UNIVERSIDADE da MADEIRA

SOFTWARE REQUIREMENT SPECIFICATION

for

Masters Thesis Application

Version 0.2

Written by Nelson Vieira

February 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 the main cases that will be entered. Stakeholders tell us who communicates with the system directly and 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 their 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 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 you 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:

- **Programmers/designers 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.
- **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 Software requirements

2.1 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.

2.2 Technology requirements

Technology requirements describe what both hardware and software must be used in order for a system to be realisable. 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. For example, it is decided that the system to be implemented should run on iOS smartphones, this implies that the active stakeholder has to have an iOS smartphone to be able to use the system.
The technology requirements that have been identified are as follows:

- Firestore or similar database server
- Programming languages: Dart
- Accessible on any smartphone

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 and as such it is the chosen framework for this application.

3 Contextual diagram

The contextual diagram aims to establish links between the system under development 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. The contextual diagram is based on showing the actions performed with the application.

User:

→ Receives:
- Profiles/Information of the IoT devices
- Georeference map of the IoT devices

→ Sends:
- Information update

Application Manager:

→ Receive:
- Information from IoT devices

→ Sends:
- Information management
- Geo-referenced map management

3.1 Data flow diagrams

A data flow diagram shows how information flows between the various entities in the system and their relationships. 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 manager of the application is responsible for its maintenance (adding new data, correcting or deleting incorrect data) and security and for the veracity of the information.

3.2 Swimlane diagram (high level)

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. 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 manager, as mentioned above, can modify the devices' data.

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	Text
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	Text
3	Search devices	User	**User**: The system should allow the user to search for devices by name	Text
4	Query devices through parameters	User	**User**: The system should allow the user to consult devices of only a certain type, data collected, general location	Text
5	Query statistics of the devices	User	**User**: The system should allow consulting statistics of devices	Text
6	Add a device	User	**User**: The system should allow the user to add a new device with name, category, data collected, location, etc.	Text
7	Delete a device	App Manager	**App Manager**: The system should allow the manager to delete a device	Text
8	Edit a device	User	**User**: The system should allow the user to change some data of a device	Text
9	Create account	User	**User**: The system shall allow a user to create an account.	Text

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. [2] 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 Manager Requirements

RM.2.1 - The system shall allow the manager to delete an entity;

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. [2]

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 interchangeable messages between systems and one or more actors. [3]
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 - entity 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

4.3 O âmbito e visão do projeto

Text

4.4 Stakeholders

5 Requisitos de software

6 Priorização de requisitos

Em relação à priorização de requisitos, é usada a técnica Quality Function Deployment proposta por Cohen em 1995, que serve para estimar a prioridade de um grupo de requisitos. Baseado no benefício da inclusão de uma feature/requisito, da penalização da mesma não ser incluída e ainda o custo e riscos associados à implementação. Com o método MoSCoW, a partir dos features iniciais é feita ainda uma redução para facilitar o uso da tabela de Quality Function Deployment.
Nesta abordagem são usados os valores 0 e 1. No caso de 1 significa que o requisito/feature da coluna é mais prioritário que o da linha e se for 0 o contrário verifica-se.

Tabela MoSCoW	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 6.1: Tabela de priorização usando a técnica MoSCoW

Após esta seleção inicial é criada uma tabela de priorização onde é pedido , numa escala de 1-9, para classificar o benefício e penalização de cada requisito. É também estimado o custo e risco de implementação associado a cada feature.

