



**POLITECNICO**  
**MILANO 1863**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

---

# Requirement Analysis and Specification Document (RASD)

---

TRACKME

- v1.0 -

*Authors:*

**Avila**, Diego

**Schiatti**, Laura

**Virdi**, Sukhpreet

903988

904738

904204

November 11<sup>th</sup> , 2018

---

# Contents

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Context . . . . .	1
1.2	Purpose . . . . .	1
1.3	Scope . . . . .	2
1.3.1	Description of the given problem . . . . .	2
1.3.2	World and shared phenomena . . . . .	4
1.3.3	Goals . . . . .	5
1.4	Definitions, Acronyms, Abbreviations . . . . .	5
1.4.1	Definitions . . . . .	5
1.4.2	Acronyms . . . . .	6
1.4.3	Abbreviations . . . . .	6
1.5	Revision history . . . . .	6
1.6	Document structure . . . . .	7
<b>2</b>	<b>Overall description</b>	<b>8</b>
2.1	Product perspective . . . . .	8
2.2	Product functions . . . . .	10
2.3	User characteristics . . . . .	11
2.4	Assumptions, dependencies and constraints . . . . .	12
2.4.1	Domain assumptions . . . . .	12
<b>3</b>	<b>Specific requirements</b>	<b>13</b>
3.1	External interface requirements . . . . .	13
3.1.1	User interfaces . . . . .	13
3.1.2	Hardware interfaces . . . . .	16
3.1.3	Software interfaces . . . . .	17
3.1.4	Communication interfaces . . . . .	17
3.2	Functional requirements . . . . .	17

3.2.1	Use case diagrams . . . . .	21
3.2.2	Use cases description . . . . .	24
3.2.3	Activity diagrams . . . . .	40
3.2.4	Sequence diagrams . . . . .	40
3.2.5	Requirements traceability matrix . . . . .	42
3.3	Performance requirements . . . . .	44
3.4	Design constraints . . . . .	44
3.4.1	Standards compliance . . . . .	44
3.4.2	Hardware limitations . . . . .	45
3.4.3	Any other constraint . . . . .	45
3.5	Software system attributes . . . . .	45
3.5.1	Reliability . . . . .	45
3.5.2	Availability . . . . .	45
3.5.3	Security . . . . .	45
3.5.4	Maintainability . . . . .	45
3.5.5	Portability . . . . .	45
<b>4</b>	<b>Formal analysis using Alloy</b>	<b>46</b>
<b>5</b>	<b>Effort spent</b>	<b>47</b>
<b>6</b>	<b>References</b>	<b>48</b>

---

# List of Figures

---

1.1	High-level Description of the problem . . . . .	3
2.1	Domain Model . . . . .	9
2.2	Statechart Diagrams . . . . .	10
3.1	UI: TrackMe's Home Page . . . . .	13
3.2	UI: Login Page . . . . .	14
3.3	UI: Individual User's Registration page . . . . .	14
3.4	UI: Third Party's Registration page . . . . .	15
3.5	UI: Registered user's Dashboard . . . . .	15
3.6	UI: Registered user's Dashboard who can choose to Accept or Reject the requests	16
3.7	UI: Third party user's Dashboard to make requests . . . . .	16
3.8	Data4Help Use Case Diagram . . . . .	22
3.9	AutomatedSOS Use Case Diagram . . . . .	23
3.10	Track4Run Use Case Diagram . . . . .	24
3.11	Data4Help - Accept Request S7equence Diagram . . . . .	40
3.12	Data4Help - Access to bulk data and subscribe Sequence Diagram . . . . .	41
3.13	Track4Run - Accept invitation Sequence Diagram . . . . .	42

---

# List of Tables

---

1.1	Revision history timeline . . . . .	7
3.1	Traceability matrix . . . . .	44
5.1	Time spent by all team members . . . . .	47
5.2	Time spent by each team member . . . . .	47

---

# Introduction

---

## 1.1 Context

Nowadays, due to the availability of a huge variety of smart electronic devices, more and more applications are developed to help people in their day-to-day activities. In the healthcare field, wearable devices such as smartwatches are highly useful since they can be used to collect information about general well-being of users by means of mobile sensor technologies. As expected, measured data has several possible applications including, patient diagnostics and treatment or research motivations.

**TrackMe** is a company that develops health-monitoring devices devoted to measure and record different parameters related to the health status of a person (i.e. body temperature, blood pressure, heart pulse rate and percentage of O<sub>2</sub> in the blood) and also their location. TrackMe health smartwatch is synchronized with an app that gives users access to their data and stats.

## 1.2 Purpose

Taking into account the long list of currently available wearable devices, **TrackMe** is aware that the market is plenty of healthcare wearable devices, therefore, to stand out against its competitors, the key ingredient is improving customers experience, and doing so requires understanding their needs and how they interact with the system, in order to provide personalized recommendations and correctly oriented new services.

After analysing users behaviour, TrackMe decided to focus on third-party companies and profit from users data in a direct way (i.e. extend its business model by implementing **data trading**). This is, provide the collected data (i.e. location and health status) to third parties by means of a new software-based service called **Data4Help**.

Moreover, after some time, TrackMe realizes that a good part of its third-party customers want to use the data acquired through Data4Help to offer a personalized SOS service to elderly people, and decides to build a new service called **AutomatedSOS** to provide a personal alarm service to subscribed customers by monitoring their health status.

Finally, TrackMe realizes that another great source of revenues could be the development of a service to track athletes participating in a run. In this case, the service, called **Track4Run**, will allow run organizers to define the path, TrackMe wearable-devices users to enrol, and spectators to see on the map the position of all runners during the run.

## 1.3 Scope

### 1.3.1 Description of the given problem

The TrackMe environment will be composed by three systems, whose scope are defined in this section.

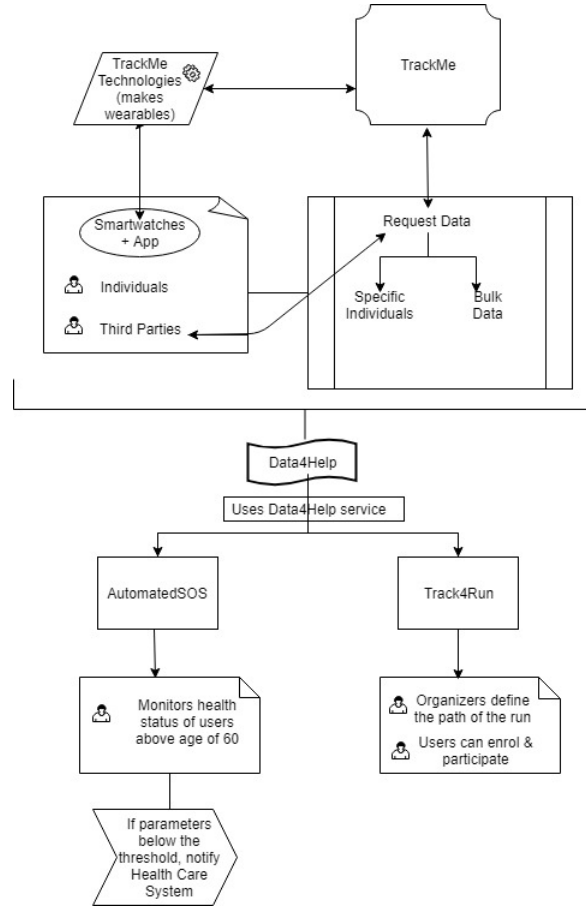


Figure 1.1: High-level Description of the problem

**TrackMe** develops its own health-monitoring smart-watch and bases the assumption that all registered individuals own the same to retrieve the necessary raw data (body temperature, blood pressure, heart pulse rate, percentage of  $O_2$  in the blood, current location) as input for the service **D4H**. TrackMe provides the user an interface for the registration of individuals as well as third parties. Individuals who register, agree to TrackMe acquiring their data. They are wirelessly connected to each other. We presume the data to be posted using a compatible application that comes with the health-monitoring device. As mentioned before, it also supports the registration of third parties. While doing so, we acknowledge the company is legally established by validating its certificate, who can thereafter request for the data of some specific individuals (using SSN) to whom the request will accordingly be sent. The individuals have the choice to subsequently accept or reject it. Alternatively, they can also ask for bulk data based on criteria filtered and provided by the system (such as age, gender, country, province) which will be handled directly by TrackMe. If the request for data acquisition is approved, TrackMe offers these third party customers to subscribe to the



new data, in real-time.

**ASOS** is built on top of Data4Help to provide an opportunity to users above the age of 60, to subscribe to a new SOS service. AutomatedSOS monitors the health status of these users. When vital parameters such as heart rate, blood pressure, body temperature, percentage of  $O_2$  in the blood are below certain thresholds, the health care system is automatically notified, which accordingly handles the arrival of an ambulance to the location. It must be noted that this post-notification management cannot be tracked.

