

POLITECNICO DI MILANO

School of Industrial and Information Engineering

**Computer Science and Engineering**



**POLITECNICO**  
MILANO 1863

# **TRACKME RASD**

## **Requirements Analysis and Specification Document**

Software Engineering 2 Project

The project was made by

**Luca Alessandrelli 846260**

**Andrea Caraffa 919970**

**Andrea Bionda 921082**

Version 2.0 - 2018/2019

---

**Deliverable:** RASD  
**Title:** Requirements Analysis and Specification Document  
**Authors:** Luca Alessandrelli, Andrea Caraffa, Andrea Bionda  
**Version:** 2.0  
**Date:** 19-October-2018  
**Download page:** <https://github.com/lucaalexandrelli/AlessandrelliCaraffaBionda.git>  
**Copyright:** Copyright © 2017, Luca Alessandrelli, Andrea Caraffa, Andrea Bionda – All rights reserved

---

# Contents

<b>Table of Contents</b>	<b>3</b>
<b>1 Introduction</b>	<b>5</b>
1.1 Purpose	5
1.2 Scope	5
1.2.1 Goals	5
1.2.2 World Phenomena	6
1.3 Definitions, Acronyms, Abbreviations	6
1.4 Revision History	7
1.5 Document Structure	7
<b>2 Overall Description</b>	<b>8</b>
2.1 Product perspective	8
2.2 Product functions	10
2.2.1 Data4Help - Providing data to third parties	10
2.2.2 AutomatedSOS - Sending ambulance request in critical situation	10
2.2.3 Track4Run - Run management	10
2.3 User characteristics	10
2.4 Assumptions, dependencies and constraints	11
2.4.1 Text Assumptions	11
2.4.2 Domain Assumptions	12
<b>3 Specific Requirements</b>	<b>13</b>
3.1 External Interface Requirements	13
3.1.1 User Interfaces	13
3.1.2 Hardware and Software Interfaces	19
3.1.3 Communication Interfaces	19
3.2 Scenarios	19
3.3 Functional Requirements	20
3.3.1 Use Case Diagram	23
3.3.2 Use Cases	25
3.4 Sequence Diagram	35
3.5 Performance Requirements	38
3.6 Design Constraints	38
3.6.1 Standards compliance	38
3.6.2 Hardware limitations	38
3.7 Software System Attributes	38
3.7.1 Reliability	38
3.7.2 Availability	39
3.7.3 Security	39
3.7.4 Maintainability	39
3.7.5 Portability	39
<b>4 Formal Analysis Using Alloy</b>	<b>40</b>
4.1 Code	40
4.2 Results	44
4.3 Generated World	44

<b>5</b>	<b>Effort Spent</b>	<b>45</b>
5.0.1	Luca Alessandrelli	45
5.0.2	Andrea Caraffa	46
5.0.3	Andrea Bionda	47
<b>6</b>	<b>Reference Documents</b>	<b>48</b>

# 1 Introduction

## 1.1 Purpose

The following Requirements Analysis and Specification Document examines a possible solution for a specific system-to-be provided by the TrackMe company. Therefore, this document contains the description of the scenarios, the use cases that describe them, and the models describing requirements and specification for the system-to-be.

Data4Help is a location-based health information service-to-be that allows third parties to monitor the location and health status of individuals. The given problem is to design and develop this service and other two services, AutomatedSOS and Track4Run, which exploit the features offered by the first one.

AutomatedSOS is a service-to-be thought to help elderly people. Constantly monitoring the health status of the subscribed customers, this service sends to the user's location an ambulance as soon as the recorded values are anomalous, for example when some health parameters are below certain thresholds.

Finally, Track4Run is a service-to-be that tracks athletes participating in a run. The service, allows organizers to define the path for the run, participants to enroll to the run and spectators to see on a map the position of all the runners during the run.

## 1.2 Scope

### 1.2.1 Goals

- Data4Help

- G.1 Acquire user's position and health status.
- G.2 Provide to third parties user's position and health status.
  - G.2.1 Provide data on demand to non-subscribed third parties.
  - G.2.2 Provide data in real-time to subscribed third parties.
- G.3 Allow third parties two different ways to get user's data.
  - G.3.1 Allow third parties to get data of a single person.
  - G.3.2 Allow third parties to get data of a group of people.
- G.4 Provide data in an anonymous way, to protect user's privacy.

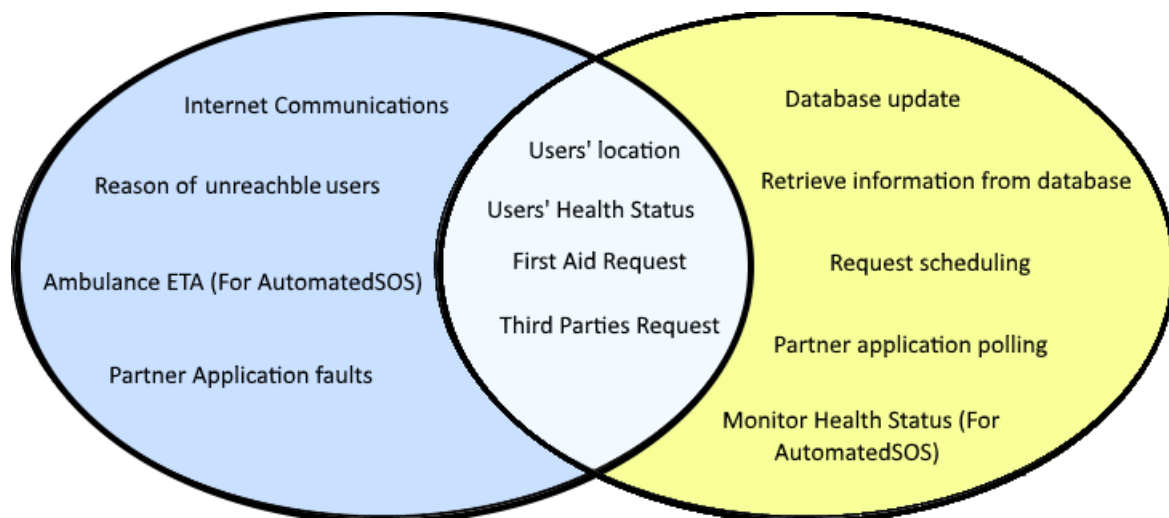
- AutomatedSOS

- G.5 Retrieve user's position and health status.
- G.6 Monitor user's health parameters.
- G.7 Send an ambulance to user's location whenever certain parameters are below the threshold.

- Track4Run

- G.5 Retrieve user's position and health status.
- G.8 Allow promoters to manage a run.
  - G.8.1 Allow promoters to define the path for the run.
  - G.8.2 Allow promoters to invite athletes to the run.
- G.9 Allow athletes to enroll on a specific run.
- G.10 Allow spectators to watch in real time the position of every athlete in a specific run.

### 1.2.2 World Phenomena



## 1.3 Definitions, Acronyms, Abbreviations

### • Definitions

- (a) Single request: request to access to the data of some specific individuals.
- (b) Group request: request to access to anonymized data of group of individuals.
- (c) Live/real-time acquisition/: third parties can access to data as soon they are ready, through service updates.
- (d) On demand acquisition: third parties can access to data when they request them.
- (e) Subscribers: third parties allowed to receive live acquisition about preselected user/group.
- (f) User credentials: information that an individual has to provide to become a registered user: name, surname, date of birth, address, email, telephone number, job, marital status and fiscal code.
- (g) Third parties' credentials: information that a company has to provide to become a registered one: company name, p.iva.
- (h) Race information: all the information about the run such as name, date, promoters, maximum number of participants and race path.
- (i) Partner Application: Application installed on users' device, not necessarily developed by TrackMe, that is in charge with retrieve location and health status.

## 1.4 Revision History

This is a report on all versions of the document along with the reason of the updates/changes.

Version	Changes	Motivation
2.0	Corrected orthographic errors.	/

## 1.5 Document Structure

This document is composed by 6 sections:

**Section 1** gives an introduction to the problem and describes the purposes of the services-to-be provided by TrackMe. The scope of the application is defined by describing the application domain and listing the goals.

**Section 2** presents the overall description of the project. Product perspective subsection presents the class diagram describing the domain model used by all the three services. In addition, that subsection include a state diagram that analyzes the process of making a request to access the users' data. User characteristics subsection list the actors interested in using these services.

**Section 3** specifics the requirements identified, both functional and non functional. The first subsection includes the external interface requirements, showing user interfaces with several mockups. Some scenarios describing specific situations are then listed here. 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.

**Section 4** includes the alloy model and the discussion of its purpose. Also, a world generated by it is shown.

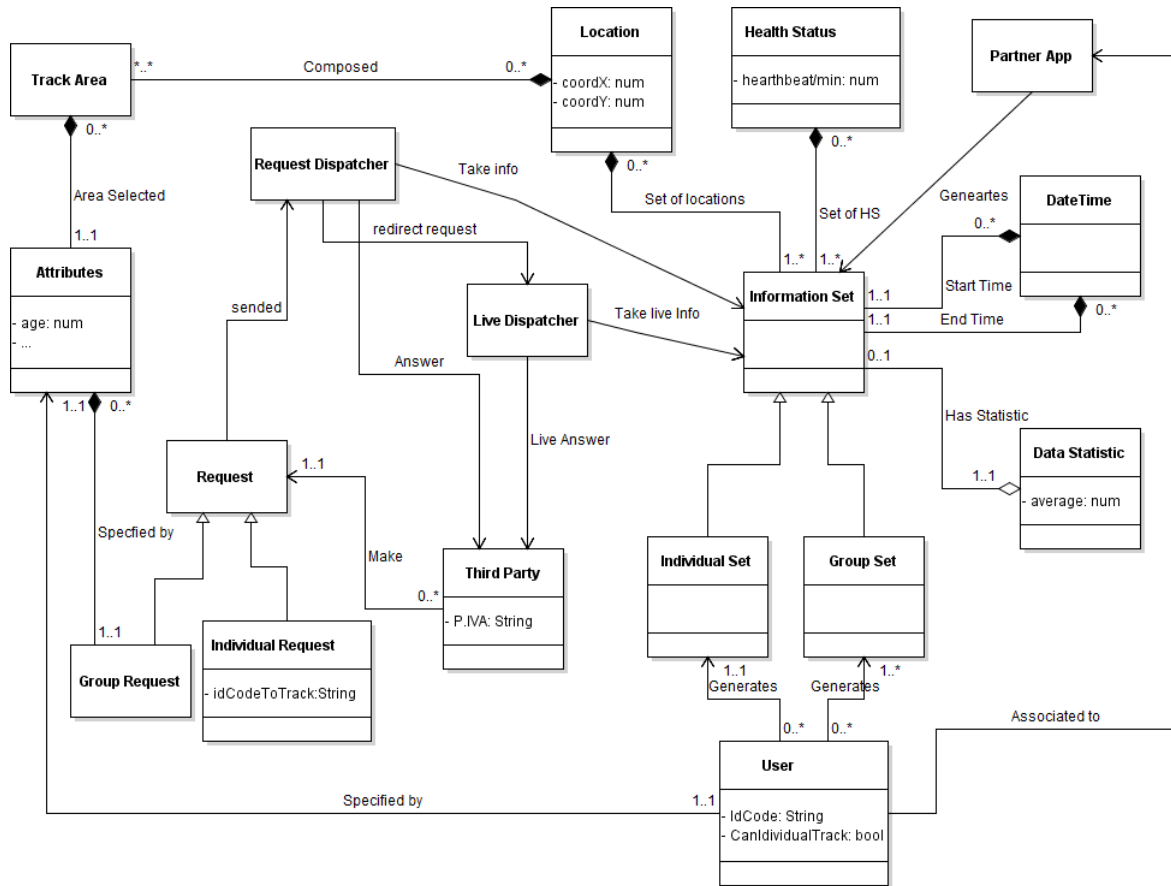
**Section 5** shows the effort spent by each group member while working on this project.

