TrackMe project Julián Cuéllar Mangut
Javier Fernández Rodríguez

POLITECNICO
MILANO 1863

Requirement Analysis and Specification Document

Deliverable:	RASD
Title:	Requirement Analysis and Verification Document
Authors:	Julián Cuéllar Mangut, Javier Fernández Rodríguez
Version:	1.0
Date:	11-November-2018
Download page:	https://github.com/javferrod/CuellarFernandez
Copyright:	Copyright © 2018, Julián Cuéllar, Javier Fernández

Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose

The purpose of this document is to give a detailed specification for the TrackMe software product. Along the following pages the goals, requirements, constraints and interfaces of the project will be explained. The intention of this document is not only to be proposed to the customer but to be used as the ground for the development of the product.

This document could be used as a contractual basis for the realisation of the project.

TrackMe enhances the flowing of data from the users to the clients, enabling the companies to make right choices about their users thanks to the analysis of the data. These data may encompass location, heart rate, age and so. The data needs to be recollected in a time basis, allowing the clients to analyse their evolution trough the time.

1.2 Scope

The recollection of the data should be carried out in an automated fashion, using the sensors available in mobile devices such as mobiles (smartphones) and smart-watches.

Such automation should not undermine the privacy rights of the users. The system should provide mechanism for the users to grant or deny their approval for the recollection and treatment of their data.

The main goal of the system is to provide tools for the analysis of the recollected data. Therefore, the system should provide a dashboard and an API to the clients, allowing them to navigate and query the available data.

The goals of the system are summarised in table 1.

ID	Goal
GL1	The system should provide accounting and authorisation for users and clients.
GL2	The system should store the recollected data.
GL3	The system should recollect the data using the sensors available in the users' devices, asking the user directly the information when no sensor is available for recollecting the information (for example, weight).
GL4	The system should recollect the data from the users at time intervals.
GL5	The system should store and display the data in a time series format, allowing the client to consult the changes in the parameters along the time.
GL6	The system should allow the clients to easily query the already recollected data of the users.
GL7	The system should allow the clients to query the data of an specific user.
GL8	The system should allow the clients to subscribe to a query, providing new data as arrives.
GL9	The system should protect the privacy of the users. A data batch displayed to a client should not enable the differentiation between individuals.
GL10	The system should allow users to monitor some of their parameters, alerting the emergency system when any of these parameter gets out of a threshold.

Table 1: Goals

1.3 Definitions, acronyms and abbreviations

1.3.1 Definitions

- **Administrator:** Worker of TrackMe with access to the entire platform without restrictions.
- **Alarm:** A warning given to the user and emergency services when health parameters are lower than necessary.
- **Client:** Individual or company that pays TrackMe to get access to the data of the users of TrackMe.
- **Data batch:** Collection of data from different users that comply with a query based on logical constraints made by a client.
- **Emergency system:** External system that performs the emergency warning functions for the AutomatedSOS service.
- **JJ Software:** Software company responsable of the implementation, operation and maintenance of TrackMe software.
- **Logical constraint:** Boolean sentence made upon a Parameter, the queries are formulated as a set of constraints. The data batch ensuing of a query complies with all the logical constraints of the query.
- **Measure:** Data obtained by the users' devices and communicated to the server. It is classified in parameters.
- **Optimum internet connection:** Connection with at least 10 Mb/s and a latency below 20 ms.
- **Parameter:** Type of data in which the recollected data is organised. These parameters may include blood pressure, location.
- **Payment system:** External system that allows us to make the different payments to clients.
- **Plan ID:** Unique identifier of a Plan. A Plan is an object of Stripe which represents a monthly cost associated to the services offered by TrackMe.
- **Subscription ID:** Unique identifier of a Subscription. A Subscription is an object of Stripe representing the association of a Client with a Plan. When a subscription is created, the client is charged the amount of the Plan in a monthly basis.
- **Android Permission Request:** Message presented to an Android user requesting allowance of the user to collect data from some sensor, as the location.
- **Track:** This is the tracking of data, people or any type of item that can be traced and tracked.
- **Token:** Unique identifier which replace username and password, have a limited duration in time.
- **User:** Individual that installs the TrackMe application and give TrackMe permission to collect and sell their personal data.

1.3.2 Acronyms

- **API**: Application Program Interface.
- **GPS**: Global Positioning System.
- **HTTPS**: Hyper Text Transfer Protocol Secure.

- **SSL**: Secure Sockets Layer.
- **TLS**: Transport Layer Security.

1.3.3 Abbreviations

- **403 error**: Forbidden HTTP error code.
- **bpm**: Beats per minute.
- **CVC**:Card Verification Code
- **M/F**: Male/Female.

1.4 Revision history

- **V 1.0:** First version of the document directed at consumers and developers.

1.5 Reference Documents

The various pages and documents referred to in this document can be found in page 37.

During the reading of the document you will find notation to the different references that have been used.

1.6 Document Structure

The document has been structured in different sections and subsections as can be seen on page 4.

During this section we will try to explain in more detail what is dealt with in each section or subsection.

The document is divided into 5 large sections which are divided into multiple subsections, some subsections are further divided into subsections.

- The first section, is based on an introduction in which talk about the idea that is followed to carry out this document and the project, the objectives that are wanted to reach and that must fulfill the final project as well as different definitions, acronyms or abbreviations that are going to be found throughout the project so that no doubt is generated in the reader.

  This section begins on page 7.

- The second section make a more in-depth perspective of the product by treating in a very exact way which users the application is going to be aimed at, dividing them into different groups to name and observe in a more detailed way their different characteristics, we observe the different functions that the project is going to present and how these are fulfilling the different objectives, also observe in detail what external systems are going to be used and what external problems are found to fulfill these objectives.

  This section begins on page 11.

- The third section gets more into how the application is inside, leaving behind the aerial view of the previous sections. In this section we talk about how the application will look using different designs to see the final idea of both the mobile application and the web application as well as the smartwatch. It also deals with the hardware and software interfaces as well as the different requirements expected from the project at a functional and performance level. It presents the reader with different use cases and diagrams so that he or she has no doubt when using the different services, it deals with the different limitations observed when trying to adapt the applications and services to the different devices and finally it is observed how the system is going to be maintained at the level of security, maintenance, portability and different factors.

This section begins on page 17.

- The fourth section contains the alloy model of the services which will be used in their future development.

  This section begins on page 31.

- The last section contains the times used for the development of this document.

  This section begins on page 35.

At the end of the document the references used are included as well as two appendices on the parameters and intervals used by TrackMe.

2 Overall Description

2.1 Product perspective

The recollection of the data will be carried out by the mobile devices of the users, upon installation of the application of TrackMe. Before the recollection of data takes place, the users have to register to the system and give their approval.

To enable the analysis of the data to the clients, TrackMe provides centralised tools by means of an online dashboard and an API endpoint. The latter is intended to facilitate the integration of TrackMe platform with client's ones.

The payment of the clients for accessing the data is handle by a third party system and integrated in TrackMe. The request of ambulances is performed by a third party service, presumably an public service offered by the government.

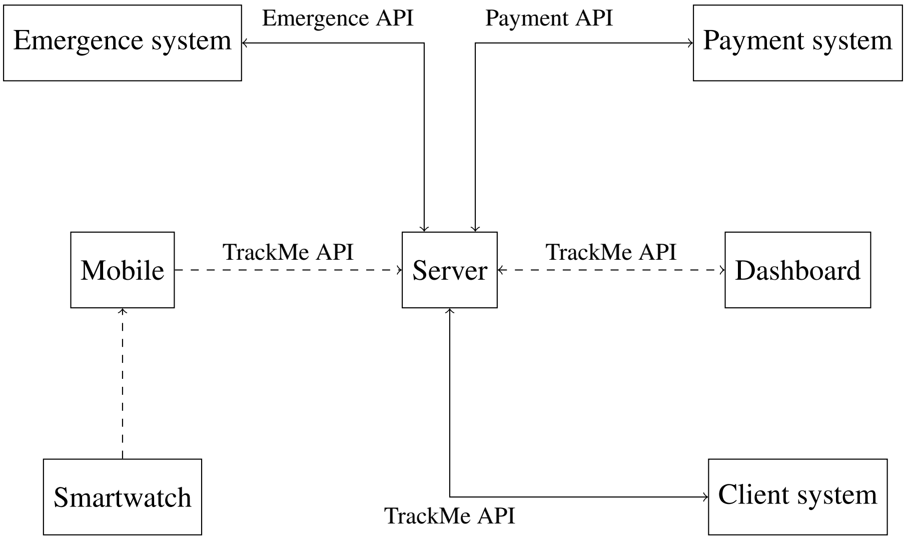


Figure 1: Overview of the system

An overview of the system is shown in figure 1. The dashed lines represents connections between elements of the TrackMe system while the solid ones corresponds to outside ones. These connections are discussed more profoundly in sections 2.5.2 and 3.1.2.

The components connected by dashed lines are internal of the system and therefore no requirements will be impose on these interfaces. However, stating the general architecture of the solution removes vagueness and allows the requirements to be specified in a more concise way.

2.2 Product functions

The main functions of the product are detailed below. The requirements stated later pursue the implementation of these functions.

