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 1.2 - 2018/2019

Deliverable:	RASD
Title:	Requirements Analysis and Specification Document
Authors:	Luca Alessandrelli, Andrea Caraffa, Andrea Bionda
Version:	1.2
Date:	9-December-2018
Download page:	https://github.com/lucaalessandrelli/AlessandrelliCaraffaBionda.git
Copyright:	Copyright © 2018, Luca Alessandrelli, Andrea Caraffa, Andrea Bionda – All rights reserved

Contents

List of Figures

List of Tables

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

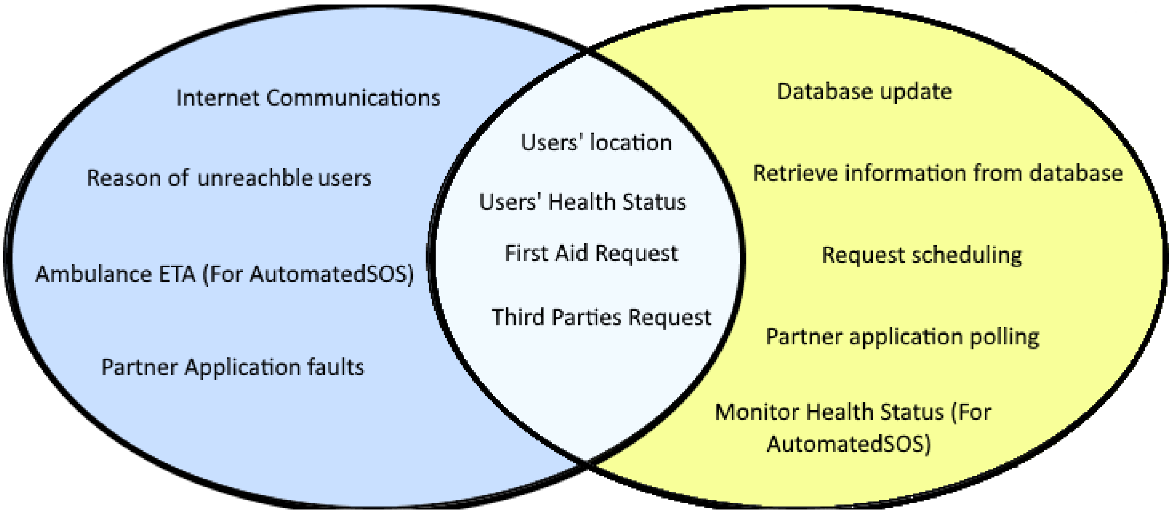

Internet Communications
Reason of unreachble users
Ambulance ETA (For AutomatedSOS)
Partner Application faults
Users' location
Users' Health Status
First Aid Request
Third Parties Request
Database update
Retrieve information from database
Request scheduling
Partner application polling
Monitor Health Status (For AutomatedSOS)

Figure 1: World Phenomena

1.3 Definitions, Acronyms, Abbreviations

- Definitions

  (a) Individual monitoring request: request to access to the data of some specific individuals.

  (b) Group monitoring request: request to access to anonymized data of groups 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) 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.

  (f) Third parties' credentials: information that a company has to provide to become a registered one: company name, p.iva.

  (g) Run information: all the information about the run such as name, date, promoters, maximum number of participants and race path.

  (h) Partner Application: Application installed on users' device, not necessarily developed by TrackMe, that is in charge with retrieve location and health status.

  (i) Security Number / Fiscal Code: a Social Security Number (SSN) is a nine-digit number issued to U.S. citizens and its primary purpose is to track individuals. Fiscal Code is the equivalent of Social Security Number in Italy.

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
1.1	Corrected orthographic errors.	/
	Added text assumptions, modified use cases, mock-ups, requirements and class diagram.	In the first version of the document Data4Help would also answer to third party requests with data retrieved by AutomatedSOS, leading to major exploitation of very sensitive data.
	Modified Product Functions, requirements, text assumptions and use cases.	In the first version of the document Data4Help provided only raw data as answers to third parties' requests. Statistics could also be useful.
	Modified requirements, mock-ups, product functions, text assumptions and use cases.	In the first version of the document the logic behind promoting and enrolling to a run event wasn't explained enough.