**Section 6** includes the reference documents.

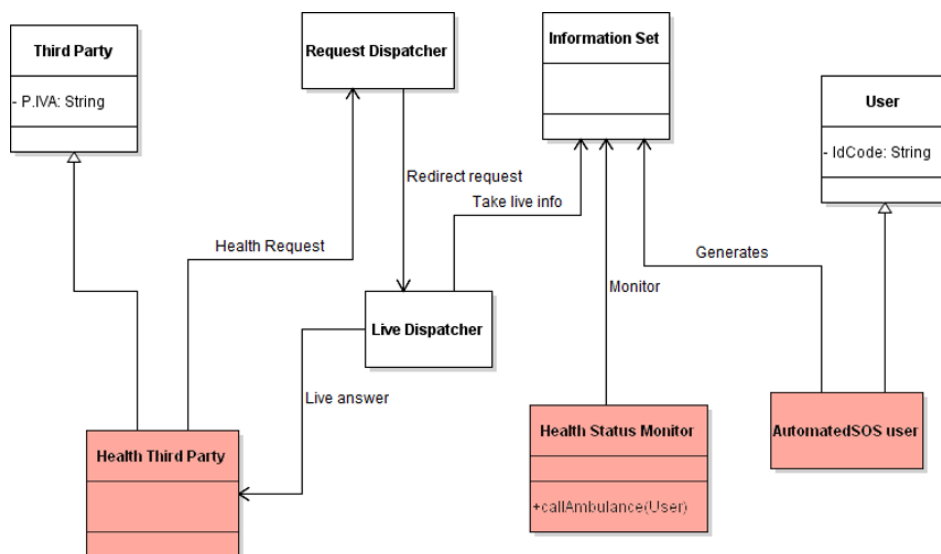
## 2 Overall Description

### 2.1 Product perspective

The following Class Diagrams represent the three services and the domain model where they work. White class are dedicated to perform Data4Help service, the other two applications are listed below.

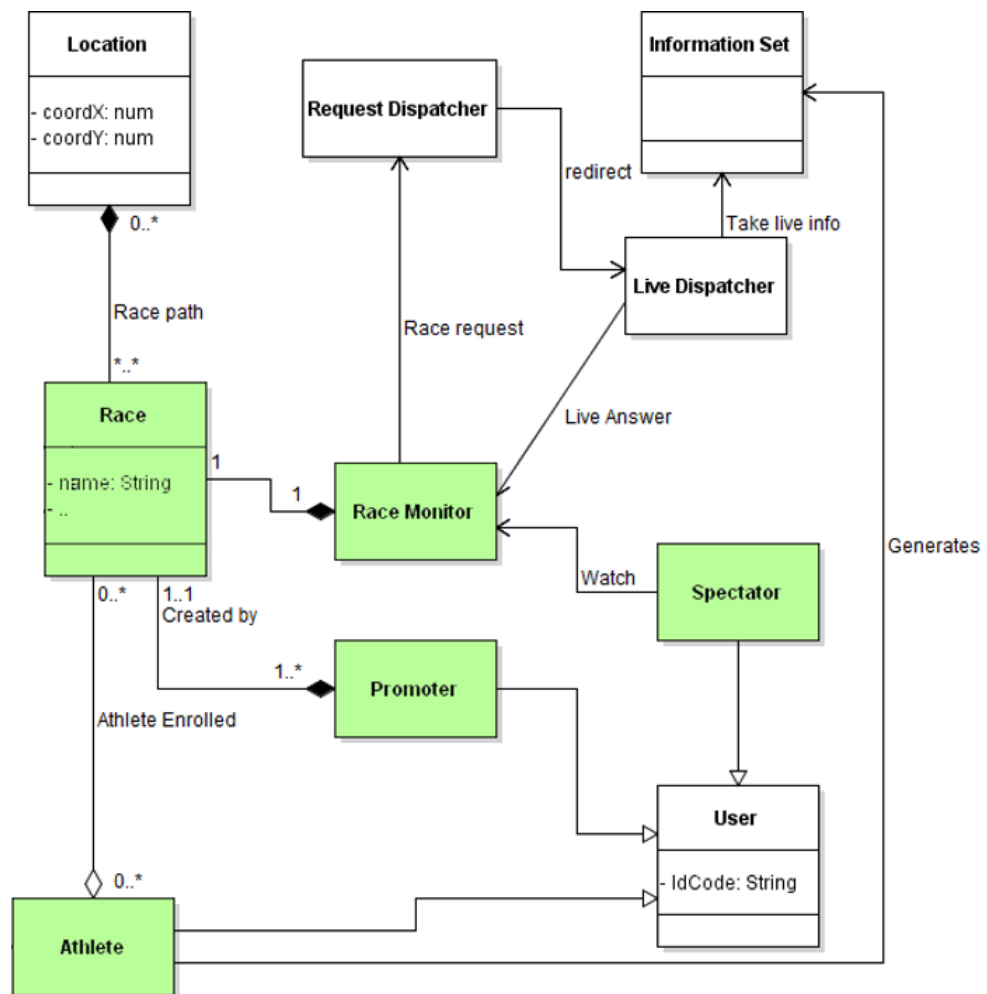


Red classes are dedicated to **AutomatedSOS** application, supported by Data4Help service (white ones).

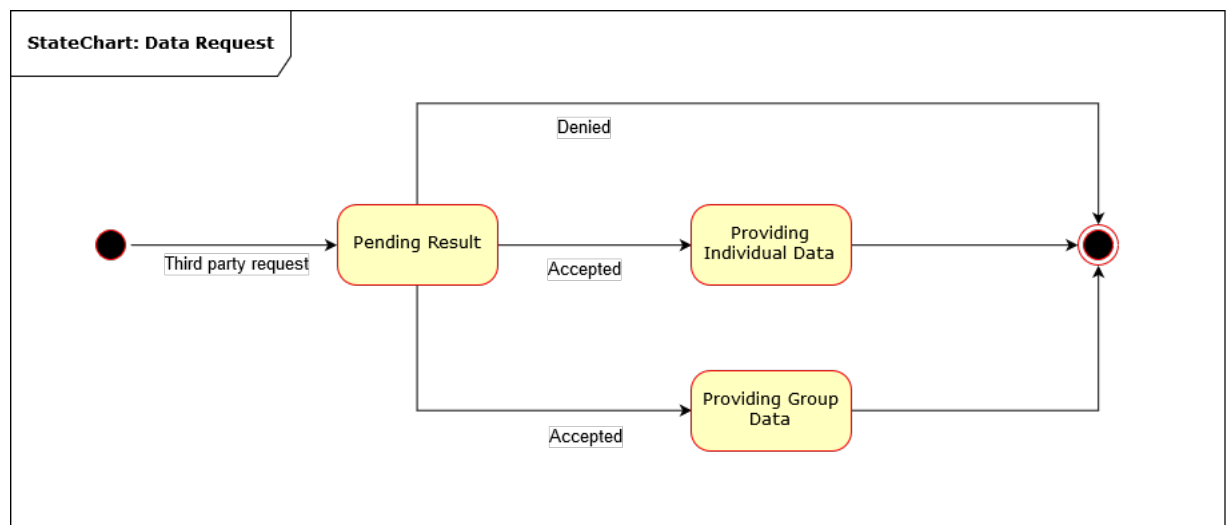




Green classes are dedicated to **Track4Run** application, supported by Data4Help service (white ones).



The following State Chart represents the behaviour of a Data Request Object.



## **2.2 Product functions**

The systems-to-be under analysis have to offer several functions. Below, the main functions provided by each system are more precisely specified, considering all the aspects emerged from the previous list of goals.

### **2.2.1 Data4Help - Providing data to third parties**

This is the core function that Data4Help has to ensure. After collecting users position and health status information from external partner applications, Data4Help provides these data to the third party interested in having them. Data4Help provides data on demand sending to the third party all the available data about an individual (or a group of individual) collected so far. So the third party is provided with all the data about a user collected until now. In addition, Data4Help offers a providing data service in real time, allowing the third party to subscribe to new data and to receive them as soon as they are produced.

### **2.2.2 AutomatedSOS - Sending ambulance request in critical situation**

AutomatedSOS monitors the health status of the subscribed customers and, when such parameters are below certain threshold, sends to the location of the customer an ambulance, guarantying a reaction time of less than 5 seconds from the time parameters are below the threshold.

Therefore, the main function offered by AutomatedSOS is sending an ambulance request, with the relative user position, to the nearest hospital to the user. In order to optimize the times, the ambulance request contains all the data about the user health status. Providing these information, when rescue arrives, it can immediately act accordingly to the received data.

### **2.2.3 Track4Run - Run management**

Track4Run offers three different functionality for its users, which can be all grouped under the 'run management' function. A user can be a promoter, in this case the user can create the event run, which will be visible to every other users. Once created a run, the promoter can define the path in an interactive way, that is by drawing the path directly on a map. Track4Run allows the promoter to set other additional information, like the start time or an overall description of the run. Finally, the promoter can invite to the run all the participants.

The athletes have to be user too. Once received a run request, the athlete can enroll the run or reject it. In the first case, Track4Run tracks in real time the participant position for all the run through a GPS device. Therefore the athlete must wear this device, that for example can be a smartwatch.

A user can also be a simple spectator and see on a map the position of all runners during the run. A spectator is also provided with the main information about the participants and with live time laps.

## **2.3 User characteristics**

1. Third party: Company interested in retrieving useful data from TrackMe's users. Usually, this information can be relevant for marketing strategy.
2. User: Individual whose data are acquired from TrackMe through Data4Help service and are provided to third parties. AutomatedSOS is a service thought for elderly people, while Track4Run is a service thought for athletes, promoters and spectators of runs. User's privacy is protected by each service.

## 2.4 Assumptions, dependencies and constraints

In the specification document certain parts are not specified and a bit ambiguous. So we decided to make the following assumptions.

### 2.4.1 Text Assumptions

- **Data4Help**

- (a) User's data are collected from partner applications or from the other two TrackMe applications installed on user's devices.
- (b) Partner applications can be all the sport assistant apps, gps assistant apps or all the other applications that can retrieve location and health status of individual for such reason.
- (c) All the partner applications require to submit user credentials.
- (d) When the partner application is installed and credentials are submitted, the user is required to accept privacy policy, composed in two parts:
  - i. The first, mandatory, user accept to be tracked in group mode.
  - ii. The second, optional, user accept to be tracked in single mode.
- (e) Individual monitoring requests are not accepted or denied one by one by the specific user. If the user agreed on the treatment of his data as information of an individual (second part of privacy policy) all individual request by third parties are automatically accepted.
- (f) Data are collected from partner application only when they are active on user's device.
- (g) Only third parties that are registered to Data4Help can request the monitoring service.
- (h) Groups are characterized by its members' attributes (age, gender, city, ...).
- (i) Health status parameters that can be acquired are all the ones supported by a standard smart-watch as: Heart Rate, Blood Pressure, Pedometer, Calories Calculation.

- **AutomatedSOS**

- (a) AutomatedSOS exploit only smartwatches devices to retrieve all the information needed.
- (b) AutomatedSOS is an application that needs to be installed into the user's device.
- (c) All data retrieved by AutomatedSOS are sent to Data4Help.
- (d) In order to keep under systematic review the user's health status, all the historical information about the user are received by Data4Help's Database.
- (e) This service can be used only by elderly people (70+) or by who really need it, in order to avoid useless waste of resources.
- (f) User can see all personal information that have been sent to the Data4Help service.