Feature		Benefício relativo	Penalização relativa	Valor Total	Valor %	Custo relativo	Custo %	Risco relativo	Risco %	Prioridade
Eliminar uma entidade	7	7	6	20	10,10	1	2,56	1	2,70	1,92
Adicionar uma entidade	6	8	8	22	11,11	1	2,56	1	2,70	2,11
Navegar no mapa	1	9	9	27	13,64	8	20,51	1	2,70	0,59
Pesquisar entidades	3	8	8	26	13,13	6	15,38	9	24,32	0,33
Consultar entidades através de parâmetros	4	9	7	25	12,63	4	10,26	2	5,41	0,81
Selecionar entidade no mapa	2	7	7	25	12,63	5	12,82	1	2,70	0,81
Editar uma entidade	8	6	6	18	9,09	1	2,56	9	24,32	0,34
Consultar estatísticas das entidades	5	7	6	18	9,09	5	12,82	4	10,81	0,38
Criar conta (associada à entidade)	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 6.2: Tabela de priorização das features

Utilizando este método obtemos os requisitos ordenados por prioridade:

Rank	Feature	# Feature	Prioridade
1	Adicionar uma entidade	6	2,11
2	Eliminar uma entidade	7	1,92
3	Consultar entidades através de parâmetros	4	0,81
4	Selecionar entidade no mapa	2	0,81
5	Navegar no mapa	1	0,59
6	Consultar estatísticas das entidades	5	0,38
7	Editar uma entidade	8	0,34
8	Pesquisar entidades	3	0,33
9	Criar conta (associada à entidade)	9	0,19

Table 6.3: Requisitos mais prioritários ordenados

6.1 Critérios de aceitação

Para mais facilmente conseguir-se testar se as features mais prioritárias que foram escolhidas anteriormente foram bem implementadas foram criados estes critérios de aceitação para cada uma delas. Estes critérios ajudam-nos a perceber as condições mínimas para que esta aplicação possa ser considerada um MVP, "minimum viable product", ou seja, para que este projeto tenha os requisitos mínimos possíveis de forma a que este seja considerado production ready.
Para estes critérios de aceitação foi considerado o seguinte:

- Funcionalidade de alto nível que tem de estar presente para que o sistema seja usável
- Critérios não funcionais e métricas de qualidade que têm de ser satisfeitas
- Possibilidade de problemas em aberto ou defeitos (podemos garantir que nenhum defeito ou TBD esteja presente para o sistema ser aceite)
- Restrições legais ou contratuais (que têm de ser satisfeitas para o sistema ser aceite)

6.1.1 Features

Eliminar uma entidade

- O sistema deverá permitir a eliminação de uma entidade

Adicionar uma entidade

- O sistema permite ao Gestor do Portal adicionar uma entidade com os seguintes parâmetros: nome, NIF, morada, contactos (site, email, telefone, redes sociais), natureza jurídica, estatuto jurídico, data de criação, âmbito geográfico, área de intervenção

Navegar no mapa

- O sistema consegue representar as entidades da região no mapa

Pesquisar entidades

- O sistema permite pesquisar entidades pelo nome

Consultar entidades através de parâmetros

- O sistema permite pesquisar entidades pelos seguintes parâmetros: NIF, morada, natureza jurídica, estatuto jurídico, data de criação, âmbito geográfico, área de intervenção

Selecionar entidade no mapa

- O sistema permite a selecionar uma entidade no mapa

Editar uma entidade

- O sistema permite a edição de uma entidade
- O sistema guarda na base de dados as modificações que foram realizadas

Consultar estatísticas das entidades

- O sistema permite que o utilizador possa consultar estatísticas relativas às entidades

Criar conta (associada à entidade)

- O sistema consegue criar um novo perfil do tipo Entidade
- A Entidade tem que inserir o seu nome, NIF, morada, nº de telefone, email e uma password
- O sistema consegue detectar se o email já está em uso
- O sistema consegue detectar se o NIF da entidade já está em uso
- O sistema consegue enviar email de confirmação de criação de perfil
- O utilizador consegue confirmar criação de perfil

6.2 Protótipo

Para o protótipo é realizado primeiramente vários drafts com ferramentas de design, como Figma ou . Depois, com as ferramentas referidas anteriormente em **Requisitos de Tecnologia** (PHP, HTML, CSS e Javascript), é criado um website.

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

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