**T4R** allows 'fit' users to participate in any upcoming run. If the user desires to participate they can accept the request and enrol themselves through a redirected link. We assume that organizers define a valid path viewable to all users before the run. Spectators (read : all users) can view the position of all the participants on the map during the run.

### 1.3.2 World and shared phenomena

- **World Phenomena**

In order to better understand which entities are relevant for the system and how they interact, it is essential to describe the real world events that are involved, they are

- TrackMe wearable devices.
- Individuals sharing their personal data.
- Third-party customers willing to use the data acquired through the devices.
- Healthcare system .

- **Shared Phenomena**

The shared phenomena is composed by all the relevant interactions between the world and the system. Every interaction is part of a relationship between entities in the real world and Trackme environment. The main ones are listed below.

- The physical health parameters collected by the Trackme devices (i.e. blood pressure, body temperature, etc.), and stored by Data4Help
- Individuals location collected by the Trackme device, and stored by Data4Help
- Healthcare system, that let AutomatedSOS send the alarms

- The running circuit defined in Track4Run
- The current location of the athletes participating in a run.

### 1.3.3 Goals

The goals are divided according to each service TrackMe wants to offer to its customers:

- **Data4Help**

- [G1] The individual could allow (or refuse) Data4Help to use their data.
- [G2] The third-party company should be able to access data of a specific individual
- [G3] The third-party company should be able to access anonymized data of groups of individuals under certain constraints.
- [G4] The third-party company should be able to access anonymized data of groups of individuals under certain constraints.

- **AutomatedSOS**

- [G5] Provide a service capable to send an advice to the health care service when a individual's parameters are below a defined threshold.

- **Track4Run**

- [G6] Run organizers could define the path for a given run and TrackMe users can enrol to it.
- [G7] Run spectators could track the position of all the runners during the race.

## 1.4 Definitions, Acronyms, Abbreviations

### 1.4.1 Definitions

- **Data trading:** Generate revenue from user data in a much more direct way, by selling user data to a third party.
- **Health status:** Collection of the last measured overall physical health parameters of a user or a group of users.

- **Remote monitoring:** Remote Monitoring (RMON) is a standard specification that facilitates the monitoring of network operational activities through the use of remote devices known as monitors or probes(here, we are using smartwatches).
- **Wearable device:** Devices that can be used to collect data and monitor users' overall physical health, such as body temperature, blood pressure, heart pulse rate, etc.
- **Third party company:** Customer who needs to know the health status of the population for different purposes (e.g. health insurance companies)

### 1.4.2 Acronyms

- RASD: Requirement Analysis and Specification Document
- D4H: Data4Help
- ASOS: AutomatedSOS
- T4R: Track4Run
- SSN: Social Security Number

### 1.4.3 Abbreviations

- $[Gn]$ : n-goal.
- $[Dn]$ : n-domain assumption.
- $[Rn]$ : n-functional requirement.
- $[UCn]$ : n-use case.

## 1.5 Revision history

It is important to keep track of the revisions made to this document:

---



---

Version	Last modified date
1.0	11 <sup>th</sup> November, 2018

---

Table 1.1: Revision history timeline

## 1.6 Document structure

This document is divided in six parts, each one devoted to approach each one of the steps required to apply requirements engineering techniques.

- Chapter 1 gives an introduction to the problem and describes the purpose of the application TrackMe. The scope of the application is defined by stating the goals and description of the problem.
- Chapter 2 presents the overall description of the project. The product perspective includes details on the shared phenomena and the domain models.
- Chapter 3 contains the external interface requirements, including: user interfaces, hardware interfaces, software interfaces and communication interfaces. Furthermore, the functional requirements are defined by using use case and sequence diagram. The non-functional requirements are defined through performance requirements, design constraints and software system attributes. All this aspects will be useful for the development team.
- Chapter 4 includes the alloy model and the discussion of its purpose. Also, a world generated by it is shown.
- Chapter 5 shows the effort spent by each group member while working on this project.
- Chapter 6 includes the reference documents.

---

# Overall description

---

## 2.1 Product perspective

In the previous section, the scope of the application was delimited and explained in a shallow way, but at this point it is useful to include further details on the shared phenomena and a domain model as a visual representation of the system.

The addition of brand-new services to TrackMe requires to enlarge the existing model in such a way that it can include the abstraction of those features. To explain in detail the way data will be organized in the upcoming system, the structure of TrackMe up to now will be treated as a "black box". This means, only those parts of the whole data model that will allow us to obtain users' basic information and collected data (required by Data4Help) will be considered.

On the other hand, AutomatedSOS and Track4Run are treated as third-parties that make requests for the data that Data4Help offers. Every time a user agrees to activate any of those services, a new request is sent to Data4Help to obtain his data.

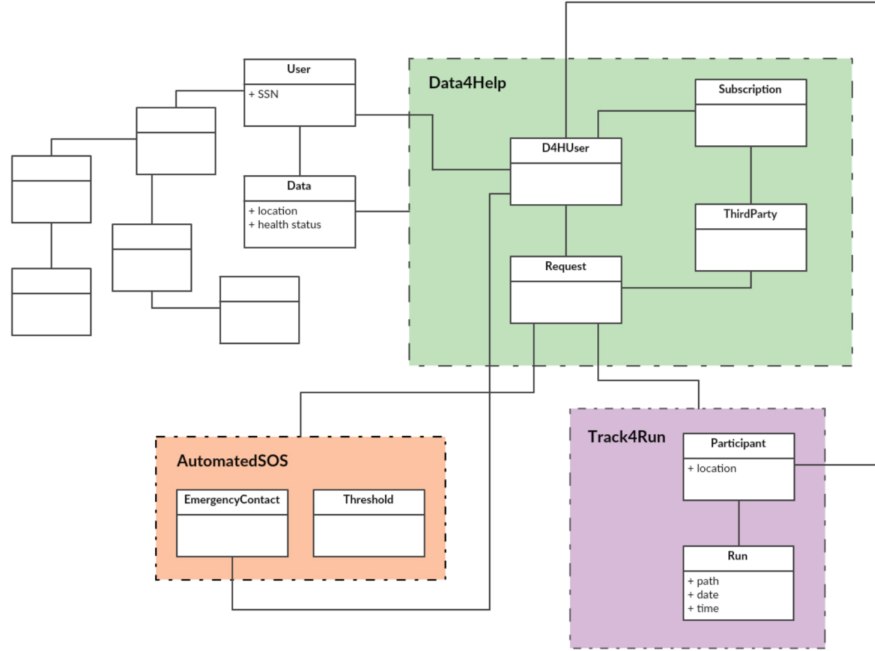


Figure 2.1: Domain Model

So as to understand the main events happening in the system, it is useful to provide a visual representation using state chart diagrams (described in the figure below). For **D4H** the main task is handling requests, and they can be in different states (unprocessed, pending, rejected/approved and completed). On the other hand, **ASOS** is a background process in charge of checking the health condition of a person, therefore it monitors the health status of the given individual until an anomaly happens, notifies it and after, loops back to the initial state. Finally, **T4R**, needs to tackle the "how to setup a run and make it available for the spectators" task, then this chain of events can also be modelled.

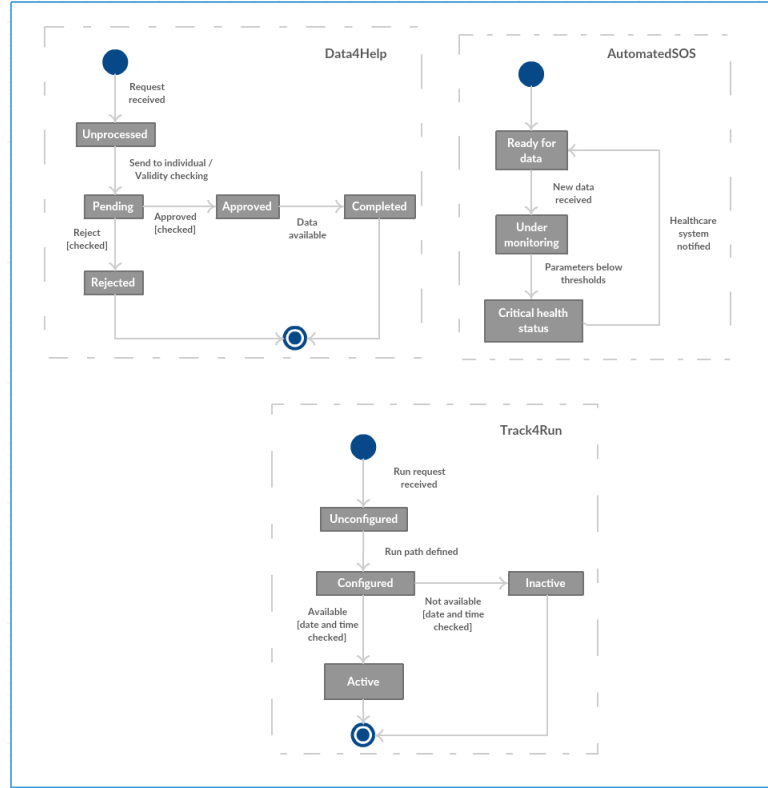


Figure 2.2: Statechart Diagrams

## 2.2 Product functions

The TrackMe environment is composed, as said before, by a set of 3 services, with D4H as the leading service. ASOS and T4R are going to be build on top of D4H, and will make use of all of its functionalities. Below, the main features of each service are listed, and a description is offered.