- **Track4Run**

- (a) During the registration to the application the user is asked to accept or deny the treatment of his data by Data4Help service.
- (b) The application has three functions:
  - i. Promoter: allow the user to manage a run.
  - ii. Athlete: allow the user to participate to a run. In order to be an athlete the request of data treatment by the Data4Help service need to be accepted.
  - iii. Spectator: Allow the user to watch in real time the positions of all the athletes in a given run.

- (c) Any user can organize an event.
- (d) All the events can be spectated by users.
- (e) All users invited to a run can accept or discard the request.
- (f) Race path are always composed by citizen routes (never in private circuits or stadiums)

## 2.4.2 Domain Assumptions

### • Data4Help

- D.1 User's information are collected from partner applications or from the other two TrackMe applications installed on users' devices.
- D.2 All the partner applications require to submit user credentials.
- D.3 The identification (fiscal code, social security number) and the secondary data (attributes) given by the individual during the registration are correct.
- D.4 Devices used to monitor individuals always report correct values.
- D.5 Partner application always report correct values to Data4Help.
- D.6 In order to perform an individual request, third parties has to know the user's fiscal code or security number.
- D.7 Security number and fiscal code are not information given to third parties by Data4Help.

### • AutomatedSOS

- D.4 Devices used to monitor individuals always report correct values.
- D.9 The user always dresses a smartwatch on which AutomatedSOS is installed and running.
- D.10 The first aid system is always up and ready to receive messages from AutomatedSOS.
- D.11 The ambulance successfully reach the location of the individual.
- D.12 The ambulance always get to the location in the minimum amount of time.

### • Track4Run

- D.4 Devices used to monitor individuals always report correct values.
- D.13 During a run athletes always wear a smartwatch on which Track4Run is installed.
- D.14 The path defined by the organizer actually exist.
- D.16 If an athlete enroll to a run then athlete also participates to the run.
- D.17 All athletes have their tracking devices with them and the application is enabled for the entire duration of the run.
- D.18 Athletes never go out of the defined path.

## 3 Specific Requirements

### 3.1 External Interface Requirements

#### 3.1.1 User Interfaces

- Data4Help

The third parties interested in having location and health status information of individuals can make the request on the Data4Help's Website. Since the individuals do not need any particular Data4Help's App for their data retrieval, Data4Help does not offer any other user interface besides its Website. On the Website thought, for the third parties it is possible to make both group and individual requests and to view all the data provided to the specific third party. The third parties in order to send a data request must register through the registration function offered by the Website.

The following mockups represent a basic idea of what the Data4Help's Website will look like in the first release.

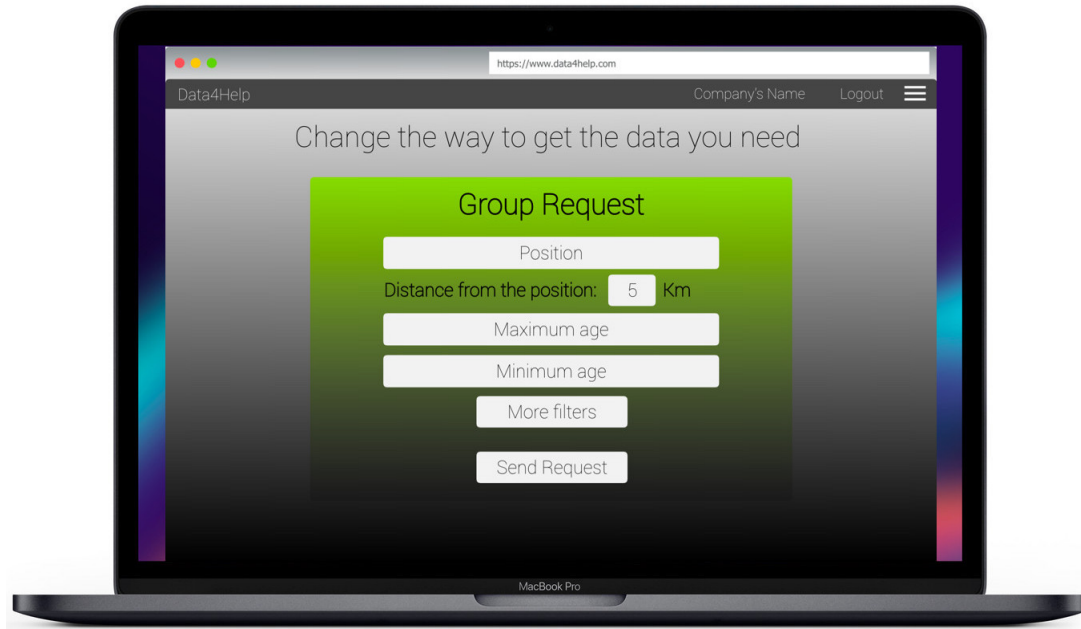


Figure 1: Group monitoring request.



Figure 2: Individual monitoring request.

- AutomatedSOS

TrackMe offers to AutomatedSOS users an App for smartwatches, with which the users can see their location and health status information. No interface is offered to the third parties since they interact exclusively with Data4Help's Interface.

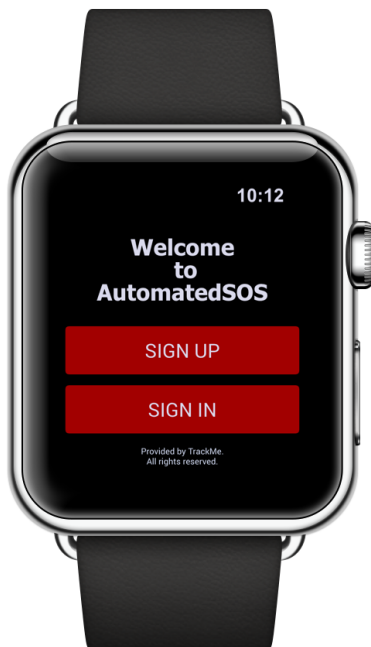


Figure 3: Welcome page that the user see in the first App access. Sign up and Sign in are two possibilities.

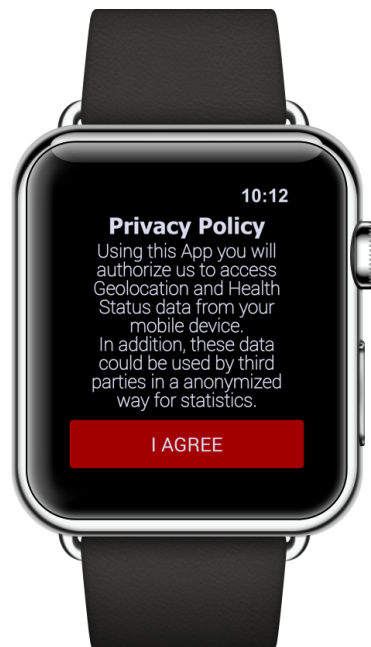


Figure 4: Privacy policy conditions regarding Data4Help's treatment of data and group monitoring request.



Figure 5: Privacy policy conditions regarding individual monitor request, the user can agree to these terms or not.

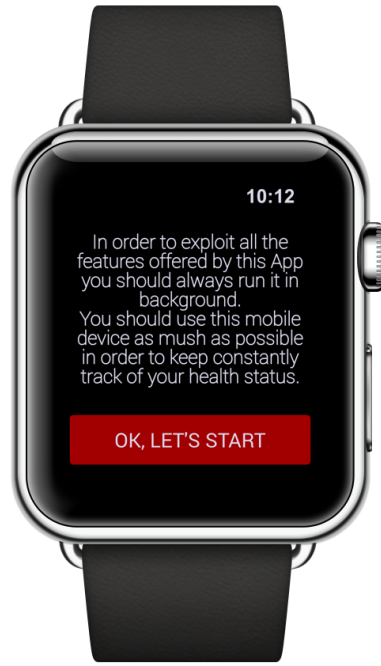


Figure 6: Message that communicates to the user the importance of wearing the smartwatch.



Figure 7: Main menu showing the various functions offered to user.



Figure 8: Fields the user have to fill to complete the registration.



Figure 9: Warning message that communicates to the user that an ambulance has been sent to the user location.

- **Track4Run**

Track4Run users can use an App for smartphone and another one for smartwatches. The first one could be used by everyone, while the second one is made only for the athletes. Like for AutomatedSOS, there is not any interface provided for the third parties.

The mockups showed in the next page represent a basic idea of what the Track4Run's App for smartphone will look like in the first release.



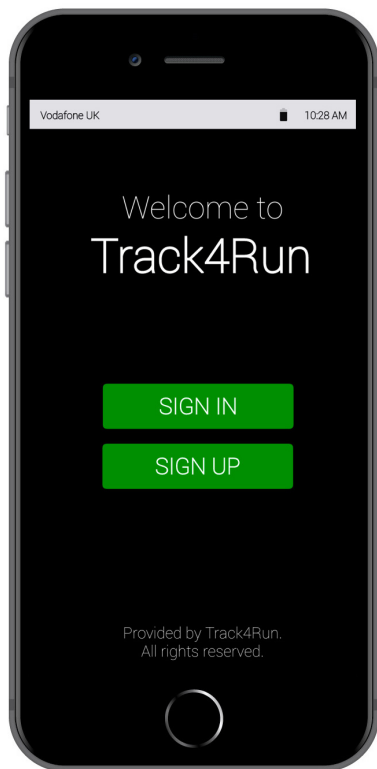


Figure 10: Welcome page.

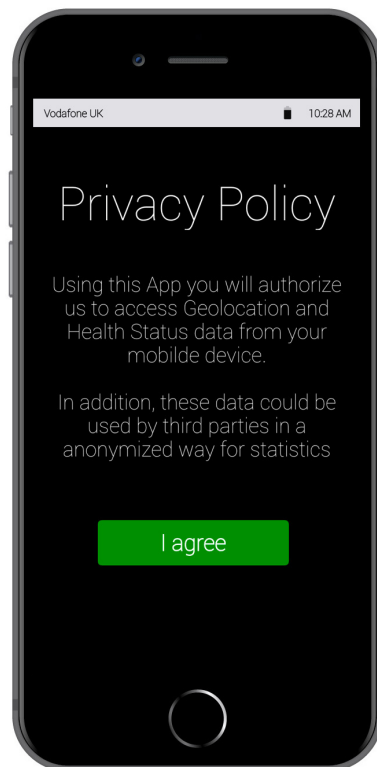


Figure 11: Privacy policy conditions 1.



Figure 12: Privacy policy conditions 2.

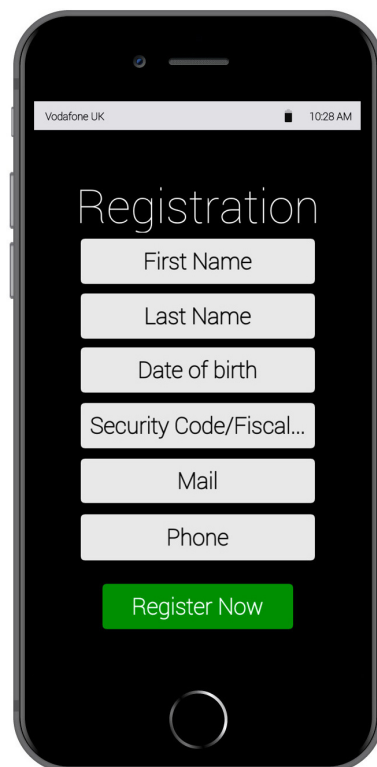


Figure 13: Registration form.

