Politecnico di Milano
AA 2018-2019

Computer Science and Engineering
Software Engineering II

Requirements Analysis and Specification Document

Stefano Bagarin
Alessandra Pasini

Document version: 1.0
November 11, 2018

Contents

Section 1

Introduction

1.1 Purpose

The aim of this RASD is to provide a well detailed description of the *TrackMe*'s system. This is done by supplying a precise presentation of the proposed solution and its purposes by giving all goals that we are intent to reach with it and all domain assumptions and requirements necessary to achive them. This document has been redacted to be used by users and third parties with the objective to make the *TrackMe*'s system clear.

The system is designed as two separated software applications: one, called *Data4Help and AutomatedSOS*, is used for track and collect the user's health parameters and location and to call the SOS system in case of health parameters lower than the threshold values; the other, called *Data4Help* used by third parties for require, receive and analize data of single users or groups of users.
The users' appilcation *D4H and ASOS* is design to be an health tracker; it has the aim to collect and monitor his/her health parameters during the day and, in case of values lower than the threshold ones, to contact an SOS service and to give it the user's location. On the other hand the third parties's app *D4H* is design to make requests of single users' data or group of users' data, to receive them in case of request's acceptance or a notification in the opposite case.

1.2 Goals

The goals of *TrackMe*'s system are the following:

[**G1**] Allows a person to register and to have a personal area to which he/she can access with his credentials.

[**G2**] Allows the third party to register and to have a personal area to which it can access with his credentials.

[**G3**] Allows the third party to require data.

- [**G3.1**] Third party can require single person's data.
- [**G3.2**] Third party can require anonymized data of group of people.

[**G4**] Allows the user to accept or not to let a third party to have access to his/her data.

[**G5**] Allows third party to subscribe for new data.

[**G6**] Allows third party to see users' or groups of users' data obtained through a successful request.

[**G7**] Allows users to monitor their health parameters.

[**G8**] Allows users to activate or deactivate the *ASOS* service on top of *D4H*.

[**G9**] Allows an unhealty user to receive quick help if have the *ASOS* service activated on his account.

1.3 Scope

1.3.1 Description of the given problem

As already said in the previous section *TrackMe*'s system as the aim to provide different applications to the different actors.

- *D4H and ASOS* app is designed to provide the users with an overview of their health parameters by obtaining them from the device associated to the smartphone that runs it. The device should be able to collect health parameters such as the heart beat and send them to the main app running on the smartphone. If the user has activated the *ASOS* service everytime that a new value is collected the app checks if it is higher or not than the trashold value:
  - if it is upper it is stored in the database;

  - if it is lower the *ASOS* service will be activated and it will contact the outside SOS service in less than 5 seconds from the time the parameters started to be below the threshold. By contacting the SOS service *ASOS* will also give the location of the user who needs medical help.

  The user can always choose to activate the *ASOS* service by subscribing to it in the designated area. The user can also monitor his/her health parameters by looking the specific app's area.
  Anytime a third party will ask to accede to the single user's data the app will send a notification to the specific user to get permission to share his/her data. The user can accept or deny the request.

- *D4H* app is designed to provide the third parties with the possibility to require single user's or group of users' data. If the third party wants to accede to the data of a specific user it must have his/her secure number or fiscal code. The app is than in charge to ask to the specific user if he/she wants to share his/her data with the third party: if the user accepts, data will be shared; if he/she denies, the third party will receive a notification saying that.
  To obtain data of groups of users the third parties must specify some contraints to define the type of data and the type of uses they are interested in; this means that the query can be personalized depending on which type of data the third parties need. The app accepts those requests and sends data if and only if the quantity of users that respect the query's parameters is heigher than 1000; this threshold has been imposed to guarantee users' privacy. in case the number of users is lower than 1000 the third party will receive a notification saying that.
  Throught this app a third party can also subscribe for new data, once a request has been accepted the app asks if it would like to receive more data related to the single user or to the group of users of the request.

1.3.2 World and Machine

The aim of this subsection is to describe the problem following the World and Machine model. The World domain is the set of all real-life events in which the Machine will be introduced and in which its actions will be observed. The Machine domain is the set of all phenomena which it can control such as algorithms, devices, inputs from the world.
Those two domains are connected via Shared Phenomena which are World's events and Machine's actions which are observable by both domains.

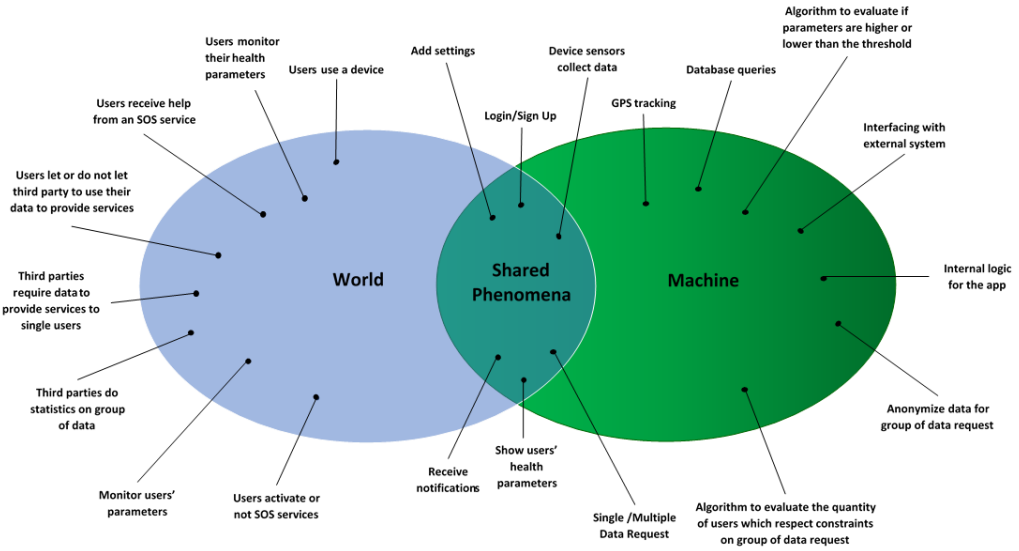

Figure 1.1: Figure 1: World and Machine diagram

1.4 Definitions, Acronyms and Abbreviations

1.4.1 Definitions

- **TrackMe's system:** the whole software, it's a compact way to refer to all different services that must be developed.
- **Smartphone:** HW that runs the app and the location service.
- **Device:** HW that collects all user's health parameters.
- **Personal data:** name, surname, date of birth, address, fiscal code and sex of an individual.
- **Third party data:** name, type of society, p. iva, fiscal code, address.
- **Data:** location and health parameters collected by the user's device.
- **Location:** real time GPS position.
- **Health parameters:** data about the user's health status collected by his/her device, such as heart beat, vein pressure, body temperature ..
- **User:** an individual user of the application who provides data.
- **Third party:** somebody that uses the application because interested in obtaining data of users or groups of users.

- **Unhealty user:** user who has health parameters below a certain threshold.
- **Credentials:** the unique couple of e-mail and password associated to each user or third part.
- **SOS service:** the service contacted to send help to the user in case of health parameters below thresholds.

1.4.2 Acronyms

- **RASD:** Requirement Analysis and Specification Document.
- **GPS:** Global Positioning System.
- **HW:** hardware.
- **API:** Application Programming Interface.
- **D4H:** Data4Help.
- **ASOS:** AutomatedSOS.

1.4.3 Abbreviations

- **[Gn]:** n-th goal
- **[Dn]:** n-th domain assumption
- **[Rn]:** n-th requiremnt
- **App:** appllication

1.5 References

This document is strictly based on the specification concerning the RASD assignment for the Software Engineering II project, part of the course held by professor Matteo Rossi and Elisabetta Di Nitto at the Politecnico di Milano, A.Y 2018/2019.

1.6 Document Structure

This document consist in four sections:

- **Section 1- Introduction:** the aim of this chapter is to give a general introduction and short overview of the *TrackMe*'s system by listing goals that must be reached; the scope of the two application and some important semantic definitions necessary to fully understand the document.
- **Section 2 - Overall Description:** the aim of this section is to show a general perspective about the structure and the functioning of the software. This is achieved by presenting some descrictive diagrams, the descriptions of the major functions of the software and the assumptions under which it will work.
- **Section 3 - Specific Requirements:** the aim of the third section is to specify all requirements necessary to reach goals listed in chapter one; provide some possible scenarios which show some possible usages of the two app ; define the use cases by providing also some explicative diagrams and describe all possible limitation related to hardware and software. It also provide a description of external interfaces, in particular users' and third parties' interfaces are described through some mockups.
- **Section 4 - Alloy:** contains the alloy model of and the discussion of its purposes. It also provide a possible world generated by the model.
- **Appendix A:** contains the software and tools used, the bibliography and all efforts spent by each group component.

Section 2

Overall Description

2.1 Product Perspective

The main goals of the *Track Me*'s system are to provide to the third parties a big pool of users from which can gather data (offered by the base service of *D4H*) and to provide to the users a timely medical assistance in case of illness (offered by the SOS service on top of *D4H*). The system is designed to be used as two different application: one for users and one for third parties. The users' application sends data about them to *Track Me*'s servers where then they can be collected by third parties throught theirs application by making a request for receiving data of a specific users or anonimized data of multiple users. *Track Me* uses user's data also for garanting a timely medical assistance to the users who have activated the *ASOS* service in case of drop of theirs health parameters gathered by theirs application by contacting an external SOS service.

