its programming language. Flutter has better performance than React Native or a PWA stack and as such it is the chosen framework for this application.

3.3 Requirements table

The requirements table identifies each feature and links each feature to an origin.
This is important to make it easier to manage requirements in the future. Knowing the

origin of the creation of the requirements makes it easier to refer back to the source to clarify any questions.
A table was created with all the features found. For each feature it was identified the users to which it is applied and the source.
For the features found were described the most appropriate requirements for them.

R#	Feature	Applicable stakeholders	Description	Source
1	Navigate the map	User	**User**: The system should allow the user to scroll through the map of devices	[2]
2	Select device on the map	User	**User**: The system should allow the user to select a device on the map to view more information	[2]
3	Query devices through parameters	User	**User**: The system should allow the user to consult devices of only a certain type, data collected, general location	[2]
4	Query statistics of the devices	User	**User**: The system should allow consulting statistics of devices	[2]
5	Add a device	User	**User**: The system should allow the user to add a new device with name, category, data collected, location, etc.	[2]
6	Delete a device	App Administrator	**App Administrator**: The system should allow the administrator to delete a device	[2]
7	Edit a device	User	**User**: The system should allow the user to change some data of a device	[2]
8	Create account	User	**User**: The system shall allow a user to create an account.	[2]
9	Select privacy choices	User	**User**: The system shall allow the user to select their privacy choices for a certain device if the device allows for it.	[2]
10	See more information about privacy in IoT	User	**User**: The system shall allow the user to check what the terms used in the application mean.	Survey results

Table 3.1: Requirements table

3.4 Functional requirements

Functional requirements define the functions of a system or its components, where functions are specifications or behaviours between system outputs and inputs. [1] These describe what developers have to implement so that users can complete tasks (user requirements), which in turn satisfy business requirements. [3] Functional requirements are essential to the success of a project. After building the tracing table the functional requirements that were needed were extracted for each feature and grouped appropriately according to the following groups:

3.4.1 User Requirements

RU1.1 - The system should allow the user to scroll through the devices map;
RU1.2 - The system shall allow the user to select a device on the map to view more information;
RU1.3 - The system shall allow the user to search for a device by name;
RU1.4 - The system should allow the user to consult devices of only a certain category, data collected, etc.;
RU1.5 - The system must allow consulting statistics of the devices;
RU1.6 - The system should allow the user to add a new device with name, category, data collected, location, etc.;
RU1.7 - The system should allow the user to change some data of a device;
RU1.8 - The system shall allow a user to create an account.

3.4.2 Administrator Requirements

RM.2.1 - The system shall allow the administrator to delete a device;

3.4.3 System requirements

RS.4.1 - The system must show statistics related to the devices;

4 Use cases

4.1 Use cases diagram

The use cases diagram provides a high level visualisation of the user requirements. The box represents the system boundary. An actor's arrow for a use case indicates that he is the primary actor for it.
The primary actor initiates the use case and derives the primary value from it. An arrow goes from a use case to a secondary actor, where it participates in some way in the successful completion of the use case. [3]

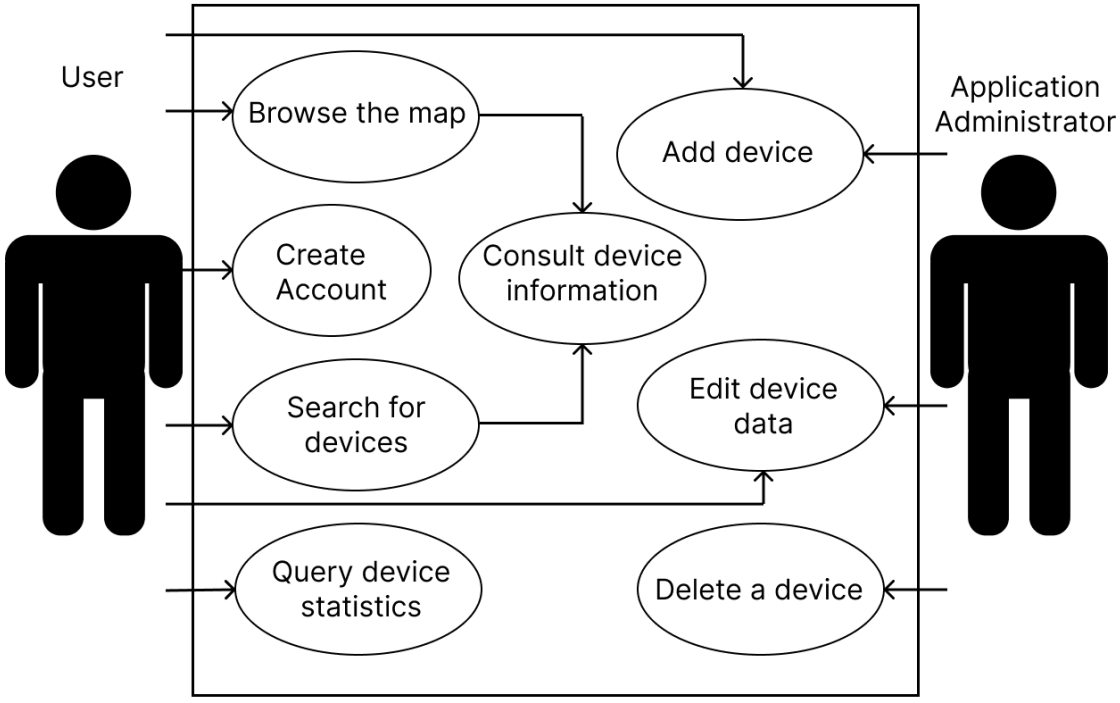

Figure 4.1: Use cases diagram

4.2 Use cases

In Software Engineering, a use case is a type of classifier representing a coherent functional unit provided by the system, subsystem, or class manifested by sequences of interchange-