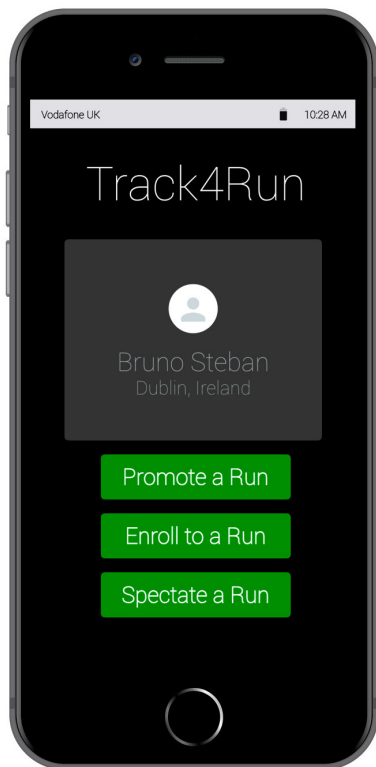


Figure 14: Main user menu.

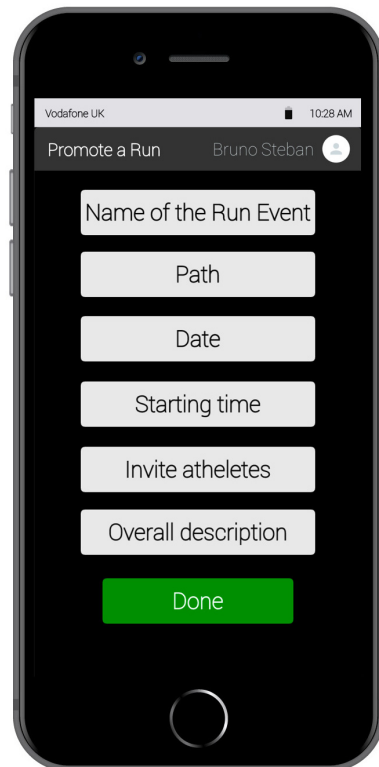


Figure 15: Promote a run view.

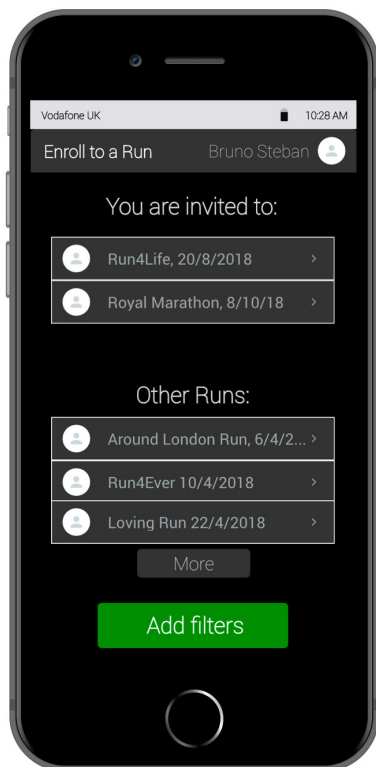


Figure 16: Enroll to a run view.

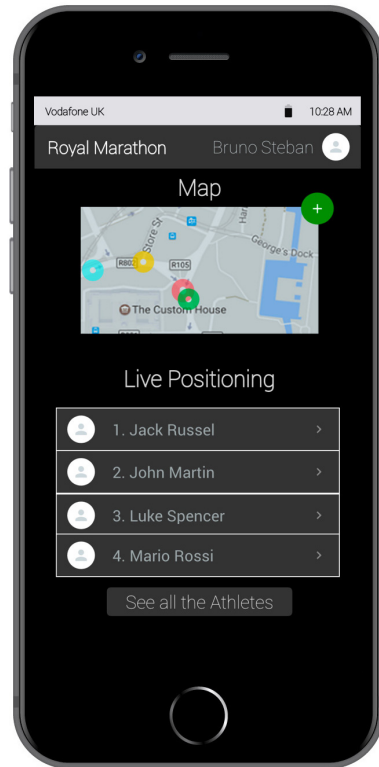


Figure 17: Spectate a run view.

### 3.1.2 Hardware and Software Interfaces

None of the three services-to-be offer any hardware or software interface to the external world. In order to send an ambulance request to the nearest hospital, AutomatedSOS uses a platform provided by the hospital. All the user data needed for the ambulance intervention (such as the location, health status and so on) are sent, for example, by email. For this reason none of the three services under analysis offer any communication interface.

### 3.1.3 Communication Interfaces

## 3.2 Scenarios

- **Scenario 1**

The company StatisticsDispenser is interested into weekly providing public statistics about people living in London. For this reason the company, which is already registered to Data4Help, need to send a group monitoring request. After logging into the Data4Help website, StatisticsDispenser open the group request section. The website loads a new page where the company can filter groups through some attributes regarding his members like the age, the gender, the city and many more. For the specific purpose StatisticsDispenser chooses only to filter people who live in London and people who's age is between 20 and 60. Then, due to the fact that the company need future data, StatisticsDispenser subscribes to the group. From now on every time new data is available the system sends a notification to StatisticsDispencer.

- **Scenario 2**

Mark often goes for a run so he decides to download an app to track his progress. The app he installed on his smartphone is a Partner Application with Data4Help. After he registers to this app he's asked if he wants to give his information to the company TrackMe and also if he wants his data to be treated also as individual data. Right after he accepts the policy he's asked to create a Data4Help account or to link an existing one to the application. Mark never created a Data4Help account so he decides to go through the registration process. He fills all the attributes fields required for the sign up and confirm the registration. Data4Help creates the account and saves all the attributes that Mark filled in. Data4Help is now ready to receive Mark's data from the partner application.

- **Scenario 3**

Bob is 77 years old and lately he's having heart problems. Under his son's advice he decided to use the AutomatedSOS application to receive immediate aid in case of need. One month later Bob doesn't feel okay and his heartbeat value goes below the threshold. AutomatedSOS recognizes that Bob is in a critical health state and quickly sends a report to First Aid system containing useful information like his current location and the reason he's in danger. After the First Aid system received the report, it immediately sends an ambulance to Bob's location and then sends an acknowledge message to AutomatedSOS. The application shows a message on Bob's device to let him know that an ambulance is coming for aid.

- **Scenario 4**

Mario promotes run events for a living so in order to simplify the process he goes through everyday he decides to download Track4Run on his smartphone. The famous company AdiDas designate Mario to promote a run that takes place once a year in Milan. Mario log into the app and enter the "Promote a Run" section, inserts and confirms all the information needed. Track4Run creates the run and makes it available to athletes to enroll in.

- **Scenario 5**

Lately Eddie and his friends are bored of what they usually do so they decided to participate to a

different activity. More precisely they want to create a run event using Track4Run App and see who's the fastest at running. Unfortunately Eddie got ill the day before the event but he's just too curious of seeing who of his friends is going to win. For this reason as soon as the run starts he logs into Track4Run and enters the "Spectate a Run" section and select the run created with his friends. A few moments later the map appears on the app with all the athletes positions on it letting Eddie see how the run is proceeding in real time.

### 3.3 Functional Requirements

- Data4Help

- G.1 Acquire user's position and health status.**

- D.1 User's information are collected from partner applications or from the other two TrackMe applications installed on users' devices.
    - D.2 All the partner applications require to submit user credentials.
    - D.3 The identification (fiscal code, social security number) and the secondary data (attributes) given by the individual during the registration are correct.
    - R.1 Retrieve user credentials inserted into partner application as group attributes.
    - R.2 Allow users already registered in Data4Help world to sign in with their account without providing user credentials again.
    - R.3 Allow individuals to agree the privacy policy (first part) so that they can be tracked in group mode through installed application.
    - R.4 During the registration allow individuals to specify if they are also interested to be tracked in single mode (agree the second part of privacy policy) through installed application.
    - R.5 After the registration, the system provides the password to access the data through provided e-mail.
    - D.4 Devices used to monitor individuals always work and report the correct values.
    - D.5 Partner application always report correct values to Data4Help.
    - R.6 The system has to correctly receive data from partner applications installed on user's device.

- G.2 Provide to third parties, the user's position and health status.**

- R.7 Allow third parties to register to Data4Help service specifying all their credentials.
    - R.8 For each user registered the system has to automatically retrieve and store data from partner applications with a resolution of 10 minutes; independently from the requests reached.
    - R.5 After registration the system provides, through provided e-mail, the password to access the data.

- G.2.1 Provide data on demand to non-subscribed third parties.**

- R.9 The system has to collect all the useful data that match the request.
      - R.10 The system has to send to the third party all the data collected until the moment of the request.

- G.2.2 Provide data in real-time to subscribed third parties.**

- R.11 Allow third parties to subscribe to groups or individuals in order to receive live data.
      - R.12 Provide to subscribed third parties data as soon as they are available by the system.

- G.3 Allow third parties two different ways to get users' data.**

- G.3.1 Allow third parties to get data of a single person.**

- D.6 In order to perform an individual request, third parties has to know the user's fiscal code or security number.
- D.7 Security number and fiscal code are not information given to third parties by Data4Help.
- R.13 Allow third parties to insert the fiscal code of the user he wants to track.
- R.14 Deny third parties to receive single mode information about users that have not accepted the second part of the privacy policy.
- R.15 Collect all the useful data retrieved by Data4Help that are produced by the interested users.
- R.16 Send all the collected information to request applicant.

**G.3.2 Allow third parties to get data of a group of people.**

- R.17 Allow third parties to insert search area and attributes in which they are interested to restrict their field of search.
- R.18 Deny third parties to receive information if the provided information can hurt users' privacy, for this purpose group request under 1000 users involved are rejected.
- R.15 Collect all the useful data retrieved by Data4Help that are produced by the interested users.
- R.16 Send all the collected information to request applicant.

**G.4 Provide data in an anonymous way, to protect users' privacy.**

- R.14 Deny third parties to receive single mode information about users that have not accepted the second part of the privacy policy.
- R.18 Deny third parties to receive information if the provided information can hurt users' privacy, for this purpose group request under 1000 users involved are rejected.

• **AutomatedSOS**

**G.5 Retrieve user's position and health status.**

- R.19 Allow users to be tracked from AutomatedSOS filling up the registration and agreeing to privacy policy.
- D.4 Devices used to monitor individuals always report correct values.
- D.9 The user always dresses a smartwatch on which AutomatedSOS is installed and running.
- R.20 The application has to interact with Smartwatch/Smartphone APIs in order to retrieve location and health status.

**G.6 Monitor user's health parameters.**

- R.20 The application has to interact with Smartwatch APIs in order to retrieve location and health status.
- R.21 The application has to retrieve users' health status every 2 seconds in order to guarantee a reaction time of 5 seconds.
- R.22 The application sends to Data4Help service all the data retrieved in live acquisition.
- R.23 The application gets from Data4Help service all the historical data about the user.

**G.7 Send an ambulance to user's location whenever certain parameters are below the threshold.**

- R.24 The application has to control health status with data retrieved in local to immediately realize whether certain parameters are critical.
- R.25 The application sends an ambulance request to the nearest hospital whenever parameters are critical.
- D.10 The first aid system is always up and ready to receive messages from AutomatedSOS.

- R.26 Supply to the hospital the user's location and all the useful information to provide efficient first aid.
- R.27 In the case no answer arrives from the hospital the software must repeat another time the request until an answer is reached.
- D.11 The ambulance successfully reach the location of the individual.

- **Track4Run**

**G.5 Retrieve user's position and health status.**

- R.28 Allow users to be tracked from Track4Run filling up the registration and agreeing to privacy policy.
- D.4 Devices used to monitor individuals always report correct values.
- R.20 The application has to interact with Smartwatch/Smartphone APIs in order to retrieve location and health status.