The following diagram is provided to better describe the *Track Me*'s system:

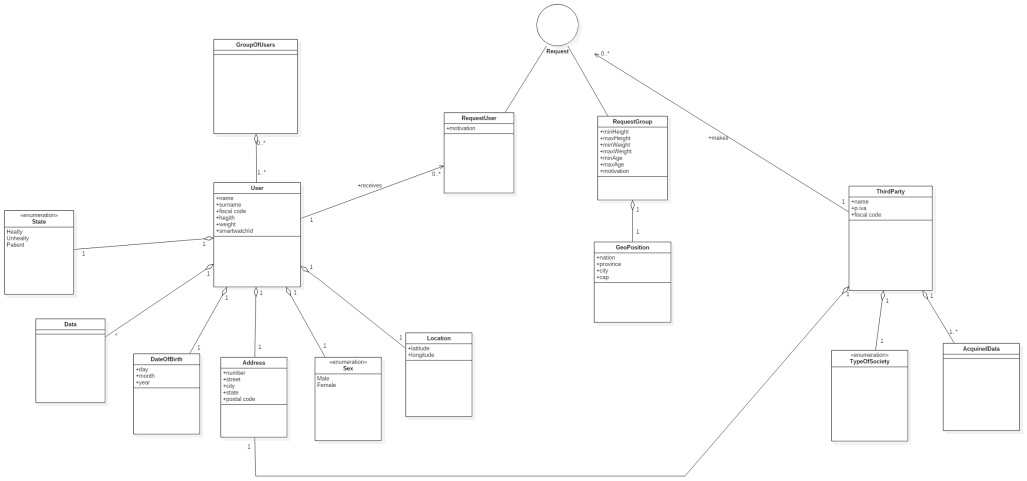

Figure 2.1: Figure 1: Class Diagram

As shown, all the system is centered on garanting data to third parties of users or groups of users and garanting medical assistance for the users who have activated the SOS service (the attribute "state" of the user is intended right for this scope).

In the next diagrams the processes of acquisition data of users and groups of users by third parties are analyzed:

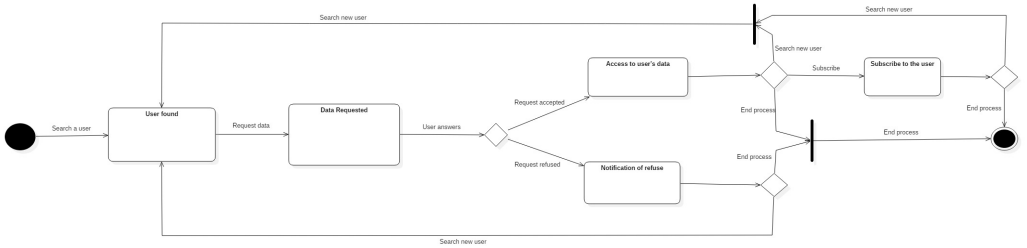

Figure 2.2: Figure 2: State diagram of the process of acquisition data of users by third parties.

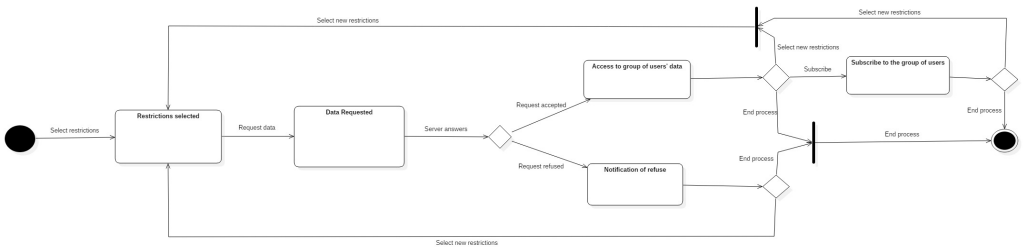

Figure 2.3: Figure 3: State diagram of the process of acquisition data of groups of users by third parties.

Notice that both diagrams describe the sequence of actions to make a single or multiple requests and relavite subscriptions to users/groups of users. Notice also that the two diagrams have a similar structure but are conceptually different and takes different actors (in particular no user takes an action in the second diagram).

2.2 Product Functions

In this section are specified the major function of the *TrackMe*'s system according to the presented goals.

- **Request data to users**
  Third parties must be able to request data directly to the users if and only if they can provide to the app the fiscal code or the social security number of the user and the reason why they would like to see the user's parameters. The requestes about single users can be done in the apposite area of third parties' app.
  When a user receives a request for his data, he must be able to reply positively or negatively selecting the relative icon on the data request notification which contains the reason of the request .
  Data collected throught a successfully request will be shown in a graph representing the trend of the user's health rate along the current day, week, month and year in the area of third parties' app about collected data.

- **Request data to groups of users**
  Third parties must be able to request data of groups of users to *D4H*. The groups of users must be specified by selecting the range of age, weight, height and address in which the third parties are interested. The requestes about groups of users can be done in the apposite area

of third parties' app.
The requestes about groups of users must be take in charge by *D4H* which will procede to guarantee the data only if the users who respect the ranges specified by the third parties are more than 1000.
Data of groups of users received by third parties must be strictly anonymized.
Data collected throught a successfully request will be shown in a graph representing the trend of the users' health rate along the current day, week, month and year in the area of third parties' app about collected data.

- **Subscription service**
  Third parties must be able to subscribe to new data of users or groups of users after a successfully request for his/them data by selecting the apposite icon on the data request accepted notification.
  Third parties subscribed to new data of users or groups of users will receive them as soon as they are produced and can be seen in the area of third parties' app about collected data.

- **Monitor health parameters**
  Users must be able to check their health parameters in real time. Those parameters will be shown in the apposite area of the users' app in a graph representing the trend of the users' health rate along the current day, week, month and year; in this area will be also possible to see the weight and height parameter set by the user.

- **Medical assistance service**
  Users must be able to activate or deactivate the *ASOS* service on their account by the *ASOS* area of the users' app.
  Users with the *ASOS* service activated on their account will have a guaranteed tempestive medical assistance in case of drop of their health parameters below a certain threshold. The medical assistance is guaranteed to be informed about the unhealty user in a maximum of 5 seconds from the drop of the user's health parameters. *TrackMe* will give to the SOS service in charge of the medical assistance the location, the health parameters and the personal data of the unhealty user to guarantee the best possible service.

2.3 User Characteristics

The following actors are the users of those two applications:

- **User:**
  a person successfully registered to *D4H* and *ASOS* and that is able to use all its services to monitor his/her health parameters.
- **Third Party:**
  a company, a foundation or any other type of institution that is successfully registered to *D4H* and that is able to use all its services to obtain data and see users' or groups of users' health parameters.

2.4 Assumptions and Dependencies

The domain assumptions of *D4H* are the following:

[**D1**] The user has a device linked to his smartphone.

[**D2**] The user's smartphone can provide an accurate enough current location.

[**D3**] The user's device can provide accurate enough healt parameters.

[**D4**] The user's smartphone can provide constantly data to *D4H*.

[**D5**] There is an SOS service that has the capability to receive emergency calls by *ASOS*.

- [**D5.1**] The SOS service has the capability to receive data about the unhealty user.
- [**D5.2**] The SOS service has the capability to send assistance to the unhealty user.

Section 3

Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interface

The following mockups represent a basic idea of how the users' mobile app will look like in the first release.

Figure 3.1: Mock up - Login screen.

Figure 3.2: Mock up - Registration.

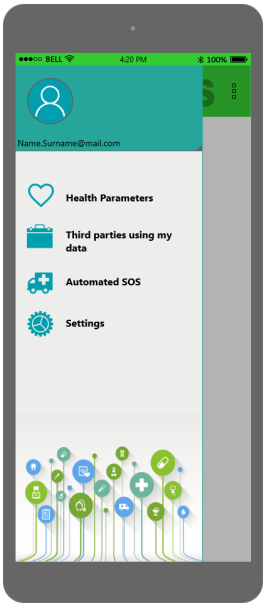

Figure 3.3: Mock up - Menu.

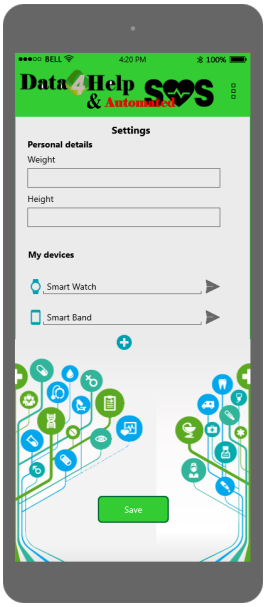

Figure 3.4: Mock up - Settings.