Table 1: Version 1.1

Version	Changes	Motivation
1.2	Corrected orthographic errors.	/
	Modified use-cases and use-case diagrams	In the new version of the document a discrepancy about the subscription to a group got fixed.
	Modified requirements	In the new version of the document the role of each requirements is better specified deleting all the ones with the same meaning.
	Modified mock-ups	In this version of the document a new web page for third parties got added inside the "User Interface" section.
	Modified definitions	

Table 2: Version 1.2

1.5 Document Structure

This document is composed by six sections:

1. Introduction	This section 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.
2. Overall Description	This section 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 includes a state diagram that analyzes the process of making a request to access the users' data. *User characteristics* subsection lists the actors interested in using these services.
3. Specific Requirements	This section specifics the requirements identified, both functional and non functional. The first subsection includes the external interface requirements, showing user interfaces with several mock-ups. 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.
4. Formal Analysis Using Alloy	This section includes the alloy model and the discussion of its purpose. Also, a world generated by it is shown.
5. Effort Spent	This section include information about the number of hours each group member has worked for this document.
6. Reference Documents	This section contains the list of reference documents.

Table 3: Document Structure

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.

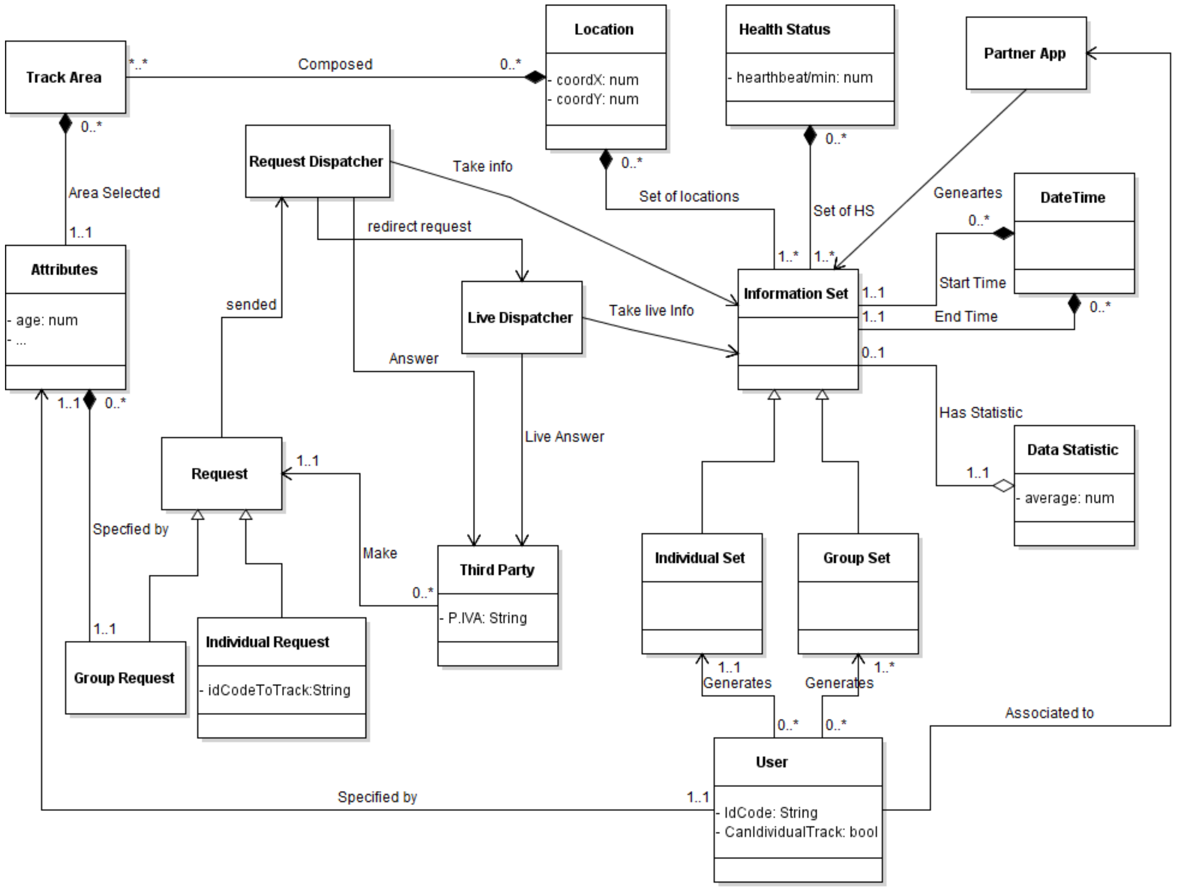


Figure 2: Data4Help Class Diagram

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

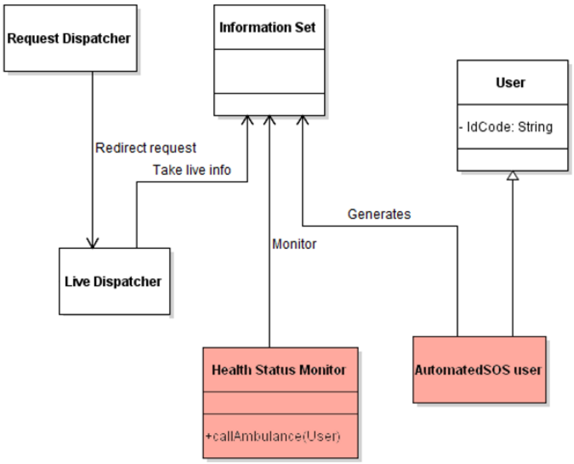


Figure 3: AutomatedSOS Class Diagram

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

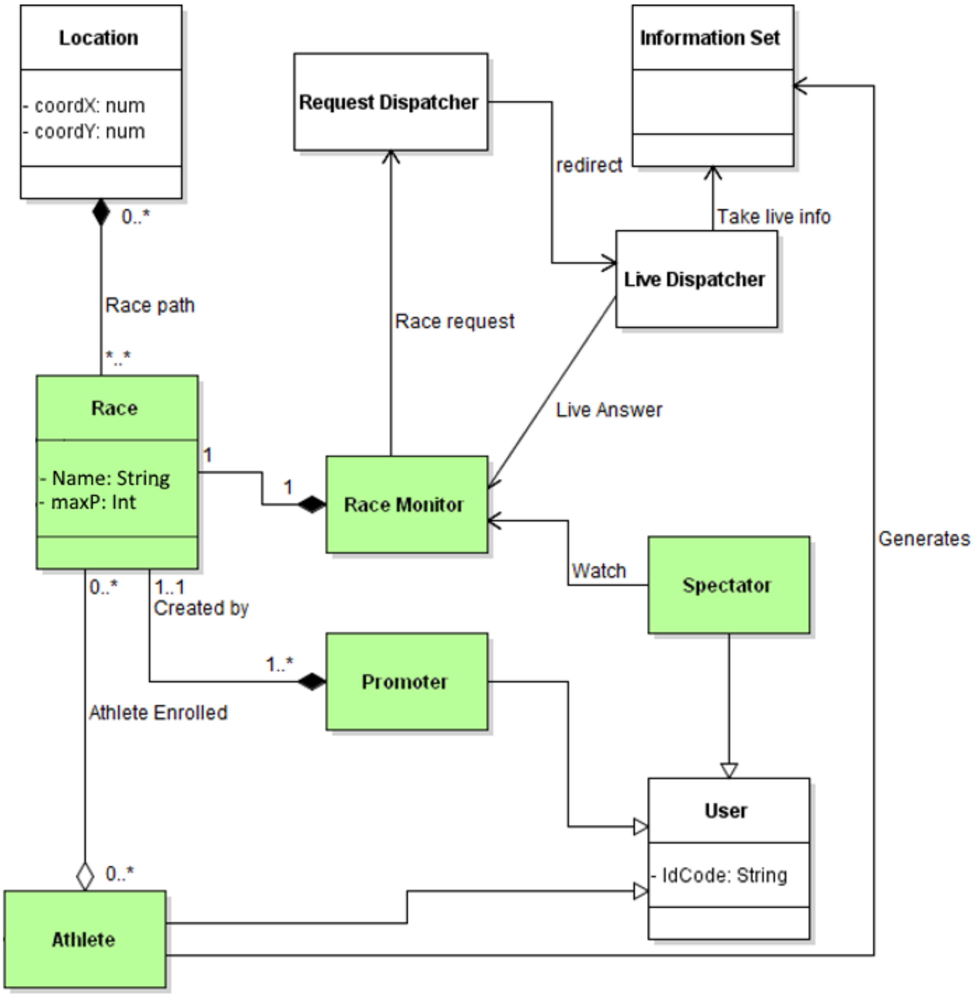


Figure 4: Track4Run Class Diagram

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

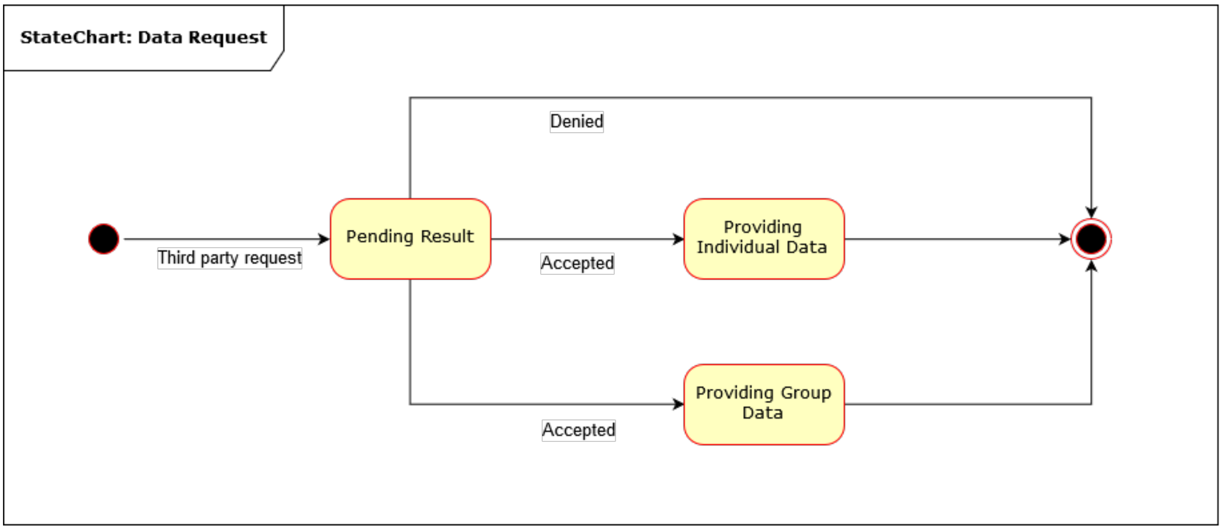


Figure 5: Data Request Object State Chart

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. TrackMe in order to offer a complete solution to third parties send them also statistics about data.

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. AutomatedSOS retrives data directly from the user's smartwatch and send them to Data4Help. In order to keep under control the user's health status AutomatedSOS performs monitoring using these retrieved data and other hystorical data got from Data4Help, the latter are used to have a profile of the user's health status.

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. Every run can be enrolled by everyone, so the scope of inviting a person is just to promote the run and to facilitate the enrollment process.