**G.8 Allow user to manage a run.**

- R.29 Allow users to create a run providing all the general information about the competition.
- D.4 Devices used to monitor individuals always report correct values.

**G.8.1 Allow promoters to define a path for the run.**

- R.30 Allow promoters to define a path for the run by selecting the routes inside a map.
- D.14 The path defined by the organizer actually exist.

**G.8.2 Allow promoters to invite athletes to the run.**

- R.31 Allow promoters to send a participation request.

**G.9 Allow athlete to enroll on a specific run.**

- R.32 Allow the user to see all the runs he/she has been invited to.
- R.33 Allow the user to make a participating request for a run he/she has not been invited to.
- D.16 If an athlete enroll to a run then athlete also participates to the run.

**G.10 Allow spectators to watch in real time the position of every athletes in a specific run.**

- D.17 All athletes have their tracking devices with them and the application is enabled for the entire duration of the run.
- R.34 Allow user to select a run to be viewed.
- R.35 The application requests to Data4Help the position of all the other athletes involved.
- R.36 The application receives and displays the position of all the other athletes involved.
- D.18 Athletes never go out of the defined path.
- D.13 During a run athletes always wear a smartwatch on which Track4Run is installed.

### 3.3.1 Use Case Diagram

- Data4Help



- AutomatedSOS



- Track4Run





### 3.3.2 Use Cases

- Data4Help Use Cases

Name	Sign Up
Actors	Third Party
Entry Conditions	TRUE
Event Flow	<ul style="list-style-type: none"><li>(a) The Third Party enters the sign up section.</li><li>(b) The system shows to the third party all the attributes fields needed for the registration.</li><li>(c) The Third Party fills all the attribute fields.</li><li>(d) The Third Party confirms all the attributes inserted stating he wants to register.</li><li>(e) The system creates and saves the third party's account.</li></ul>
Exit Condition	The third party's account has been created and the third party is now registered.
Exceptions	<ul style="list-style-type: none"><li>• If the system notices that the attributes used in the registration are already linked to an existing account then a warning is generated saying that there is already a third party registered with the given attributes.</li></ul>

Name	Sign In
Actors	Third Party
Entry Conditions	TRUE
Event Flow	<ul style="list-style-type: none"><li>(a) The Third Party enters the sign in section.</li><li>(b) The system shows to the third party all the credentials fields needed for the log in.</li><li>(c) The Third Party fills all credentials fields and confirms he wants to log in.</li><li>(d) The system accepts the log in request.</li></ul>
Exit Condition	The third party is now logged in.
Exceptions	<ul style="list-style-type: none"><li>• If the third party inserts invalid log in credentials a warning is generated saying the credentials are invalid.</li></ul>

Name	Request Individual Monitoring
Actors	Third Party
Entry Conditions	The third party is logged in.
Event Flow	<ul style="list-style-type: none"> <li>(a) The Third Party enters the Individual Request section.</li> <li>(b) The system shows to the third party all the information fields needed for the identification of the individual.</li> <li>(c) The Third Party fills all the attribute fields and confirms he wants to track that specific individual.</li> <li>(d) The system shows all the individual's information that have been collected until the moment of the request.</li> </ul>
Exit Condition	The request's outcome is shown to the third party.
Exceptions	<ul style="list-style-type: none"> <li>• If the inserted attributes are not linked to any user account then a warning message is displayed saying that the individual is not registered.</li> <li>• If the individual that correspond to the attributes inserted didn't accept the individual treatment of data policy then a warning message is displayed saying that the request is rejected.</li> </ul>

Name	Request Group Monitoring
Actors	Third Party
Entry Conditions	The third party is logged in.
Event Flow	<ul style="list-style-type: none"> <li>(a) The Third Party enters the Group Request section.</li> <li>(b) The system shows to the third party all the information fields needed for defining the group.</li> <li>(c) The Third Party insert the search area.</li> <li>(d) The Third Party inserts all the attributes.</li> <li>(e) The system accepts the request.</li> <li>(f) The system shows all the group's information that have been collected until the moment of the request.</li> </ul>
Exit Condition	The request's outcome is shown.
Exceptions	<ul style="list-style-type: none"> <li>• If the group request get rejected by the system a warning message will be displayed saying the request is rejected.</li> </ul>
Special Requirements	The system rejects group monitoring requests when the group's information can compromise users' privacy. For this purpose requests of groups composed by less than 1000 users get rejected.

Name	Subscribe To A Group
Actors	Third Party
Entry Conditions	The third party has just sent an accepted monitoring request to a group.
Event Flow	<ul style="list-style-type: none"> <li>(a) The Third Party requests to follow the selected group.</li> <li>(b) The system links the third party to the group.</li> <li>(c) The system sends new data to the third party.</li> </ul>
Exit Condition	The Third Party is subscribed to the selected group.
Exceptions	<ul style="list-style-type: none"> <li>• If the third party is already subscribed a warning message is shown saying the subscription have been already done.</li> </ul>

Name	Sign Up From Partner App
Actors	User, Partner Application
Entry Conditions	The user accepted the treatment of data policy.
Event Flow	<ul style="list-style-type: none"> <li>(a) The user starts the sign up function on the partner app.</li> <li>(b) The Partner Application shows to the user all the attributes fields needed for the registration.</li> <li>(c) The User fills all the attribute fields.</li> <li>(d) The Partner Application sends to the system the attributes inserted by the user.</li> <li>(e) The system receives by the partner application all the attributes inserted by the user.</li> <li>(f) The system creates the user's account and saves the received data.</li> </ul>
Exit Condition	The system registered the user.
Exceptions	<ul style="list-style-type: none"> <li>• If the system notices that attributes used in the registration are already linked to an existing account then a message is sent back to the partner application in order to let the user know what happened.</li> </ul>

Name	Link Account To The Partner App
Actors	User, Partner Application
Entry Conditions	The user accepted the treatment of data policy and already has an existing account to link to the partner application.
Event Flow	<ul style="list-style-type: none"> <li>(a) The user starts the account linking function on the partner app.</li> <li>(b) The Partner Application shows to the user all the credential fields needed for the linking process.</li> <li>(c) The User fills all the credential fields.</li> <li>(d) The Partner Application sends to the system the credentials inserted by the user.</li> <li>(e) The system receives by the partner application all the credentials inserted by the user.</li> <li>(f) The system sends back to the partner application the outcome of the operation.</li> </ul>
Exit Condition	The system registered the user.
Exceptions	<ul style="list-style-type: none"> <li>• If the system notices that the credentials received are not linked to an existing account then a message is sent back to the partner application in order to let the user know what happened.</li> </ul>

Name	Send User's Data
Actors	User, Partner Application
Entry Conditions	The user accepted the treatment of data policy and the partner application is running on the user's device.
Event Flow	<ul style="list-style-type: none"> <li>(a) The Partner Application collects user's data.</li> <li>(b) The Partner Application sends the user's data to the the system.</li> <li>(c) The system receives and saves the user's data sent by the partner application.</li> </ul>
Exit Condition	The system saved the user's data.
Exceptions	None

## • AutomatedSOS Use Cases

Name	Sign Up
Actors	User
Entry Conditions	The User has AutomatedSOS installed on his smartwatch.
Event Flow	<ul style="list-style-type: none"> <li>(a) The User enters the sign up section of the app.</li> <li>(b) The User accepts the treatment of data policy.</li> <li>(c) The system shows to the user all the attributes fields needed for the registration.</li> <li>(d) The User fills all the attribute fields.</li> <li>(e) The User confirms all the attributes inserted stating he wants to register.</li> <li>(f) The system creates and saves the user's account.</li> </ul>
Exit Condition	The user's account has been created and the user is now registered.
Exceptions	<ul style="list-style-type: none"> <li>• If the User does not accept the treatment of data policy then a warning is generated saying that ,in order to register, the policy must be accepted.</li> <li>• If the system notices that attributes used in the registration are already linked to an existing account then a warning is generated saying that there is already an individual registered with the given credentials.</li> </ul>

Name	Sign In
Actors	User
Entry Conditions	The User has AutomatedSOS application installed on his smartwatch.
Event Flow	<p>(a) The User enters the sign in section of the app.</p> <p>(b) The system shows to the user all the credentials fields needed for the log in.</p> <p>(c) The User fills all credentials fields and confirms he wants to log in.</p> <p>(d) The system accepts the log in request.</p>
Exit Condition	The User user is now logged in.
Exceptions	<ul style="list-style-type: none"> <li>• If the user inserts invalid log in credentials a warning is generated saying the credentials are invalid.</li> </ul>

Name	See Acquired Data
Actors	User
Entry Conditions	The User is logged in.
Event Flow	<p>(a) The User enters the Acquired Info section of the app.</p> <p>(b) The system gets all the user's information that have been retrieved by the application until that moment.</p> <p>(c) The system displays the user's information.</p>
Exit Condition	All the information retrieved by the system are shown on the app.
Exceptions	<ul style="list-style-type: none"> <li>• If the system do not find information about the user then a warning message is shown to the user saying that until now the application did not record any information.</li> </ul>

Name	Set Preferences
Actors	User
Entry Conditions	The User is logged in.
Event Flow	<ul style="list-style-type: none"> <li>(a) The User enters the Preferences section of the application.</li> <li>(b) The User can add or remove certain health parameters in order to personalize the monitoring profile.</li> <li>(c) The User can also change certain parameters threshold.</li> <li>(d) The User confirms all the changes done.</li> <li>(e) The system saves all the changes made by the user.</li> </ul>
Exit Condition	The parameters are correctly updated as the user wants them to be.
Exceptions	<ul style="list-style-type: none"> <li>• If the user does not confirm the changes then all parameters remain the same as before.</li> </ul>

Name	Send Ambulance Request
Actors	AutomatedSOS, First Aid
Entry Conditions	A critical health parameter value is below the threshold.
Event Flow	<ul style="list-style-type: none"> <li>(a) AutomatedSOS sends to First Aid a report that contains all important information about the user like his current location, his gender, his age, his health profile, and the list of parameters that got below the threshold.</li> <li>(b) First Aid immediately sends an ambulance to the user's location.</li> <li>(c) First Aid sends an acknowledge message to AutomatedSOS.</li> <li>(d) AutomatedSOS displays on the app a warning message saying that an ambulance is currently heading to the user's location.</li> </ul>
Exit Condition	A warning message is shown saying that an ambulance is currently heading to the user's location.
Exceptions	<ul style="list-style-type: none"> <li>• If no acknowledge message is received by AutomatedSOS after the form has been sent, as soon as a certain time out expires AutomatedSOS re-send the form with updated information.</li> </ul>
Special Requirements	The form need to be sent to First Aid with a reaction time of less than 5 seconds from the time the parameters are below the threshold.

