POLITECNICO
MILANO 1863

TrackMe

Software Engineering II - Prof. Elisabetta Di Nitto

Requirements Analysis and Specification Document

Michele Gatti, Federica Gianotti, Mathyas Giudici

Document version: 1.0
November 5, 2018

Deliverable:	RASD
Title:	Requirement Analysis and Specification Document
Authors:	Michele Gatti, Federica Gianotti, Mathyas Giudici
Version:	1.0
Date:	November 5, 2018
Download page:	https://github.com/MathyasGiudici/GattiGianottiGiudici
Copyright:	

Contents

Section 1

Introduction

1.1 Purpose

The goal of the Requirement Analysis and Specification Document (RASD) is to give a clear description of the system that is going to be developed, its functional and non-functional requirements, its constraints and its domain. Moreover, it provides information about the relationship between the system taken into account and the external world by providing use cases and scenarios. Finally it gives a more formal specification of the most relevant features of the system to be using the Alloy language. Generally this type of document is mainly addressed to developers, programmers, testers, project managers and system analysists, but it can be useful also for final users. Track Me is a company that wants to develop three different but connected software-based services:

- **Data4Help**: a service that allows third parties to monitor the location and health status of individuals. Through this service third parties can request the access both to the data of some specific individuals, who can accept or refuse sharing their information , and to anonymized data of group of individuals, which will be given only if the number of the members of the group is higher than 1000, according to privacy rules.

- **AutomatedSOS**: a service addressed to elderly people which monitors the health status of the subscribed customers and, when such parameters are below a certain threshold (personalized for every user using the data from Data4Help), sends to the location of the customer an ambulance, guaranteeing a reaction time less than 5 second from the time the parameters are below the threshold.

- **Track4Run**: a service to track athletes participating in a run. It allows organizers to define the path for a run, participants to enroll to a run and spectators to see on a map the position of all runners during the run. This service will exploit the features offered by Data4Help.

1.1.1 Goals

The three applications of the system have in common the following goals:

- [**G.1**]: Allow unregistered user to sign in to access to the application;
- [**G.2**]: Allow registered user to log in and access to the application;
- [**G.3**]: Allow registered user to manage his/her profile;

The description given above can be summarized as a list of goals, specific for each service.

Data4Help:

- [**G.4**]: Allow registered third parties to request data of a single individual;
- [**G.5**]: Allow registered third parties to request data of a group of people;

AutomatedSOS:

- [**G.6**]: Allow data acquisition through smart watches (or similar);
- [**G.7**]: Allow monitoring the health status of an individual registered user;
- [**G.8**]: Allow sending location of an individual registered user to an ambulance if his/her parameters are below a certain threshold;

Track4Run:

- [**G.9**]: Allow registered user to become organizers or athletes of a run;
- [**G.10**]: Allow organizers to define the date and the path for a new run;
- [**G.11**]: Allow organizers to delete a run;
- [**G.12**]: Allow registered athletes to enrol to a run;

- **[G.13]**: Allow registered athletes to delete an enrolment of a run;
- **[G.14]**: Allow unregistered user to access as spectator;
- **[G.15]**: Allow registered/unregistered user to see on a map the position of all runners during a run;

1.2 Scope

According to *The World and the Machine* [4] we can divide every system into two parts:

- The **machine**, which is the portion of system to be developed;
- The **world**, which is the portion of the real-world affected by the machine.

As a consequence we can classify phenomena in three different types:

- **World phenomena**: phenomena that the machine cannot observe;
- **Machine phenomena**: phenomena located entirely in the machine;
- **Shared phenomena**: phenomena that can be controlled by the world and observed by the machine or controlled by the machine and observed by the world;

Below we give an analysis of world and shared phenomena:

World phenomena

- A user turns on data connection;
- A user wears his smartwatch during a day;
- The batteries of the smartwatch of a user run out;
- A user turns on the GPS;
- An enrolled runner for a run takes part in it;
- A runner wears his smartwach during a run.

Shared phenomena

- New user registeres to Data4Help service;
- A Data4Help registered user logs into the system;
- A user recives a request to share his data;
- A user accept/decline a request to share his data;
- A third party requires data of a specific user;
- A third party requires data of a group of users;
- A user subscribes to AutomatedSOS service;
- An ambulance is called as a consequence of specific acquired data from the system;
- A Data4Help user access to Track4Run for the first time;
- A Track4Run user organizes a new run;
- A Track4Run user enrols for a run;
- An unregistered user access as a spectator to a run.

Machine phenomena

- The machine interfaces with external software and hardware systems;
- The machine manages database queries;
- The machine manages the 3G/4G Internet connection;
- The machine manages the Bluetooth connection;
- The machine manages GPS tracking.

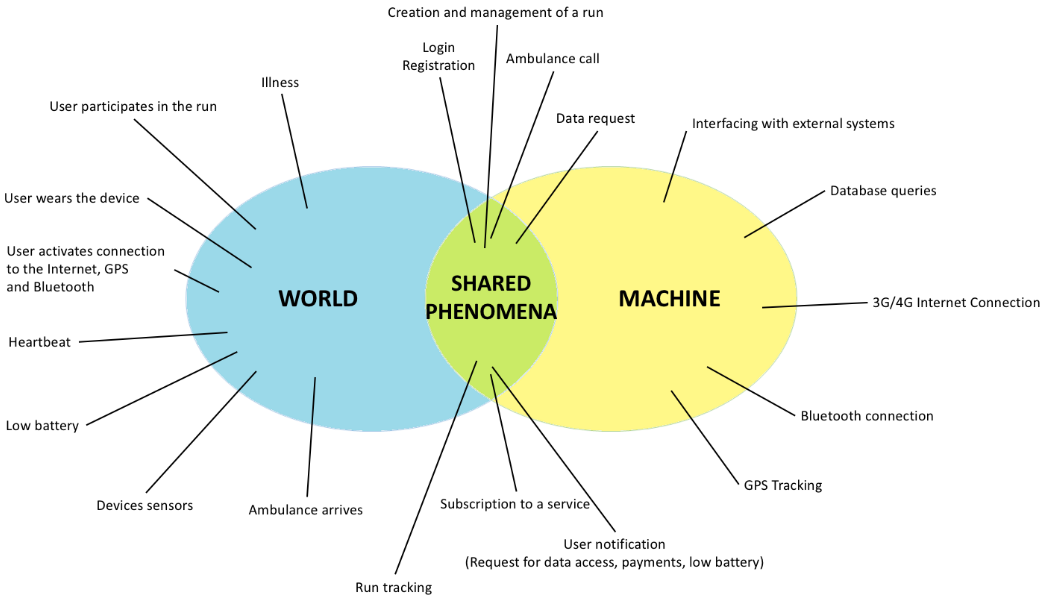

Figure 1.1: World and Machine model

1.3 Definitions, Acronyms and Abbreviations

API: Application Programming Interface;

GPS: Global Positioning System;

Organizer: A registered user that organizes a run, defining date and path;

OS: Operating System;

RASD: Requirement Analysis and Specification Document;

Run: An event that is organized by one organizer, at which one or more people can partecipate and that can be followed by one or more spectators;

Runner: A registered user that enrols for a run;

Spectator: Unregistered user that access to Track4Run to follow a run;

System: The software system-to-be, including all of its services;

Third party: Any external organization that wants to access to data acquired by Data4Help;

UML: Unified Modeling Language;

User: Any person, registered or not, who accesses to one of the applications (for Data4Help there is a special user called *Third party*);

VAT: Value Added Tax.