2.2.1 Recollection of users' data

The related goals are GL1, GL2, GL3 and GL4.

The recollection of the users' data will be carried out by the users' devices[1] and stored by the server of TrackMe. The data will be grouped by parameters. The parameters are defined in section A.

An important distinction has to be made between fixed parameters and temporary ones. The former includes data that not changes, like the birth of date, codice fiscale, name, residence and so on. The latter will be recollected at a fixed interval of time and stored with a time-stamp associated with them. Location and heart rate are in this group.

These temporary parameters are not limited to the information recollected by the sensors of the users' devices. Some parameters, as for example the weight, may need the manual collaboration of the user to be recollected. This may be accomplished by promping the user to introduce the information in the application by himself. These kind of parameters will be called manual parameters.

This function is the cornerstone of the entire system, as the rest of the functionalities are devoted to consuming the recollected data.

2.2.2 Querying data

The related goals are GL5, GL6, GL7, GL9 and GL9.

The data stored in the server will be available to the clients. The system should allow clients to build a query based on the available parameters. This query will return a data batch[2].

The clients will be provided with two ways of accessing the data. The first one will be by mean of an online dashboard[3] and an API.

The latter is intended to erase the interconnection between the system of the client and the TrackMe one. Because of this, the API functionalities will be limited to querying data.

No matter what way the client use, two access of the data should be distinguish:

- **Anonymous query (Group search)**: Query that return a data batch with enough entries to be considered anonymous, i.e. not individuals can be distinguish.
- **Individual query**: Query by an identifier related univocally to an user. The client must elicit the permission of the user before getting the data.

This distinction is tightly related to goals GL7 and GL9. The related goal is GL10.

The charge to the clients for the access will be by means of a subscription.

2.2.3 Ambulance service (AutomatedSOS)

The related goal is GL10.

AutomatedSOS is an extra service offered to the users. The system will monitor a set of parameters further detailed in 2.5.3. When some of these parameters goes out of a threshold predefined by TrackMe, an alarm will be raised to the Emergency System, attaching the location of the user, the phone number and the phone number of a family member.

The aforementioned alarm will be communicated to the Emergency System by electronic means, using an API offered by the Emergency System.

[1] i.e. smartwatch and smartphone

[2] See data batch definition on 1.3.1

[3] i.e. a web page

2.3 User characteristics

There are three different types of actors in the system. Their definitions can be found in 1.3.1. Through these sections, each type will be analysed and an attempt of characterising them based on demographics will be made.

2.3.1 Users

The potential users can be segmented by age:

- **Generation Z**: Born between 2000's and onwards. 95% of persons in this age interval have a smartphone. They prefer the smartphone over the laptop [1]. They are comfortable sharing personal data in order to get a more personalised experience [2].
- **Millennials**: Born between 1980 and 2000's, the so-called *digital natives* are very confident in their abilities with digital interfaces and are used to fast-paced interactions.[3]. They read less than the average user [3], which leads to a rapid acceptation of the terms of usage of any application.
- **Generation X**: Born between 1980 and 1960. According to [4], 85% of *Gen Xers* owns a smartphone, which makes this generation a good segment for TrackMe product. They are generally influenced by convenient purposes and will keep aside digital products [5] if they do not feel comfortable with them.
- **Baby boomers**: Born between 1946 and 1964. The market segment of *Boomers* is very interesting since they have a great acquisition power. However, the penetration of fitness tracking devices are very scarce in this demographic interval. In fact, *Boomers* usually found difficulties in using the health monitor devices [6].

With these segments in mind, TrackMe system will be target to the three first segments. Specially taking into account that Generation Z and Millennials are not concerned about the lack of privacy. Generation X users have a great purchase power, but will be more reticent to the acceptation of the terms of usage.

AutomatedSOS will be oriented to Baby Boomers. Since they usually found difficulties in the usage of the fitness tracking devices, the application should be simple and concise.

2.4 Clients

Clients are assumed to be companies interested in offering a better service through the use of data. The main sectors in which the TrackMe system shall focus are:

- **Insurance:** They will be able to present to the users insurances more adapted to them (for example thanks to their location if they make many trips they will be able to offer them insurances more dedicated to it) as well as discounts by the state of health that the users may present.

  It is also presented as an advantage for the insurer since thanks to the different data they can calculate the profitability that a user presents to them.
- **Marketing:** They allow companies to provide more personalised marketing and advertising services for each user thanks to the lifestyle data of each one.
- **Banks:** They can offer better offers in mortgages or loans to the users thanks to the different data that TrackMe offers since they are able to know the state of health and the type of life that their clients have.

Thanks to TrackMe, the bank has an advantage since it can calculate the situation in which its clients will find themselves during the repayment of the loan or mortgage.

- **Wellness and health applications:** They can use TrackMe's data for a better control of their clients and thus present a more personalised information to each user.

2.5 Assumptions, dependencies and constrains

2.5.1 Assumptions

ID	Description
DA1	Users' mobile have GPS sensor.
DA2	Users' smarthwatch have an accurate hearth rate sensor.
DA3	Users eventually accepts Android Permission Request of the necessary sensors.
DA4	Users will answer the questions prompted by the application with reliable information.
DA5	Users will grant permissions to the application to work in the background.
DA6	Users's devices will have internet connection when the application will send the data to the server.
DA7	The user that owns the smarthphone in which the application is installed is the same user which wears the smarthwatch.
DA8	The users provides veridic data when registering.
DA9	Clients provides a card account with enough funds.
DA10	The emergency system API process a request in less than one second.
DA11	The user do not move more than 20 meters between the call to the emergency API an the arrival of the ambulance.

Table 2: Domain assumptions

2.5.2 Dependencies

TrackMe system relies in two external systems to perform several tasks:

- **Stripe**: Payment system that allows TrackMe system to charge the clients a monthly fee for the access to the data. Stripe stands out over the competitors because its API, which is recognized as one of the best in the business [7].
- **Emergency system**: To offer AutomatedSOS, TrackMe system needs a programmatic way to reach the emergency institutions. Therefore, an API is needed to rise an alarm when required.

Tables 3 and 4 contains the format and endpoints needed for the proper functioning of TrackMe product.

ID	Method	URL	Parameters	Return	Description
SI1	POST	/ambulance	location, phone number, phone number of a family member, triggered parameter	estimated time	Call when an emergency service is needed, the important data of the customer is send to the Emergency system

Table 3: Software interfaces of Emergency API

ID	Method	URL	Parameters	Return	Description
SI2	POST	/v1/tokens	Card Nº, Card expiration, CVC	Card ID	Register a card to be used in SI3. One time use.
SI3	POST	/v1/customer	Card ID, email	Customer ID	Register a client in the payment system, returns the ID that identifies the client in the payment system.
SI4	POST	/v1/subscriptions	Customer ID, Plan ID	Subscription ID	Starts charging the client the amont indicated by the Plan ID
SI5	DELETE	/v1/subscriptions	Subscription ID	Subscription ID	Register a client in the payment system, returns the ID that identifies the client in the payment system.

Table 4: Software interfaces of Payment API

The official documentation of Stripe can be consulted in `https://stripe.com/docs`.

2.5.3 Constraints

To ensure the proper work of TrackMe system, several constraints will be in place. The constraints will be formulated upon the different components of the system stated in figure 1.

ID	Element	Value	Motivation
C1	Operating system	Android 7.0 to 9.0	These versions agglutinate the 54% of the Android users [8]. The versions left behind are not supported anymore by Google.
C2	Minimum RAM	1GB	The application will not be very demanding in terms of RAM. However, the proper functioning is only ensure to devices with 1GB of RAM or more.
C3	Minimum free space	50MB	The application will occupy 50MB or less in the internal storage of the users' devices.
C4	Orientation	Vertical	The application will not adapt to a change in device orientation. Will be always in vertical.
C5	Background activity	Yes	The application will only work properly if background activity is allowed by the users' devices.
C6	Smartwatch	AndroidWear	The application will only recollect health status data if the smartwatch paired with the user device is powered by AndroidWear.

Table 5: Mobile application

ID	Element	Value	Motivation
C7	Operating system	Ubuntu Server 16.04	The proper functioning of the server will be only ensure in Ubuntu Server 16.04. Other operating systems may be consider upon approval of JJ Software.
C8	Minimum RAM	32 GB	The server will need to withstand an important workload, therefore 32GB of RAM is needed.
C9	Minimum disk space	200 GB	This space will be needed to store all the data of the users in a secure way.
C10	Minimum bandwidth	500 Mb/s	To offer a good throughput, 500 Mb/s are needed to coupe with the users and clients request.

Table 6: Server

ID	Element	Value	Motivation
C11	Supported browsers	Safari 10, Firefox 50, Chrome 60, IE 11, Edge 10, Opera	The dashboard will properly work in all modern browsers, from the versions stated on the left and above.
C12	Minimum display	1024px x 784px	This is the minimun display resolution in which the dashboard will look as in the mockups.