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 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 smartwatch as: Heart Rate, Blood Pressure, Pedometer, Calories Calculation.

  (j) The answers to both Individual Requests and Group Requests contain also some statistics other than raw data.

  (k) Live acquisition request expires after one month from the moment that is formulated.

- 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 any case, data retrieved by AutomatedSOS won't be sent to third parties if they perform an individual request. Thus TrackMe doesn't offer additional services for third parties through AutomatedSOS.

  (e) In order to keep under systematic review the user's health status and have a broad view of it, some historical information about the user are received by Data4Help's Database.

  (f) This service can be used only by elderly people (70+) or by who really need it, in order to avoid useless waste of resources.

(g) 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) All run events are seen as public runs, so every user can enroll to future runs.
  - (d) Any user can organize an event and also invite other users to enroll to the run.
  - (e) All the events can be spectated by users.
  - (f) All users invited to a run can accept or discard the request.
  - (g) 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

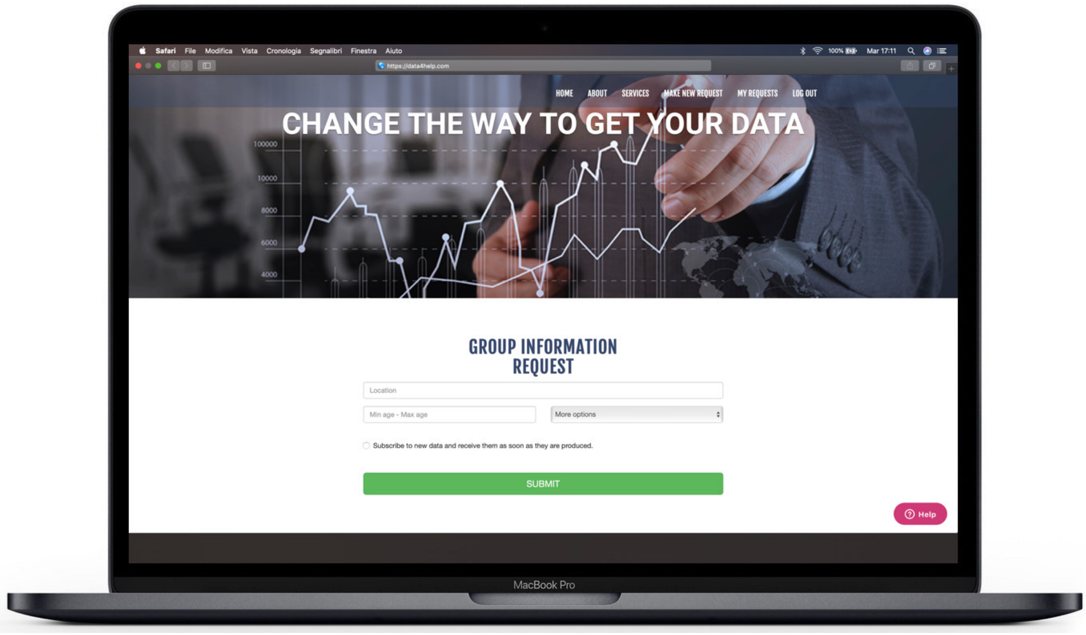
3.1 External Interface Requirements

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 retrievement, 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.