able messages between systems and one or more actors. [4]
Using this technique describes the tasks that users need to perform with the system or the user-system interaction that may be important to some stakeholders. They also help in testing by checking that the functionality has been implemented correctly. The use case uses UML (Unified Modeling Language) notation.

ID and Name:	UC-01 Device information query
Created By:	Nelson Vieira 20/02/2023
Primary Actor:	User
Description:	The user makes a device information query
Trigger:	The user wants to search device information
Pre-conditions:	N/A
Post-conditions:	POST-1. The user finds device information
Normal Flow:	**1.0 Query information of a device on the map** 1. The user browses the map 2. The user clicks on the icon to show some information about the device 3. The user clicks on the device pop-up
Alternative Flow:	**1.1 Device information search** 1. User enters device name 2. The user chooses the device he wants from a list generated from the search performed
Alternative Flow:	**1.2 Alternative search for information from a device** 1. The user selects one of the parameters: a) Category b) Data collected 2. The user chooses the device he wants from a list generated from the search carried out
Exceptions:	**1.0.E1 The API is not working** 1. The system displays an alert message: "We are having connection problems, please wait for a while"
Priority:	High
Business Requirements:	N/A
Assumptions:	N/A

Table 4.1: Use case 1 - device information query

ID and Name:	UC-02 Device statistics query
Created By:	Nelson Vieira 20/02/2023
Primary Actor:	User
Description:	The user queries the statistics of the devices
Trigger:	The user wants to find statistics of devices
Pre-conditions:	N/A
Post-conditions:	POST-1. The user finds statistics of devices
Normal Flow:	**2.0 Device statistics query** 1. User selects statistics tab 2. The user can only select certain parameters, such as: a) Category b) Location
Alternative Flow:	N/A
Exceptions:	N/A
Priority:	High
Business Requirements:	N/A
Assumptions:	N/A

Table 4.2: Use case 2 - statistics query

ID and Name:	UC-03 Add a device
Created By:	Nelson Vieira 22/02/2023
Primary Actor:	User
Description:	Addition of a new IoT device in the application
Trigger:	The user wants to add a new IoT device
Pre-conditions:	N/A
Post-conditions:	POST-1. A new IoT device is added to the application
Normal Flow:	**3.0 Add a device** 1. The user enters the following data of a new IoT device: a) Name b) Type of data collected c) Category d) Photos 2. The user clicks submit 3. The user adds the location of the IoT device on the map
Alternative Flow:	N/A
Exceptions:	**3.0.E1 The device is already in the database** 1. The system displays an error message
Priority:	High
Business Requirements:	N/A
Assumptions:	N/A

Table 4.3: Use case 3 - add a device

ID and Name:	UC-04 Edit a device's data
Created By:	Nelson Vieira 22/02/2023
Primary Actor:	User
Description:	Editing the data of an IoT device in the application
Trigger:	The user wants to edit an IoT device's data
Pre-conditions:	N/A
Post-conditions:	POST-1. The data that has been changed appears in the application
Normal Flow:	**4.0 Edit a device's data** 1. The user can change any of the following device data: a) Name b) Type of data collected c) Category d) Photos 2. The user clicks on submit
Alternative Flow:	N/A
Exceptions:	**4.0.E1 The device to be edited has been deleted in the meantime** 1. The system displays an error message 2. The system prohibits editing
Priority:	High
Business Requirements:	N/A
Assumptions:	N/A

Table 4.4: Use case 4 - edit a device's data

ID and Name:	UC-05 Delete a device
Created By:	Nelson Vieira 22/02/2023
Primary Actor:	App Administrator
Description:	Deleting a device in the application
Trigger:	The administrator wants to delete a device
Pre-conditions:	PRE-1. The device to be deleted must be in the application's database
Post-conditions:	POST-1. The device is deleted from the application
Normal Flow:	**5.0 Delete a device** 1. The administrator deletes a device, through: a) ID of device b) Name of device 2. The administrator confirms the deletion 3. The system deletes the device
Alternative Flow:	N/A
Exceptions:	**5.0.E1 The device to be deleted no longer exists in the database** 1. The system displays an error message 2. The system prohibits deletion
Priority:	High
Business Requirements:	N/A
Assumptions:	It is assumed that the administrator has database access

Table 4.5: Use case 5 - delete a device

5 Requirement prioritisation

Regarding the prioritization of requirements, the Quality Function Deployment technique proposed by Cohen in 1995 [5] is used to estimate the priority of a group of requirements. This is based on the benefit of including a feature/requirement, the penalty of not including it, and also the cost and risks associated with implementation. With the MoSCoW method [6], the initial features are reduced to facilitate the use of the Quality Function Deployment table.
In this approach the values 0 and 1 are used. In case of 1 it means that the column requirement/feature is a higher priority than the row one and if it is 0 the opposite is true.