Table 7: Clients' Dashboard

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- **Web Application Interface**

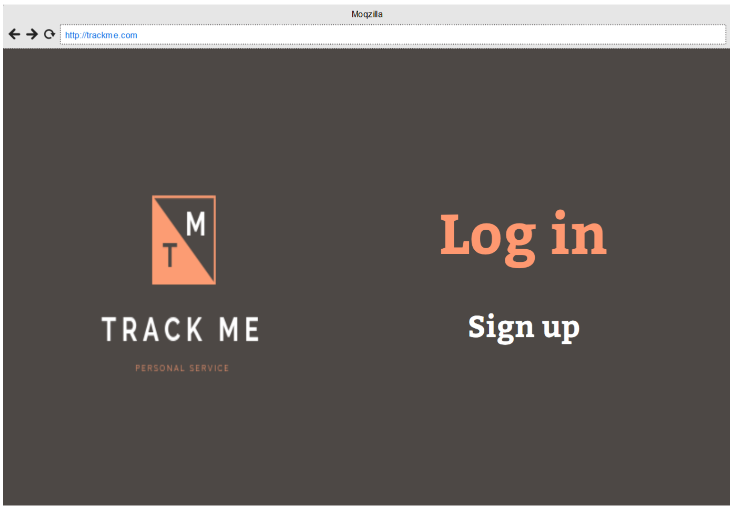


Figure 2: Log in and sing up on the Web Application

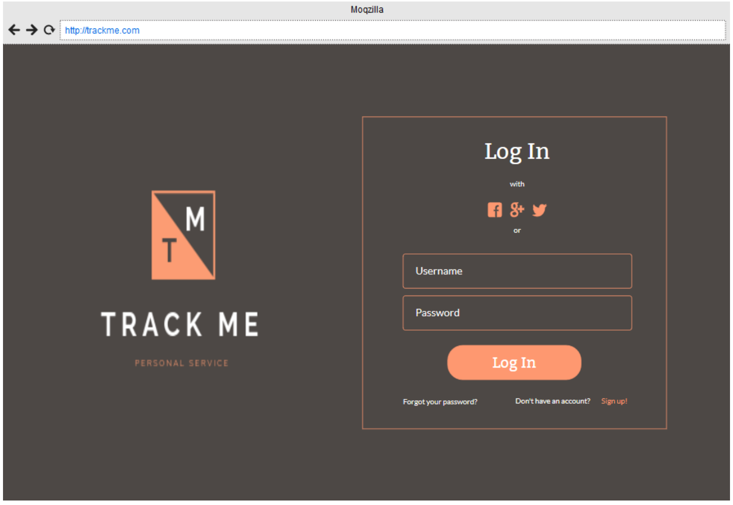


Figure 3: Log in screen of the Web Application

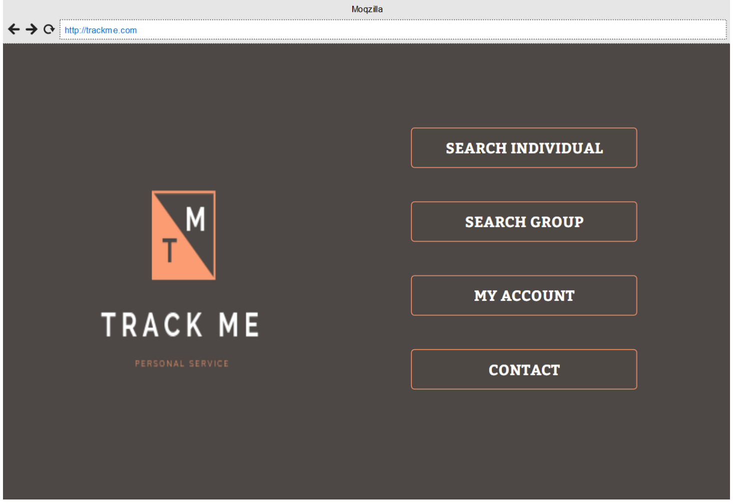


Figure 4: Principal page of the Web Application

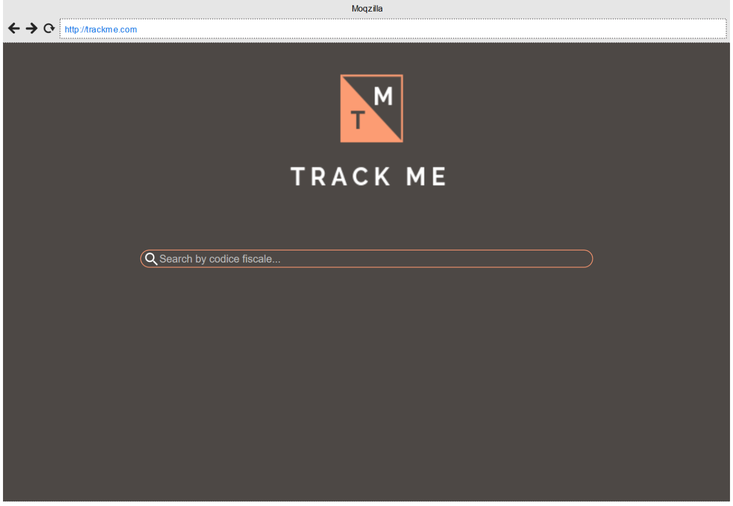


Figure 5: Search for individual data in the Web Application

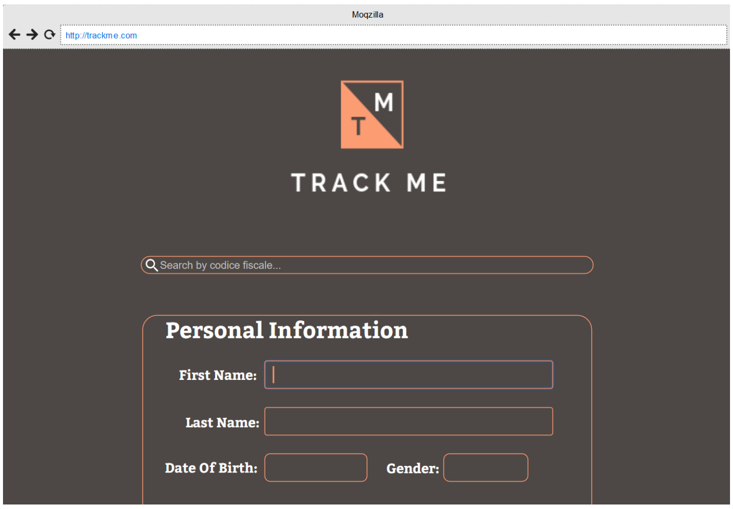
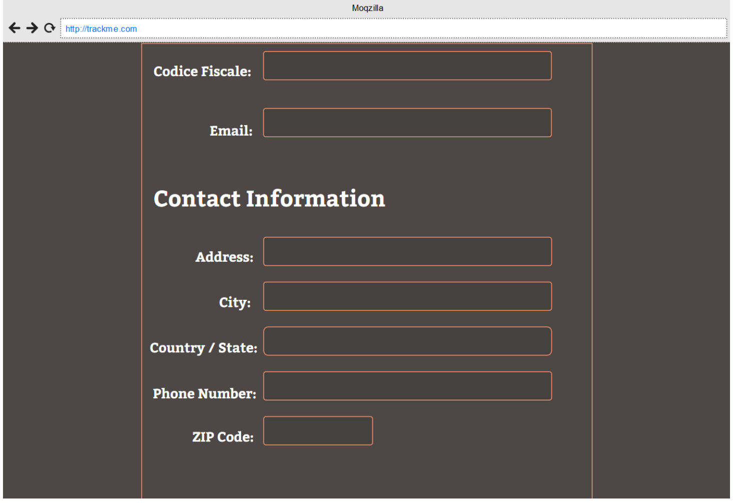


Figure 6: Results of the individual search in the Web Application (First part)



Codice Fiscale:

Email:

Contact Information

Address:

City:

Country / State:

Phone Number:

ZIP Code:

Figure 7: Results of the individual search in the Web Application (Second part)

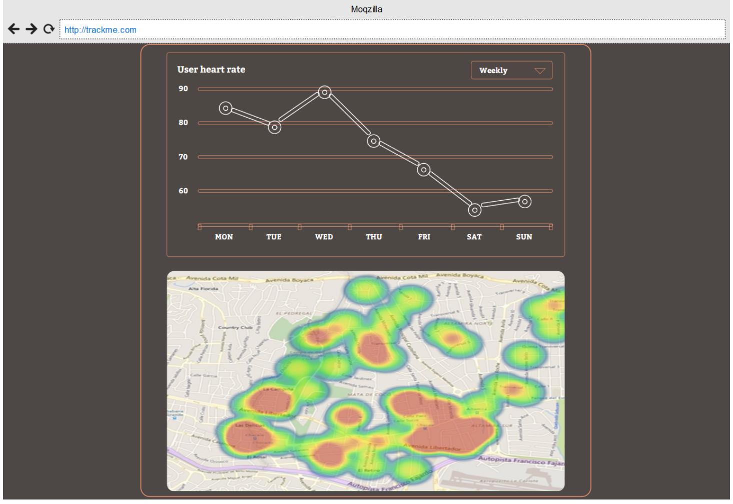
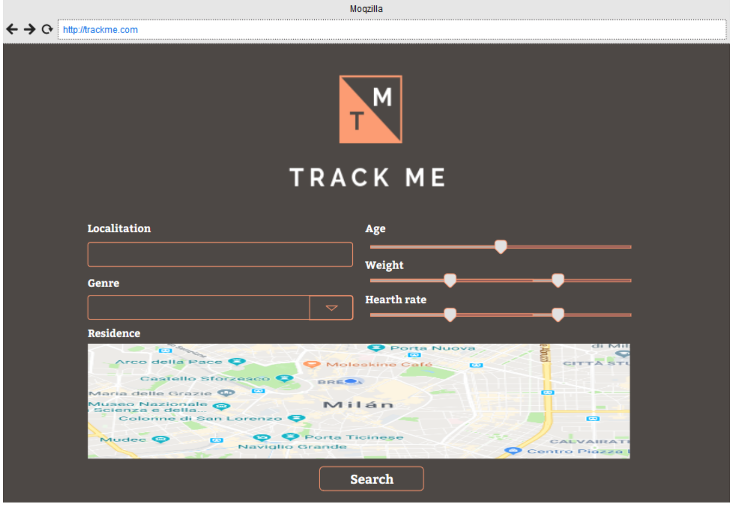