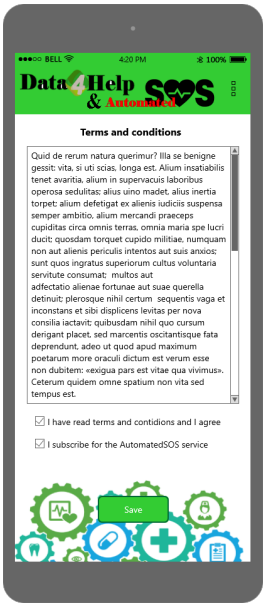

Figure 3.5: Mock up - Subscribe to *ASOS*.

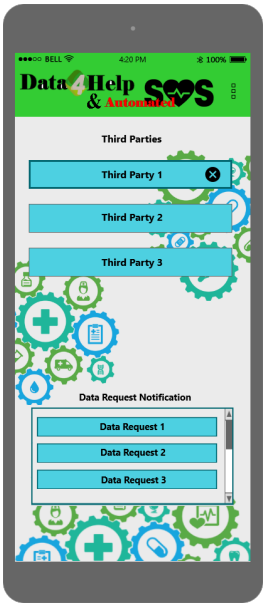

Figure 3.6: Mock up - Third party using my data.

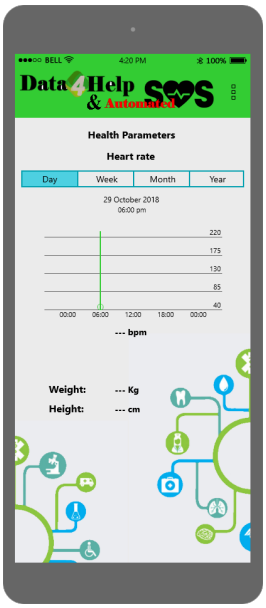

Figure 3.7: Mock up - Health parameter.

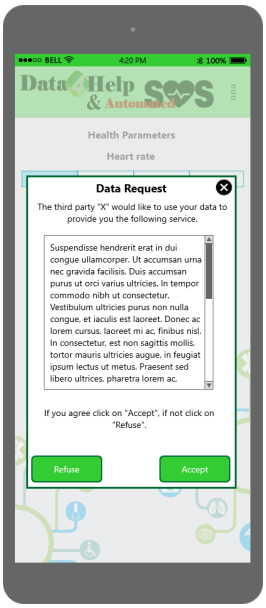

Figure 3.8: Mock up - Allert to accept or deny .

3.1.2 Thrid Party Interface

The following mockups represent a basic idea of how the third party's mobile app will look like in the first release.

Figure 3.9: Mock up - Login screen.

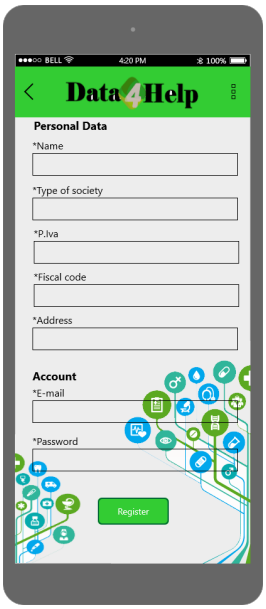

Figure 3.10: Mock up - Registration.

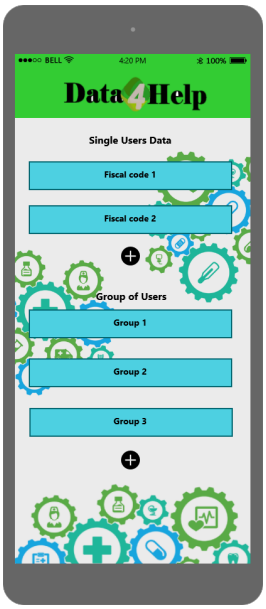

Figure 3.11: Mock up - Main scene.

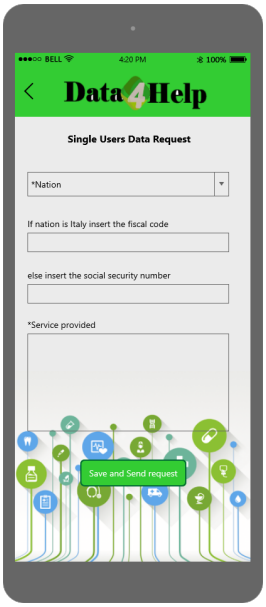

Figure 3.12: Mock up - Single data request.

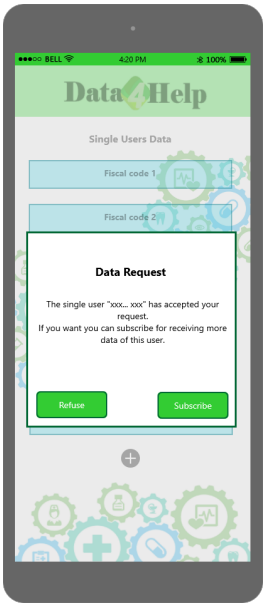

Figure 3.13: Mock up - Positive notification.

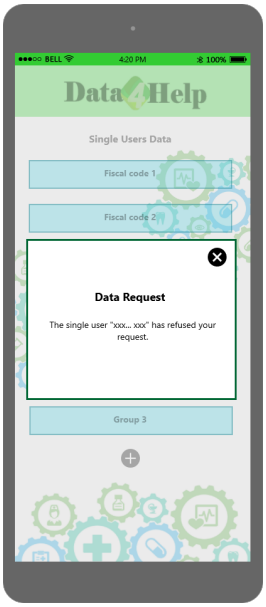

Figure 3.14: Mock up - Negative notification.

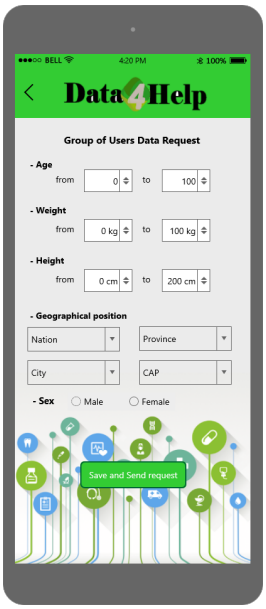

Figure 3.15: Mock up - Group of data request.

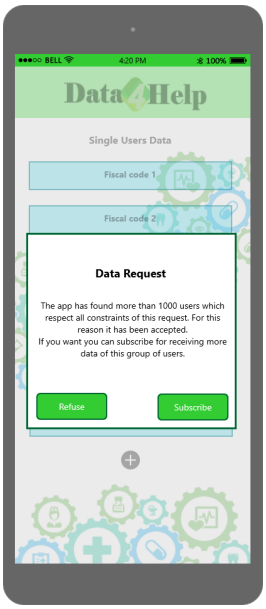

Figure 3.16: Mock up - Positive notification.

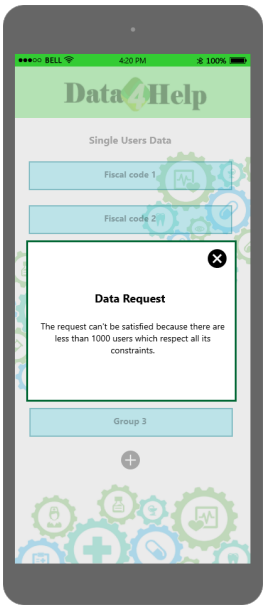

Figure 3.17: Mock up - Negative notification.

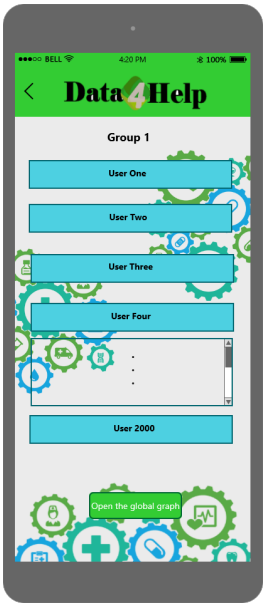

Figure 3.18: Mock up - Users in a group.

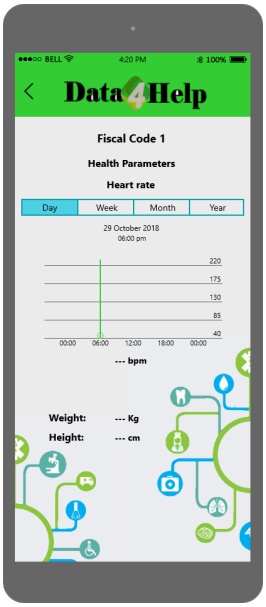

Figure 3.19: Mock up - Single user's parameter.

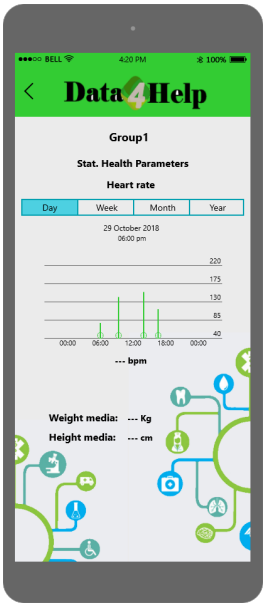

Figure 3.20: Mock up - Statistic par. of a group.