- **Data4Help**

D4H will be the leading service, and the features it will provide are mostly related to registered third party companies. Companies will be able to access different types of data from the individuals wearing the TrackMe devices. They will be able to subscribe to a specific individual data, or to a group of anonymized individuals' data, as long as certain restrictions are fulfilled. The individuals, on the other hand, will be able to accept or reject the request of accessing his/her data, and the third party companies will be notified of the individuals' decision.

- **AutomatedSOS**

ASOS is a complementary service offered to the senior range of users, and it will be built on top of the D4H service. All the elderly individuals of D4H will receive a request to subscribe to this service, whose main feature is to contact the individual's National Health Care Service every time any critical health parameter is under or above a defined threshold.

- **Track4Run**

T4R is the last service offered by TrackMe, and it will, also, be build on top of D4H. Designed as a service for run organizers and runners, who operate the TrackMe devices. The run organizers will be able to define a running circuit, and send invitations to the TrackMe device users; The individuals will be able to register to any particular competition they prefer. Furthermore, during the duration of each race, all spectators will be able to spot, through the T4R site, the location of each registered individual in the circuit.

## 2.3 User characteristics

The target users of the **TrackMe** system are:

- **Individuals:**

- who can **register** and **allows** TrackMe to store, analyse and process their data;
- can **manage** the requests if some third party users wants access to their data individually;
- users above the age of 60, can **avail** themselves of ASOS service by using the subscribe option available on their dashboard;
- they can **participate** into the upcoming runs;
- **Organizers and spectators** are also categorized as individuals. Organizers take the initiative of organising runs and defining the path, such that, other individuals are able to participate, whereas, spectators are the audience;

- **Third party users:**

- can **register** and make **requests** for the data required;
- can **subscribe** to the data after acquiring;



- **ASOS and T4R** are the third party companies **managed directly by** TrackMe;

Therefore, all the constraints derived from these characteristics must be satisfied from the **TrackMe** system, as much as possible.

## 2.4 Assumptions, dependencies and constraints

### 2.4.1 Domain assumptions

Domain assumptions help to make clear what it is expected from the external environment and allows constraining the software-driven machine so as to stay in the domain of interest.

- [D1] TrackMe guarantees that the wearables can provide sufficient accuracy and sensitivity when monitoring individuals.
- [D2] TrackMe addresses data protection and integrity against possible attacks.
- [D3] TrackMe devices are up and running during monitoring.
- [D4] The data collected is directly related to the individuals' by their SSN and is structured according to the data scheme required by D4H.
- [D5] The provided SSN by the individual is valid and trustable.
- [D6] Out of coverage scenarios cannot be handled by ASOS so as to guarantee a 5 seconds reaction.
- [D7] The organizers meet all the legal requirements and permissions necessary to set up a run.

# Specific requirements

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

## 3.1 External interface requirements

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

### 3.1.1 User interfaces

The following mock-ups represent a basic idea of what the web application will look like after the first release:

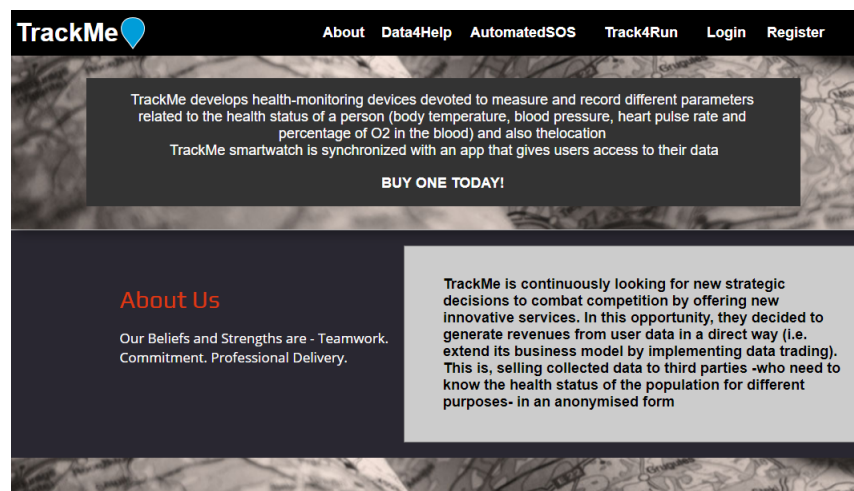


Figure 3.1: TrackMe's Home Page

As when new customers visit the home page of TrackMe, they can read about the work

that our company does, what services we offer, what benefits users can achieve by joining the community. In addition, they can buy the wearables and get more information, how to use them.

Next, comes the web page through which users can Login into the system (if already registered). And if not, they can register themselves through the Register web-page. There are separate register forms for the Individual users and the Third-party users visible below:

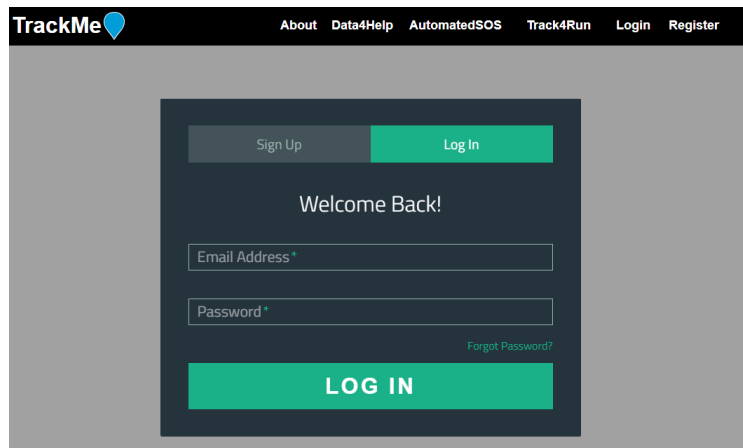
The screenshot shows the TrackMe website's login interface. At the top is a black navigation bar with the TrackMe logo and links for About, Data4Help, AutomatedSOS, Track4Run, Login, and Register. The main content area has a dark gray background. A central white box contains a 'Sign Up' button and a green 'Log In' button. Below these is the text 'Welcome Back!'. There are input fields for 'Email Address \*' and 'Password \*', with a 'Forgot Password?' link next to the password field. At the bottom of the white box is a large green 'LOG IN' button.

Figure 3.2: Login Page

Registration is free for all the users and it doesn't take much time to complete the forms, as we can see below:

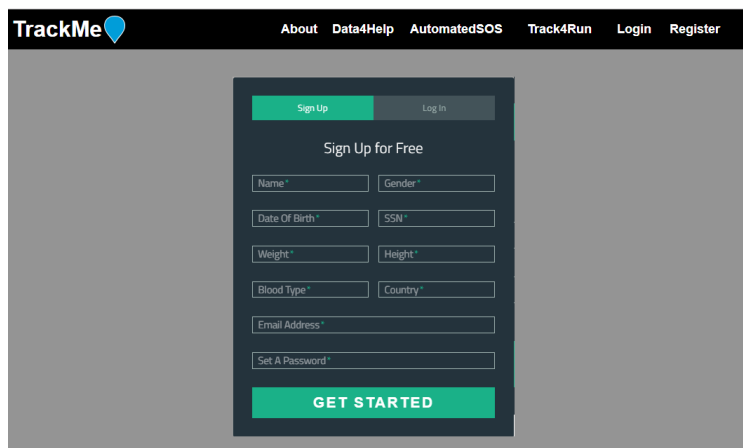
The screenshot shows the TrackMe website's individual user registration form. The navigation bar is the same as in the login page. The main content area has a dark gray background. A central white box is titled 'Sign Up for Free'. It contains several input fields: 'Name \*' and 'Gender \*' (side-by-side), 'Date Of Birth \*' and 'SSN \*' (side-by-side), 'Weight \*' and 'Height \*' (side-by-side), 'Blood Type \*' and 'Country \*' (side-by-side), 'Email Address \*', and 'Set A Password \*'. At the bottom of the white box is a large green 'GET STARTED' button.

Figure 3.3: Individual User's Registration page

Figure 3.4: Third Party's Registration page

In Figure below, its personal dashboard for the individual users, who can manage and view the requests and are able to view who all third parties have subscribed to their data.