Figure 8: Results of the individual search in the Web Application (Third part)



Localitation

Genre

Residence

Age

Weight

Hearth rate

Search

Figure 9: Search for group data in the Web Application

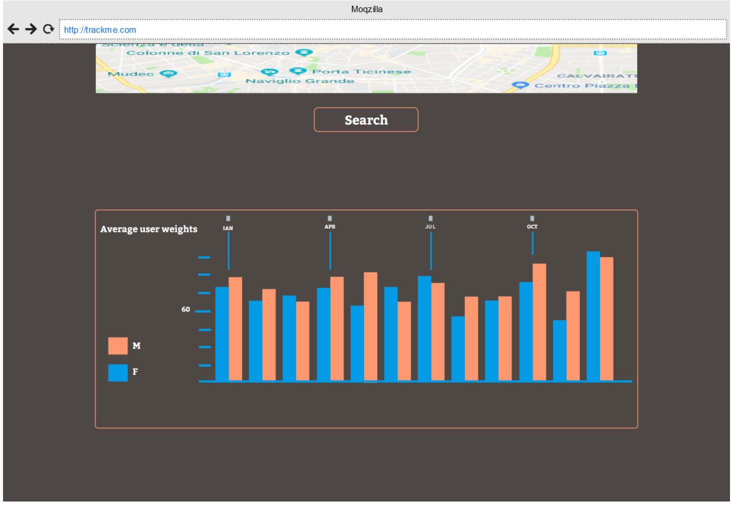


Figure 10: Result of the group search in the Web Application (First part)

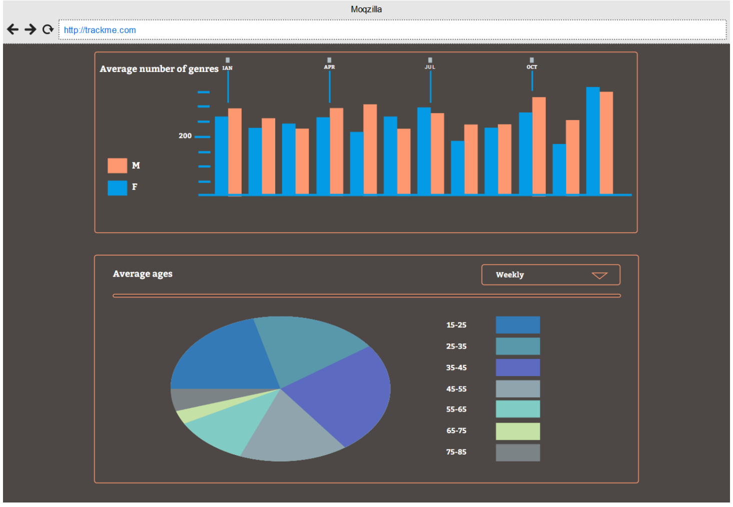


Figure 11: Result of the group search in the Web Application (Second part)

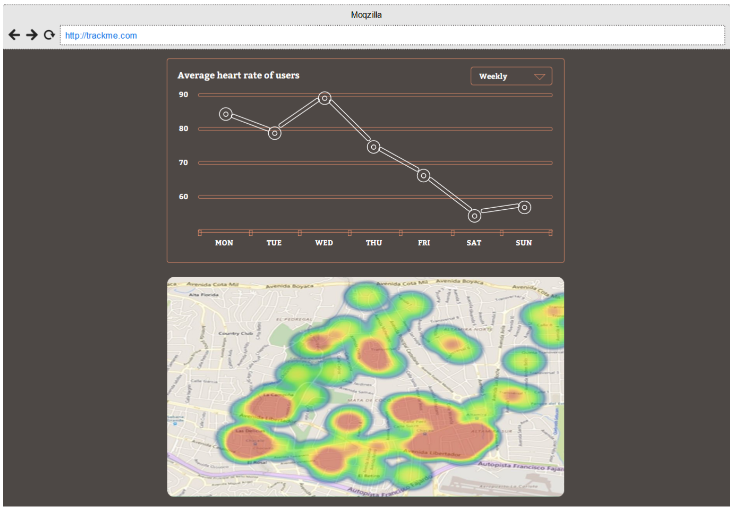
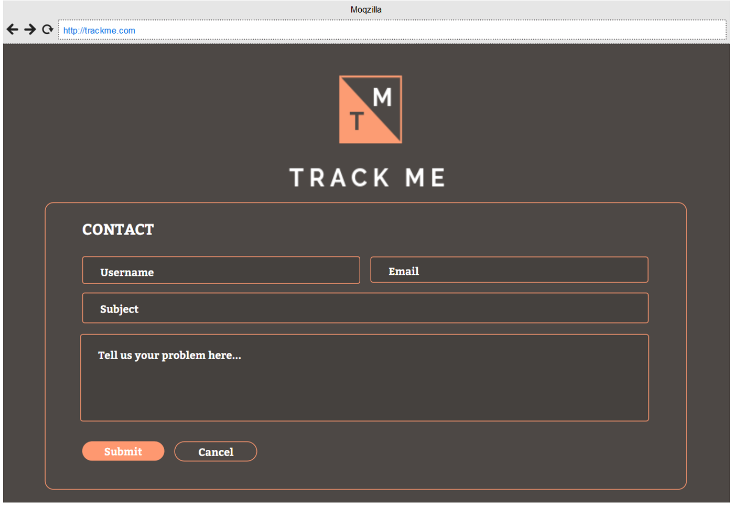


Figure 12: Result of the group search in the Web Application (Third part)



Tell us your problem here...