GROUP INFORMATION REQUEST

Location

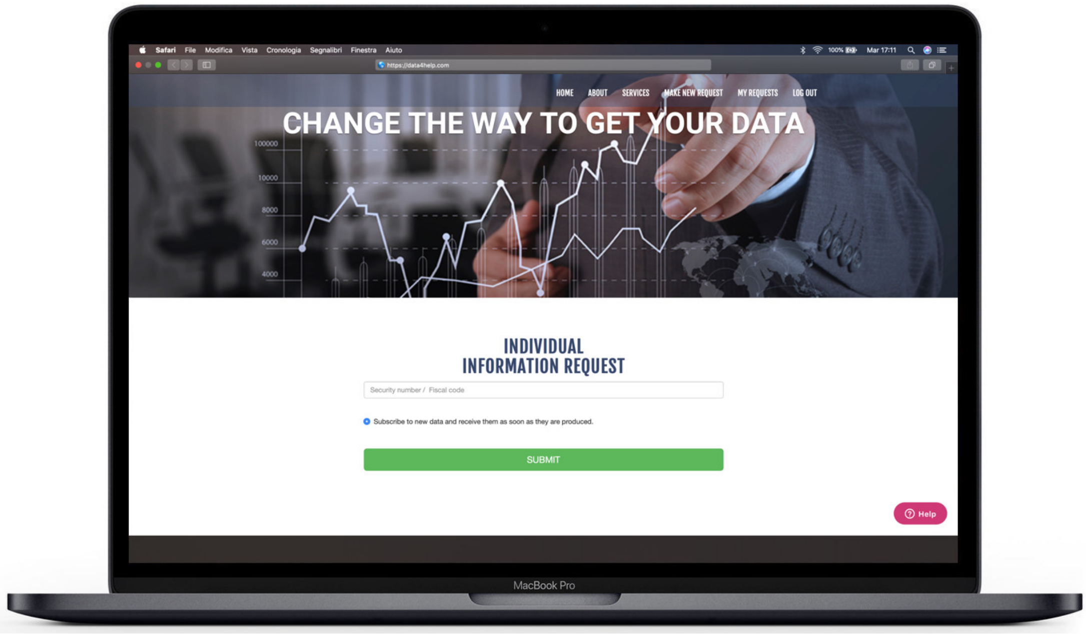
Min age - Max age

More options

○ Subscribe to new data and receive them as soon as they are produced.

SUBMIT

Figure 6: Group Monitoring Request Mockup



CHANGE THE WAY TO GET YOUR DATA

INDIVIDUAL
INFORMATION REQUEST

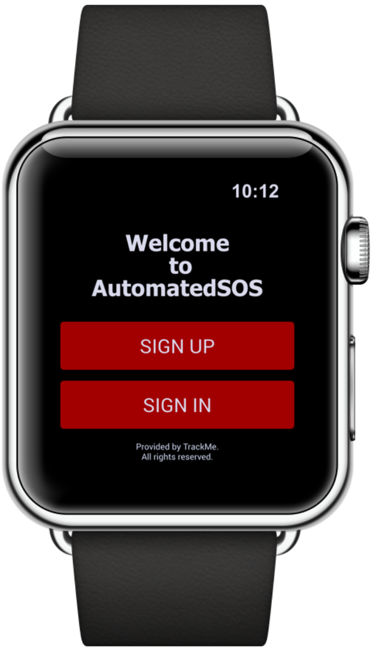
Security number / Fiscal code

● Subscribe to new data and receive them as soon as they are produced.

Figure 7: Individual Monitoring Request Mockup

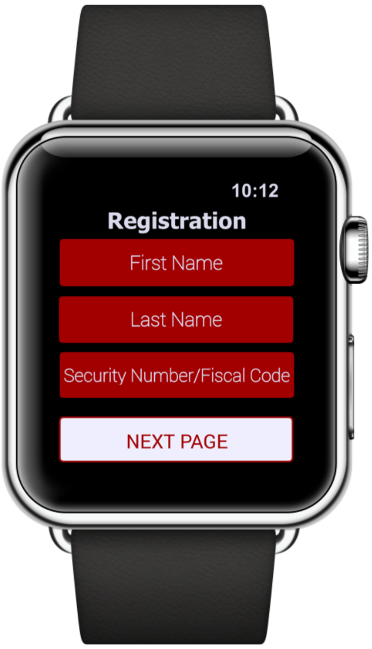
- AutomatedSOS

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.