#	Date	Time	Business Name
3326	10/21/2013	3:29 PM	AutomatedSOS
3325	10/21/2013	3:20 PM	Track4Run
3324	10/21/2013	3:03 PM	ABC

Figure 3.5: Registered user's Dashboard

In the next figure, we can see the customer has options to choose to accept or reject the request for data acquisition. As soon as an action is taken upon the request, it gets deleted from the page.

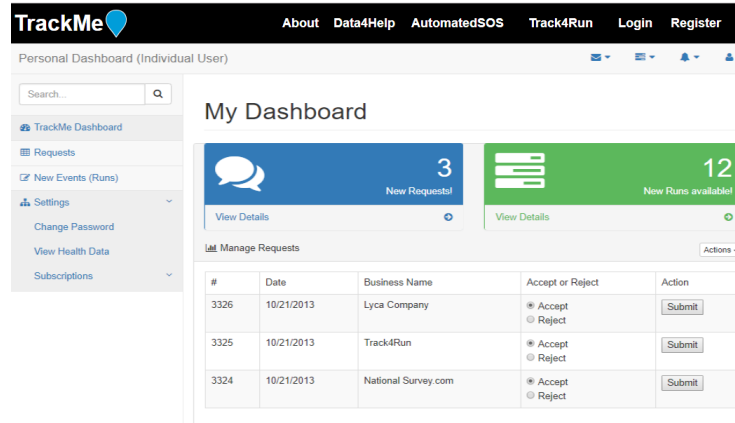


Figure 3.6: Registered user's Dashboard who can choose to Accept or Reject the requests

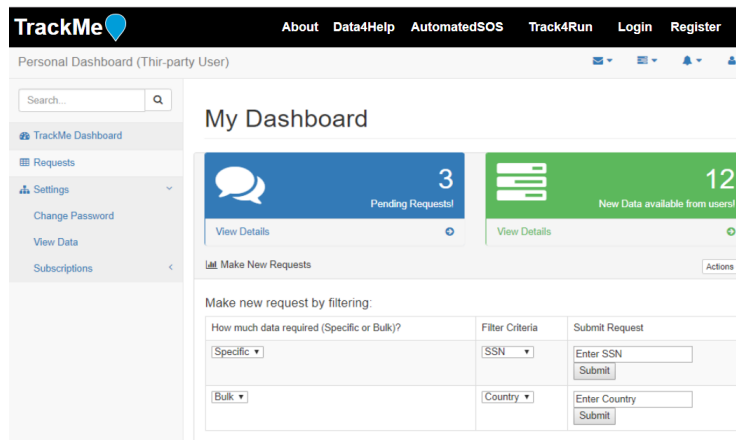


Figure 3.7: Third party user's Dashboard to make requests

In the figure above, Third party users can make request for the data using the filtering criteria provided by the system.

### 3.1.2 Hardware interfaces

No such direct hardware interfaces are required by our system, as this is a web portal application. The physical GPS is managed by the smart-watch wearables and managed through its application. And the hardware connection to the database server is managed by the underlying operating system on the web server. Thus, it can be easily accessible from any device or any location if **Internet service** is available.

### 3.1.3 Software interfaces

Web applications are by nature distributed applications. Specifically, web applications are accessed with a web browser and are popular because of the ease of using the browser as a user client.

Software Interfaces considered in TrackMe system are as follows:

- **Data4Help:** A service developed by TrackMe which collects all the data of the individuals and provides the data if necessary; It provides an interface to the third party users to make request;
- **AutomatedSOS:** A Third party company organised directly by TrackMe which provides support in case of emergency to the subscribed users;
- **Track2Run:** A Third party company organised directly by TrackMe which provides an interface to individuals to enrol into the run and allows the audience to view the position of the runners during the run;

### 3.1.4 Communication interfaces

The communication between different parts of the system is important since they depend on each other. However, in the way communication is achieved is not important for the system and is therefore handled by the underlying operating system for web portal.

## 3.2 Functional requirements

The functional requirements are those which are the fundamental actions of the system. As before, every requirement is divided by subsystem (D4H, ASOS and T4R), and the relation with each goal is shown.

[G1] The individual could allow (or refuse) Data4Help to use their data.

[D2] TrackMe addresses data protection and integrity against possible attacks.

[D4] The data collected is directly related to the individuals' by their SSN and is structured according to the data scheme required by D4H.

[D5] The provided SSN by the individual is valid and trustable.

[R1] The system must allow an individual to register a new account.

- [R2] The system must allow an individual to access to its account.
- [R3] The system must allow an individual to accept or reject its requests of accessing personal data.
- [R4] The system must be able to communicate with TrackMe database in order to obtain the health status and location of an individual.
- [G2] The third-party company should be able to access data of a specific individual.
  - [D2] TrackMe addresses data protection and integrity against possible attacks.
  - [D4] The data collected is directly related to the individuals' by their SSN and is structured according to the data scheme required by D4H.
  - [D5] The provided SSN by the individual is valid and trustable.
  - [R5] The system must allow a third party company to register a new account.
  - [R6] The system must allow a third party company to access to its account.
  - [R7] The system must allow a third party company to search for an individual health status and location using his/her SSN.
  - [R8] The system must be able to notify the individual that a third party company wants to access its data.
- [G3] The third-party company should be able to access anonymized data of groups of individuals under certain constraints.
  - [D2] TrackMe addresses data protection and integrity against possible attacks.
  - [D3] TrackMe devices are up and running during monitoring.
  - [D4] The data collected is directly related to the individuals' by their SSN and is structured according to the data scheme required by D4H.
  - [R5] The system must allow a third party company to register a new account.
  - [R6] The system must allow a third party company to access to its account.
  - [R9] The system must allow a third party company to filter data of an anonymized group of individuals by country, age, gender and blood type parameters.
  - [R10] The system must be able to anonymize health status and location of a group of individuals.

- [G4] The third-party company should be able to access anonymized data of groups of individuals under certain constraints.
- [D2] TrackMe addresses data protection and integrity against possible attacks.
- [D3] TrackMe devices are up and running during monitoring.
- [D4] The data collected is directly related to the individuals' by their SSN and is structured according to the data scheme required by D4H.
- [D5] The provided SSN by the individual is valid and trustable.
- [R5] The system must allow a third party company to register a new account.
- [R6] The system must allow a third party company to access to its account.
- [R11] The system must allow a third party company to subscribe to an individual health status and location.
- [R12] The system must allow a third party company to subscribe to data of an anonymized group of individuals
- [G5] Provide a service capable to send an advice to the health care service when a individual's parameters are below a defined threshold.
- [D1] TrackMe guarantees that the wearables can provide sufficient accuracy and sensitivity when monitoring individuals.
- [D3] TrackMe devices are up and running during monitoring.
- [D4] The data collected is directly related to the individuals' by their SSN and is structured according to the data scheme required by D4H.
- [D5] The provided SSN by the individual is valid and trustable.
- [D6] Out of coverage scenarios cannot be handled by ASOS so as to guarantee a 5 seconds reaction.
- [R13] The system must be able to send a request for accessing an individual's data when he/she is older than 60 years old.
- [R14] The system must be able to monitor, and compare against defined thresholds, the health status of an individual.
- [R15] The system must be able to contact the Healthcare service associated to an individual.



- [G6] Run organizers could define the path for a given run and TrackMe users can enrol to it.
- [D1] TrackMe guarantees that the wearables can provide sufficient accuracy and sensitivity when monitoring individuals.
- [D2] TrackMe addresses data protection and integrity against possible attacks.
- [D3] TrackMe devices are up and running during monitoring.
- [D4] The data collected is directly related to the individuals' by their SSN and is structured according to the data scheme required by D4H.
- [D7] The organizers meet all the legal requirements and permissions necessary to set up a run.
- [R16] The system must allow a participant to register a new account.
- [R17] The system must allow a participant to access to its account.
- [R18] The system must allow an organizer to register a new account.
- [R19] The system must allow an organizer to access to its account.
- [R20] The system must allow an organizer to create a race event.
- [R21] The system must allow an organizer to define the running circuit of a race event.
- [R22] The system must allow an organizer to send invitations to the participants to enrol in a race event.
- [R23] The system must allow a participant to accept or reject an invitation to a race.
- [G7] Run spectators could track the position of all the runners during the race.
- [D1] TrackMe guarantees that the wearables can provide sufficient accuracy and sensitivity when monitoring individuals.
- [D2] TrackMe addresses data protection and integrity against possible attacks.
- [D3] TrackMe devices are up and running during monitoring.
- [D4] The data collected is directly related to the individuals' by their SSN and is structured according to the data scheme required by D4H.
- [R16] The system must allow a participant to register a new account.
- [R17] The system must allow a participant to access to its account.
- [R18] The system must allow an organizer to register a new account.