1.4 Reference documents

——————— TODO ———————

1.5 Overview

This document is structured as follows:

Section 1: Introduction. A general introduction to the goals, the phenomena and the scope of the system-to-be. It aims giving general but exaustive information about what this document is going explain.

Section 2: Overall Description.

Section 3: Specific Requirements.

Section 4: Effort Spent. A summary of the worked time by each member of the group.

At the end there is the bibliography.

Section 2

Overall Description

2.1 Product Perspective

2.1.1 User Interfaces

Standard Users

Standard users can use two different smartphone applications: *AutomatedSOS* and *Track4Run*. Both of them should be very easy to use and should allow the user to connect the smartphone to the smartwatch or to the chosen device. *AutomatedSOS* is mainly used by elderly people so it should have large buttons and large writing and it shouldn't ask to the user to interact a lot with the device. *Track4Run* is mainly used by young people so it should be more interactive and allow the sharing of the track on social networks and other social options such as comparing race data with friends. Standard users can also access services provided by TrackMe using a web application. Using it they can manage their accounts in a more comfortable way, verify requests for accessing their data, create new route and follow a race watching players position on the map (in *Track4Run* service).

Special Users

Third parties who want to analyse data collected from *Data4Help* can access the service using a web application. The web application lets special users to insert a request for data. If the request is accepted, it allows the download of the asked data. The system should also offer an online support to help user in using the service.

2.1.2 Hardware Interfaces

- Web applications (both the one for standard users and the one for special users) must be accessible using a computer with characteristic specified in section 2.4.4.
- Smartphone on which the app must work must provide to the app an Internet connection used to send data to TrackMe servers. At least one between the smartphone and the smartwatch must have a GPS antenna built in. The wearable device must also integrate a reasonably precise heartbeat sensor.

2.1.3 Software interfaces

- Web applications (both the one for standard users and the one for special users) must be compatible with the most popular browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari.
- Mobile apps for standard users must be available for both iOS and Android devices and must be compatible with most of the smartwatch and other health devices available on the market regardless of the operating system used by the device (using the API made available to programmers by producers).
- Application backends store data collected in a relational DBMS.
- Web applications access and show data by accessing the relational DBMS.
- Web applications for third parties has to interface also with a payments broker in order to receive money from companies who want to get data from the system.
- Web application and Track4Run have to interface also with Maps in order to generate the path for the run and to virtually follow a run.

2.2 Product Functions

The system is composed by several applications.

AutomatedSOS

AutomatedSOS is designed for elderly people and allows you to make an automatic call for help if the application detects a dangerous state of health. To use this app, few user interactions are required. In particular the user can:

- Register to the service.
- Log-in to the service.
- Respond to requests to access to their personal data by a third party.
- Report a false alarm following an emergency call with an alert to the nearest ambulance.
- Manage personal account and send a request to delete all the acquired data.
- Connect the health device such as smartwatch, smart band, heart rate sensors with Bluetooth.
- Pause data monitoring.

The app will autonomously monitor the health status of the user and make an emergency call to the nearest ambulance in case of emergency.

Track4Run

Track4Run is designed to track athletes participating in a run. Using it the user can:

- Register to the service.
- Log-in to the service.
- Respond to requests to access to their personal data by a third party.
- Manage personal account and send a request to delete all the acquired data.
- Connect the health device such as smartwatch, smart band, heart rate sensors with Bluetooth.
- Pause data monitoring.

- Share performance data via popular social networks such as Facebook, Instagram, Twitter, etc.

The app will autonomously track the health status and the position of the athlete.

Web application for standard users

Using it they can:

- Register to the service.
- Log-in to the service.
- Respond to requests to access to their personal data by a third party.
- Manage personal account and send a request to delete all the acquired data.
- Crate a path to use in *Track4Run*.
- Send an invitation to join in a run.
- Follow a competition watching the position of the athletes on the map.

Web application for special users

Using it they can:

- Register to the service.
- Log-in to the service.
- Send a request to access data of a standard user.
- Send a request to access data of a group of people.
- Manage past requests.
- Download data obtained after a request has been accepted.

——ADD SCHEMA PRODUCT FUNCTIONS——

2.3 User Characteristics

Those applications have different targets.

AutomatedSOS

This mobile app is thought for elderly people. It is not necessary the user is a “tech addicted” because a familiar can setup the system and let it work autonomously.

Track4Run

This mobile app is thought for athletes. It is most dedicated to young people who use frequently tech products.

Third party WebApp

This application is thought for companies who want to analyse data collected by the app. They could be statistics or pharmaceutical companies, hospitals, etc.

2.4 Constraints

2.4.1 Anonymous data collection

Companies who want to analyse data from a group of people without asking the permission to every single person must make a request for anonymized data of a group of at least 1000 people.

2.4.2 Privacy

Before allowing a company to access a user’s data, it is necessary to get a formal permission by the user.

2.4.3 Regulatory Policies

When a new user register to the service he must accept the privacy policy in order to use the application. He must be informed about personal and sensible data collection like his position and his health parameters. In every moment the user can ask TrackMe to delete all the collected data about him. All the collected data must be kept safe and must not be accessible by unauthorized person. Also, third parties who access the service must guarantee the security of the data. The whole process must comply with the GDPR regulations for the protection of users' personal data.

2.4.4 Hardware limitations

In order to use the service, user's hardware should comply to these minimum requirements:

Mobile application

- Smartphone
  - iOS or Android operative system
  - UMTS/4G Internet connection with a minimum speed of 1Mb/s
  - Bluetooth antenna
  - GPS antenna
  - 300 Mb available memory
  - Dual-core processor
  - 1 Gb RAM
- Smartwatch / other health device
  - Bluetooth antenna
  - Heartbeat sensor

Web application

- Computer
  - Internet connection with a minimum speed of 1Mb/s
  - Browser application
  - 720p monitor resolution

2.4.5 Parallel operation

The system must guarantee the simultaneous use of the mobile app by at least 100,000 users and the simultaneous use of the web app by at least 10,000 users. Consequently, the DBMS must be able to process a large number of simultaneous transactions.

2.4.6 Reliability requirement

The system reliability, seen as the probability that components and performances will meet the requirement during a specified period of time, must be at least 95%, considering a period of one month.