MoSCoW Table	1	2	3	4	5	6	7	8	9
1		0	0	0	0	0	1	1	0
2	1		1	1	0	1	1	0	0
3	1	0		0	0	1	1	0	0
4	1	0	1		0	1	1	0	0
5	1	1	1	1		1	1	1	1
6	0	0	0	0	0		1	0	0
7	0	0	0	0	0	0		0	0
8	1	1	1	1	0	1	1		0
9	1	1	1	1	0	1	1	1	
Total	6	3	5	4	0	7	8	2	1

Table 5.1: Prioritisation table using the MoSCoW technique

After this initial selection, a prioritisation table is created where it is measured, on a scale of 1 to 9, to rank the benefit and penalty of each requirement. The cost and implementation risk associated to each feature is also estimated.

Feature		Benefício relativo	Penalização relativa	Valor Total	Valor %	Custo relativo	Custo %	Risco relativo	Risco %	Prioridade
Delete a device	7	7	6	20	10,10	1	2,56	1	2,70	1,92
Add a device	6	8	8	22	11,11	1	2,56	1	2,70	2,11
Navigate the map	1	9	9	27	13,64	8	20,51	1	2,70	0,59
Search devices	3	8	8	26	13,13	6	15,38	9	24,32	0,33
Query devices through parameters	4	9	7	25	12,63	4	10,26	2	5,41	0,81
Select device on map	2	7	7	25	12,63	5	12,82	1	2,70	0,81
Edit a device	8	6	6	18	9,09	1	2,56	9	24,32	0,34
Consult devices' statistics	5	7	6	18	9,09	5	12,82	4	10,81	0,38
Create account	9	6	5	17	8,59	8	20,51	9	24,32	0,19
Total		67	62	**198**	100,00	**39**	100,00	**37**	100,00	

Table 5.2: Features prioritisation table

Using this method we get the requirements sorted by priority:

Rank	Feature	# Feature	Prioridade
1	Add a device	6	2,11
2	Delete a device	7	1,92
3	Query devices through parameters	4	0,81
4	Select device on map	2	0,81
5	Navigate the map	1	0,59
6	Consult devices' statistics	5	0,38
7	Edit a device	8	0,34
8	Search devices	3	0,33
9	Create account	9	0,19

Table 5.3: Highest priority requirements ordered

5.1 Acceptance Criteria

To make it easier to test whether the highest priority features that were chosen previously were well implemented, these acceptance criteria were created for each of them. These criteria help us understand the minimum conditions for this application to be considered an MVP, i.e., for this project to have the minimum possible requirements in order for it to be considered production ready.
For these acceptance criteria the following was considered:

- High-level functionality that must be present for the system to be usable
- Non-functional criteria and quality metrics that have to be satisfied
- Possibility of open problems or defects (we can guarantee that no defects or TBD is present for the system to be accepted)
- Legal or contractual restrictions (that have to be met for the system to be accepted)

5.1.1 Features

Delete a device

- The system must allow the deletion of a device

Add a device

- The system allows the user to add a device

Navigate the map

- The system can represent the devices on the map

Search devices

- The system allows you to search for devices by name

Query devices through parameters

- The system allows searching devices by the following parameters: type, data collected, category

Select device on map

- The system allows you to select a device on the map

Edit a device

- The system allows the editing of a device
- The system saves in the database the changes that have been made

Consult devices' statistics

- The system allows the user to consult statistics concerning the devices

Create account

- The system allows the creation of an account
- The user has to enter its username, email and a password
- The system can detect whether the email is already in use
- The system can send a profile creation confirmation email
- User can confirm profile creation

5.2 Prototype

For the prototype several drafts are first made with design tools, such as Figma or . Then, with the tools previously mentioned in **Technology Requirements**, an application is created.

Bibliography

[1] R. Fulton and R. Vandermolen, "Chapter 4: requirements—writing requirements," *Airborne Electronic Hardware Design Assurance: A Practitioner's Guide to RTCA/DO-254*, pp. 89–93, 2017.

[2] N. Vieira and M. Barreto, "Empowering users' privacy rights in the internet of things," Funchal, Madeira, 2023.

[3] K. Wiegers and J. Beatty, *Software Requirements*. Pearson Education, 2013.

[4] A. S. f. P. Affairs, "Use cases," https://www.usability.gov/how-to-and-tools/methods/use-cases.html, accessed: 2022-04-08.

[5] L. Cohen, *Quality function deployment: how to make QFD work for you*. Prentice Hall, 1995.

[6] D. Clegg and R. Barker, *Case method fast-track: a RAD approach*. Addison-Wesley Longman Publishing Co., Inc., 1994.