3.1.3 Hardware Interface

This is a software application whose main function is to collect data from single user, share them with third parties and give a medical service.
The app for users requires using a smartphone that can access to internet, that can exploit GPS services to track their location and give it to the SOS service in case of unhealty users and bluetooth to connect the smartphone to the device which will collect health parameters. All sensors needed to collect and share health parameters are already installed on the device.
The app for third parties only requires a smartphone or a computer which has access to the internet in oder to communicate with the system servers.

3.1.4 Software Interfaces

The server side of the app needs the following software products:

- Java EE (the latest available version is the 8.2 and it is possible to download it at `https://goo.gl/T0IdR0`);
- MySQL (the latest available version is the 5.7 and it can be download at `https://dev.mysql.com/downloads/mysql/5.7.html`).

It is also necessary a software that will be able to automatically connect the server to the informative system of E112, the european emergency number(USA and some other states have a software which automatically redirect the E112 call to their emercency centers). By a research this software is provided only by some automobilistic houses like BMW and Volvo and by a new device called eCall that has been introduced in Europe as mandatory on vehicle of classes M1 and N1 since the 31-03-2018.

3.1.5 Communication Interface

Users and third parties communicate with the server via *HTTPS* protocol. A safe and stable communication must be always guarantee between the server and the SOS service everytime that a user becomese an unhealthy one; this is necessary in order to guarantee a really high performance and to send the correct data.

3.2 Scenarios

3.2.1 Scenario 1

Matteo bought a new smart watch which is compatible with his smartphone and, because of this, he would like to download on his phone an application to monitor his health parameters. A friend of him recommended the *D4H* and *ASOS* app and Matteo immediately downloaded it. After he registered himself by giving some personal data, his e-mail and by choosing a password he added some information in settings:

- Matteo inserted his weight and height;
- he linked his smart watch to the smarphone throught the application to obtain the health parameter;

Then he accepted to activate the *ASOS* service by going in the *ASOS* are in the main menu. Matteo started to use this application and now he can constantly check his health parameters and see all third parties which has required and which use his data.

3.2.2 Scenario 2

The Health-Tech society would like to obtain data of group of users to analyze and to do some inference on them without spend money to develop a

software to collect data from users.This society decides to download the *D4H* application. After the registration done by giving some personal data, e-mail and a chosen password Health-Text logins to his personal area and can start to require data of a group of users. To require those data the society puts some constraint to obtain data of a specific group of users:

- users must have an age between 20 and 30;
- users can be both male or female;
- the society wants to analyze data of users with a weight between 50 kg and 70 kg;
- users must have a height between 160 cm and 180 cm;
- they must live in Lombardy;

After a while the app notifies that the request has been accepted because the number of users that respect the constraints is higher than the treshold applied to guarantee the users' privacy.

3.2.3 Scenario 3

The Health-Tech society would like to obtain data of a group of users to analyze them. It specifies those constraints for the group of users:

- users must have an age between 16 and 18;
- users must be male;
- the society wants to analyze data of users with a weight grater than 100 kg;
- users must have a height between 160 cm and 180 cm;
- they must live in Sondrio;

After a while the app notifies that the request has been denied because the number of users which respect the constraints is smaller than the treshold applied to guarantee the users' privacy.

3.2.4 Scenario 4

The Health-Tech society would like to provide some personalized services to a user. An employee enters in the society's *D4H* personal area and does a request for obtaining the specific user's data. To do it Health-Tech must have a NumberId related to the user or it must have his/her fiscal code and it must explain the reason why it is requiring those data. After a while the app notify that the user has accepted the request and so the society starts to receive data and subscribes for obtaining new data from the user.
The app constantly sends the user's data to the Health-Tech personal area.

3.2.5 Scenario 5

The Health-Tech society would like to provide some personalized services to a user. An employee enters in the society's *D4H* personal area and does a request for obtaining the specific user's data. To do it Health-Tech must have a NumberId related to the user or it must have his/her fiscal code and it must explain the reason why it is requiring those data. After a while the app notify that the user didn't accept the request; the request is delated and the employee close the app.

3.2.6 Scenario 6

Diana is at university when the *D4H* and *ASOS* app sends her a notification requiring to accept or deny the possibility to share her data to a third party. She has chosen to deny it because she was not interested in the proposed service.

3.2.7 Scenario 7

Edoardo is at work when the *D4H* and *ASOS* app sends him a notification requiring to accept or deny the possibility to shere his data to a third party. He has chosen to accept it bacause he is interested in the proposed service. As soon as he has accepted he goes in the menu and enters in the area which shows all third parties. He can see that the new one has been added immediately.

3.2.8 Scenario 8

Rosaria is walking around the street, she is going to take a coffee with her niece when she has an heart attack. Fortunately she is dressing up her smart

watch which takes over that her health parameters are lower than the threshold. The app contacts the SOS service and sends to it Rosaria's personal details and data and her location in less than 5 seconds from the detaction. The SOS service immediately sends her an ambulance which brings her to the hospital in time to save her.

3.3 Functional Requirements

[G1] Allows a person to register and to have a personal area to which he/she can access with his credentials.

- **[R1]** A person can register in the application by providing his personal data, a unique e-mail and a password.
- **[D1]** The user has a device linked to his smartphone.
- **[R2]** A user can log in by providing his e-mail and password.
- **[R3]** Each user has a personal area associated with his credentials.
  - **[R3.1]** A user can see the third parties subscribed to his data in his personal area.
  - **[R3.2]** A user can add weight and height in his personal area.
  - **[R3.3]** A user can link his device to the app in his personal area.

[G2] Allows the third party to register and to have a personal area to which it can access with his credentials.

- **[R4]** A third party can register in the application by providing his third party data, a unique e-mail and a password.
- **[R5]** A third party can log in by providing his e-mail and password.
- **[R6]** Each third party has a personal area associated with his credentials.
  - **[R6.1]** A third party can see the data of users and groups of users obtained from his personal area.
  - **[R6.2]** A third party can require data to users or groups of users from his personal area.

[G3] Allows the third party to require data.

- **[D2]** The user's smartphone can provide an accurate enough current location.

[**D3**] The user's device can provide accurate enough healt parameters.

[**D4**] The user's smartphone can provide constantly data to *D4H*.

[**G3.1**] **Third party can require single person's data.**

[**R7**] A third party can require user's data by providing his fiscal code and the reason of the request.

[**G3.2**] **Third party can require anonymized data of group of people.**

[**R8**] A third party can require anonymized data of groups of users by providing constraints that define users.

[**R8.1**] The group of user to which require data can be specified by any combination of geographic areas, range of ages, sex, range of weights and range of heights.

[**R8.2**] Requests of data related to groups of users are accepted by *TrackMe* if and only if the amount of users which respect the given constraints is equal or grater than 1000.

[**G4**] **Allows the user to accept or not to let a third party to have access to his/her data.**

[**R9**] A user is notified by the *D4H*'s app when a third party makes a request to have access to the user's data.

[**R10**] A user can reply to a third party interested to his data from his personal area within 30 days.

[**G5**] **Allows third party to subscribe for new data.**

[**R11**] After a request has been accepted, the third party can decide to subscribe to the data of that user or group of users.

[**R12**] A user is notified by the *D4H*'s app when a third party decides to subscribe for his data.

[**R13**] A user can stop the subscribe to his data by a third party in any moment from his personal area.

[**R14**] If a user stops the subscribe of a third party to his data, the third party will be notified by the *D4H*'s app.

[**G6**] **Allows third party to see users' or groups of users' data obtained through a successful request.**

[**R15**] Third parties can see the obtained data directly on their personal area.

[**R16**] Third parties are notified by the *D4H*'s app if their's request for new data has been rejected.

[**G7**] **Allows users to monitor their healt parameters.**

[**R20**] A user can see his health parameters in his personal area.

[**G8**] **Allows users to activate or deactivate the *ASOS* service on top of *D4H*.**

[**R17**] Users can activate or deactivate the *ASOS* service from their's personal area.

[**G9**] **Allows an unhealty user to receive quick help if have the *ASOS* service activated on his account.**

[**D5**] There is an SOS service that has the capability to receive emergency calls by *ASOS*.

[**D5.1**] The SOS service has the capability to receive data about the unhealty user.

[**D5.2**] The SOS service has the capability to send assistance to the unhealty user.

[**R18**] ASOS is able to contact the SOS service in case of the rise of an unhealty user in 5s.

[**R19**] ASOS is able to send to the SOS service the unhealty user's data and personal data.