Figure 13: Contact zone of the web application

- **Application Interface**

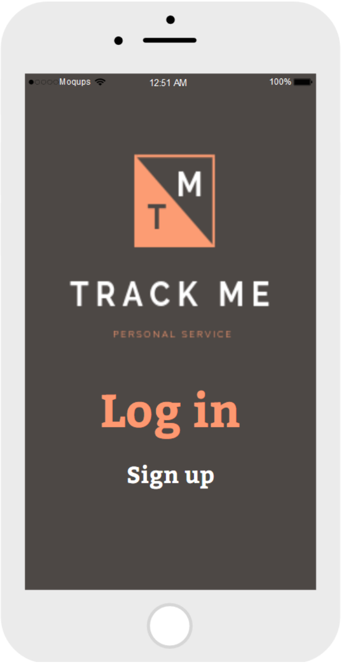


Figure 14: Log in and sing up window of the application

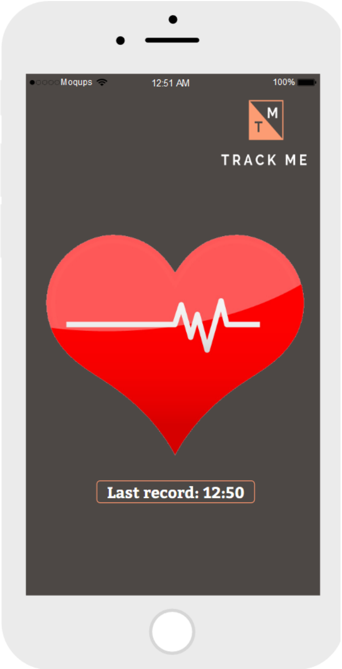


Figure 15: Main window of the application

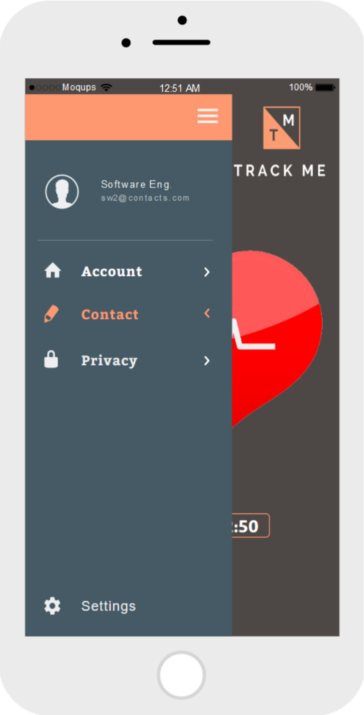


Figure 16: Main application window with slider tab

- **Smartwatch Interface**

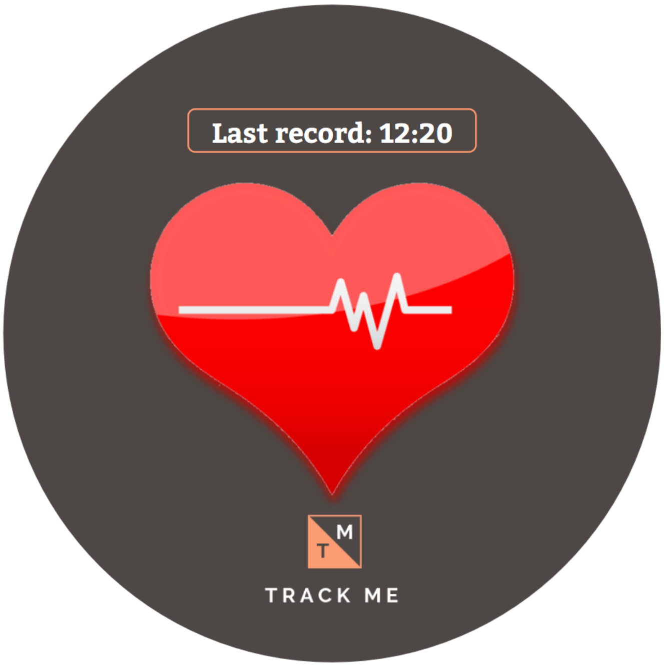


Figure 17: Smartwatch window

3.1.2 Software Interfaces

In this section, the API offered by TrackMe will be detailed. This API is intended to be use for the clients who seeks an integration with their own systems.

ID	Method	URL	Parameters	Return	Description
SI6	POST	/login	Username, password	Client token	Returns a token that will be use to authenticate the user in the next API calls.
SI7	POST	/query	Client token, query	Data batch	Return a data batch containing all the entries that corresponds to the query.
SI8	POST	/search-user	Client token, User ID	Data of the user	Returns the available data of the searched user if the client has permission

Table 8: Software interfaces of TrackMe API for the client

3.1.3 Hardware Interfaces

TrackMe is presented as a software that is based on an Android application, because of it does not require special hardware other than the one named above, this hardware is more than anything a mobile device such as a Smartphone or a Smartwach. As discussed below it is necessary that the mobile device has an Internet connection as well as Bluethoot (if Smartwatch is used) and GPS (for tracking the location).

The system will run on any off-the-shelf hardware that supports Linux x64. Therefore, no special requirements and interfaces are needed. The relevant constraints for the users' devices are stated in section 2.5.3.

3.2 Use cases

This section is meant to clarify the normal usage of the platform by the different actors of the system.

ID	UC1
Name	Sing up of Clients.
Actor	Client.
Entry conditions	The client have to be on the web application.
Events flow	1. The client have to click on the button of sing up of the web application (figure 2). 2. Fill in the necessary information requested in the form that will appear as well as a form of payment. 3. After the confirmation the system will save the data and the client will be registered.
Exit conditions	The client will be registered and able to work with TrackMe.
Exceptions	1. The form of payment is not accepted or is incorrect. 2. The client is already registered. 3. The client has not filled in one of the necessary information fields or a field is not filled in correctly.

Table 9: Sing up of a Client use case

ID	UC2
Name	Log in of Clients.
Actor	Client.
Entry conditions	1. The client have to be registered on TrackMe. 2. The client have to be on the web application.
Events flow	1. Press the log in button in the web application (figure 2). 2. Complete the username and password sections of the log in window (figure 3). 3. After clicking on the log in button, if it is correct the username and password will access to user account and the system will redirect to the main window (figure 4).
Exit conditions	The client will access his account.
Exceptions	Username and password do not match or do not exist.

Table 10: Log in of a Client use case

ID	UC3
Name	Sing up of Users.
Actor	User.
Entry conditions	The user must have the application installed and be in it.
Events flow	1. The user have to click on the button of sing up of the application (figure 14). 2. Fill in the necessary information requested in the form that will appear. 3. After the confirmation the system will save the data and the user will be registered.
Exit conditions	The user will be registered and able to use TrackMe services.
Exceptions	1. The username already exists in the system. 2. The email already exists in the system. 3. The user has not filled in one of the necessary information fields or a field is not filled in correctly.

Table 11: Sing up of a User use case

ID	UC4
Name	Log in of Users.
Actor	User.
Entry conditions	1. The user have to be registered on TrackMe. 2. The user needs to have the application installed.
Events flow	1. Press the log in button in the application (figure 14). 2. Complete the username and password sections of the log in window. 3. After clicking on the log in button, if it is correct the username and password the user will access be redirected to the main window (figure 15).
Exit conditions	The user will access his account.
Exceptions	Username and password do not match or do not exist.

Table 12: Log in of a User use case

ID	UC5
Name	Search of an individual.
Actor	Client.
Entry conditions	The client have to be registered on TrackMe and log in on the web application.
Events flow	1. Click on the individual data search button. (Figure 4) 2. Enter the Codice Fiscale in the search area that appears on the new page (in the area where it is requested). (Figure 5) 3. The system will show the data of the user if the client have the necessary permissions, another search can be performed also. (Figures 6, 7, 8)
Exit conditions	The client will be able to see the information of the requested user.
Exceptions	1. The Codice Fiscale does not exist. 2. The Codice Fiscale is misspelled. 3. The client does not have the permissions to view the user's data.

Table 13: Case of use of individual data search

ID	UC6
Name	Querying group data.
Actor	Client.
Entry conditions	The client have to be registered on TrackMe and log in on the web application.
Events flow	1. Click on the group data search button of the web application (figure 4). 2. Client will be redirected to a search page, figure 9, where a query can be formulated. 3. After pressing the search button the system will show the information of the users (anonymously) who meet the criteria given (figures 10, 11 and 12).
Exit conditions	The client will be able to see the data of the group that fulfills the given requirements.
Exceptions	1. Insert a search criteria that the set of users that satisfy them is less than 1000. 2. Any of the search criteria given is misspelled or does not exist.

Table 14: Case of use of group data search

The following state charts are mean to clarify the operation of the system.

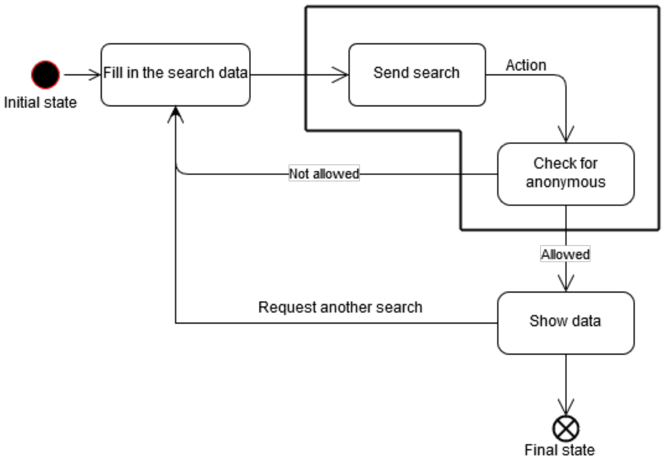


Figure 18: Statechart group search

In figure 18 it can be saw the state chart diagram of the group search which corresponds to use case UC14 and figure 9.

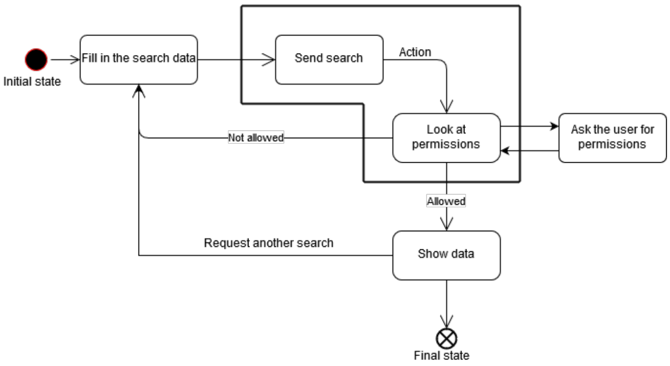


Figure 19: Statechart individual search

In figure 19 it can be saw the state chart diagram of the group search which is assigned to the use case UC13 and figures 5, 6, 7 and 8.

The following sequence diagram attempts to show the main workings of the application for better understanding.