Welcome
to
AutomatedSOS

Provided by TrackMe.
All rights reserved.



Registration

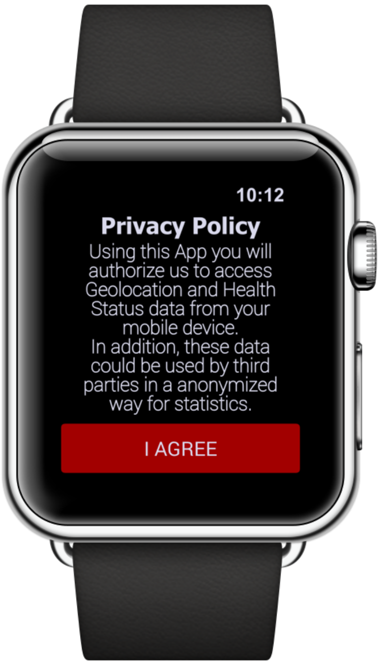
First Name

Last Name

Security Number/Fiscal Code

Figure 8: Welcome Page Mockup

Figure 9: Registration Form Mockup

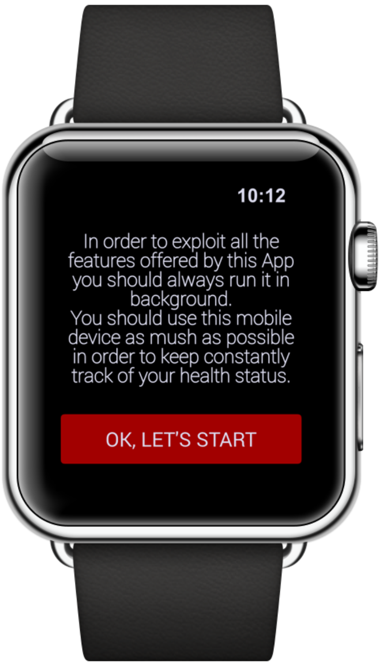


Privacy Policy

Using this App you will authorize us to access Geolocation and Health Status data from your mobile device.

In addition, these data could be used by third parties in a anonymized way for statistics.

Figure 10: Privacy Policy Mockup



In order to exploit all the features offered by this App you should always run it in background.

You should use this mobile device as mush as possible in order to keep constantly track of your health status.

Figure 11: Usage Conditions Mockup

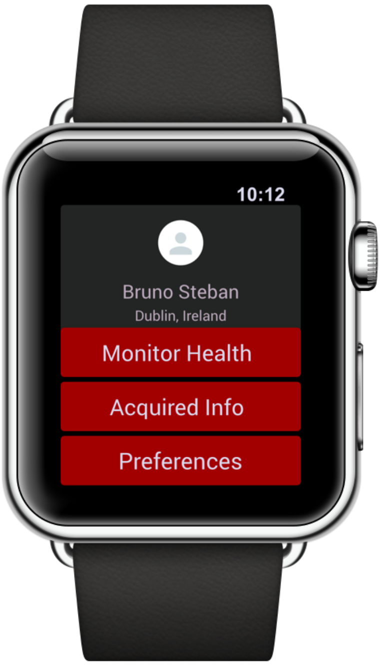
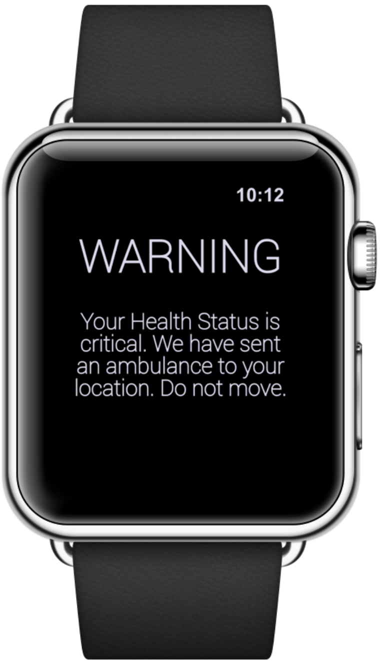


Figure 12: Main Menu Mockup



WARNING

Your Health Status is critical. We have sent an ambulance to your location. Do not move.

Figure 13: Warning Message Mockup

- 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 AutomatedS0S, there is not any interface provided for the third parties.