3.3.1 Use Cases

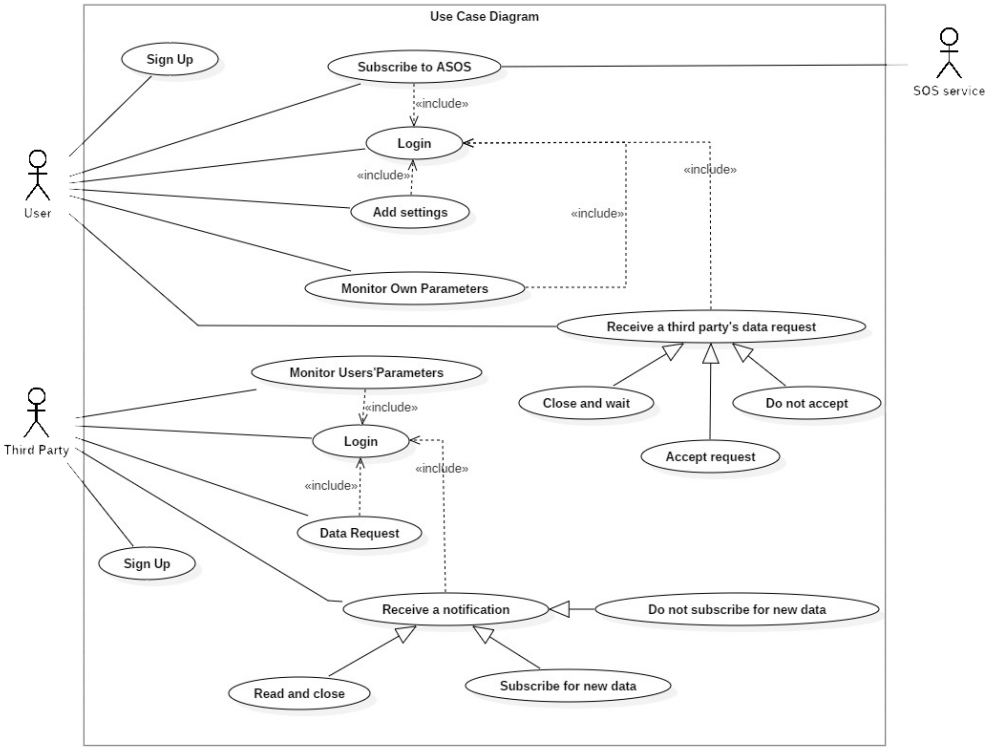

Figure 3.21: Figure 1: Use Case Diagram

Sign up

Name	Sign Up
Actors	User - Third party
Entry conditions	The individual or the third party has downloaded the app on his/her smartphone.

Event Flow	1. The individual opens the app on his/her smartphone; 2. He/she clicks on the *Register here* link; 3. Fills all the mandatory fields and provide all his/her personal data; 4. Clicks on "Register" button; 5. The system saves all personal data and creates a new personal area.
Exit conditions	The user or the third party has a personal area and now is able to use the application.
Exceptions	1. The user or the third party is already signed up; 2. The user or the third party doesn't fill all the mandatory fields; 3. The e-mail has already been registered. All the exceptions are handled by notifying the user and taking him back to the sign up activity.
Goals	G1 - G2
Requirements	R1 - R3 - R4 - R6

Login

Name	Login
Actors	User - Third party
Entry conditions	The user or the third party has the application installed; has succesfully registered and has a personal area.

Event Flow	1. The user or the third party opens the app on his/her smartphone; 2. Writes his/her credentials; 3. Clicks on "Login" button;
Exit conditions	The user or the third party is succesfully redirected to his/her main page.
Exceptions	1. The user or the third party doesn't register yet; 2. The user or the third party enters an invalid Username; 3. The user or the third party enters an invalid Password; 4. He/she doesn't fill all fields. All the exceptions are handled by notifying the user and taking him/her back to the login activity.
Goals	G1 - G2
Requirements	R2 - R3.1- R5 - R6.1

Add settings details

Name	Add settings details
Actors	User
Entry conditions	The user has the application installed; has successfully registered; has a personal area and has a device that can be linked to the smartphone.

Event Flow	1. The user opens the app on his/her smartphone; 2. He/she login; 3. Opens the Menu by the three points icon in the main scene; 4. Clicks on the *Settings* link; 5. Inserts his/her weight and height; 6. Adds his/her device; 7. Clicks the *Save* button.
Exit conditions	The app saves successfully all personal data and the device starts to send health parameters.
Exceptions	1. The user exits from the Settings area without saving, this exception is handled by notifying it to the user; 2. The user doesn't link any personal device, this exception can't be handled but the user notices it because without a linked device the app can't run correctely.
Goals	G1
Requirements	R3.2 - R3.3

Monitor own parameters

Name	Monitor his/her own parameters
Actors	User

Entry conditions	The user has the application installed; has succesfully registered; has a personal area, has added settings details and has a device linked to his/her smartphone.
Event Flow	1. The user login;
Exit conditions	The app opens the main scene in which the user can see his/her health paramters by graphs and numbers.
Exceptions	-
Goals	G7
Requirements	R20

Receive a third party's data request

In the figure 3.21 there are three use cases which derives from this abstract one; their difference is given by the last step of the event flow and the exit condition. Because of this reason we have chosen to compress them in just one table.

Name	Receive a third party's data request
Actors	User
Entry conditions	The user has the application installed; has succesfully registered; has a personal area and has a device linked to his/her smartphone.
Event Flow	1. The user receives a notification on his/her smartphone; 2. He/she login and finds an alert; 3. Reads the alert; 4. Click on the *Accept* button or on the *Refuse* button or on the *Exit* icon.

Exit conditions	The allert is closed and the choise is saved: 1. if he/she has clicked on the *Exit* icon the notification is added in the *Data Request Notification* area; 2. if he/she has clicked on the *Accept* button the third party is added in the *Third Party* area; 3. else it is delated.
Exceptions	1. The user doesn't accept neither refuse before the time bound. This exception is resolved by considering the non answer as a negative one.
Goals	G4
Requirements	R3.1 - R9 - R10

Receive a notification (third party)

In the figure 3.21 there are three use cases which derives from this abstract one; their difference is given by the last step of the event flow and the exit condition. Because of this reason we have chosen to compress them in just one table.

Name	Receive a notification
Actors	Third Party
Entry conditions	The third party has the application installed; has succesfully registered and has a personal area.

Event Flow	1. The third party receives a notification; 2. The third party login and finds an alert; 3. Reads the alert; 4. Clicks on the *Subscribe* button or on the *Refuse* button or on the *Exit* icon.
Exit conditions	The allert is closed and the choise is saved: 1. if the third party has clicked on the *Subscribe* button the single user data or the group of users data are added in the main scene and the third party will always receive new data ; 2. if the third party has clicked on the *Refuse* button the single user data or the group of users data are added in the main scene and the third party won't receive new data ; 3. else nothing happens.
Exceptions	-
Goals	G5 - G6
Requirements	R6.1 - R11 - R12 - R13 - R14 - R15 - R16

Subscribe to *ASOS*

Name	Subscribe to *ASOS*
Actors	User
Entry conditions	The user has the application installed; has succesfully registered; has a personal area and has a device linked to the smartphone.

Event Flow	1. He/she login; 2. Opens the menu and clicks on the *ASOS* link; 3. Checks the check boxes; 4. Clicks on the *Save* button.
Exit conditions	The app saves the choise and activate the *ASOS* service.
Exceptions	-
Goals	G8 - G9
Requirements	R17 - R18 - R19

Data request

Name	Data request
Actors	Third Party
Entry conditions	The third party has the application installed; has succesfully registered and has a personal area.
Event Flow	1. The third party login; 2. Clicks on the plus icon in the main scene: if the request is for single user data the first one else the second; 3. Fills all the mandatory fields with the required data ; 4. Clicks on the *Save and Send request* button.
Exit conditions	The app saves the request and starts to elaborate it.

Exceptions	1. The third party doesn't fill any field. The exception is handled by notifying the third party and taking him back to the Request Data activity.
Goals	G3 - G3.1 - G3.2
Requirements	R6.2 - R7 - R8 - R8.1

Monitor users' parameters

Name	Monitor users' parameters
Actors	Third Party
Entry conditions	The third party has the application installed; has succesfully registered; has a personal area and has received data of some single users and some group of users.
Event Flow	1. The third party login; 2. Clicks on a *Fiscal code X* button or on a *Group X* button;
Exit conditions	The app opens the health parameter of the user associated to the clicked fiscal code or to the group of users associated to the considered group.
Exceptions	-
Goals	G6
Requirements	R6.1 - R15

3.3.2 Sequence Diagrams

In this section are reported the most meaningful sequence diagrams in relation to some scenarios previously presented.

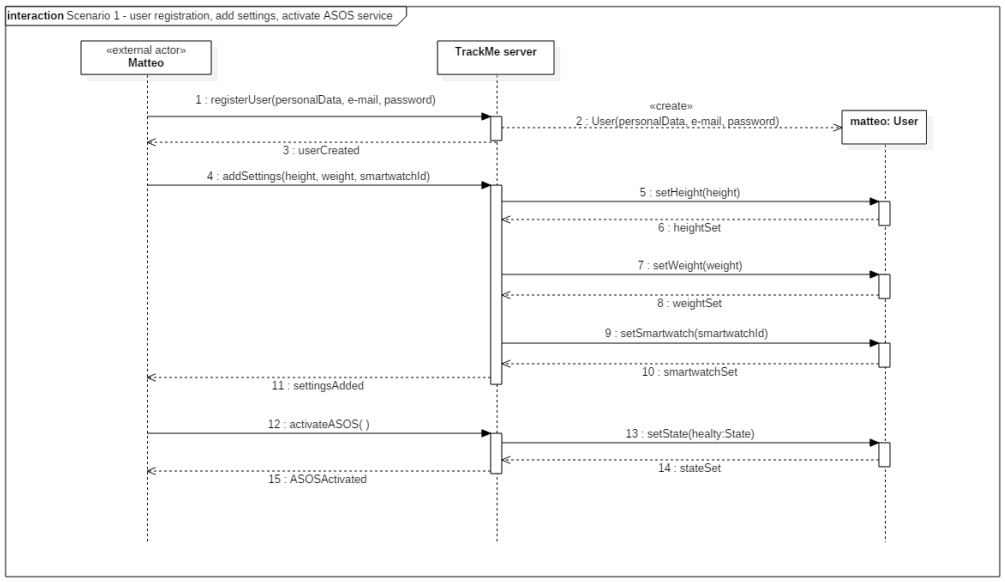

Figure 3.22: Figure 1: Sequence diagram relative to the scenario 1.