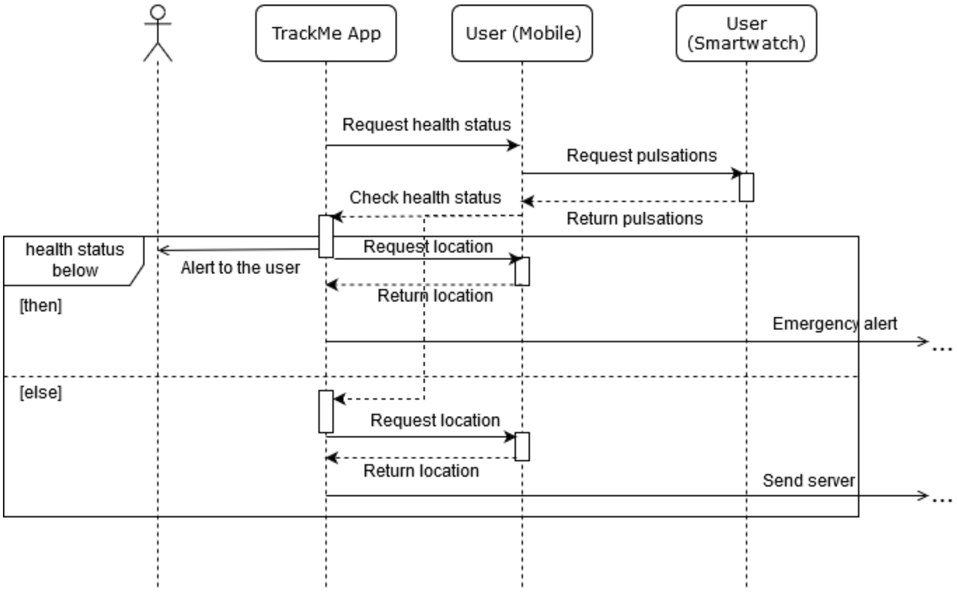


Figure 20: Track sequence diagram

Figure 20 shows the sequence diagram of the tracking carried out by the application each time interval and as in the case of parameters below what is necessary, an alarm is sent to the user and to the emergency services.

The following class diagram summarizes how the system is organized and what main structures it contains.

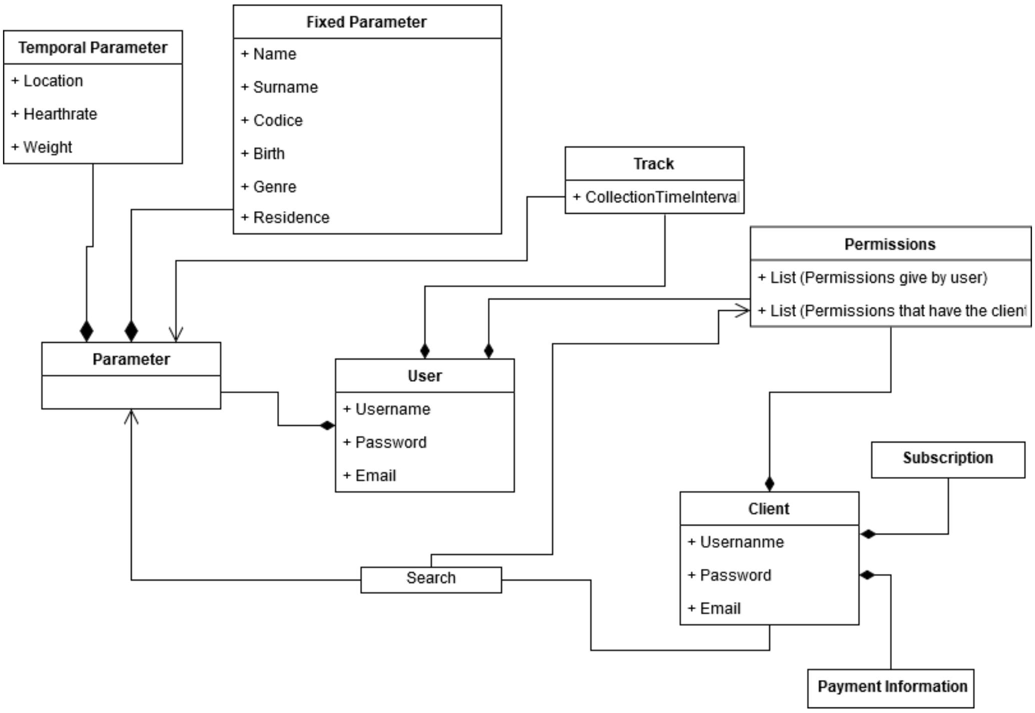


Figure 21: Class diagram of the system

Figure 21 shows the main structures of the system as they are related to each other and how the main functions (Search and Track) are related to the system to be performed.

3.3 Functional Requirements

The functional requirements are divided in three categories. The requirements of the user application, table 15; the requirements related with clients, table 16 and 17, and finally the AutomatedSOS requirements, table 18.

ID	Goal	Description
FR1	GL1	When an user opens the application and no login had been performed, the system shall show the welcome page (figure 14).
FR2	GL1	When the welcome page is shown, the system shall show two buttons (figure 14). When clicked, one of them shall redirect to the login page and the other to the registration page.
FR3	GL1	When the registration page is completed, the system shall show the terms and conditions page and only users that accept the terms and conditions will successfully registered.
FR4	GL3	When the user logs in for the first time in the application, the application shall check what sensors are available an issue an Android Permission Request for each of them.
FR5	GL3	If the user declines an Android Permission Request, the application shall issue again an Android Permission Request for the same sensor.
FR6	GL3 GL4	The system shall poll the available sensors in the background at fixed time intervals and store the measures in the server.
FR7	GL4	The fixed intervals at which each sensor shall be polled are stated in 26.
FR8	GL3 GL4	The system shall prompt the user to introduce the manual parameters at fixed time intervals and store the measures in the server.
FR9	GL4	The fixed intervals at which the manual parameters shall be asked to the user are stated in 26.
FR10	GL7	When a client has sent a request for access, the system shall display a notification in the user's device showing the name of the client which requires the permission and a button to accept.

Table 15: Functional requirements of user application

ID	Goal	Description
FR11	GL6	A query consists of a set of parameters with associated logical constraints. The result of the query must comply all the logical constraint in the query.
FR12	GL6	The numerical parameters' logical constraints can be equal (=), greater (>), greater or equal (=>), smaller (<) and smaller or equal (=<). The PM7 parameter do not follow this requirement.
FR13	GL6	The PM7 parameter's logical constraint is expressed as a set of points in which the searched values are geographically inside.
FR14	GL6	The system should provide specific inputs adapted to the type of data to introduce the logical constraints of the query. Table 27 states the parameters and its inputs.
FR15	GL6 GL9	When the client introduces a query from the dashboard page (figure 9) and the number of entries that fullfil the query are equal or more than 1000, the system shall show the data in page (Figures 10, 11 and 12).
FR16	GL6 GL9	When the client introduces a query from the dashboard page (figure 9) and the number of entries that fullfil the query are less than 1000, the system shall warn the client about the impossibility to show the results in page.
FR17	GL8	When the system is showing a data batch in the dashboard that fullfils a query (figures individual search: 6, 7 and 8; figures group search: 10, 11 and 12) and new data that also fullfils the query arrives, the system shall update the view of the data without intervention of the client.
FR18	GL7	When the client introduces a codice fiscale from the dashboard page (figure 5), the user exists and the client have already obtained the permission of the user, the system shall return the data associated to the individual.
FR19	GL7	When the client introduces a codice fiscale from the dashboard page (figure 5), the user exists and the client do not have the permission of the user, the system shall prompt the client to ask permission to the user.
FR20	GL7	When the client introduces a codice fiscale from the dashboard page (figure 5), and the user do not exists, the system shall prompt the client to ask permission to the user.
FR21	GL7	When the client is requesting permission to a concrete user in page and clicks on *Yes*, the system shall emit a to the appropriate user application requesting their permission.
FR22	GL7	When a user approves the request of access made by a client, the system shall store that permission.
FR23	GL7	The system shall show the client a list of all users that had give their permission of access in page in descending alphabetical order.

Table 16: Functional requirements of the client dashboard

In table 17 the phrase *When a message with a correct format reach* is used often. The correct format reefers to the one stated in section 3.1.2 for each corresponding SI interface.

ID	Goal	Description
FR24	GL1	When a message with a correct format reach the interface SI6 with an existing pair of username and password, the system shall replay with a token that will identify the client in the next api calls. The token have a validity of 3 days.
FR25	GL6, GL9	When a message with a correct format reach the interface SI7 with a well form query and the result of the query have 1000 entries or more, the system shall replay with a data batch that complies the logical constraints expressed in the query.
FR26	GL6, GL9	When a message with a correct format reach the interface SI7 with a well form query and the result of the query have less than 1000 entries, the system shall replay with a 403 error.
FR27	GL7	When a message with a correct format reach the interface SI8 with a valid codice fiscale, the user exists and the client have already obtained the permission of the user, the system shall return the data associated to the individual.

Table 17: Functional requirements of the client API

ID	Goal	Description
FR28	GL10	When a parameter sent by an user's application arrives at the server and is below a defined threshold and the user is sign up in AutomatedSOS, the system shall rise an alarm to the Emergency System using interface SI1 within 5 seconds.

Table 18: Functional requirements of the AutomatedSOS service

3.4 Performance Requirements

ID	Description
PR1	The system shall be able to process at least 500 request per second.
PR2	The system shall be able to respond to a query made by a client within 5 seconds in an optimum internet connection between client and system scenario.
PR3	The system shall be able to store all the data gathered from the users at least during 3 months.
PR4	The system shall be able to handle at least 10000 active users.
PR5	The system shall be able to handle at least 100 active clients.

Table 19: Performance requirement

3.5 Design Constraints

3.5.1 Standards compliance

Since TrackMe system do not have to handle any critical situation nor have connection with the real world, the applicability of standards is limited.

Although law is not a standard strictly speaking, since the impact of the rules about privacy and usage of data is noticeable in TrackMe product, a review of the applicable legislation must be made.