## • Track4Run Use Cases

Name	Sign Up
Actors	User
Entry Conditions	The User has Track4Run application installed on his device.
Event Flow	<ul style="list-style-type: none"> <li>(a) The User enters the sign up section of the app.</li> <li>(b) The User accepts the treatment of data policy.</li> <li>(c) The system shows to the user all the attributes fields needed for the registration.</li> <li>(d) The User fills all the attribute fields.</li> <li>(e) The User confirms all the attributes inserted stating he wants to register.</li> <li>(f) The system creates and saves the user's account.</li> </ul>
Exit Condition	The user's account has been created and the user is now registered.
Exceptions	<ul style="list-style-type: none"> <li>• If the User does not accept the treatment of data policy then a warning is generated saying that ,in order to register, the policy must be accepted.</li> <li>• If the system notices that attributes used in the registration are already linked to an existing account then a warning is generated saying that there is already an individual registered with the given credentials.</li> </ul>



Name	Sign In
Actors	User
Entry Conditions	The User has Track4Run application installed on his smartwatch.
Event Flow	<ul style="list-style-type: none"> <li>(a) The User enters the sign in section of the app.</li> <li>(b) The system shows to the user all the credentials fields needed for the log in.</li> <li>(c) The User fills all credentials fields and confirms he wants to log in.</li> <li>(d) The system accepts the log in in request.</li> </ul>
Exit Condition	The User user is now logged in.
Exceptions	<ul style="list-style-type: none"> <li>• If the user inserts invalid log in credentials a warning is generated saying the credentials are invalid.</li> </ul>

Name	Promote A Run
Actors	User
Entry Conditions	The User is logged in.
Event Flow	<ul style="list-style-type: none"> <li>(a) The User enters the Promote a Run section of the app.</li> <li>(b) The system shows to the user a new tab where the user can define all the important information about the run and also invite athletes.</li> <li>(c) The system creates and saves the run's information.</li> <li>(d) The system automatically sends notifications to all athletes specified by the promoter asking them if they want to participate to the run.</li> </ul>
Exit Condition	The run event has been created and added to the list of promoted runs.
Exceptions	<ul style="list-style-type: none"> <li>• If the user does not insert critical information (like the path, the name or the date) a warning message is shown saying that critical parameters are missing.</li> </ul>

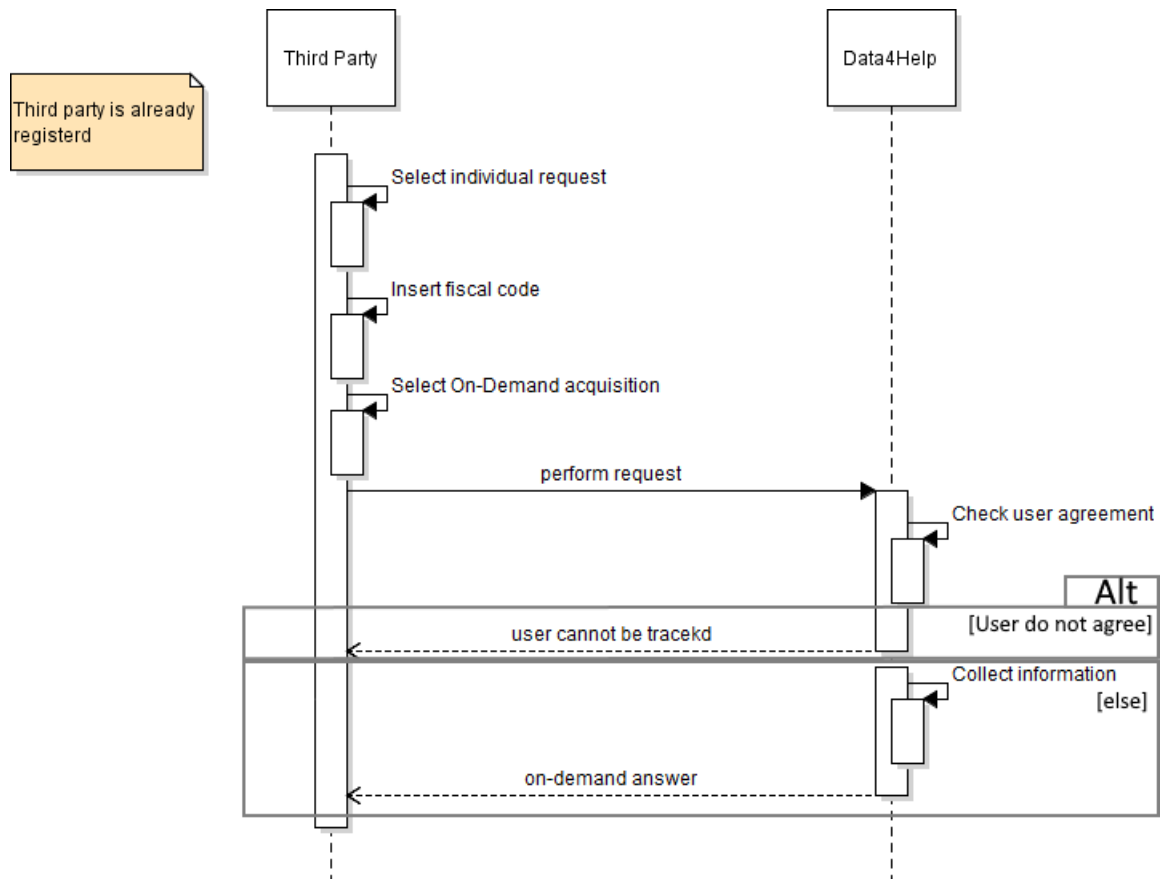
Name	Enroll To A Run
Actors	User
Entry Conditions	The User is logged in.
Event Flow	<ul style="list-style-type: none"> <li>(a) The User enters to the "Enroll to a Run" section of the app.</li> <li>(b) The system shows to the user the list of all the created runs that will take place in the future.</li> <li>(c) The User can filter the runs with some attributes.</li> <li>(d) The User chooses the run he wants to participate to.</li> <li>(e) The system adds the user to the list of athletes enrolled to the run.</li> </ul>
Exit Condition	The user is now enrolled to the run.
Exceptions	<ul style="list-style-type: none"> <li>• If the number of athletes has already capped the max amount in the run chosen by the user then a warning message is displayed saying that no more athletes are allowed to participate to the run.</li> </ul>

Name	Spectate A Run
Actors	User
Entry Conditions	The User is logged in.
Event Flow	<ul style="list-style-type: none"> <li>(a) The User enters the Spectate a Run section of the app.</li> <li>(b) The system shows a new tab where the list of all live runs is visible.</li> <li>(c) The User can filter the runs with some attributes.</li> <li>(d) The User chooses the run he wants to spectate.</li> <li>(e) The system shows to the user the map of the run and also the position of all athletes in real time.</li> </ul>
Exit Condition	The system is showing to the user the map and the athletes positions.
Exceptions	None.

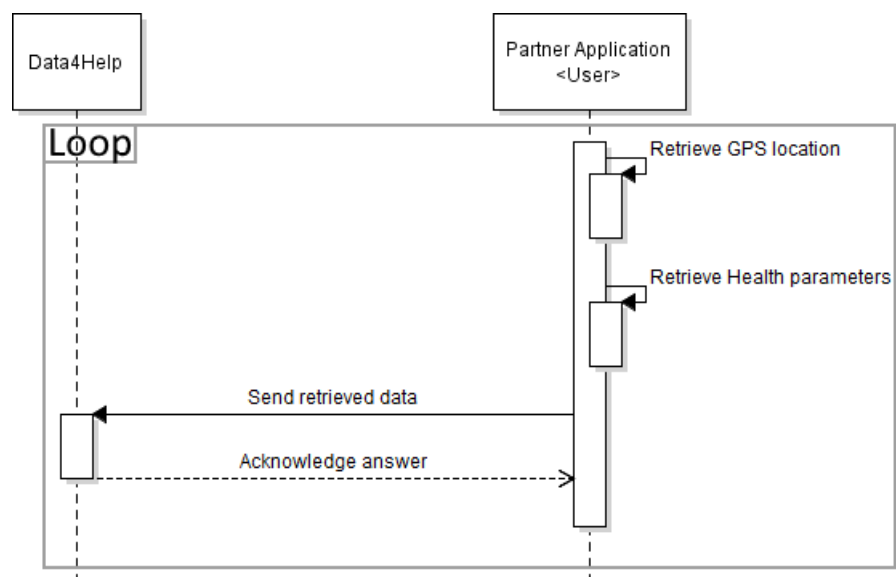
## 3.4 Sequence Diagram

- Data4Help

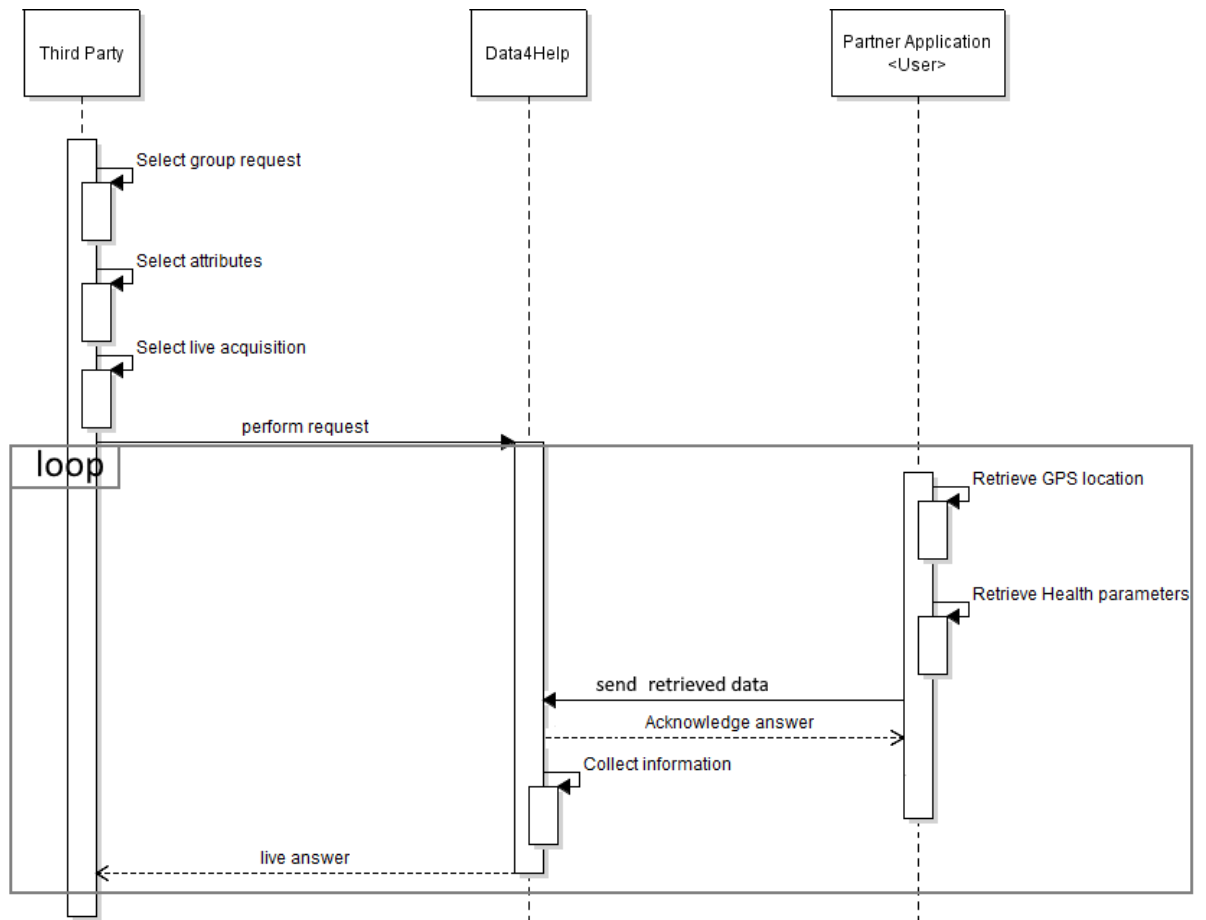
Individual request with on-demand acquisition performance.