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

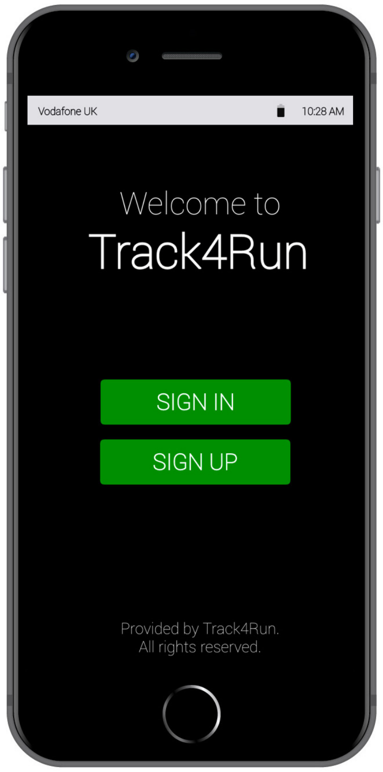


Figure 14: Welcome Page Mockup

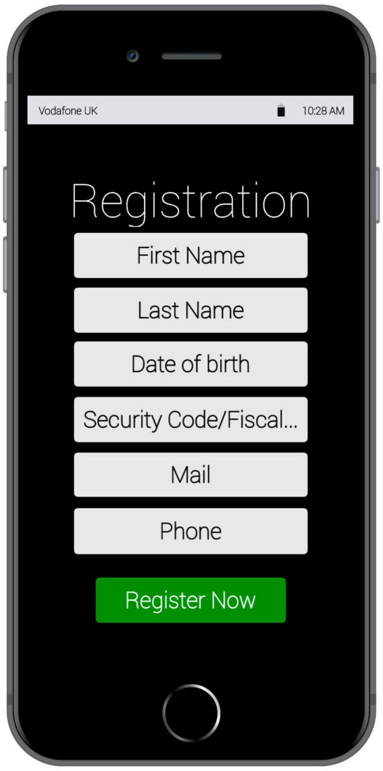
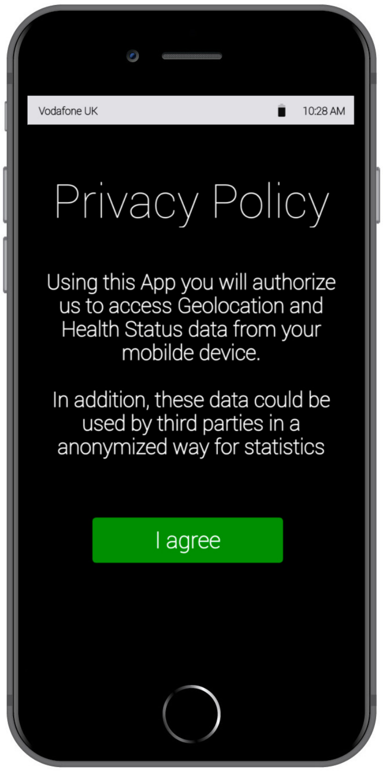


Figure 15: Registration Form Mockup



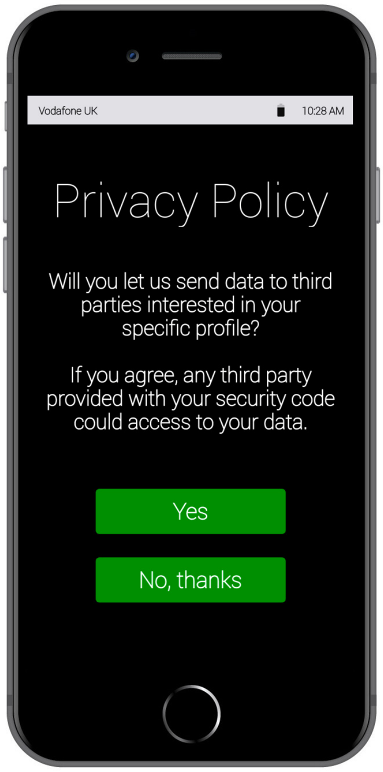
Vodafone UK 10:28 AM

Privacy Policy

Using this App you will authorize us to access Geolocation and Health Status data from your mobilde device.