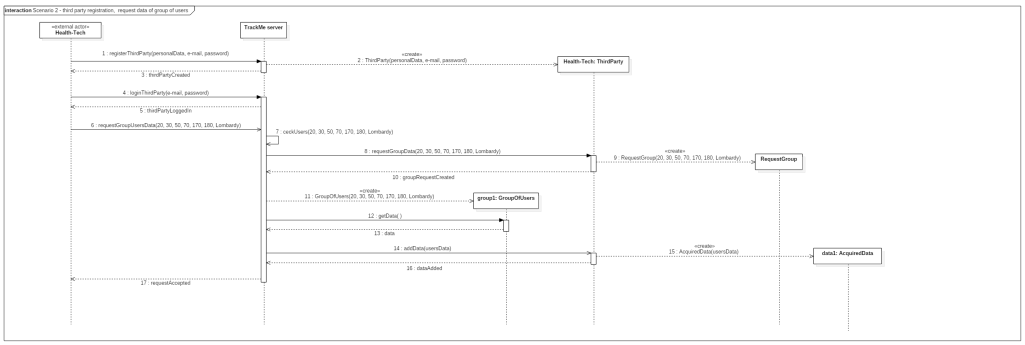

Figure 3.23: Figure 2: Sequence diagram relative to the scenario 2.

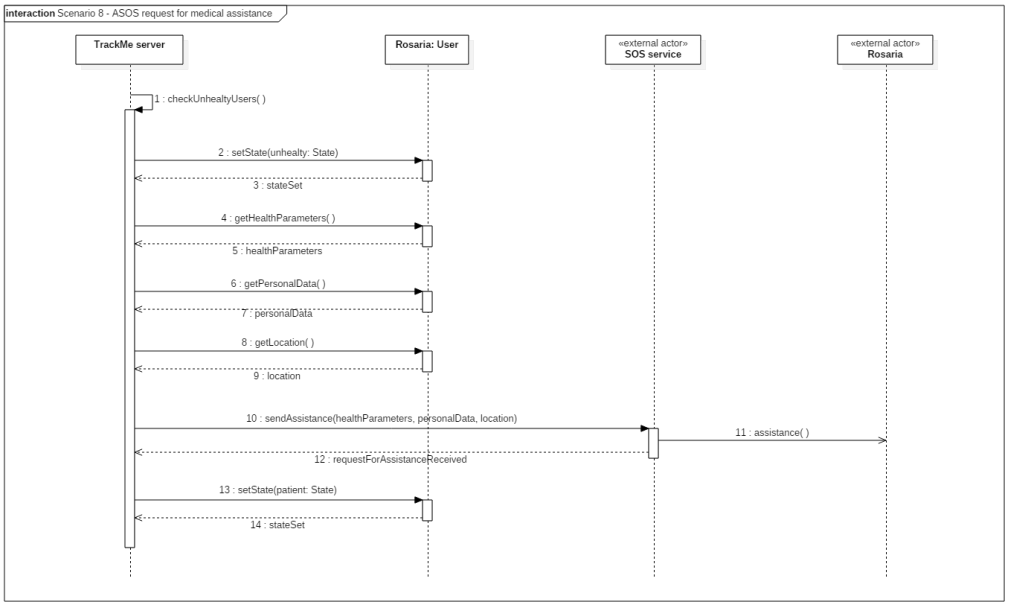

Figure 3.24: Figure 3: Sequence diagram relative to the scenario 8.

3.4 Performance Requirements

The system is provided to serve fairly great number of users and third parties simultaneously. Users will rely on the application in order to monitor their health parameters and to receive help in case of parameters lower than the threshold. Third party will rely on the application in order to require data of single user or group of users.

3.5 Design Constraints

3.5.1 Regulatory policies

The system must be allowed by users and third parties to collect, process and store their personal data; it also must be capable to delete them if a user or a third party requires it. While signing up users must give the permission to the system to share their data in an anonymized way and in the personal privacy respect.
Users and third parties must use the system in a propert way respecting law and policies.

3.5.2 Hardware limitations

The app for users requires a smartphone and a device that are compatible and which can be linked one to each other.
The smartphone must have the following properties:

- GPS;
- bluetooth;
- 3G or 4G internet connection;
- Android or IOS system.

The app for third parties requires a smartphone which has a 3G or 4G connection and an Android or IOS system.

3.6 Software System Attributes

3.6.1 Reliability

The application must be available 24/7. Negligibly small concessions might be tolerated.

3.6.2 Availability

In order to guarantee high degree of availability, the system should have more than one server because in this way, in case a server fails, the other one will be ready to take over. The system is expected to be available 99.99 percent of the time for the users which have subscibed to *ASOS*.

3.6.3 Security

Users and third parties credential should be confidentially stored and encrypted with high-security encryption. It must be guarantee to users that their data will be anonymized if they will be shared with a third party which has required a group of users data.

3.6.4 Maintainability

The app is going to be flexible and easy to maintain, it must be capable to facilitate addition of new features and options. For that purpose we will use clear code following the design patterns as much as possible.

Complete and detailed documentation will be provided in order to keep maintainability on the highest level.

3.6.5 Portability

The app will be written in Java EE so it could be exectued on any platform which support the Java Virtual Machine and runs JEE. The mobile app will have to work on both Android and IOS systems, the web one on the latest version of the main browsers as Google Chrome and Firefox.

Section 4

Alloy

4.1 Alloy Model

In this section is given the alloy model of data requests, which can be both request of data of a single user or request of data of a users' group.
It has been choise to model the data requests' use case because, in our opinion, it rappresent one of the most crucial and important activities that can be done throght *trackMe*'s system and because it needs to follow a few constraints which could be open to interpretation if explained by using natural language only.
The following contraints are insert in the model:

- data requests do not exist without a third party;
- a group of users doesn't exist if a data of a users' group request hasn't been done by a third party;
- each parameter retaled to a request can exist only if there's a request related to it;
- a user is in a group of users only if it respects all data requests' constraints.
- a group of users' request is accepted if and only if the number of users respecting all request's constraints is equal or grater than 1000.
- there are no constraints which define when and how a single data request is accepted, it depends from an undefined user choise.