2.4.7 Availability requirement

The system should be available 24/7 in order to guarantee the service and to manage emergency situations.

2.4.8 Criticality of the system

AutomatedSOS

An error in the system could result in an unnecessary call for an ambulance or failure to call in an emergency.

Track4Run

An error in the system could cause a non-optimal use of the service.

2.5 Assumptions and Dependencies

From now on the following assumptions are given for guaranteed:

1. Users of the app have a phone with an iOS or Android operating system.
2. Users of the app have a phone with working GPS module with an uncertain of ±1 meter.
3. All users enjoy a stable Internet connection.
4. The user accepts or refuses the request for access to his data within 24 hours from what is forwarded. After this period the request is considered rejected.
5. Each user is identified with a unique code.
6. Once the request is accepted by the user, the system provides the applicant with the data within 24 hours.
7. Users enter the correct data during registration.

8. The user autonomously recharges the smartphone and the smartwatch when it signals that the battery is low.
9. A user cannot participate in two races at the same time.
10. Every request for access to a user's personal data by third parties must be explicitly accepted by the user.
11. When a third party accesses the personal data of a user is responsible for any unauthorized disclosure of data.

Behavioural during devices recharge time:

1. If the smartphone is charging in a fixed position near the user, the wearable device remains connected via Bluetooth to the smartphone and the *AutomatedSOS* service is not interrupted.
2. If the smartphone is charging in a fixed position, the *Track4Run* service is interrupted.
3. If the smartphone is charging using a transportable battery, all the services keep working.
4. If the wearable device is charging, the service *AutomatedSOS* is interrupted.
5. If the wearable device is charging, the service *Track4Run* keep working (available only data about position).

2.6 Future Extensions

The addition of new features will be evaluated in the future. The possible proposals are:

1. The creation of a new application with the purpose of collecting user data, without offering services such as *AutomatedSOS* and *Track4Run*. Users will be paid to use the application consistently. This operation considerably increases the number of users and also collects data from those subjects that are currently excluded (people who are not elderly and do not practice sports).
2. The possibility for the user to share their sanitary facilities for free with their doctor.

Section 3

Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interfaces of *AutomatedSOS* and *Track4Run* must be intuitive and user-friendly in order to permit an easy interaction with all the services offered by the systems.
Moreover both the application and the web site must support multiple languages.

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communication Interfaces

3.2 Functional Requirements

3.2.1 Individual Sign In

Purpose
Anyone who wants to subscribe to one or both services offered by *Data4Help* must go through the registration process, which can be carried out either through *AutomatedSos* and *Track4Run* apps or through the web site. The process requires exactly the same steps regardless the platform through which it is carried out:

1. The new user is required to fill in all the fields in which she/he is asked for his name, his surname, his date of birth, his city of birth, his city of residence, his occupation, and a valid e-mail address;

2. The user must accept the conditions regarding his privacy, in particular the collection of his data by *Data4Help* and the sharing of them in anonymous way with third parties.

After that the system will check the correctness of the inserted data, in particular it will check that the user isn't already registerd and that the inserted e-mail isn't already used by someone else. If the result of this control is positive the registration is authorized and the user will receive a confermation e-mail to the specified e-mail address with the password she/he has to use to access to all *Data4Help* services.

Scenario 1
Sara would like to register her grandmother to *AutomatedSos* to not worry about her helth status when they are not together. She opens the browser on his personal computer and search for *Data4Help* web site, then she clicks on the "*Sign In*" button, which is located in the main page. She passes through the steps of the registration process, inserting his grandmother data and accepting the required conditions. Finally, if the inserted data are accepted by the system, she receives the confirmation e-mail.

Scenario 2
Marco wuold like to organize an amateur run with his friends and remembers that someone told him something about a new application called *Track4Run* so he decides to try it. He downloads the app on his smartphone and turn it on. The first page that is shown to him contains the "*Sign In*" button and the "*Log In*" one, he presses on the first one and starts his registration process. He doesn't use his personal e-mail address, but an e-mail address

he has in common with his brother that they usually use to make purchase online. Unexpectedly he is informed by the system that the insert e-mail is already registered in the database and so he has to change it and this time he inserts his personal e-mail address. This time the registration is succesfull and he receives the confermation e-mail.

Use Case
The *Individual Sign In* use case is analyzed in Table 3.1.

Functional requirements

1. The system must not accept an e-mail address that is already used by an already registered user;
2. The system must not authorize the registration untill all the fields are filled up;
3. The system must not authorize the registration untill the required conditions aren't accepted;
4. The system must send the confirmation e-mail to the inserted e-mail address with the password when "*Submit*" button is clicked only if all the inserted data are acceptable and the required conditions has been accepted;
5. The system must let the **Individual user to be** leave the registration process at anytime.

Actor	**Individual user to be**
Goal	**[G.1]**
Input Condition	A person wants to subscribe to one of *Data4Help* services
Event Flow	1. The **Individual user to be** opens the main page of *Data4Help* web site from his personal computer or of *AutomatedSos* or *Track4Run* apps from his smartphone; 2. The **Individual user to be** clicks on "*Sign in*" button; 3. The system shows the form the **Individual user to be** has to fill up; 4. The **Individual user to be** fills up the form with his name, his surname, his date of birth, his city of birth, his city of residence, his occupation and an e-mail address; 5. The **Individual user to be** accepts the required conditions; 6. The **Individual user to be** clicks on "*Submit*" button; 7. The system checks wheter the inserted information are acceptable or not; 8. The **Individual user to be** receives a confirmation e-mail containing the password he has to use to access to all *Data4Help* services.
Output Condition	The system tells the **Individual user to be** that his registration is completed
Exceptions	• If functional requirements 1 or 2 are not satisfied the process goes back to step 4; • If functional requirement 3 is not satisfied the process goes back to step 5; • If the **Individual user to be** decides to leave the registration process this one is aborted.

Table 3.1: *Individual Sign In* use case

3.2.2 Third Party Sign In

Purpose
Any third party who wants to subscribe to *Data4Help* must go through the registration process, which can be carried out through *Data4Help* web site. The process requires several mandatory steps:

1. The third party which aim to become a new member is required to fill in all the fields in which it is asked for informations about the company itself like: its business name, its VAT number, its legal address, its billing address, its corporate e-mail address and the sector in which it operates;

2. The third party must also provide the data of its legal representative, in particular his name, his surname, his office address, his phone number, his e-mail address and his fiscal code;

3. The third party must select the preferred payment method;

4. The third party must accept different conditions:

   - It must assume responsibility in case of unauthorized disclosure of user data;
   - It must accept Milan as the place of jurisdiction in the case of a legal dispute.

After that the system will check the correctness of the insrted data, in particular it will check that the third party isn't already registered. If the result of this control is positive the registration is authorized and the third party will receive a confirmation e-mail to the specified e-mail address with the password it has to use to access to *Data4Help* services. From now we will refer to the third party that wants to become a new member as "**Special user to Be**" to distinguish it from an Individual user.