- [R19] The system must allow an organizer to access to its account.
- [R20] The system must allow an organizer to create a race event.
- [R23] The system must allow a participant to accept or reject an invitation to a race.
- [R24] The system must allow any spectator of a run to view in a map the participants' location
- [R25] The system must allow a spectator to click on a participant location in order to view his/her health status.

### 3.2.1 Use case diagrams

In the current section the use cases for all the subsystems are shown.

- **Data4Help**

In the Figure 3.8 the Data4Help use cases are shown. The most important use cases are *Manage requests*, *Access individual data*, *Access bulk data*, and *Send a request*. As shown in the figure, the actors related to this system are the *Individual*, who is the user of the TrackMe wearable device, and the *Third party company*, which is the actor who will access the individual's data.

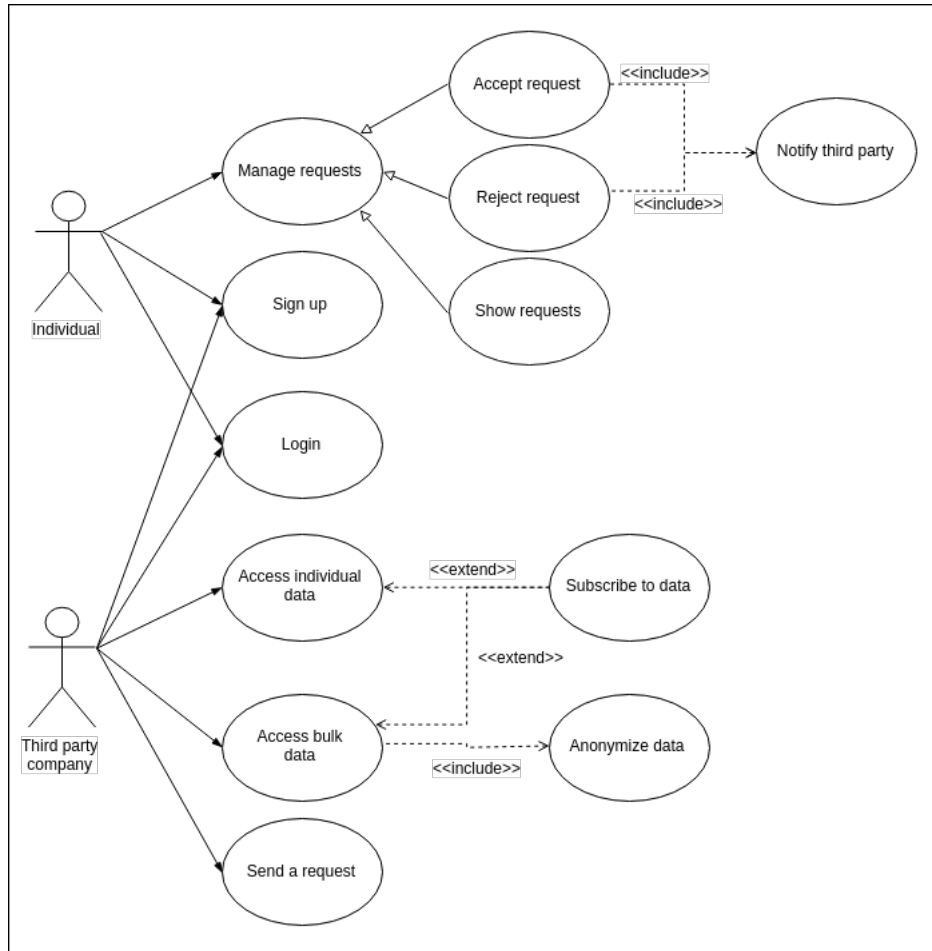


Figure 3.8: Data4Help Use Case Diagram

### • AutomatedSOS

In the Figure 3.9 AutomatedSOS use cases are shown. In the case of ASOS, the system has two passive actors, one of them is the Healthcare service, that is affected by the use case *Contact Health care service*, and D4H, which is affected by the use case *Send request for subscription*. On the other hand, the use case *Receive individual's data* is affected by D4H, since is that service which sends the individual's information to ASOS.

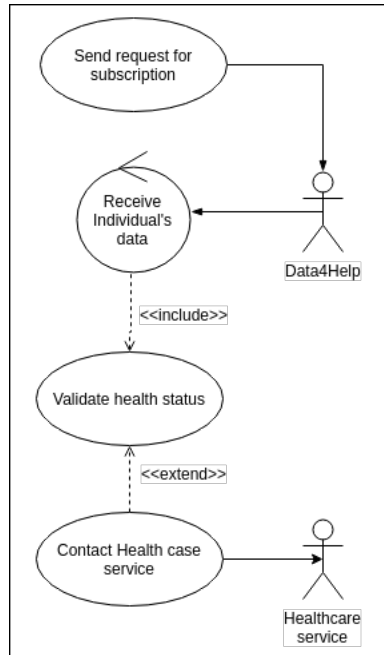


Figure 3.9: AutomatedSOS Use Case Diagram

- **Track4Run**

In the Figure 3.10 the Track4Run use cases are shown. The most important use cases are *Manage invitations*, and *Create a run*. In this system, the actors are the *Participant*, who is the TrackMe wearable device user, the *Organizer*, who will setup the run, and the *Spectator*, who are all the non-users that may track and watch the location of the participants during a given run.

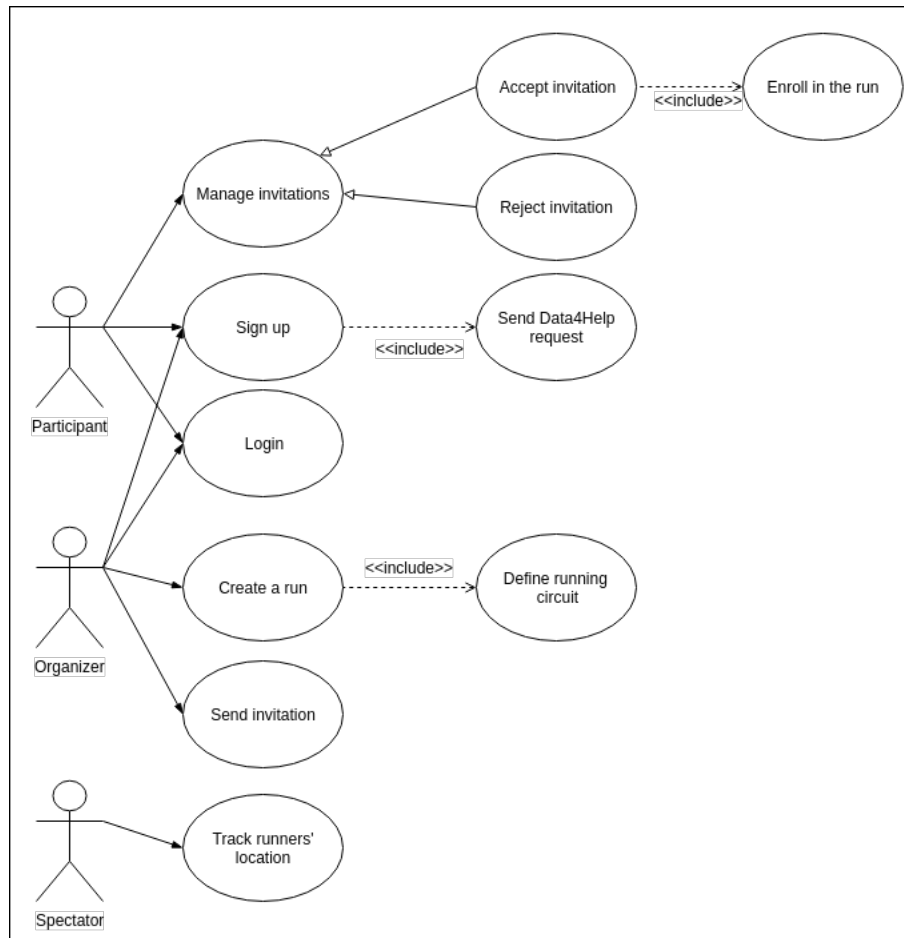


Figure 3.10: Track4Run Use Case Diagram

### 3.2.2 Use cases description

In the following section a description of each use case is provided. For every use case, an ID is defined, as well as the entry conditions, steps to accomplish the exit condition and any exception that may occur. As in previous sections, all the use cases are described based on the subsystem they belong to.