Since May 2018, the EU General Data Protection Regulation replaces the previous legislation, enforcing a homogeneous set of rules in the whole Europe Union. The following excerpt is obtained from [9]:

The request for consent must be given in an intelligible and easily accessible form, with the purpose for data processing attached to that consent. Consent must be clear and distinguishable from other matters and provided in an intelligible and easily accessible form, using clear and plain language. It must be as easy to withdraw consent as it is to give it.

Therefore, the registration process must be clear, stating in a natural language the terms of usage. Users needs to accept one per one the collection of each peace of information that TrackMe is collecting. Users also can exercise their right to withdraw their consent. A section in the users' application offering to withdraw their permission must be present. Please, notice the Privacy section in figure 16.

3.5.2 Hardware limitations

There are no hardware limitations, it is only necessary for the user to present a mobile phone with the possibility of installing mobile applications (a smartphone) and if a health check is required, the user must have a Smartwatch.

In addition to what has been said for the mobile phone, it is also necessary that the user has:

- Stable Internet connection (2G, 3G, 4G).
- Bluetooth for connection to the Smartwatch if required.
- GPS for user location.

The Smartwatch the only requirement is to be able to measure the heart rate as well as the different parameters to perform health monitoring.

3.6 Software System Attributes

3.6.1 Reliability

The system shall preserve the integrity of the saved data no matter what circumstances may happen.

3.6.2 Availability

The system shall be available 99.99% of the time. Programmed maintenance downtime communicated to TrackMe on behalf of JJ Software at least one month ahead will not account in the availability time.

In the event that an external factor causes TrackMe services to fall, planned emergency measures will be taken to ensure that the system continues to operate at all times.

3.6.3 Security

For security, the HTTPS protocol will be used, which applies SSL/TLS cryptography [10] that will allow the passage between the server and the different services in a secure way, not allowing attacks or loss of information from TrackMe clients and users.

3.6.4 Maintainability

The operation and maintenance of the TrackMe system will be realised by JJ Software under the terms agreed on the Maintenance and Operation contract.

The routine tasks as minor fixes will be performed with no charge to TrackMe. Minor fixes includes light tasks with a budget time of two hours of work monthly.

One important aspect to highlight is the Terms of Conditions that will be presented to users and clients. The Terms of Conditions shall be easy to change and update in order to adapt the TrackMe system to legislation changes.

3.6.5 Portability

The portability degree depends on each system's component. The mobile application shall be portable between Android platforms that fullfills the requirements stated in table 5. The server component shall be portable between Linux platforms that fullfills the requirements stated in table 6. The online dashboard shall work in any browser that fullfills the rules stated in table 7

4 Formal Analysis Using Alloy

The analysis made in alloy is focus on the relationships between the data, the queries and the individual search made by the clients.

The Query is the most complex concept, the constraints represents logical constraints made on a Parameter, as for example age less than 18 years and so on. Some values will comply that constraint and therefore the user who owns that values will be matched by the query and their data displayed as a result of the query.

Only queries with more than 2 entries are allowed, in the final implementation this level will be raised to 1000. Individual searches are only allowed when a codice matches and the Client already have permission from the user owner of the codice.

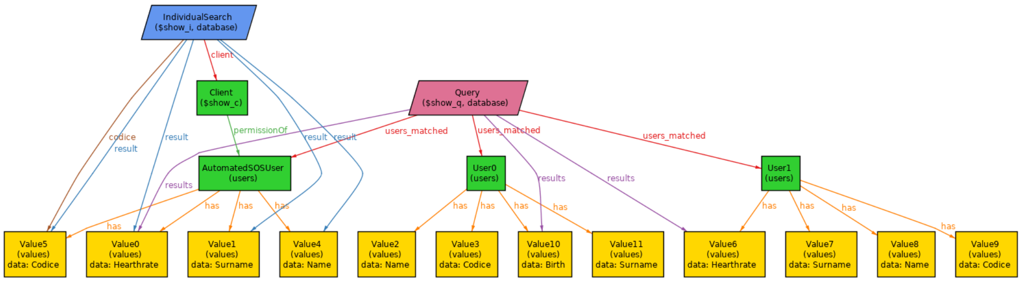

IndividualSearch ($show_i, database)
client
Client ($show_c)
permissionOf
Query ($show_q, database)
users_matched
codice
result
results
AutomatedSOSUser (users)
User0 (users)
User1 (users)
has
Value5 (values) data: Codice
Value0 (values) data: Hearthrate
Value1 (values) data: Surname
Value4 (values) data: Name
Value2 (values) data: Name
Value3 (values) data: Codice
Value10 (values) data: Birth
Value11 (values) data: Surname
Value6 (values) data: Hearthrate
Value7 (values) data: Surname
Value8 (values) data: Name
Value9 (values) data: Codice

Figure 22: Example of generated world with 3 Users, 1 Query and 1 Individual Search

There are several remarkable facts in figure 22. As can be saw, the Query only returns the fields that are not protected by privacy issues. In contrast, the Individual Search return all the information of the User matched by the codice since the client have permission. Also, all users have exactly one value of the codice, name and surname and a variable amount of the temporal data as the hearth rate.

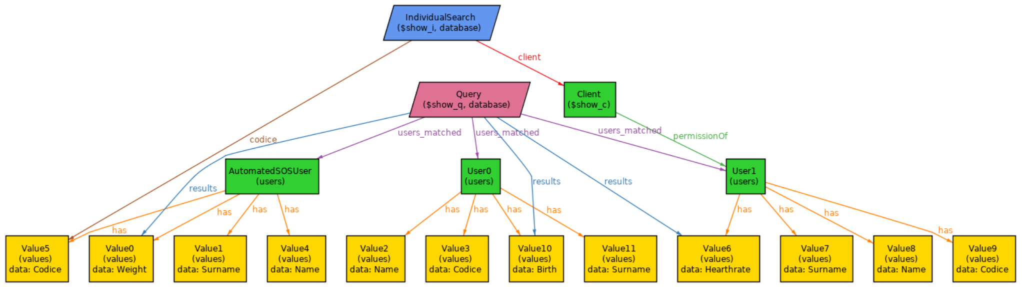

IndividualSearch ($show_i, database)
client
Client ($show_c)
Query ($show_q, database)
codice
users_matched
permissionOf
results
AutomatedSOSUser (users)
User0 (users)
User1 (users)
has
Value5 (values) data: Codice
Value0 (values) data: Weight
Value1 (values) data: Surname
Value4 (values) data: Name
Value2 (values) data: Name
Value3 (values) data: Codice
Value10 (values) data: Birth
Value11 (values) data: Surname
Value6 (values) data: Hearthrate
Value7 (values) data: Surname
Value8 (values) data: Name
Value9 (values) data: Codice

Figure 23: Example of generated world with only 1 User

Figure 23 shows how the Individual Search behaves since the client do not have permission of the user. Therefore the results are empty.