Scenario 1
PharmaAnalisi SPA wants to acquire data of a group of young people in order to do an analysis about the kind of life they conduct. It opens the browser and search for *Data4Help* web site, then it clicks on the "*Third Party Sign In*" button, which is located in the main page. It passes through the steps of the registration process, inserting all the required data but forgotting to accept one of the conditions. As a consequence the system won't permit it to conclude the registration process, so it checks again and figure out what was missing, it accepts the condition and submit its registration. Finally, it receives the confimation e-mail.

Use Case
The *Third Party Sign In* use case is analyzed in Table 3.2.

Functional requirements

1. The system must not accept an e-mail address that is already used by an already registered third party;
2. The system must not accept a business name that is already used by an already registered third party;
3. The system must not accept a VAT number that is already used by an already registered third party;
4. The system must not authorize the registration untill all the fields are filled up;
5. The system must not authorize the registration untill the preferred payment method has been selected;
6. The system must not authorize the registration untill the required conditions aren't accepted;
7. The system must send the confirmation e-mail to the inserted e-mail address with the password when "*Submit*" button is clicked only if all the inserted data are acceptable and the required conditions has been accepted;
8. The system must let the **Special user to be** leave the registration process at anytime.

Actor	**Special user to be**
Goal	**[G.1]**
Input Condition	A third party wants to subscribe to *Data4Help* services
Event Flow	1. The **Special user to be** opens the main page of *Data4Help* web site. 2. The **Special user to be** clicks on "*Sign in (Third party)*" button; 3. The system shows the form the **Special user to be** has to fill up; 4. The **Special user to be** fills up the form with its business name, its VAT number, its legal address, its billing address, its corporate e-mail address and the sector in which it operates; 5. The **Special user to be** selects the preferred payment method; 6. The **Special user to be** accepts the required conditions; 7. The **Special user to be** clicks on "*Submit*" button; 8. The system checks wheter the inserted information are acceptable or not; 9. The **Special user to be** receives a confirmation e-mail containing the password it has to use to access to *Data4Help* services.
Output Condition	The system tells the **Special user to be** that its registration is completed
Exceptions	• If functional requirements 1,2,3 or 4 are not satisfied the process goes back to step 4; • If functional requirement 5 is not satisfied the process goes back to step 5; • If functional requirement 6 is not satisfied the process goes back to step 6; • If the **Special user to be** decides to leave the registration process this one is aborted.

Table 3.2: *Third Party Sign In* use case

3.2.3 Individual Log In

Purpose
The main goal of the login feature is to allow the access to one of the services of *Data4Help* to any registered user. To access to an application the user has to fill out the credential form where e-mail and password are required. Moreover, there is a *Forgot password?* section where a user could recover his/her password via e-mail. An e-mail is sended to the user with a temporary password that user will change once logged in.
Moreover, at the first login in one of the application the *Individual User* must associate to the system one device (like smartwatch or similar) to allow the system to trace his/her data. This process is very important in *AutomatedSOS* application.

Scenario 1
Francesca loves running. When she has heard about *Track4Run* application she downloaded it immediately. Her friend Clara told her about a charity run for the following weekend, so Francesca opened *Track4Run*, she clicked on *Log in*. She inserted her e-mail address and password and clicked on the login button. Everything was correct, so she entered in the system and could enroll the run.

Scenario 2
One year ago Tommaso, Aldo's grandchild, installed on Aldo's phone *AutomatedSOS*. Yesterday, Aldo bought a new phone, he downloaded the app but he forgot his password so he couldn't log in the application. He clicked on *Forgot password?* he inserted his e-mail address and clicked on *Restore my password*. He received a mail with the provisional password and he became able to access to the system. After the access he was forced to change his provisional password into a new one.

Scenario 3
According to the scenario presented in Section 3.2.1 where we told about the registration of Sara's grandmother.
Sara helps her grandmother to do the first log in *AutomatedSOS*. After inserting of credentials and a succesfull login, Sara has to match the grandma's smartwatch with the application. *AutomatedSOS* has a wizard to help users: Sara turns on the bluetooth in each devices (smartwatch and phone); after the phone matching with the smartwatch via bluetooth (system matching), Sara has to do the matching with *AutomatedSOS*. On the smartwatch's screen

there is a six digits number and Sara puts this number in *AutomatedSOS* application and she clicks on *Done* button. After that smartwatch and phone are matched and *AutomatedSOS* is able to watch the health status of Sara's grandmother.

Use Case
The *Generic Individual Log In* use case is analyzed in Table 3.3.
The *First Individual Log In* use case is analyzed in Table 3.4.

Functional requirements

1. The **Individual User** must be already registered in the system in order to log in successfully;
2. The **Individual User** has to remember his/her e-mail address and password in order to log in successfully;
3. The password inserted by the **Individual User** must correspond with the e-mail address;
4. If the **Individual User** inserts wrong credential could not be able to access to the system;
5. If the **Individual User** clicks on *Forgot password?*, the system sends a new password to the **Individual User** e-mail address if and only if the e-mail address is valid and registered to the system;
6. After a password restoring through *Forgot password?* operations, the system must allow the access with the temporary password and then it has to force the **Individual User** to change the temporary password into a new one;
7. (First access only) The inserted matching number by the **Individual User** must correspond with the visualized on the device's screen.

Actor	**Individual user**
Goal	**[G.2]**
Input Condition	The **Individual User** is already registered to the system and want to log in
Event Flow	1. The **Individual User** open one of the applications (*AutomatedSOS* or *Track4Run*); 2. The **Individual User** clicks on *Log In* button; 3. The **Individual User** fills in the fields with his/her e-mail address and password; 4. The **Individual User** clicks on the login button.
Output Condition	The system allows the login of the **Individual User** and loads the dashboard of the **Individual User**.
Exceptions	• If functional requirements 1 or 3 are not satisfied the system notifies the **Individual User** with an error message and the process goes back to step 3; • If the **Individual User** inserts wrong credentials for three times the system notifies him/her with an e-mail.

Table 3.3: *Generic Individual Log In* use case

Actor	**Individual user**
Goal	**[G.2]**
Input Condition	The **Individual User** is already registered to the system and want to log in
Event Flow	The first part of the event flow is already explained in Table 3.3. 1. The **Individual User** has to match a device to *AutomatedSOS* or *Track4Run* application; 2. The **Individual User** switchs on the bluetooth in each devices; 3. The **Individual User** connects the mobile phone with the health device (system connection via bluetooth); 4. The **Individual User** inserts the six digits number visualized on the screen of the health device; 5. The **Individual User** clicks on the *Done* button;
Output Condition	The system allows the login of the **Individual User** and loads the dashboard of the **Individual User**.
Exceptions	All already explained exceptions in Table 3.3 are still valid. • If functional requirement 7 is not satisfied the system notifies the **Individual User** with an error message and the process goes back to step 2;

Table 3.4: *First Individual Log In* use case

3.2.4 Third Party Sign In

Purpose
The main goal of the login feature is to allow the access to one of the services of *Data4Help* to any registered user. To access to an application the user has to fill out the credential form where e-mail and password are required. Moreover, there is a *Forgot password?* section where a user could recover his/her password via e-mail. An e-mail is sended to the user with a temporary password that user will change once logged in.