- **Data4Help**

**ID:** [UC1]

**Name:** Manage requests

**Actor:** Individual

**Entry conditions:**

1. A logged in individual

**Event flow:**

1. The individual clicks on Requests button
2. The system shows the page with a list of all pending request for the individual (**Show requests** [UC2] )
3. The individual selects the desired action (accept or reject) and clicks the Submit button
4. The system **Accepts the request** ( [UC3] ) or **Rejects the request** ( [UC4] )

**Exit conditions:**

- The request is set as accepted or rejected

**Exceptions: -**

---

**ID:** [UC2]**Name:** Show requests**Actor:** Individual**Entry conditions:**

1. A logged in individual
2. A request to access the individual's health status and location

**Event flow:**

1. The system gets all the pending requests
2. The system shows all the pending requests to the individual

**Exit conditions: -****Exceptions:**

1. If there are no pending requests, the system should show a message to the user

---

**ID:** [UC3]**Name:** Accept the request**Actor:** Individual**Entry conditions:**

1. A logged in individual
2. A request to access the individual's health status and location
3. The individual clicked on the submit button to accept a request

**Event flow:**

1. The system sets the pending request as Accepted
2. The system **Notifies the third party** ( [UC5] )

**Exit conditions:**

1. The request is set to Accepted

**Exceptions: -**

---

**ID:** [UC4]**Name:** Reject the request**Actor:** Individual**Entry conditions:**

1. A logged in individual
2. A request to access the individual's health status and location
3. The individual clicked on the submit button to reject a request

**Event flow:**

1. The system sets the pending request as Rejected
2. The system **Notifies the third party** ( [UC5] )

**Exit conditions:**

1. The request is set to Rejected

**Exceptions: -**

**Scenario:** A TrackMe device user, who is already registered in D4H, has receive a notification in his cellphone saying that a company he never heard of wants to access his health data and location. Since it is an unknown company, he logs in into D4H website, goes to the Manage Requests sections, and rejects the suspicious request.

---

**ID:** [UC5]

**Name:** Notify third party

**Actor:** Individual

**Entry conditions:**

1. A logged in individual
2. A request is set to Accepted or Rejected

**Event flow:**

1. The system adds a notification of the individual's decision to the third party company

**Exit conditions:**

1. The system shows a new notification in the Third party company dashboard

**Exceptions:** -

---

**ID:** [UC6]

**Name:** Sign up

**Actor:** Individual, Third party company

**Entry conditions:** -

**Event flow:**

1. The Individual or the Third party company enters to the sign up page of the website
2. The Individual or the Third party company completes all the mandatory fields
3. The Individual or the Third party company clicks on the "Get started" button

**Exit conditions:**

1. The system creates an account for an Individual or a Third party company

**Exceptions:**

1. If either the Individual or the Third party company are already registered, an error message will be shown



2. If there some of the fields in the registration form that are not filled in, an error message will be shown

---

**ID:** [UC7]

**Name:** Login

**Actor:** Individual, Third party company

**Entry conditions:** -

**Event flow:**

1. The Individual or the Third party company enters to the login page of the website
2. The Individual or the Third party company completes the email address and password fields
3. The Individual or the Third party company clicks on the "Log in" button

**Exit conditions:**

1. The system redirects the Individual or a Third party company to the profile page

**Exceptions:**

1. If the credentials are wrong, an error message will be shown
2. If the password or the email are missing, an error message will be shown

---

**ID:** [UC8]

**Name:** Send a request

**Actor:** Third party company

**Entry conditions:**

1. The Third party company is logged in

**Event flow:**

1. The Third party company clicks on the Requests button
2. The system shows a form with a filter criteria, which contains the SSN field
3. The Third party company fills in the SSN field and clicks submit

**Exit conditions:**

1. The system sends a notification to the Individual

**Exceptions:**

1. If the SSN is not valid, an error message will be shown
2. If the SSN does not corresponds to an active Individual, an error message will be shown

**Scenario:** The company HelpMe needs updates on the health status of one of its clients in order to offer a particular service. To be able to access the data of the client, makes a request for accessing the health data and location of the user, using his SSN. To do so, an employee logs in into D4H, and sends a request to the client using the D4H website.

---

**ID:** [UC9]

**Name:** Access individual data

**Actor:** Third party company

**Entry conditions:**

1. The Third party company is logged in
2. The Third party has sent a request to access an specific Individual health status and location

**Event flow:**

1. The Third party company clicks on the View data button
2. The system shows a form with a filter criteria, which contains the SSN field
3. The Third party company fills in the SSN field and clicks submit
4. The system gets the health status and location of the given SSN
5. The system shows the health status and location of the Individual
6. The third party company may click on Subscribe, which extends to the use case [UC10]

**Exit conditions:**

1. The system shows the health status and location of the given Individual

**Exceptions:**

1. If the SSN is not valid, an error message will be shown
2. If the SSN does not corresponds to an active Individual, an error message will be shown
3. If the SSN does not corresponds to an Individual who accepted the previously sent request, an error message will be shown

---

**ID:** [UC10]**Name:** Subscribe to data**Actor:** Third party company**Entry conditions:**

1. The Third party company is logged in
2. The Third party has made a search
3. The Third party company clicks on the Subscribe button

**Event flow:**

1. The system creates a subscription between the Third party company and saved query, that can be an specific SSN or a more general query

**Exit conditions:**

1. The system starts to notify the Third party company of new incoming data of the Individual or the anonymized group of individuals

**Exceptions:** -

**Scenario:** A company needs to subscribe to the health data of one of its users. To do so, an employee logs in into D4H website, looks for the users health data and location using its SSN, and subscribes to the query. Once the employee does that, the company will receive any update of its user's health status and location.

---

**ID:** [UC11]**Name:** Access bulk data**Actor:** Third party company**Entry conditions:**

1. The Third party company is logged in

**Event flow:**

1. The Third party company clicks on the View data button
2. The system shows a form with different filter criteria, which contains country, age, gender and blood type fields
3. The Third party company selects the different filters
4. The system **Anonymize the data**( [UC12] )
5. The system shows the health status and location of the anonymized group of individuals
6. The third party company may click on Subscribe, which extends to the use case [UC10]

**Exit conditions:**

1. The system shows the health status of the anonymized group of individuals

**Exceptions:**

1. If the data cannot be anonymized, an error message will be shown
2. If there is not enough data, an error message will be shown

---

**ID:** [UC12]**Name:** Anonymize data**Actor:** Third party company**Entry conditions:**

1. The Third party company is logged in
2. The Third party company has made a search for a bulk of data

**Event flow:**

1. The system gets the data based on the filters the Third party company has provided
2. The system anonymize the data, by removing the location of each user

**Exit conditions:**

1. The system returns the anonymized data

**Exceptions:**

1. If there are less than 1000 individuals in the request of data, an error is returned
2. If there is not enough data, an error is returned

- **AutomatedSOS ID:** [UC13]

**Name:** Send request for subscription

**Actor:** D4H

**Entry conditions:**

- The system knows the SSN of the elderly individual to be subscribed

**Event flow:**

1. The system sends a request to Data4Help in order to subscribe to the health status and location of the given individual
2. Data4Help sends an OK response to advice the system that the request was sent

**Exit conditions:**

1. Data4Help placed a request for subscription to the individual

**Exceptions:**

---

**ID:** [UC14]

**Name:** Receive individual's data

**Actor:** D4H

**Entry conditions:**

- The individual is subscribed to ASOS

**Event flow:**

1. Every time the health status and location of the subscribed individual, D4H sends the data to ASOS
2. The system **Validate the health status** ( [UC(] 15)) of the individual

**Exit conditions:**

**Exceptions:**

---

**ID:** [UC15]

**Name:** Validate health status

**Actor:** -

**Entry conditions:**

- The individual is subscribed to ASOS
- ASOS has received new health data of the individual
- The system has several thresholds configured

**Event flow:**

1. The system validates the health condition of the individual using the configured thresholds
2. The system **Contacts the healthcare service** ( [UC() 16]) if needed

**Exit conditions:**

**Exceptions:**

---

**ID:** [UC16]

**Name:** Contact healthcare service

**Actor:** Healthcare service

**Entry conditions:**

- Some parameters of the health data of the individual are above or below the threshold

**Event flow:**

1. The system gets the healthcare service contact of the given individual
2. The system contacts the healthcare service, and sends the health data and location of the given individual

**Exit conditions:**

- The system has contacted the healthcare service
-

**Exceptions:** -

**Scenario:** An individual subscribed to ASOS service is having a hart attack. ASOS has validated the individual's health status and compared the parameters with the defined thresholds, and has decided to call the Health care service. To do so, it contacts the Health care service associated to the individual, using a previously defined protocol and API, and sends to them all the data of the individual.