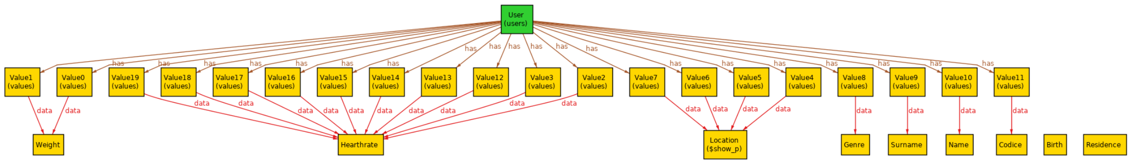


Figure 24: Example of generated world with an Individual Search with no results

Finally 24 focus on the number of values own by an user. The fixed parameters have, at most one value whilst the temporal ones have zero or more values.

```
abstract sig Parameter{}

abstract sig FixedParameter extends Parameter {}
abstract sig TemporalParameter extends Parameter{}

one sig Codice, Name, Surname, Birth, Genre,Residence  extends FixedParameter{}
one sig   Location, Hearthrate, Weight extends TemporalParameter{}

sig Value {}

sig User{}

sig AutomatedSOSUser extends User{}

sig Client{
        permissionOf: set User
}


one sig Database {
        users: set User,
        values: set Value,

        has:  User one -> Value,
        belongs: Value -> one User,


        data:  Value->one Parameter,
        reverse_data :  Parameter one->Value
}{
        belongs = ~ has
        reverse_data =~data

        //1 codice per user
        all u:users |#(has[u] & reverse_data[Codice])=1
        //1 name per user
        all u:users |#(has[u] & reverse_data[Name])=1
        //1 surname per user
        all u:users |#(has[u] & reverse_data[Surname])=1

        //at most 1 residence per user
        all u:users |#(has[u] & reverse_data[Residence])<2
        //at most 1 birth date per user
        all u:users |#(has[u] & reverse_data[Birth])<2
        //at most 1 Genre per user
        all u:users |#(has[u] & reverse_data[Genre])<2

}
```

```
//All the values and users needs to be stored in the Database
fact {
        all v:Value | v in Database.values
        all u:User | u in Database.users
}


//Values cannot belong to different parameters at the same time
fact{
        all v:Value| one p:Parameter | v->p in Database.data
}

//Values cannot belong to different users at the same time.
fact{
        all v:Value| one u:User | u->v in Database.has
}



sig Query{
        database: one Database,

        //Represents the logical constraints inposed by the client in the query
        constraints: set (Location + Hearthrate + Weight + Residence +Birth + Genre),
        //The data that fullfils the constrained imposed.
        results: set Value,
        //The users that fullfill the constraints imposed
        users_matched: set User
}{
        //Queries with few entries will not be allowed
        #(users_matched) > 2

        /* The matched users (the one that fullfills the constraints) must have values
            for the constrained parameters. However, not all the users  will fullfill the
            constraint */
        all u:users_matched | u in database.belongs[database.reverse_data[constraints]]

        /* In results all the data of the matched users
           are present except the protected one.*/
        results = database.has[users_matched]
                  - database.reverse_data[(Codice + Name + Surname + Residence)]
}

sig IndividualSearch{
        database: one Database,
        codice: one Value,
        result: set Value,
        client: one Client,
}{
        //The codice value must belong to Codice parameter
        database.data[codice] in Codice
        result = database.has[database.belongs[codice] & client.permissionOf]
}

// CHECKS

/* If a user is matched, a constraint has to be made to
   some parameter in which the user have values stored. */
check noUserMatchedWithoutConstraints{
        all q:Query |{
                all u:q.users_matched |u in
                q.database.belongs[q.database.reverse_data[q.constraints]]
        }
} for 50

//Any codice, name, surname or residence appears in the result of a query.
check noForbiddenParametersInQuery{
        all q:Query | q.database.data[q.results]
                      not in (Codice + Name + Surname + Residence)
} for 50

//Queries with few entries will not be allowed
check noFewEntries{
        all q:Query | #(q.users_matched)>2
} for 50

//Queries with few entries will not be allowed
check noFewEntries{
        all q:Query | #(q.users_matched)>2
} for 50

//Any user can't have more than 1 codice
check noTwoCodice{
        all d:Database | {
                all u:d.users |#(d.has[u] & d.reverse_data[Codice])<2
        }
} for 50
// Ommited the rest of the checks on numer of genres, residences and so on  for simplicity

//Must be the adequate number of parameters depending of the number of users.
check mandatoryParameters{
                #(Database.users) = #(Database.reverse_data[Codice]) and
                #(Database.users) = #(Database.reverse_data[Name]) and
                #(Database.users) = #(Database.reverse_data[Surname]) and
                #(Database.users) >= #(Database.reverse_data[Birth]) and
                #(Database.users) >= #(Database.reverse_data[Residence]) and
                #(Database.users) >= #(Database.reverse_data[Genre])
} for 50
```

5 Effort Spent

Person	Task performed	Time spent
Javier	Scope	2 hours
Javier	Definitions	30 minutes
Julián	Definitions	15 minutes
Julián	Document Structure	45 minutes

Table 20: Effort spent in Section 1 (page 7)

Person	Task performed	Time spent
Javier	Product perspective	2 hours
Javier	Product functions	2 hours
Javier	Users characteristics - Users	1 hour
Julián	Users characteristics - Clients	20 minutes
Javier	Assumptions	30 minutes
Julián	Assumptions	30 minutes
Javier	Dependencies	1 hour
Javier	Constraints	1 hour

Table 21: Effort spent in Section 2 (page 11)

Person	Task performed	Time spent
Julián	Web Application Interface	11 hours
Julián	Application Interface	30 minutes hours
Julián	Smartwatch Interface	10 minutes hours
Javier	Software Interfaces	30 minutes hour
Julián	Use Cases	1 hour
Julián	Statechart	30 minutes hours
Julián	Sequence Diagram	35 minutes hours
Julián	Class Diagram	30 minutes hours
Javier	Functional requirements of user application	3 hours
Javier	Functional requirements of client dashboard	4 hours
Javier	Functional requirements of client API	30 minutes
Javier	Functional requirements of AutomatedSOS	30 minutes
Javier	Standards compliance	30 minutes
Julián	Hardware Limitations	15 minutes
Julián	Software System Attributes	20 minutes

Table 22: Effort spent in Section 3 (page 17)

Person	Task performed	Time spent
Javier	Alloy modelling	12 hours
Javier	Generation and descriptions of worlds	30 minutes

Table 23: Effort spent in Section 4 (page 31)

Person	Task performed	Time spent
Javier	Parameters	30 minutes
Javier	Intervals table	30 minutes
Javier	Inputs table	30 minutes

Table 24: Effort spent in Appendix (page 38)

References

[1] Angela Woo, “Time To Wake Up To The Next Consumer Powerhouse: Gen Z,” *Forbes*, Jul., 2018. URL: https://www.forbes.com/sites/forbesagencycouncil/2018/07/23/time-to-wake-up-to-the-next-consumer-powerhouse-gen-z/#72845e8e53f4.

[2] Emily Schiola, “Generation Z opts for personalization over privacy: What that means for Wordpress,” *Torque*, Dec., 2017. URL: https://torquemag.io/2017/12/generation-z-opts-personalization-privacy-means-wordpress/.

[3] Kate Moran, “Millennials as Digital Natives: Myths and Realities,” *Nielsen Norman Group*, Jan., 2016. URL: https://www.nngroup.com/articles/millennials-digital-natives/.

[4] Jingjing Jiang, “Millennials stand out for their technology use, but older generations also embrace digital life,” *Pew Research Center*, May, 2018. URL: http://www.pewresearch.org/fact-tank/2018/05/02/millennials-stand-out-for-their-technology-use-but-older-generations-also-embrace-digital-life/.

[5] Monika Kapoor, “Digital Behaviour of Gen X, Y and Z,” *Code Brew*, Feb., 2018. URL:https://www.code-brew.com/2018/02/07/digital-behaviour-of-gen-x-y-and-z/.

[6] Laurie M. Orlov, “Baby steps: Will Boomers Buy Into Mobile Health?,” *California Healtcare Foundation*, Dec, 2017. URL: https://www.chcf.org/wp-content/uploads/2017/12/PDF-BabyStepsBoomersMobileHealth.pdf.

[7] Brian Edmondson, “Stripe vs PayPal: Is Stripe Better than PayPal?,” *The Balance Small Business*, April, 2018. URL: https://www.thebalancesmb.com/stripe-vs-paypal-2531677.

[8] Stat Counter, “Mobile & Tablet Android Version Market Share Worldwide,” *Stat Counter*, Oct. 2018. URL: http://gs.statcounter.com/android-version-market-share/mobile-tablet/worldwide.

[9] European Union, “GDPR Regulation.” https://eugdpr.org/the-regulation/, 2018.

[10] Instant SSL, “What is HTTPS?,” URL: https://www.instantssl.com/ssl-certificate-products/https.html.

[11] Edward R. Laskowski, “What’s a normal resting heart rate?,” *Mayo Clinic*, Aug., 2018. URL: https://www.mayoclinic.org/healthy-lifestyle/fitness/expert-answers/heart-rate/faq-20057979.

Appendix A Parameters

ID	Parameter	Units	Type	Query	Individual search
PM1	Codice fiscale	String	Fixed, manual	×	✓
PM2	Name	String	Fixed, manual	×	✓
PM3	Surname	String	Fixed, manual	×	✓
PM4	Birth date	dd/mm/yyyy	Fixed, manual	year	✓
PM5	Genre	M/F	Fixed, manual	✓	✓
PM6	Residence	Latitude, longitude	Fixed, manual	Searchable, not shown	✓
PM7	Location	Latitude, longitude	Temporal, automatic	✓	✓
PM8	Hearth rate	bpm	Temporal, automatic	✓	✓
PM9	Weight	Kilograms	Temporal, manual	✓	✓

Table 25: List of parameters and its type

Appendix B Inputs and intervals associated to parameters

Parameter	Interval	Motivation
PM7	5 minutes	Necessary interval for a correct control of the state of health.
PM8	5 minutes	Since AutomatedSOS is build on top of the data recollected by TrackMe, 5 minutes allows the system monitor the health state of the user.
PM9	7 days	Since this parameters is entered manually by the user, 7 days is a period long enough to not disturb users and to collect enough data to be useful.

Table 26: Intervals at which recollection is performed

Parameter	Input	Description of input
PM4	Slider (8 to 100)	An slider with a minimum of 8 and a maximum of 100 years. The client will be able to select two numbers using two handlers. The input will formulate a query in which all the dates between the 1° of January of the actual year minus the second number and the 1° of January of the actual year minus the first number are included.
PM5	Dropdown	A dropdown with two options. The first option is M and the second F. The input will formulate a query in which if the first option is selected, the query will return data from male users. If the second option is selected, the query will return data from female users.
PM6	Map	An interactive map centred in the city of Milan. The map should allow the drawing of an area. The input will formulate a query in which all the points inside the aforementioned area are include.
PM7	Map	An interactive map centred in the city of Milan. The map should allow the drawing of an area. The input will formulate a query in which all the points inside the aforementioned area are include.
PM8	Slider (40 to 120)	An slider with a minimum of 40 and a maximum of 120 bpm, these values are based on [11]. The client will be able to select two numbers using two handlers. The input will formulate a query in which all the numbers between the first number and the second number are included.
PM9	Slider (40 to 300)	An slider with a minimum of 40 and a maximum of 300 kg. The client will be able to select two numbers using two handlers. The input will formulate a query in which all the numbers between the first number and the second number are included.

Table 27: Inputs to be displayed to the clients