Scenario
The Policlinico Cardiology Departement wants to acquire data of his/her patients, knowing their SSN. In order to see the health status of a specific patient, Francesca - head nurse - opens her laptop and she goes to *Data4Help* web site. Francesca inserts the e-mail address of the departement and the password, the access is allowed, the system shows her the dashboard and she could be able to check her patient status.

Use case and Functional Requirement
According to the **Individual Log In** [Section 3.2.1] functional requirements and use case are the same.
Exception is done only for the first access where device conncetion is not asked in this case.

3.2.5 Manage Profile

Purpose
Any user can manage his personal profile both from *Data4Help* web site and from *AutomatedSOS* or *Track4Run* applications. In particular:

- The user can change some of his personal informations: his city of residence and his occupation.
- The user can see the data acquired on him until this moment;
- The user can see the past received requests about seeing his personal data;
- The user can see the pending received requests about seeing his personal data;
- The user can change his password;
- The user can delete his profile.

Scenario 1
Chiara has just finished her studies and has just found a new job, so she wants to update the occupation field on her profile. She opens *Data4Help* web site from her personal computer, she logs in and goes in her personal area. Then she clicks on "*Edit profile*" button and the system gives her the possibility to change either her city of residence or her occupation. She changes her occupation from student to emplyed and she clicks on "*Submit changes*" button.

Scenario 2
Matteo has just finished the registration process, but he doesn't like the password he was given by the system and he wants to change it. He opens *AutomatedSOS* application on his smartphone, logs in and accesses to his personal profile. Now he clicks on "*Change password*" button and inserts the old password and the new password twice as required by the system. Finally he clicks on "*Submit changes*" button.

Scenario 3
Aldo moved to USA and so he decides to delete his profile on *Track4Run* because he was used to use it to organize amateur runs with his friends, but now he won't be able to do it anymore. He opens *Track4Run* application on his smartphone, logs in and accesses to his personal profile. Now he clicks

on "*Delete profile*" button and confirm his choice. The system removes all Aldo's information from the database.

Scenario 4
CONNECT A DEVICE

Use Case
The *Profile visualization* use case is analyzed in Table 3.5.
The *Modify personal information* use case is analyzed in Table 3.6.
The *Change password* use case is analyzed in Table 3.7.
The *Delete profile* use case is analyzed in Table 3.8

Functional requirements

1. The system must let the user view his personal profile at anytime;
2. The system must let the user upload/change his personal information at anytime;
3. The system must let the user change his password only if the old one has been inserted correctly;
4. The system must not let the user change his password if the new one has not been inserted twice;
5. The system must let the user change the device connected to his profile at anytime;
6. The system must let the user delete his profile at anytime;
7. The system must require to confirm a deleting request;
8. The system must not delete a profile if the choice isn't confirmed by the user;
9. The system must let the user leave the editing profile process at anytime;
10. The system must delete all user's personal information from its database when the user decides to delete his profile;

Actor	**User**
Goal	[**G.3**]
Input Condition	A **User** wants to view his personal profile
Event Flow	1. The **User** opens *Data4Help* web site or *AutomatedSOS* or *Track4Run* applications; 2. The **User** logs in; 3. The **User** accesses to his personal area; 4. The system shows to the **User** the data acquired on him and his pending and past requests.
Output Condition	The **User** views his personal profile with all the related information
Exceptions	None

Table 3.5: *Profile visualization* use case

Actor	**User**
Goal	**[G.3]**
Input Condition	A **User** wants to modify his personal information
Event Flow	1. The **User** opens *Data4Help* web site or *AutomatedSOS* or *Track4Run* applications; 2. The **User** logs in; 3. The **User** accesses to his personal area; 4. The **User** clicks on "*Edit profile*" button; 5. The system shows the **User** the modifiable information; 6. The **User** modifies what he wants; 7. The **User** click on "*Submit changes*" button;
Output Condition	The **User**'s information are modified
Exceptions	• If the **User** decides to leave the editing process this one is aborted.

Table 3.6: *Modify personal information* use case

Actor	**User**
Goal	**[G.3]**
Input Condition	A **User** wants to change his password
Event Flow	1. The **User** opens *Data4Help* web site or *AutomatedSOS* or *Track4Run* applications; 2. The **User** logs in; 3. The **User** accesses to his personal area; 4. The **User** clicks on "*Change password*" button; 5. The system shows the **User** the fields in which he has to insert the old and the new password; 6. The **User** inserts the old password; 7. The **User** inserts the new password twice; 8. The **User** click on "*Submit changes*" button;
Output Condition	The **User**'s password is modified
Exceptions	• If functional requirement 3 is not satisfied the system goes back to step 6; • If functional requirement 4 is not satisfied the system goes back to step 7; • If the **User** decides to leave the editing process this one is aborted.

Table 3.7: *Change password* use case

Actor	**User**
Goal	**[G.3]**
Input Condition	A **User** wants to delete his profile
Event Flow	1. The **User** opens *Data4Help* web site or *AutomatedSOS* or *Track4Run* applications; 2. The **User** logs in; 3. The **User** accesses to his personal area; 4. The **User** clicks on "*Delete profile*" button; 5. The **User** confirms his choice;
Output Condition	The **User**'s profile is deleted
Exceptions	• If functional requirement 8 is not satisfied the deleting process is aborted; • If the **User** decides to leave the deleting process this one is aborted.

Table 3.8: *Delete profile* use case

3.2.6 Individual Data Requirement

Purpose
Everytime a special user wants to access to the data of a specific individual it has to pass through some steps:

1. It has to insert the fiscal code of the targeted individual, who will receive a direct request that he/she can accept or refuse;

2. If the request has been accepted the special user will receive a notification with the payment form with the amount it has to pay to download the required data;

3. If it pays the amount due it can download the required data.

Scenario 1
AC Milan wants to acquire information about the lifestyle of Cristino Ronaldo, so it opens the browser and search for *Data4Help* web site, it logs in and then it clicks on "*Individual data requirement*" button. Then it inserts Ronaldo's fiscal code, clicks on "*Submit* button and waits a couple of days for a response. Unfortunately, Ronaldo doesn't accept the request, AC Milan receive a notification about the negative response and the process ends.