To obtain a possible world the amount of users that must be in a request of a group of users has been drastically decremented.

```
open util/boolean

--SIGNATURES

some sig ThirdParty{
        requests : set DataRequest,
}

--data request can be of two different types
abstract sig DataRequest{
        accepted: one Bool,
}

--each single user request must be related to a user and a reason
sig SingleUserRequest extends DataRequest{
        user : User,
        reason: Reason,
}

--each group users request must be related to a group of users and a set of
    ↪ constraints
sig GroupUsersRequest extends DataRequest{
        groupOfUsers : one GroupOfUsers,
        constraints : one Constraints,
}

sig User{
        age : Int,
        weight : Int,
        height : Int,
        male: one Bool,
        female: one Bool,
        geoPosition : GeoPosition,
        healthParam : HealthParam,
}{
        age >0 weight >0 height >0  male≠ female
}

sig GroupOfUsers{
        users : set User
}

--all possible bounds that can define a group of users
sig Constraints{
        minAge : lone Int,
        maxAge : lone Int,
        minWeight : lone Int,
        maxWeight : lone Int,
        minHeight : lone Int,
        maxHeight : lone Int,
        male : one Bool,
        female : one Bool,
        geoPosition : lone GeoPosition,
}{
        0 < minAge minAge ≤ maxAge
        0 < minWeight minWeight ≤ maxWeight
        0 < minHeight minHeight ≤ maxHeight
}

sig GeoPosition{}
sig Reason {}
sig HealthParam{}
```

```
--
    ↪ ------------------------------------------------------------------------------------------
    ↪
--FACT

-- data requests do not exist whitout ThirdParty
fact noDataRequestsWithoutThirdParty{
        all r: DataRequest {one tp: ThirdParty | r in tp.requests}
}

--no GroupUsersRequest with the same group of users
fact groupUsersRequestsHaveDifferentGroups{
        no r1, r2 : GroupUsersRequest | r1 ≠ r2 and r1.groupOfUsers = r2.
            ↪ groupOfUsers and r1.constraints = r2.constraints
}

--no constraints without a group of data request
fact noConstWithoutGroupOfDataRequest{
        all c: Constraints {one gur: GroupUsersRequest | c in gur.
            ↪ constraints}
}

--no GroupOfUsers without a request of group of data
fact noConstWithoutGroupOfDataRequest{
        all gou: GroupOfUsers {one gur: GroupUsersRequest | gou in gur.
            ↪ groupOfUsers}
}

--no Reason without a SingleUserRequest
fact noReasonWithoutSingleUserRequest{
        all r: Reason {one sur: SingleUserRequest | r in sur.reason}
}

--no health parameter without a user
fact noReasonWithoutSingleUserRequest{
        all hp: HealthParam {one u: User | hp in u.healthParam}
}

--no data request is related to two or more different third parties
fact groupUsersRequestsHaveDifferentGroups{
        no tp1, tp2 : ThirdParty| tp1 ≠ tp2 and { no dr:  DataRequest | dr
            ↪ in tp1.requests and dr in tp2.requests}
}

--a user is in the GroupOfUsers of a Data Request if he/she respects all
    ↪ data request's constraints
fact userInGroup{
        all u: User | all gur: GroupUsersRequest | all gou: GroupOfUsers |
            ↪ (gur.groupOfUsers = gou and u in gou.users)  implies
        (u.age ≥ gur.constraints.minAge and u.age ≤ gur.constraints.maxAge
            ↪ and u.weight ≥ gur.constraints.minWeight
        and u.weight ≤ gur.constraints.maxWeight and u.height ≥ gur.
            ↪ constraints.minHeight and u.height ≤ gur.constraints.
            ↪ maxHeight
        and u.male = gur.constraints.male and u.female = gur.constraints.
            ↪ female and u.geoPosition = gur.constraints.geoPosition)
}

--a data request of a group of users is accepted iff the number of users in
    ↪ the group is >= 1000
fact groupUsersRequestIsAccepted{
        all gur :  GroupUsersRequest | gur.accepted = True <=> #gur.
```

```
            ↪ groupOfUsers.users ≥ 1000
}

--
    ↪ ------------------------------------------------------------------------------------------
    ↪
--PRED

--add a new data request to the set of requests in ThirdParty
pred addDataRequest [tp : ThirdParty, re : DataRequest]{
        tp.requests = tp.requests + re
}

--creation of a new SingleUserRequest
pred createNewSingleUserRequest [tp: ThirdParty, sur : SingleUserRequest, u:
    ↪  User, r: Reason]{
        addDataRequest [tp, sur]
        sur.user = u
        sur.reason = r
}

--creation of a new GroupUsersRequest
pred createNewGroupUsersRequest [tp: ThirdParty, gur: GroupUsersRequest, gou
    ↪ : GroupOfUsers,  c: Constraints]{
        gur.groupOfUsers = gou
        gur.constraints = c
}

pred show {}

--
    ↪ ------------------------------------------------------------------------------------------
    ↪
--ASSERT

assert allUsersInAGroupOfDataRequestRespectTheConstraints{
        all gur : GroupUsersRequest | all u: User | u in gur.groupOfUsers.
            ↪ users implies(
        (u.age ≥ gur.constraints.minAge or no gur.constraints.minAge) and (
            ↪ u.age ≤ gur.constraints.maxAge
        or no gur.constraints.maxAge) and (u.weight ≥ gur.constraints.
            ↪ minWeight or no gur.constraints.minWeight )  and
        (u.weight ≤ gur.constraints.maxWeight or no gur.constraints.
            ↪ maxWeight) and (u.height ≥ gur.constraints.minHeight
        or no gur.constraints.minHeight) and (u.height ≤ gur.constraints.
            ↪ maxHeight or no  gur.constraints.maxHeight)  and
        (u.male = gur.constraints.male or u.female = gur.constraints.female)
            ↪  and (u.geoPosition = gur.constraints.geoPosition
        or no  gur.constraints.geoPosition))
}

assert groupOfDataRequestHasMoreThanOneThousand{
        all gur : GroupUsersRequest | gur.accepted = True implies #gur.
            ↪ groupOfUsers.users ≥ 1000
}

check  groupOfDataRequestHasMoreThanOneThousand
check allUsersInAGroupOfDataRequestRespectTheConstraints
run createNewGroupUsersRequest
run createNewSingleUserRequest
run addDataRequest
run show for 5
```

4.2 Alloy results

These are the results of the model.

```
Executing "Check groupOfDataRequestHasMoreThanOneThousand"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   0 vars. 0 primary vars. 0 clauses. 50ms.
   No counterexample found. Assertion may be valid. 0ms.

Executing "Check allUsersInAGroupOfDataRequestRespectTheConstraints"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   13790 vars. 576 primary vars. 41707 clauses. 180ms.
   No counterexample found. Assertion may be valid. 596ms.

Executing "Run createNewGroupUsersRequest"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   11863 vars. 582 primary vars. 34814 clauses. 90ms.
   Instance found. Predicate is consistent. 127ms.

Executing "Run createNewSingleUserRequest"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   11883 vars. 582 primary vars. 34867 clauses. 68ms.
   Instance found. Predicate is consistent. 73ms.
```

```
Executing "Run addDataRequest"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   11799 vars. 576 primary vars. 34695 clauses. 120ms.
   Instance found. Predicate is consistent. 70ms.

Executing "Run show for 5"
   Solver=sat4j Bitwidth=4 MaxSeq=5 SkolemDepth=1 Symmetry=20
   23852 vars. 1040 primary vars. 66833 clauses. 191ms.
   Instance found. Predicate is consistent. 149ms.

6 commands were executed. The results are:
   #1: No counterexample found. groupOfDataRequestHasMoreThanOneThousand may be valid.
   #2: No counterexample found. allUsersInAGroupOfDataRequestRespectTheConstraints may be valid.
   #3: Instance found. createNewGroupUsersRequest is consistent.
   #4: Instance found. createNewSingleUserRequest is consistent.
   #5: Instance found. addDataRequest is consistent.
   #6: Instance found. show is consistent.
```

Figure 4.1: Results of check and run commands

4.3 Alloy worlds

In this section there are three different examples of worlds descibed by the model:

- figure 4.2 highlights the metamodel related to the code written above;
- figure 4.3 provides a sample of a third party with an accepted GroupUsersRequest and a refused one. To obtain an accepted GroupUsersRequest the bound of 1000 has been reduced to 2;
- figure 4.4 has the aim to show how the model behaves with a SingleUserRequest.