---

- **Track4Run**

**ID:** [UC17]

**Name:** Sign up

**Actor:** Participant, Organizer

**Entry conditions:** -

**Event flow:**

1. The Participant or the Organizer enters to the Track4Run sign up page
2. The Participant or the Organizer fill in all the mandatory fields in the registration form
3. The Participant or the Organizer click on Register button
4. The system **Sends a Data4Help request** ( [UC19] )

**Exit conditions:**

1. The system creates an account for a Participant or an Organizer

**Exceptions:**

1. If either the Participant or the Organizer are already registered, an error message will be shown
2. If there some of the fields in the registration form that are not filled in, an error message will be shown

---

**ID:** [UC18]

**Name:** Login

**Actor:** Participant, Organizer

**Entry conditions:** -

**Event flow:**

---

1. The Participant or the Organizer enters to the login page of the website
2. The Participant or the Organizer completes the email address and password fields
3. The Participant or the Organizer clicks on the "Log in" button

**Exit conditions:**

1. The system redirects the Participant or the Organizer to the profile page

**Exceptions:**

1. If the credentials are wrong, an error message will be shown
2. If the password or the email are missing, an error message will be shown

---

**ID:** [UC19]**Name:** Send Data4Help request**Actor:** Participant**Entry conditions:**

1. The Participant has already registered

**Event flow:**

1. The system sends a notification to Data4Help in order to request health status and location of the user
2. Data4Help sends an OK response to advice the system that the request was sent

**Exit conditions:**

1. The Participant received a notification from Track4Run

**Exceptions:** -

---

**ID:** [UC20]**Name:** Manage invitations**Actor:** Participant**Entry conditions:**

1. The Participant is logged in



**Event flow:**

1. The Participant clicks on Invitations button
2. The system shows all the run invitations to the user
3. The Participant clicks either on **Accept invitation** ( [UC21] ) or **Reject invitation** ( [UC22] )

**Exit conditions:**

1. The system shows the user the pending invitations

**Exceptions: -**

---

**ID:** [UC21]**Name:** Accept invitation**Actor:** Participant**Entry conditions:**

1. The Participant is logged in
2. The Participant has clicked on Accept invitation

**Event flow:**

1. The system sets the invitation as Accepted
2. The system **Enrols the participant into the run** ( [UC23] )
3. The system shows all the run invitations to the user

**Exit conditions:**

1. The Participant is enrolled into the run

**Exceptions: -**

---

**ID:** [UC22]**Name:** Reject invitation**Actor:** Participant**Entry conditions:**

1. The Participant is logged in

2. The Participant has clicked on Reject invitation

**Event flow:**

1. The system sets the invitation as Rejected
2. The system shows all the run invitations to the user

**Exit conditions:** -**Exceptions:** -

---

**ID:** [UC23]**Name:** Enrol in the run**Actor:** Participant**Entry conditions:**

1. The Participant is logged in
2. The Participant has clicked on Accept invitation

**Event flow:**

1. The system adds the Participant into the participants list of the run
2. The system shows the Participant a confirmation message

**Exit conditions:**

1. The Participant is enrolled into the run

**Exceptions:** -

---

**ID:** [UC24]**Name:** Create a run**Actor:** Organizer**Entry conditions:**

1. The Organizer is logged in

**Event flow:**

1. The Organizer clicks on the "Create a run" button

2. The system shows a form with the following fields: Name, Start day and End time
3. The Organizer fills in all the fields and clicks the Create button
4. The system shows the **Define running circuit** ( [UC25] ) web page

**Exit conditions:**

1. The system creates the run

**Exceptions:**

1. If the Start date field is less than "Today", an error will be shown

---

**ID:** [UC25]**Name:** Define running circuit**Actor:** Organizer**Entry conditions:**

1. The Organizer is logged in
2. The Organizer has created a run

**Event flow:**

1. The system shows the Running circuit web page
2. The Organizer defines the running circuit by clicking on the map
3. Once the Organizer has finished defining the running circuit, clicks on Submit button

**Exit conditions:**

1. The system creates the running circuit and associate it with the run event

**Exceptions:** -

---

**ID:** [UC26]**Name:** Send invitation**Actor:** Organizer**Entry conditions:**

1. The Organizer is logged in

**Event flow:**

1. The Organizer clicks on "Send invitations" button
2. The system shows a list with all pending running events
3. The Organizer clicks on one running event
4. The system shows a list of all not-enrolled, Track4Run Participants
5. The Organizer selects the Participants he/she wants to invite, and clicks the Invite button

**Exit conditions:**

1. The system sends invitations to all the selected Participants

**Exceptions: -**

---

**ID:** [UC27]

**Name:** Track runners' location

**Actor:** Spectator

**Entry conditions:**

1. The Spectator is in the Track4Run site

**Event flow:**

1. The Spectator clicks on "Current events" button
2. The system shows a list of all running events which has started but has not finished
3. The Spectator selects one of the running events
4. The system shows a map with all the Participants' location

**Exit conditions: -****Exceptions: -**

**Scenario:** A marathon is taking place in the city, and a runner's family want to cheer he during the race. Since they are late and the event has already begun, they enter to the T4R website and go to the Current events web page, select the event and look for their family member.

### 3.2.3 Activity diagrams

### 3.2.4 Sequence diagrams

In the following section three sequence diagrams are described. All of them, are an important part of the system and represent the following processes: *accept a request* for accessing the individual's health status and location, *search and subscribe to a bulk of data*, and *accept an invitation* to enrol in a race.

To begin with, the *accept a request* for accessing the individual's health status and location is shown in the Figure 3.11. It is possible to observe the Manage Requests use case ( [UC1] ) flow, where the Individual asks D4H to get his/her pending requests, and the interaction between D4H and the Request collection, which returns the pending requests to the Individual. Also, once the Individual has accepted a request, it changes its status to *Accepted* and the Third party company is notified.

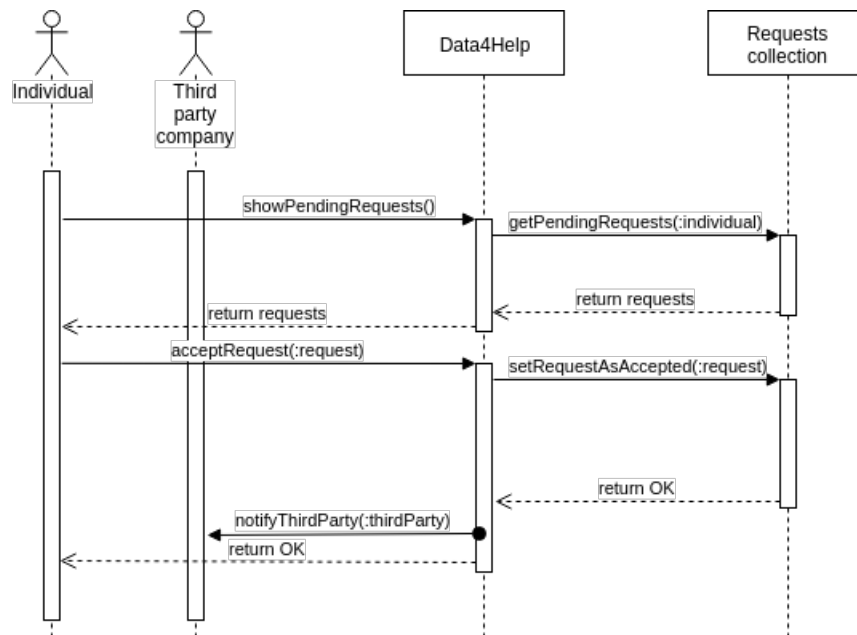


Figure 3.11: Data4Help - Accept Request Sequence Diagram

In the Figure 3.12 can be observed the sequence diagram of Access bulk data use case ( [UC11] ) and the optional Subscribe to data use case ( [UC10] ). Also in this case, it is possible to notice the action taken by the actor (the Third party company) to get data from D4H. The *search* action makes D4H to look for the data in the Data collection, anonymize

it, and return it to the actor. Optionally, the Third party company can save and subscribe to the previously used query in order to get updates.