Scenario 2
The "San Gerardo" hospital of Monza is trying *Data4Help* combined with *AutomatedSos* to monitor the health status of some of its patients after the dismissal. Lucia has been dismissed a couple of month ago and she has *AutomatedSos* installed on her smartphone. The "San Gerardo" hospital accesses to *Data4Help* from the web site, logs in, clicks on "*Individual data requirement*" button, inserts Lucia's fiscal code ì, clicks on "*Submit* button and waits a couple of days for a response. Lucia accepts the request, so the "San Gerardo" hospital receives the notification with payment form with the amount it has to pay, pays it and downloads Lucia's data.

Use Case
The *Individual Data Requirement* use case is analyzed in Table 3.9.

Functional requirements

1. The system must refuse a non existant fiscal code;

2. The system must not let the special user download the required data untill it hasn't paid the amount due;

3. The system must send a notification in case of negative response and end the process;

4. The system must send a notification with the payment form with the amount due in case of positive response;

5. The system must let the **Special user** leave the data requirement process at anytime.

Actors	**Special user** and targeted individual
Goal	**[G.4]**
Input Condition	A **Special user** wants to acquire data of an individual
Event Flow	1. The **Special user** opens the main page of *Data4Help* web site; 2. The **Special user** logs in; 3. The **Special user** clicks on "*Individual data requirement*" button; 4. The **Special user** inserts the fiscal code of the targeted user; 5. The **Special user** clicks on "*Submit* button; 6. The **Special user** recieves a notification with the response; 7. If the response is positive the **Special user** recieves the payment form with the amount it has to pay; 8. The **Special user** pays the amount due; 9. The **Special user** downloads the required data.
Output Condition	The **Special user** receives the required data
Exceptions	• If functional requirements 1 is not satisfied the process goes back to step 4; • If functional requirement 2 is not satisfied the process goes back to step 8; • If the targeted individual refuse to share his data the process is aborted; • If the **Special user** decides to leave the data requirement process this one is aborted.

Table 3.9: *Individual Data Requirement* use case

3.2.7 Group Data Requirement

Purpose
Everytime a special user wants to access to the data of a group of people it has to specify at least one of the required characteristics of it:

- It can choose among the age ranges of the members of the targeted group proposed by the system;
- It can specify the Italian city of birth of the members of the targeted group;
- It can specify the Italian city of residence of the members of the targeted group;
- It can specify the Italian province of residence of the members of the targeted group;
- It can specify the Italian province of birth of the members of the targeted group;
- It can specify the Italian region of residence of the members of the targeted group;
- It can specify the Italian region of birth of the members of the targeted group;
- It can specify the state of birth of the members of the targeted group;
- It can specify the state of residence of the members of the targeted group;
- It can specify the current occupation (students, employeds, unemployeds) of the members of the targeted group.

After this the system will check if the targetd group is composed of more than 1000 people, if it is the special user will receieve the payment form with the amount it has to pay to download the required data. Once it pays the amount due it can download the required data.

Scenario 1
PharmaAnalisi SPA wants to acquire data of a group of students living in Lombardia in order to do an analysis about the kind of life they conduct. It opens the browser and search for *Data4Help* web site, then it logs in and clicks on "*Group data requirement*" button. It specifies that the age range of

the members of the targeted group must be from 18 to 24 years old, that they should live in a city in Lombardia and that they should be students. Then it clicks on "*Submit*" button and the system accepts its request, PharmaAnalisi SPA receives the payment form with the amount it has to pay, it pays the amount due and it downloads the required data.

Scenario 2

The municipality of Sondrio wants to analyze the quality of life of the people that were born in Monza and that moved to Sondrio. It opens the browser and search for *Data4Help* web site, then it logs in and clicks on "*Group data requirement*" button. It specifies that the city of birth of the members of the targeted group must be Monza and that their city of residence must be Sondrio. Then it clicks on "*Submit*" button but unfortunately the system tells that the required data aren't accessible because the targeted group of people is composed of less than 1000 people and so the process ends.

Use Case

The *Group Data Requirement* use case is analyzed in Table 3.10.

Functional requirements

1. The system must not give data of groups of people composed of less than 1000 people;

2. The system must not let the special user download the required data untill it hasn't paid the amount due;

3. The system must let the **Special user** leave the data requirement process at anytime.

Actor	**Special user**
Goal	**[G.5]**
Input Condition	A **Special user** wants to acquire data of a group of people
Event Flow	1. The **Special user** opens the main page of *Data4Help* web site; 2. The **Special user** logs in; 3. The **Special user** clicks on "*Group data requirement*" button; 4. The **Special user** specifies the characteristics of the targeted group; 5. The **Special user** clicks on "*Submit* button; 6. If the targeted group is composed of more than 1000 people the **Special user** receieves the payment form with the amount it has to pay to download the required data; 7. The **Special user** pays the amount due; 8. The **Special user** downloads the required data.
Output Condition	The **Special user** receives the required data
Exceptions	• If the targeted group is composed of less than 1000 people the **Special user** is informed and the process is aborted; • If functional requirement 2 is not satisfied the process goes back to step 7; • If the **Special user** decides to leave the data requirement process this one is aborted.

Table 3.10: *Group Data Requirement* use case

3.2.8 Health Status Visualization

Purpose
The main feature of *AutomatedSOS* is to check the health status of the user and it has to detect any critical situation. The detection of critical situation computes a huge number of data; this computation provides several information of the health status of the user, so *AutomatedSOS* provides a service to show the health status of the user.

Scenario
Silvia is worried about her grandmother because she is tired since few days. Two month ago Silvia installed on her grandmother phone *AutomatedSOS* application. In order to calm herself, Silvia takes the phone of her grandmother, she opens *AutomatedSOS* and she logs in. In the main page of the application there is the summary of the health status and everything looks ok. To avoid any doubt Silvia clicks on the *Details* button and she checks all statistics about last week and last month value of pressure and heartbeat.

Use Case
The *Health Status Visualization* use case is analyzed in Table 3.11.

Functional requirements

1. The system must let the user view his personal health status at anytime;
2. The system must update the health status of the user at anytime it receives new data from the devices;
3. The system must stores the data in order to provide monthly and weekly statustics.

Actor	**User**
Goal	[**G.7**]
Input Condition	A **User** wants to view his/her health status
Event Flow	1. The **User** opens *AutomatedSOS* application; 2. The **User** logs in; 3. The **User** accesses to *My Health Status*; 4. The system shows to the **User** the data acquired on him/her with the relative computation of the health status.
Output Condition	The **User** views his/her personal health status monitored by *AutomatedSOS*
Exceptions	If the system does not acquire enough data to produce statistics it notifies the **User** with a warning.

Table 3.11: *Health Status Visualization* use case

3.2.9 Critical Situation

Purpose
The main feature of *AutomatedSOS* is to check the health status of the user and it has to detect any critical situation. *AutomatedSOS* monitors the health status of the subscribed customers and, when such parameters are below certain thresholds, sends to the location of the customer to an ambulance, guaranteeing a reaction time of less than 5 seconds from the time the parameters are below the threshold.