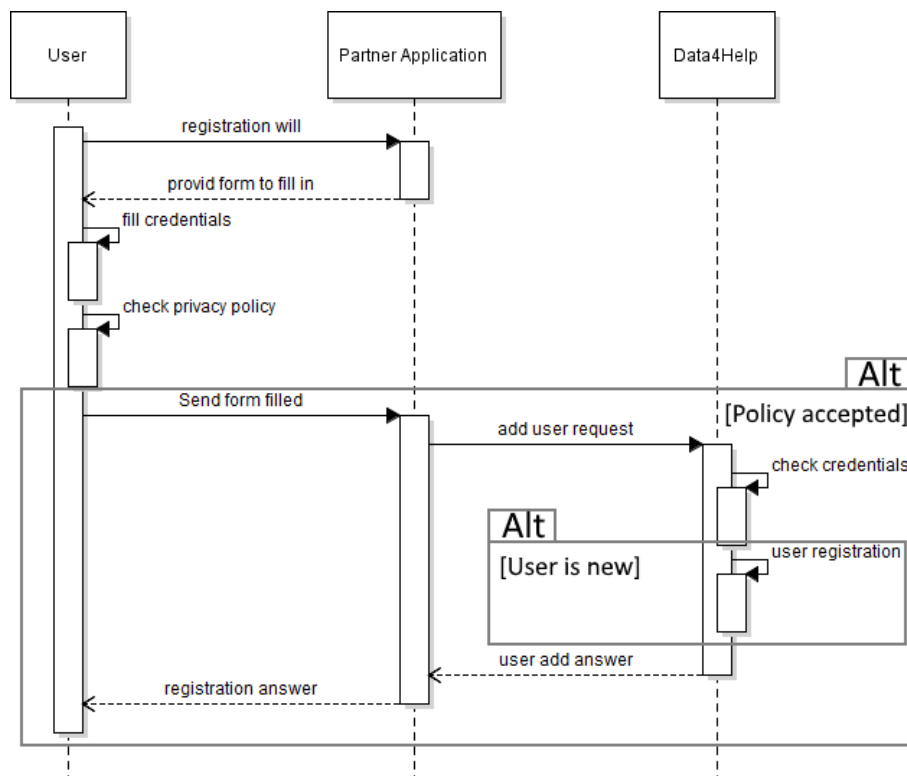
Automatic data update inside Data4Help.



Group request with live acquisition performance.

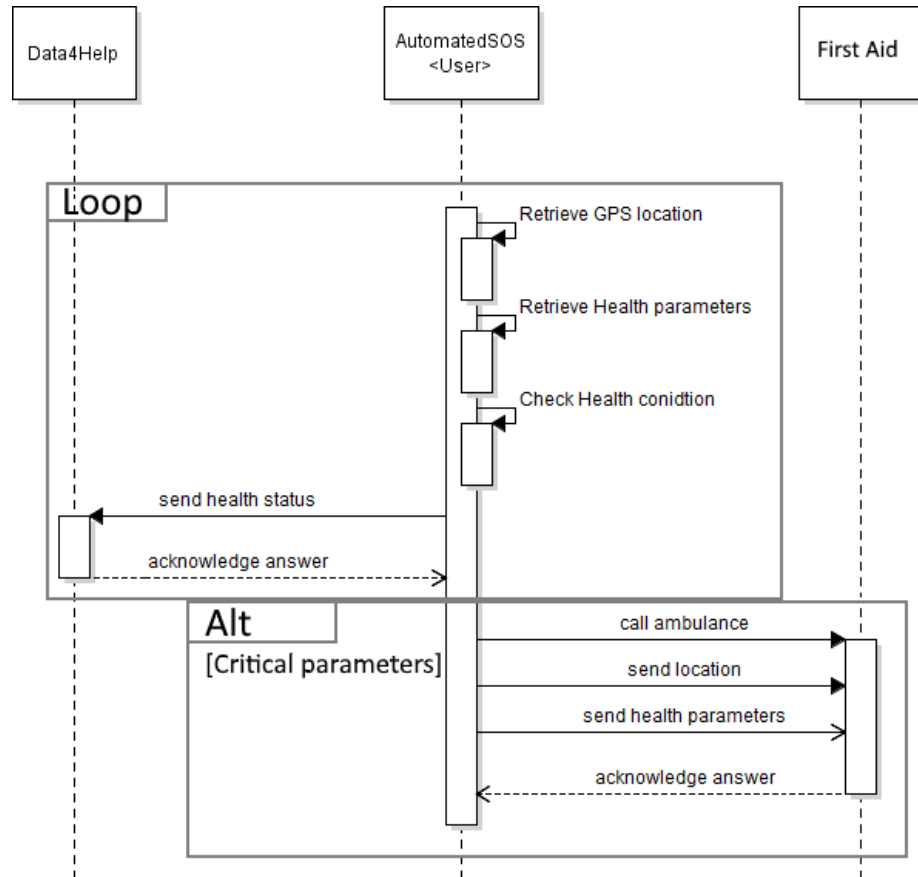


User registration performance.



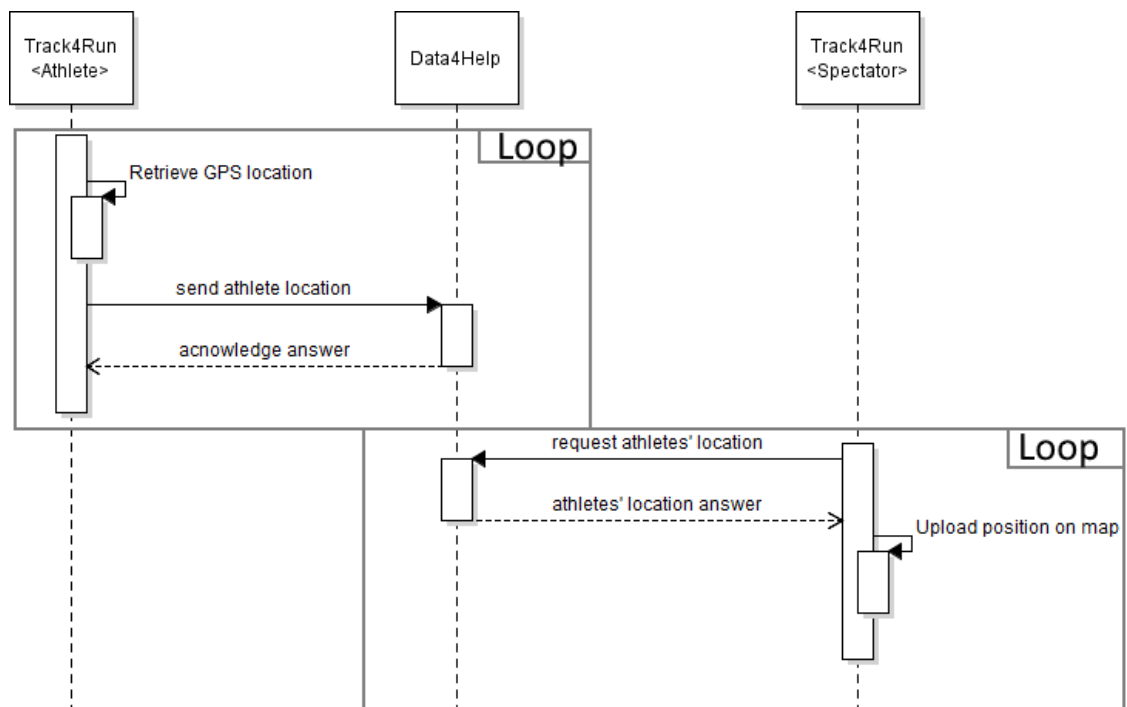
- AutomatedSOS

AutomatedSOS monitor and ambulance caller services.



- Track4Run

Track4Run automated retrieve athletes' position and update spectators' live map.



## 3.5 Performance Requirements

- Data4Help service is build to perform trend research from users that download specific partner applications. In order to perform this type of monitoring there is a high use of resources, especially in live acquisition when data must be exchanged within 2 minutes. Therefore , in its first release, this application is developed to track 10.000 users simultaneously (included users from AutomateSOS and Track4Run). Since third parties are very few in comparison to users, to serve them a less performance is required.
- AutomatedSOS is a very expensive application in terms of performance because it must monitor the individuals all the day long guarantying a data collection with an interval of 2 seconds.
- Track4Run is a variable expensive application because it has to monitor constantly during the run events (even not all day) the athletes which are participating to them.

## 3.6 Design Constraints

### 3.6.1 Standards compliance

- Partner applications request the permission to retrieve location and health status to the device, same for AutomateSOS and Track4Run.
- Data4Help requires that partner applications can use internet connection and users' mobile data to exchange information.
- AutomatedSOS requires all day internet connection in order to call an ambulance every time is needed.
- AutomatedSOS requires internet connection during all the duration of the race to the athletes and to the spectator.

### 3.6.2 Hardware limitations

- AutomatedSOS application needs to be installed on a smartwatch (smartphone is not enough) in order to acquire location and health status.
- Track4Run application requires to be installed on a smartphone or a smartwatch in order to acquire location.
- Smartphones and smartwatches must be iOS or Android platform.
- The devices must have internet connection (mobile data are mandatory).
- The devices must have GPS locator.
- Smartwatches must have Heart Rate monitor, Blood Pressure monitor, Pedometer, Calories Calculator.

## 3.7 Software System Attributes

### 3.7.1 Reliability

Data4Help service,clearly has some moments with less load (for example at night) but it's important to guarantee a 24/7 service. Some small concessions are possible during the night. In order to guaran-

tee AutomatedSOS monitoring this service must be available 24/7, in this case also during the night. Track4Run service has the same consideration of Data4Help.

### **3.7.2 Availability**

Considering that the main core of the service are data, a high level of data redundancy is necessary in order to guarantee an optimal degree of availability. This system is expected to be available 99.99% of the time.

### **3.7.3 Security**

Providing data in an anonymous way is one of the goals of the system. In order to guarantee users' privacy, the system implements certificated security communication protocol and advanced cryptography techniques. In any case users can read and agree privacy policy first.

### **3.7.4 Maintainability**

In order to guarantee maintainability the entire software project is based on Data4Help primitives (like data request, exchange and classification) that must be developed with accuracy and must be certificated. By using or extending fundamental primitives it is possible to construct incremental and interchangeable blocks that can be used to perform all the other services requested.

### **3.7.5 Portability**

Data4Help service can be reached by third parties by http requests, this way every browser can perform requests and retrieve users' data. AutomatedSOS and Track4Run applications are developed as multi-platform technology so either iOS or Android devices can run these two Apps. AutomatedSOS is developed only for Smartwatch.

## 4 Formal Analysis Using Alloy

In this section the Alloy model is represented. The main and most important features and constraints of each TrackMe module has been modeled. More precisely the correctness of goal 4 has been checked for Dat4Help ("Provide data in an anonymous way, to protect user's privacy"), for AutomatedSOS the goal 7 is checked ("Send an ambulance to user's location whenever certain parameters are below the threshold") and furthermore for Track4Run the goal 10 is checked ("Allow spectators to watch in real time the position of every athlete in a specific run").

In order to see whether the model satisfy Data4Help's privacy constraint, the minimum number of group members has been dropped from 1000 to 3 so that the complexity of tests decreases without changing the model's logic.