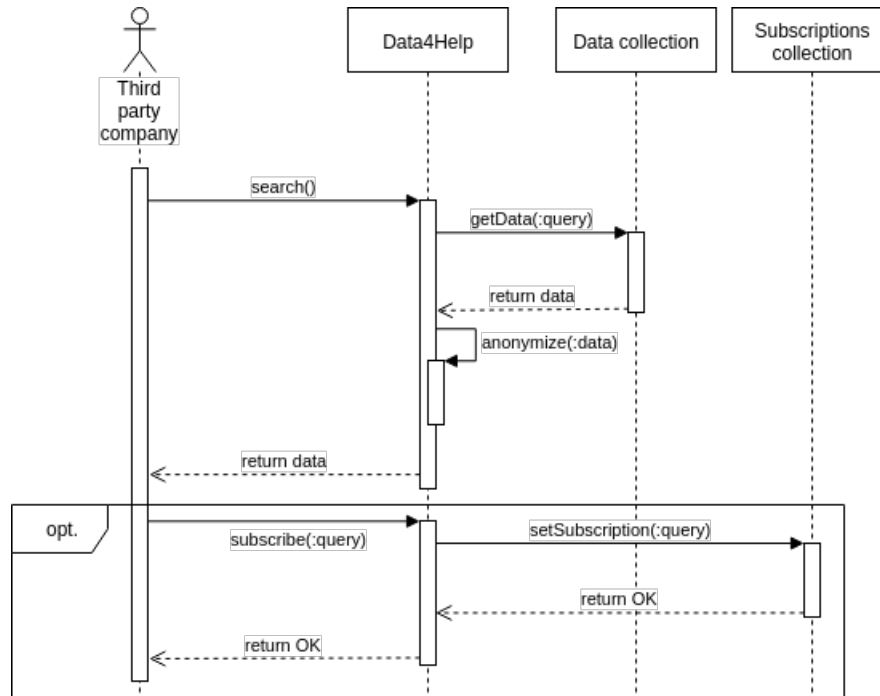


Figure 3.12: Data4Help - Access to bulk data and subscribe Sequence Diagram

Finally, the Accept invitation to a run use case ( [UC21] ) is modelled in the Figure 3.13. In this case, the Participant asks T4R to show his/her pending invitations, which are collected from the Invitations collection. Once the Participant has all the pending invitations, he/she can select one and accept the invitation, action that will enrol the Participant to the given run, and set the invitation as *Accepted*.

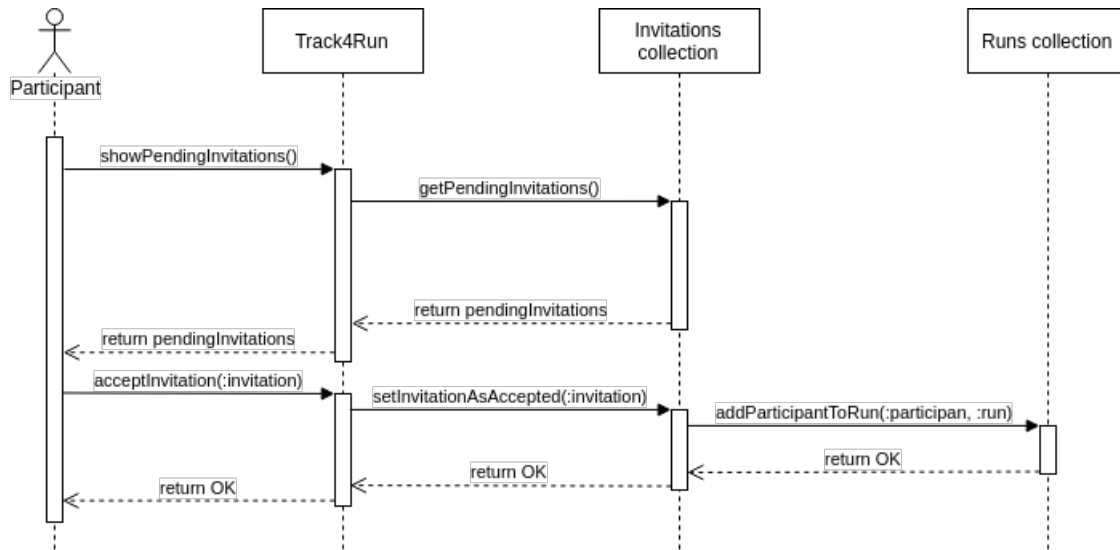


Figure 3.13: Track4Run - Accept invitation Sequence Diagram

### 3.2.5 Requirements traceability matrix

Goal Id	Requirement Id	Use case Id	Comments
[G1]	[R1]	[UC6]	
	[R2]	[UC7]	
	[R3]	[UC1]	
		[UC2]	
		[UC3]	
		[UC4]	
		[UC5]	
	[R4]	[UC9]	
[G2]	[R5]	[UC6]	
	[R6]	[UC7]	
	[R7]	[UC9]	
	[R8]	[UC8]	
[G3]	[R5]	[UC6]	
	[R6]	[UC7]	
	[R9]	[UC11]	
		[UC12]	
	[R10]	[UC12]	

[G4]	[R5]	[UC6]	
	[R6]	[UC7]	
	[R11]	[UC9]	
		[UC10]	
	[R12]	[UC10]	
		[UC11]	
		[UC12]	
[G5]	[R13]	[UC13]	
	[R14]	[UC14]	
		[UC15]	
	[R15]	[UC16]	
[G6]	[R16]	[UC17]	
		[UC19]	
	[R17]	[UC18]	
	[R18]	[UC17]	
	[R19]	[UC18]	
	[R20]	[UC24]	
	[R21]	[UC25]	
	[R22]	[UC26]	
	[R23]	[UC20]	
		[UC21]	
		[UC22]	
		[UC23]	
[G7]	[R16]	[UC17]	
		[UC19]	
	[R17]	[UC18]	
	[R18]	[UC17]	
	[R19]	[UC18]	
	[R20]	[UC24]	
	[R23]	[UC20]	
		[UC21]	
		[UC22]	
		[UC23]	
	[R24]	[UC27]	



---

	[R25]	[UC27]	
--	-------	--------	--

---

Table 3.1: Traceability matrix

### 3.3 Performance requirements

The system is provided to serve fairly great number of users simultaneously. It guarantees high level performance and maintenance operation costs very low. Technical support for installation is not required, as it is only a web application and users can access it from anywhere where the internet service is provided. Additionally, under ASOS service, we make sure the system responds and notifies the health control system within 5 seconds, when the thresholds are below the certain defined level. Users will rely on the application in order to organize their send or accept/reject requests for data and notifications, so we have to guarantee quick, reactive and correct response.

### 3.4 Design constraints

#### 3.4.1 Standards compliance

- The application is responsive that makes web pages render well on a variety of devices and window or screen sizes. Content, design and performance are necessary across all devices to ensure usability and satisfaction.
- The application complies with the web standards of the World Wide Web Consortium (W3C).
- As the Web provides many mechanisms to interlink data across systems, our system keep user's personal information private and fragmented.
- The system correctly preserves and restores user or application state within the session. The application preserves user or state when leaving the web browser and prevents accidental data loss due to back-navigation and other state changes. When returning to the web page, the application must restore the preserved state and any significant stateful transaction that was pending.
- All private data is stored in the MongoDB database.

### **3.4.2 Hardware limitations**

This is a web based application with no strict hardware limitations. The only requirement is a TrackMe smart-watch and its app that posts the data when online. Users can access the application from anywhere, from any system using the internet service.

### **3.4.3 Any other constraint**

- **Regulatory Policies**

The application will have to ask for user's position and personal data in order to have track of the health status. Social security numbers(SSN)/Fiscal code and email addresses won't be used for commercial uses.

Moreover, the system makes sure the third-party companies who registers are legally established. Thus, there are no fake companies who tries to exploit user's data.

## **3.5 Software system attributes**

### **3.5.1 Reliability**

### **3.5.2 Availability**

### **3.5.3 Security**

### **3.5.4 Maintainability**

### **3.5.5 Portability**

# Formal analysis using Alloy

---

# Effort spent

Team Work	
Task	Hours
Understanding the problem	3
Brainstorming	2
World and shared phenomena	2
Definitions, acronyms, abbreviations	X
Software system attributes	X
Alloy	X
Checking document	X
<b>Total</b>	X

Table 5.1: Time spent by all team members

Individual Work					
Diego Avila		Laura Schiatti		Sukhpreet Kaur	
Task	Hours	Task	Hours	Task	Hours
Goals	X	Context	X	Layout	X
Product functions	X	Purpose	X	Problem description	X
F requirements	X	Domain model	X	User characteristics	X
UC diagrams	X	Statechart diagrams	X	User interface	X
UC description	X	Assumptions	X	Design constraints	X
Scenarios	X	NF requirements	X	External Interface	X
	X	Effort spent	X		X
<b>Total</b>	X	<b>Total</b>	X	<b>Total</b>	X

Table 5.2: Time spent by each team member

---

# References

---

- Requirement Analysis and Specification Document: AA 2017-2018.pdf”. Version 1.0 - 26.10.2017
- Henriksen, A., Haugen Mikalsen, M., Woldaregay, A. Z., Muzny, M., Hartvigsen, G., Hopstock, L. A., Grimsgaard, S. (2018)  
Using Fitness Trackers and Smartwatches to Measure Physical Activity in Research: Analysis of Consumer Wrist-Worn Wearables. Journal of medical Internet research, 20(3), e110. doi:10.2196/jmir.9157.  
Retrieved from: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5887043/>
- IEEE. (1993). IEEE Recommended Practice for Software Requirements Specifications (IEEE 830-1993).  
Retrieved from <https://standards.ieee.org/standard/830-1993.html>
- Sloane, A. M. (2009). Software Abstractions: Logic, Language, and Analysis by Jackson Daniel, The MIT Press, 2006, 366pp, ISBN 978-0262101141.