Scenario 1
For a couple of days Vittorio felt very tired and affected of sickness. While he was walking in his house his heartbeat goes down and he lie down on the floor. Luckily he had installed *AutomatedSOS* application. The application detected a critical situation, it managed to track Vittorio position and it called the ambulace. The ambulace arrived very quickly and luckily the paramedics with a cardiac massage saved Vittorio that was carried to the hospital.

Scenario 2
Cristiano is a professional runner. He decides to install on his phone *AutomatedSOS* to keep trace of his health status and avoid any possible critical situation when he runs. One day, while he was running, *AutomatedSOS* went in an alerted status but after only 1 second the application came back in a normal status and stayed in the normal status all run duration. Probably it was an abnormal device measure of life value, so *AutomatedSOS* did not call the ambulance.

Use Case
The *Critical Situation* use case is analyzed in Table 3.12.

Functional requirements

1. The system must guarantee a reaction time of less than 5 seconds from the time it is in an alerted status;

2. The system must send the location of the customer to an ambulance;

3. The system must be in an alerted status when maximum pressure value of the customer is more than 170 mmHg and minimum pressure value is more than 100 mmHg;

Actor	**System**, **User**
Goal	[**G.8**]
Input Condition	The system goes in an alerted status
Event Flow	1. The **System** gets the GPS position of the **User**; 2. The **System** increases parameters detection with a frequence of 3 detection per second; 3. The **System** shows an alert message on the **User** device (smartwatch or similar).
Output Condition	The **System** calls an ambulance and it sends the location to the called ambulance if it is in an alert status from 3 seconds.
Exceptions	• If functional requirement 2 is not satisfied the **System** notifies the ambulance with a detected position error and it sends the last deteted position; • If functional requirements 3 and 4 are not satisfied the **System** notifies the **User** with a warning message and it invites the **User** to check the status and the conncetion of his/her devices.

Table 3.12: *Critical Situation* use case

4. The system must be in an alerted status when the heartbeat is lower than 45 bmp or it is higher than 120 bpm;

5. If the system goes in an alerted status it has to increase the life parameters detection frequency.

3.2.10 Create a Run

Purpose
Very important feature of *Track4Run* is the possibility for a *Organizer* to be able to create a new *Run.* To create a *Run* an *Organizer* must define:

- The path of the *Run* through an intercative tool;
- The date of the *Run*;
- The maximum number of participants to the *Run*;
- The expiration date to enroll the *Run*;

Scenario
Massimo want to organize a charity *Run* in his little town. He is already registered to *Track4Run* as a Runner. After he has done all permission he goes to *Track4Run* web site. With the same credential of the mobile application he logged in the system and in the dashboard he clicked on *Create a Run.* Massimo set the path trought the intercative tool, he fixed the date of the *Run* and the missing fields. When everything was correct he clicked on the *Create* button and the *Run* went on-line.

Use Case
The *Create a Run* use case is analyzed in Table 3.13.

Functional requirements

1. The system must not accept *Run* with date less than or equal to the current one;
2. The system must not accept *Run* with expiration date less than or equal to the current one;
3. The system must not accept *Run* with maximum nuber of participants less than or equal to 1.
4. The system must not accept *Run* with a path duplication greater or equal to the 80 percent in the same date of the new one;

Actor	**Organizer**
Goal	**[G.10]**
Input Condition	The **Organizer** want to create a new *Run*
Event Flow	1. The **Organizer** open *Track4Run* service through web application and he/she log in; 2. The **Organizer** clicks on *Create a Run* button; 3. The **Organizer** inserts path, date, expiration date and maximum number of participants; 4. The **Organizer** clicks on the *Create* button;
Output Condition	The system registers the new *Run* and it notifies him/her with a confirmation e-mail.
Exceptions	• If functional requirements 1, 2, 3 or 4 are not satisfied the system notifies the **Organizer** with an error message and the process goes back to step 3; • In order to prevent functional requirements 4 failure, during the building phase of the path the system continuously checks satisfaction and when the functional requirement is not satisfied it notifies the **Organizer** with a warning; • If the **Organizer** decides to leave the creation process this one is aborted.

Table 3.13: *Create a Run* use case

3.2.11 Delete a Run

Purpose
As we have the possibilty in *Track4Run* for an organizer to be able to create a new *Run*. The system must be able to manage the decision of an organizer to delete a *Run*.

Scenario
Alice want to delete a *Run* that she had organized for the next month, because the weather forecast are not the optimal. In order to avoid a rainy day Alice opened *Track4Run* web-site, she logged in and she clicked on *Manage a Run*. On top of the table there was the *Run* that she was looking for, she clicked on it and when the *Run* was opened she clicked on *Delete*. After that a mail arrived in Alice mailbox, it was the confirmation of the elimination of the *Run*.

Use Case
The *Delete a Run* use case is analyzed in Table 3.14.

Functional requirements

1. The system must not show *Run*, in *Manage a Run* section, in which an **Organizer** is not the owner.

Actor	**Organizer**
Goal	**[G.11]**
Input Condition	The **Organizer** want to delete a *Run*
Event Flow	1. The **Organizer** open *Track4Run* service through web application and he/she log in; 2. The **Organizer** clicks on *Manage a Run* button; 3. The **Organizer** looks for a *Run* through the search bar or looking to the proposed; 4. The **Organizer** clicks on the *Run* he/she wants to delete; 5. The **Organizer** clicks on the *Delete*.
Output Condition	The system delete the *Run* and it notifies him/her with a confirmation e-mail. Moreover the system must notify all enrolled people in the *Run* with a e-mail and delete his/her enrolment.
Exceptions	• The **Organizer** looks for a *Run* that is not present in the system or he/she is not the owner, the system notifies the **Organizer** with a warning message; • If the **Organizer** decides to leave the elimination process this one is aborted.

Table 3.14: *Delete a Run* use case

3.2.12 Enrol a Run

Purpose

One of the great feature of *Track4Run* is the possibility for a runner to be able to enrol a *Run*. In order to enrol a *Run* a runner must be logged in *Track4Run* application, he/she has to search the *Run* that wants in the *Enrol a Run* section and finally enrol it. However, the *Run* could be already done or the enrolling time expired.

Scenario 1

Andrea is technological boy. A few weeks ago he found in the Play Store the new app *Track4Run* and shared his discovery with his friends. Yesterday, while he was hanging out with the buddies they discovered a new run for week-end after. So, Andrea took his phone, he opened *Track4Run* app, he logged in, he clicked on *Enrol a Run* and the *Run* was on top yet. So Andrea clicked on it and when the *Run* event was opened he cliked on the *Enrol* button. After that Andrea received a confirmation e-mail of the correct enrolment.

Scenario 2

Samanta loves walking but for a few months now she started running. With her friend Federica she told about the city *Run* planned in two day. But unfortunately when she took her phone and opened *Track4Run* she discovered that the time to enrol the *Run* was expired yet.