### 4.1 Code

```
open util/integer

sig Bool {}
one sig True extends Bool {}
one sig False extends Bool {}

----- DATA4HELP SIGNATURES -----

sig FiscalCode {}

sig Location
{
    time: one Time
}

sig Time {}

sig HealthStatus
{
    time: one Time,
    parameter: some Parameter
}

sig AcquisitionSetData
{
    locationAcquisition: some Location,
    healthStatusAcquisition: some HealthStatus
}

sig IndividualPrivacyPolicy
{
    IndividualMonitoring: one Bool --Full privacy policy is accepted (individual
    ↪ monitoring)
}

sig UserAttributes
{
    address: one Location, --Location where user lives
}

sig GroupAttributes
{
    area: some Location, --Area where users live
}

sig User
{
    fiscalCode : one FiscalCode,
    policy: one IndividualPrivacyPolicy, --Status of policy acceptance
    credentials: one UserAttributes, --User's credentials
    retrievedData: some AcquisitionSetData --User's set of acquisition acquired
```



```

}

sig ThirdParty{}

abstract sig InformationRequest
{
    partyApplicant: one ThirdParty, --Third party applicant
}

sig IndividualRequest extends InformationRequest
{
    fiscalCode: one FiscalCode --Tracked User's fiscal code (lone beacuse group mode
    ↪ don't have it)
}

sig GroupRequest extends InformationRequest
{
    groupAttributes: one GroupAttributes, --Users' attributes on group search (lone
    ↪ beacuse individual mode don't have it)
}

abstract sig InformationAnswer{}

sig IndividualInformationAnswer extends InformationAnswer
{
    individualRequest: one IndividualRequest,
    user: one User,
    acquisitionData: one AcquisitionSetData
} {user.policy.IndividualMonitoring = True}

sig GroupInformationAnswer extends InformationAnswer
{
    groupRequest: one GroupRequest,
    acquisitionData: some AcquisitionSetData
} {#acquisitionData > 3} -- 3 stands for 1000

----- AUTOMATEDSOS SIGNATURES -----

sig Parameter
{
    value : one Int,
    threshold : one Int
}

sig FirstAid {}

sig Report {
    user: one User,
    receiver: one FirstAid
}

sig AmbulanceRequest
{
    time: one Time,
    report : one Report
}

----- TRACK4RUN SIGNATURES -----

sig Map
{
    athletesLocation: some Location,
    runEvent : one Run
}

sig Run
{
    athletes: some Track4RunUser,
    map: lone Map
}

```

```

sig Track4RunUser extends User {
    watchingRun: lone Run
}

----- DATA4HELP FATCTS -----

fact HealthStatusInAcquisitionSetData
{
    all h: HealthStatus | some a: AcquisitionSetData | h in a.healthStatusAcquisition
}
fact PrivacyPolicyInUser
{
    all p: IndividualPrivacyPolicy | one u: User | p in u.policy
}
fact UserAttInUser
{
    all ua: UserAttributes | one u: User | ua in u.credentials
}
fact GroupAttributesInGroupRequest
{
    all g: GroupAttributes | one ga: GroupRequest | g in ga.groupAttributes
}
fact ThirdPartyInRequest
{
    all t: ThirdParty | some r: InformationRequest | t in r.partyApplicant
}
fact AcquisitionDataInUser
{
    all a: AcquisitionSetData | one u: User | a in u.retrievedData
}
fact NoTwoAnswerForSameIndividualRequest
{
    all disj i1, i2: IndividualInformationAnswer | i1.individualRequest ≠ i2.
    ↪ individualRequest
}
fact NoTwoAnswerForSameGroupRequest
{
    all disj i1, i2: GroupInformationAnswer | i1.groupRequest ≠ i2.groupRequest
}
fact NoTwoEqualFiscalCode
{
    all disj u1, u2 : User | u1.fiscalCode ≠ u2.fiscalCode
}
fact AnswerToRequestWithRightFiscalCode
{
    all i: IndividualInformationAnswer | i.user.fiscalCode = i.individualRequest.
    ↪ fiscalCode
}
fact RightAcquisitionDataAnswerToRequest
{
    all i: IndividualInformationAnswer | one u: User | (i.user = u) and (u.
    ↪ retrievedData = i.acquisitionData)
}

fact RunnerNotSpectator
{
    all disj t1, t2 : Track4RunUser | all r: Run | (r in t1.watchingRun and t2 in r.
    ↪ athletes) implies ( t1 ≠ t2)
}

fact TwoRunnersNotSameLocation
{
    all m: Map | all l1, l2: Location | ((l1 in m.athletesLocation) and (l2 in m.
    ↪ athletesLocation)) implies l1 ≠ l2
}

----- DATA4HELP PREDICATES -----

pred IndividualAnswerRegardOnlyPolicyAgreedUsers
{
    all i: IndividualInformationAnswer | i.user.policy.IndividualMonitoring = True
}

```

```

}

pred MinimumGroupMembers
{
    all g: GroupInformationAnswer | #g.acquisitionData > 3 -- 3 stands for 1000
}

----- AUTOMATEDSOS PREDICATES -----

pred AmbulanceRequestSent
{
    all h: HealthStatus | all p: Parameter | one a: AmbulanceRequest | one u: User |
    ((h.parameter = p) and (p.value < p.threshold)) implies
    ((a.time = h.time) and (h in u.retrievedData.healthStatusAcquisition and a.report
    ↪ .user = u))
}

----- TRACK4RUN PREDICATES -----

pred SameNumberLocationsAndRunners
{
    all m:Map | all r: Run | ((m in r.map) implies (#m.athletesLocation = #r.athletes ) and
    ↪ r.athletes.retrievedData.locationAcquisition in m.athletesLocation)
}

----- DATA4HELP ASSERTIONS -----

assert UserPrivacy
{
    IndividualAnswerRegardOnlyPolicyAgreedUsers and MinimumGroupMembers
}

----- AUTOMATEDSOS ASSERTIONS -----

assert AmbulanceEmergency
{
    IndividualAnswerRegardOnlyPolicyAgreedUsers
}

----- TRACK4RUN ASSERTIONS -----

assert WatchingAthletesPosition
{
    SameNumberLocationsAndRunners
}

-----

pred show{}
check WatchingAthletesPosition for 20
check AmbulanceEmergency for 20
check UserPrivacy for 20
run show for 2 but exactly 2 GroupInformationAnswer

```

## 4.2 Results

### Executing "Check UserPrivacy for 20"

Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20  
195713 vars. 13478 primary vars. 425992 clauses. 344ms.  
No counterexample found. Assertion may be valid. 95ms.

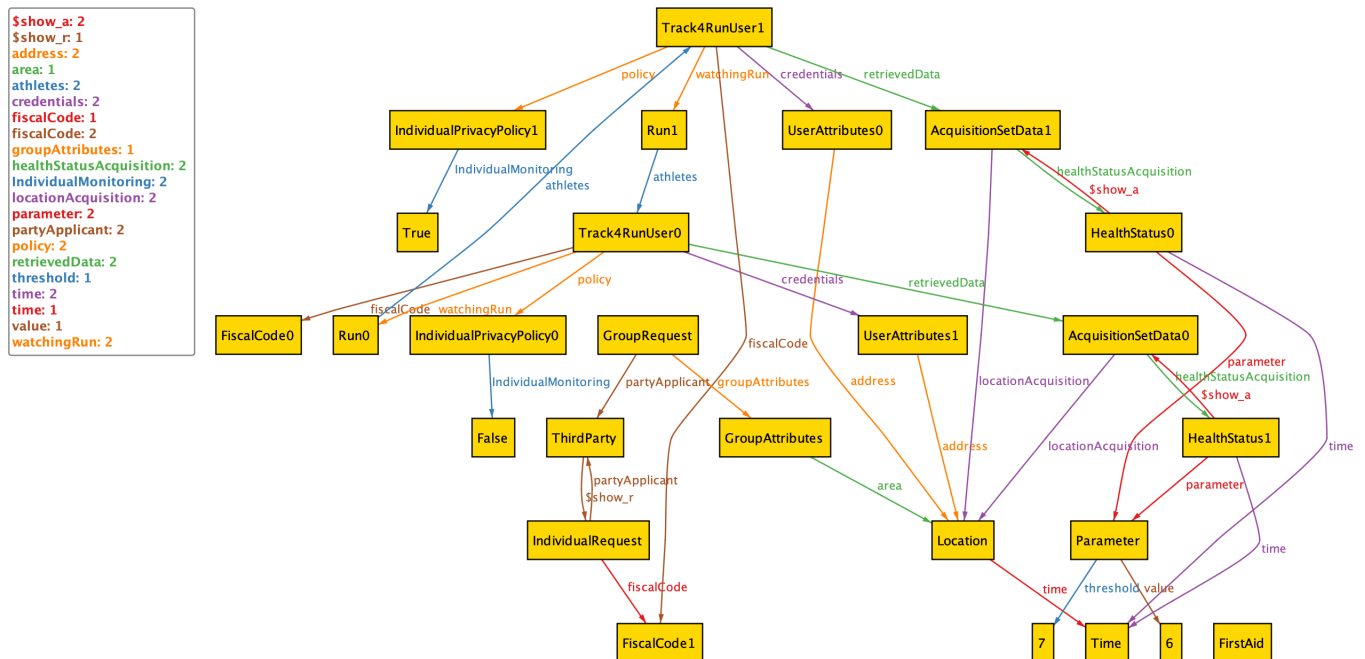
### Executing "Check AmbulanceEmergency for 20"

Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20  
197090 vars. 13498 primary vars. 413119 clauses. 300ms.  
No counterexample found. Assertion may be valid. 2409ms.

### Executing "Check WatchingAthletesPosition for 20"

Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20  
198393 vars. 13518 primary vars. 416204 clauses. 258ms.  
No counterexample found. Assertion may be valid. 523ms.

## 4.3 Generated World



## 5 Effort Spent

In this section contains information about how much hours each group member spent in working at this document.

### 5.0.1 Luca Alessandrelli

Date	Task	Hours
18/10/18	Goals	1
19/10/18	Domain Assumptions	3
20/10/18	Text Assumptions	3
24/10/18	Text Assumptions	1
24/10/18	Domain Assumptions	1
30/10/18	Text Assumptions	0.5
30/10/17	Domain Assumptions	0.5
30/10/18	State Chart	1.5
4/11/18	Goals	2
4/11/18	Text Assumptions	2
4/11/18	Domain Assumptions	2
5/11/18	State Chart	1.5
5/11/18	Use Case	4
6/11/18	Use Case	3
6/11/18	Use Case Diagram	3
7/11/18	Use Case	1.5
7/11/18	Use Case Diagram	1
8/11/18	Use Case	2
8/11/18	Use Case Diagrams	0.5
8/11/18	Scenarios	0.5
9/11/18	Scenarios	1
10/11/18	Document revision	3.5
11/11/18	Alloy	8
Text Assumptions		6.5
Goals		3
Domain Assumptions		6.5
State Chart		3
Scenarios		1.5
Use Case		10.5
Use Case Diagram		4.5
Alloy		8
Document Revision		3.5
Total		47

## 5.0.2 Andrea Caraffa

Date	Task	Hours
18/10/18	Goals	2
19/10/18	Domain Assumptions	3
20/10/18	Text Assumptions	3
21/10/18	Introduction	2
27/10/18	Goals	2
30/10/18	Product Functions	3
1/11/17	Mockups	3
3/11/18	Mockups	3
4/11/18	Goals	2
4/11/18	Mockups	2
5/11/18	External requirements	2
5/11/18	Alloy	2
6/11/18	External requirements	3
9/11/18	Revisioning	2
9/11/18	Alloy	3
10/11/18	Revisioning	3
11/11/18	Revisioning	3
11/11/18	Alloy	3
Text Assumptions		3
Goals		6
Domain Assumptions		3
Introduction		2
Product functions		3
External requirements		5
Mockups		8
Alloy		8
Document Revisioning		8
Total		46

### 5.0.3 Andrea Bionda

/	Task	Hours
	Text Assumptions	3
	Goals and Introduction	6
	Domain Assumptions	3
	Functional requirements	13
	Class Diagram	5
	Sequence Diagram	5
	Performance requirements and Constraints	3
	Alloy	7
	Total	45

## 6 Reference Documents

- Specification Document "Mandatory Project Assignment AY 2018-2019".
- Slides "Structure of RASD".
- Slides "Use of Alloy in RE".
- Use Case Diagrams created with <https://www.lucidchart.com>
- Mockups created with <https://www.fluidui.com>