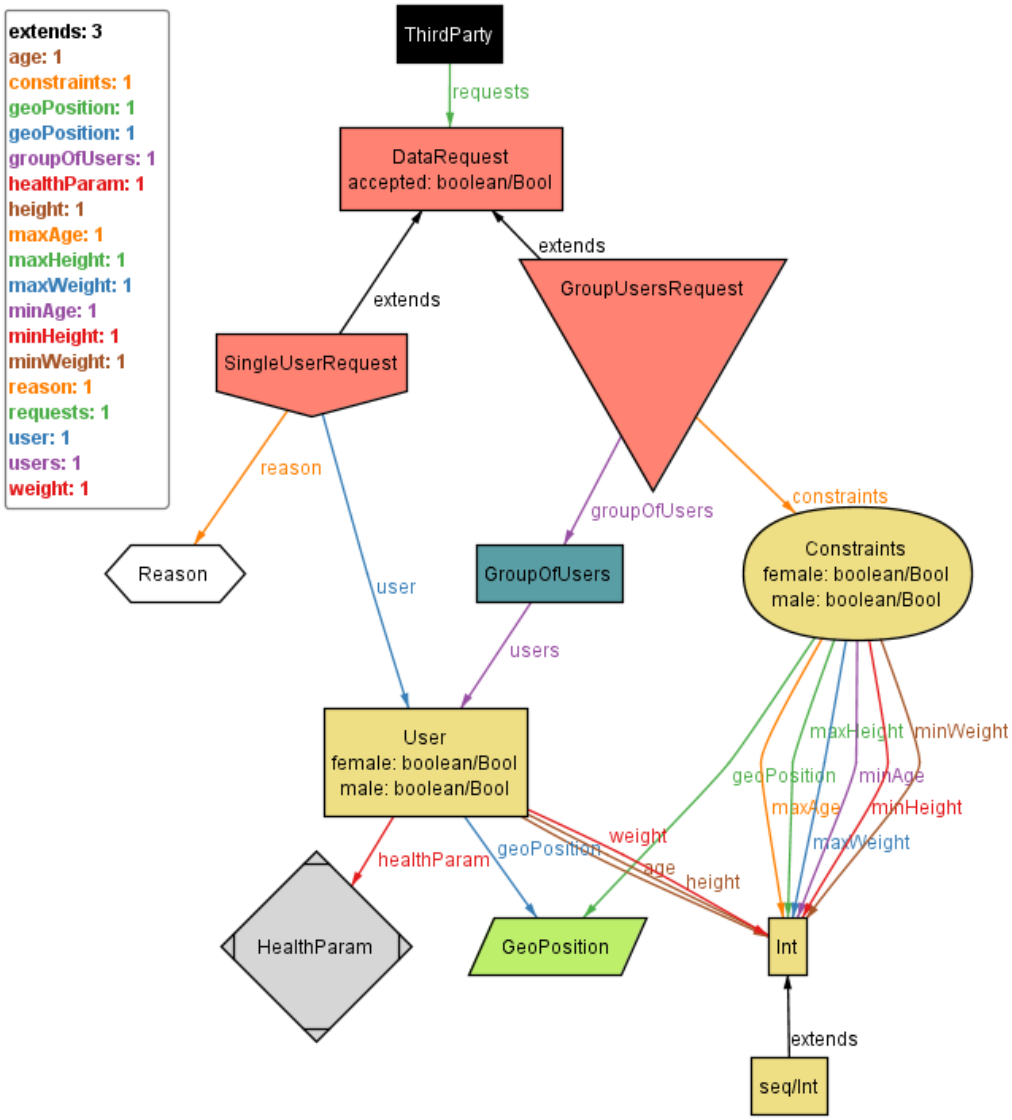

Figure 4.2: This figure represent the metamodel of the code written above

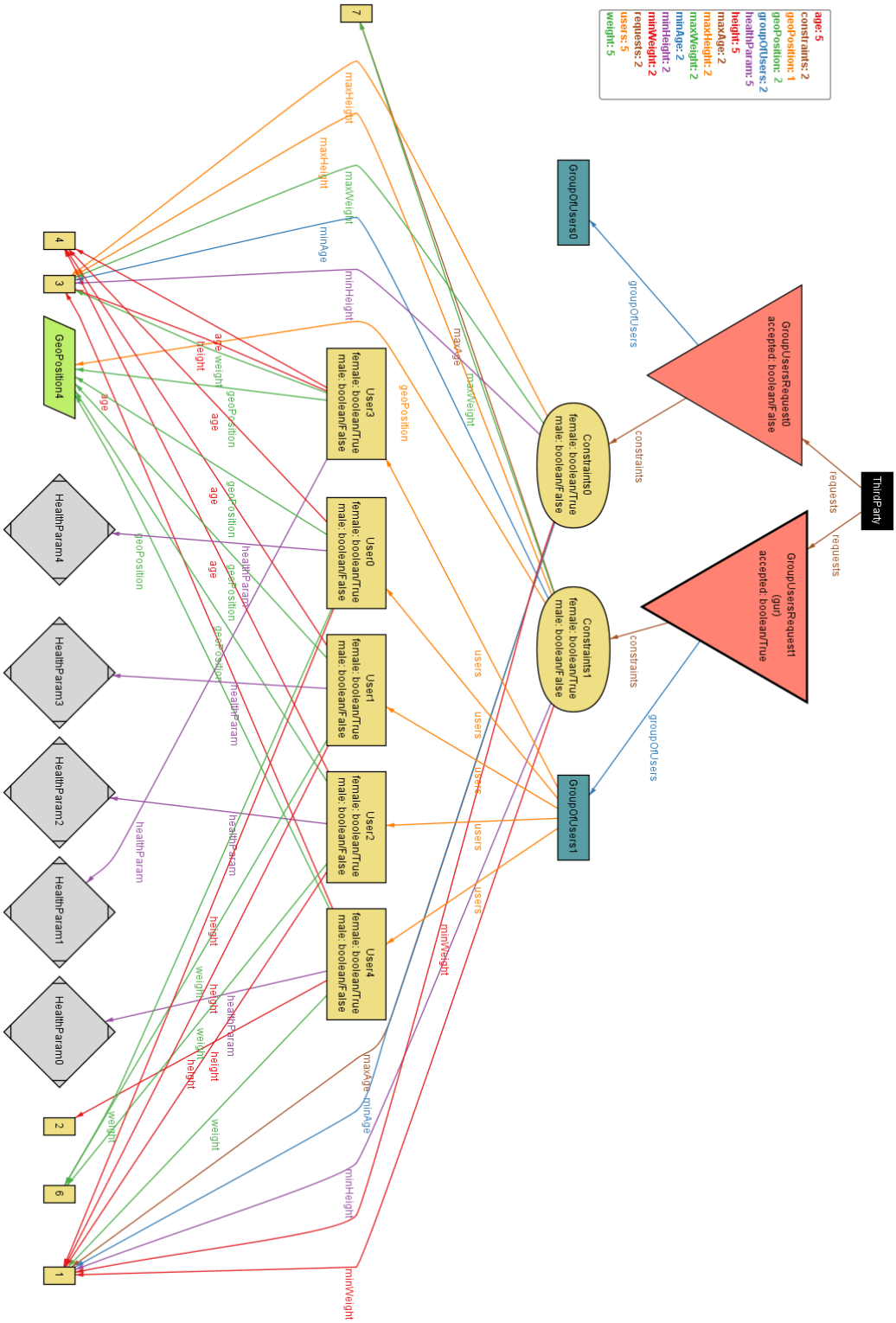

Figure 4.3: This figure represent a third party with two GroupUsersRequest

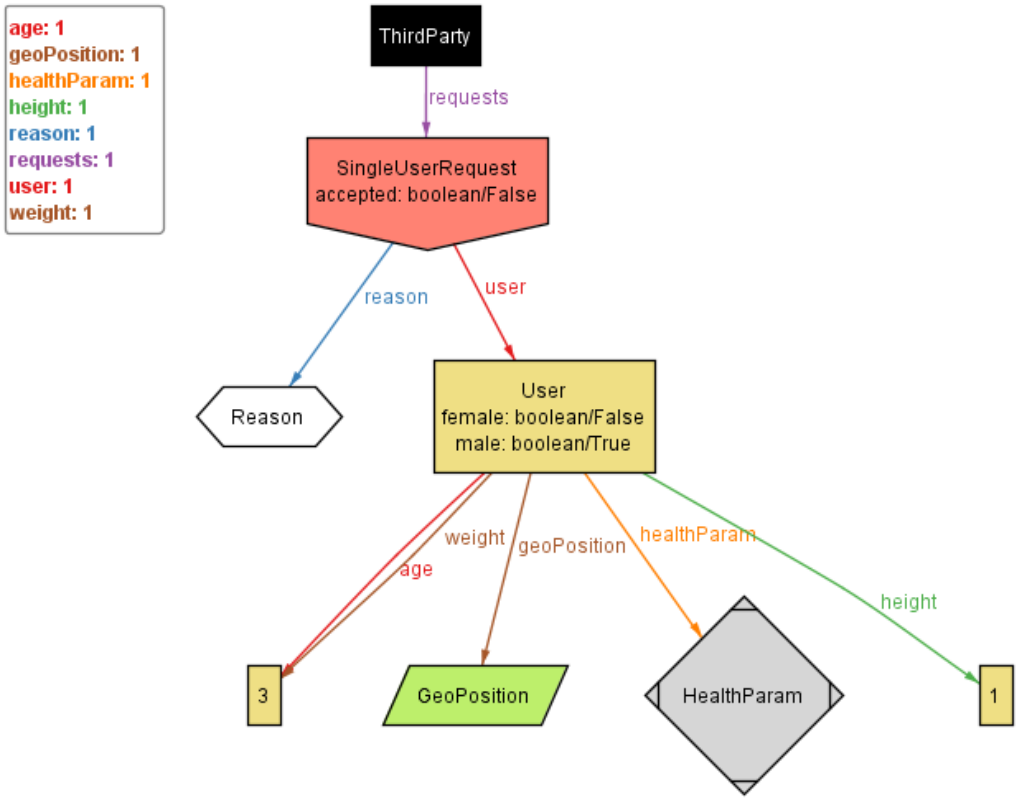

Figure 4.4: This figure represent a third party with a SIngleUSerRequest

Appendix A

Appendix

A.1 Software and tools used

The following are all tools used to produce this document:

- **LaTeX:** used as typesetting system to build this document;
- **Mockplus:** used to draw the mockups (the latest Windows' version is 3.4.1.0 and it can be found at `https://www.mockplus.com/download`);
- **StarUML:** used to draw all diagrams (the latest version is 3.0.2 and it can be found at `http://staruml.io/download`);
- **GitHub:** used to work in a distributed way and to manage different versions of the document (it can be download at `https://github.com`);
- **GitHub Desktop:** this is the official GitHub app which offers a simple and user friendly way to contribute to a git project (it can be download at `https://desktop.github.com/`);
- **Alloy Tool:** used to generate a formal model of an important part of the app (the version used is the 4.2 and it can be found at `http://alloytools.org/download.html`).

A.2 Effort Spent

The major part of the document has been produced working togheder and that's the reason way there is not a precise division of hours per sections and per group component.
The following is an approximate extimation of the number of hours of work for each group member:

- Alessandra Pasini: ~40 hrs;
- Stefano Bagarin: ~40 hrs;

A.3 Bibliography

- ISO/IEC/IEEE 29148:2011 *Systems and software engineering - Life cycle processes - Requirements engineering*
- all information related to 112 have been taken from a few official web sites:
  - `https://ec.europa.eu/digital-single-market/en/112`
  - `http://www.heero-pilot.eu/view/it/ecall.html`
  - `http://www.europarl.europa.eu/italy/it/`
    the article *Il sistema d'emergenza ecall obbligatorio dal 31 marzo sulle nuove auto*
- slides of *Software Engineering I - II*
- the alloy-style.sty library has been found at `https://github.com/Angtrim/alloy-latex-highlighting/blob/master/alloy-style.sty`