Use Case

The *Enrol a Run* use case is analyzed in Table 3.15.

Functional requirements

1. The system must not accept an enrolment for a *Run* where a **Runner** is enrolled yet.
2. The system must not accept an enrolment for a *Run* where the enrolment time is expired yet.
3. The system must not accept an enrolment for a *Run* where the maximum number of enrolment is reached;
4. The system must not accept an enrolment for a *Run* where **Runner** and *Organizer* are the same person.

Actor	**Runner**
Goal	[**G.12**]
Input Condition	The **Runner** want to enrol a *Run*
Event Flow	1. The **Runner** open *Track4Run* service through mobile application and he/she log in; 2. The **Runner** clicks on *Enrol a Run* button; 3. The **Runner** looks for a *Run* through the search bar or looking to the proposed; 4. The **Runner** clicks on the *Run* he/she wants to enrol; 5. The **Runner** clicks on the *Enrol*.
Output Condition	The system registers the enrolment of the **Runner** and it notifies him/her with a confirmation e-mail.
Exceptions	• If functional requirements 1, 2, 3 or 4 are not satisfied the system notifies the **Runner** with an error message and the process goes back to step 3; • The **Runner** looks for a *Run* that is not present in the system, the system notifies the **Runner** with a warning message; • If the **Runner** decides to leave the enrolment process this one is aborted.

Table 3.15: *Enrol a Run* use case

3.2.13 Delete an Enrolment of a Run

Purpose
As we have the possibilty in *Track4Run* for a runner to be able to enrol a *Run*. The system must be able to manage the decision of a runner to delete his/her enrolment.

Scenario
Giulia is enrolled for the annual *Run* of her neighborhood. The enrolment management was made through *Track4Run* application. Unfortunately for the day of the *Run* Giulia will be in Florence for an important work meeting. When Giulia received the meeting mail she took her phone, she opened *Track4Run* app and she clicked on *Enrolled Run*. The only row in the table was the annual *Run*, Giulia clicked on it and the she clicked on *Delete Enrolment* button. After that Giulia received a confirmation e-mail.

Use Case
The *Delete an Enrolment of a Run* use case is analyzed in Table 3.16.

Functional requirements

1. The system must not show *Run*, in *Enrolled Run* section, in which a **Runner** is not enrolled.

Actor	**Runner**
Goal	**[G.13]**
Input Condition	The **Runner** want to delete an enrolment of *Run*
Event Flow	1. The **Runner** open *Track4Run* service through mobile application and he/she log in; 2. The **Runner** clicks on *Enrolled Run* button; 3. The **Runner** looks for a *Run* through the search bar or looking to the proposed; 4. The **Runner** clicks on the *Run* he/she wants to delete the enrolment; 5. The **Runner** clicks on the *Delete Enrolment.*
Output Condition	The system deletes the enrolment registration of the **Runner** and it notifies him/her with a confirmation e-mail.
Exceptions	• The **Runner** looks for a *Run* that is not present in the system, the system notifies the **Runner** with a warning message; • If the **Runner** decides to leave the elimination process this one is aborted.

Table 3.16: *Delete an Enrolment to a Run* use case

3.2.14 Run Watching

Purpose
For each *Run*, that is present in the system, a *Specator* or an *User* must be able to watch it. The *Spectator* or the *User* could follow the positon, on the *Run* path, of any person that has enrolled the *Run* and also he/she must be able to look at the placing table of the runners.

Scenario 1
Marco is a professional runner and play for a team, unfortunately one month ago he broke his leg. Today there is an important *Run* and his teammates are enrolled. At 4 p.m., Marco will open *Track4Run* app on his mobile phone, in the dashboard he will look for the today's *Run*,he will click on it and immediately he will be "in the *Run*"; he could follow his friend on the path (watching the position on the map) and clicking on the *Placings* button he could see the current placing table.

Scenario 2
Andrea is a sport event planner. With his coworker Luca, they planned the "HOLI FUN Run" of the 7th of July in Cremona. In the *Run*'s day, Andrea was in France for a meeting so when the meeting ends he went back to his hotel, he opend his laptop and he went to *Track4Run* web page, he searched, through the search bar, the *Run* that unfortunately was just ended, however he was very happy because looking at the placing table he saw that his friend Marta won.

Use Case
The *Run Watching* use case is analyzed in Table 3.17.

Functional requirements

1. **Generic User** (that could be *Specator* or *User*) must be able to watch a *Run* that is still in progress or ended;
2. The system must continuously check the positon of the runner in order to keep the map and the placing table updated;
3. The system must be able to show all the runners that enrolled the *Run* and their GPS position is enabled and nearby the path;

Actor	**Generic User** that could be *Specator* or *User*
Goal	[**G.14**] and [**G.15**]
Input Condition	The **Generic User** want to watch a *Run*
Event Flow	1. The **Generic User** open *Track4Run* service through mobile application or web application; 2. The **Generic User** clicks on *Watch a Run* button; 3. The **Generic User** looks for a *Run* through the search bar or looking to the proposed; 4. The **Generic User** clicks on the *Run* he/she wants to watch.
Output Condition	The system loads the *Run* environment (map, path and placing table) and shows it to the **Generic User**.
Exceptions	• The **Generic User** looks for a *Run* that is not present in the system, the system notifies the **Generic User** with a warning message; • If the connection of the **Generic User** application is lost and the system couldn't be able to recover it the process goes back to step 3.

Table 3.17: *Run Watching* use case

4. The system must notify the end of a *Run* to a **Specator** or an **User** that is watching it;

5. The system must be able to compute the placing table through the GPS position of the runner enrolled in the *Run*.

Section 4

Effort Spent

4.1 Michele Gatti

Task	**Hours**
Analysis of given RASD	*3*
Purpose and Goals	1
Product Perspective and Product Functions	5
User Characteristics and Constraints	2
Assumptions and Dependencies	3
The World and the Machine	3
Team revision	1

4.2 Federica Gianotti

Task	**Hours**
Analysis of given RASD	*3*
Purpose and Goals	4
Scope, Definitions, Acronyms and Abbreviations	2
Team revision	1
Functional Requirements	12

4.3 Mathyas Giudici

Task	**Hours**
Analysis of given RASD	*3*
GitHub and LaTeX setup	2
Purpose and Goals	2
Scope, Definitions, Acronyms and Abbreviations	2
Team revision	1
Functional Requirements	14
User Interface Mockup	1,5

Appendix A

Appendix

Bibliography

[1] Elisabetta Di Nitto - Software Engineering 2 Slides (AY 2018/2019) *Project goal, schedule and rules*

[2] ISO/IEC/IEEE 29148:2011 *Systems and software engineering - Life cycle processes - Requirements engineering*

[3] IEEE 830:1998 *Recommended Practice for Software Requirements Specifications*

[4] M.Jackson & P. Zave, *The World and The Machine*, 1995