In addition, these data could be used by third parties in a anonymized way for statistics

Figure 16: Privacy Policy pt.1 Mockup



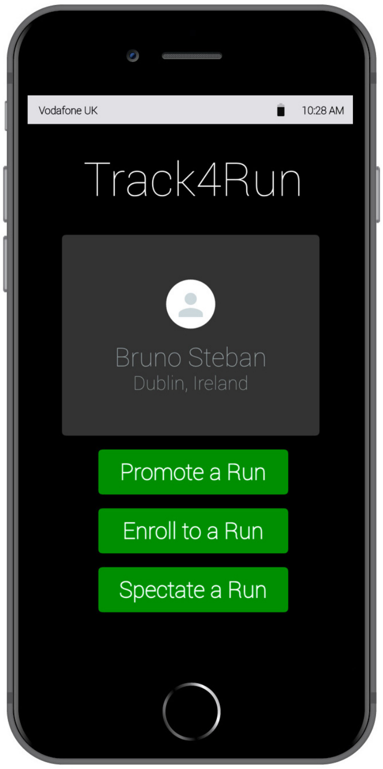
Vodafone UK 10:28 AM

Privacy Policy

Will you let us send data to third parties interested in your specific profile?

If you agree, any third party provided with your security code could access to your data.

Figure 17: Privacy Policy pt.2 Mockup

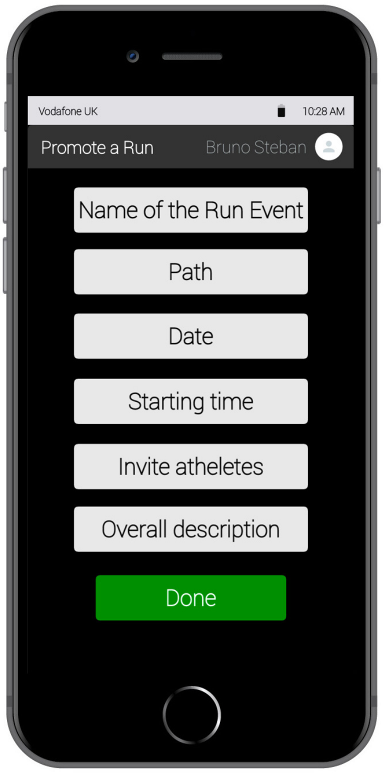


Vodafone UK 10:28 AM

Track4Run

Bruno Steban
Dublin, Ireland

Figure 18: Main Menu Mockup



Vodafone UK 10:28 AM

Promote a Run Bruno Steban

Figure 19: Promote a Run View Mockup

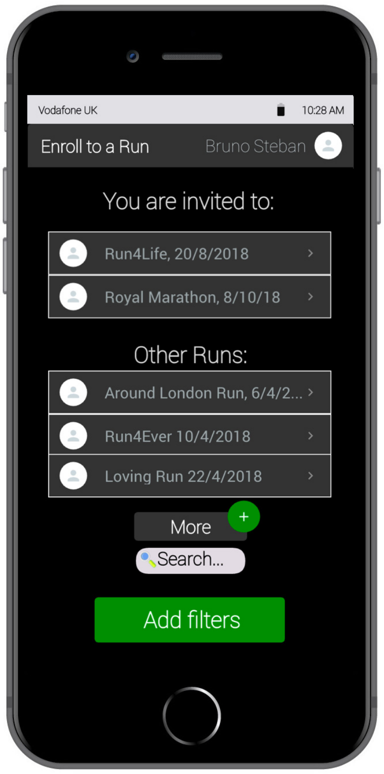


Figure 20: Enroll to a Run View Mockup

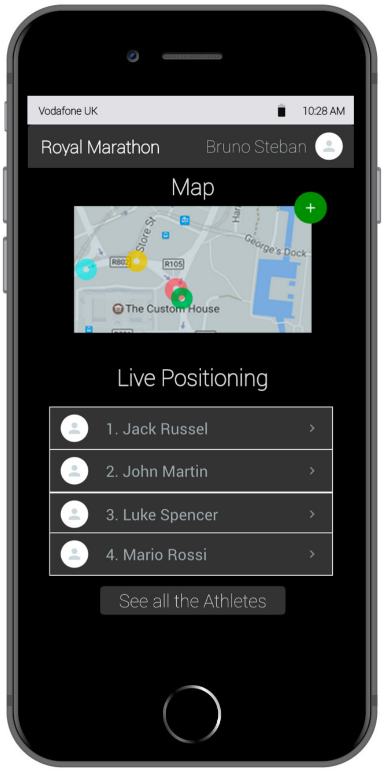


Figure 21: Spectate a Run View Mockup

3.1.2 Hardware, Software and Communication 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.

3.2 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 curios 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.

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.5 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.

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

R.6 Allow third parties to register to Data4Help service specifying all their credentials.

R.7 The system should allow third parties to send information requests.

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

R.8 The system has to collect all the useful data that match the request.

R.9 The system has to generate a statistic on data selected

R.10 The system has to send to the third party all the raw data collected until the moment of the request.

R.11 The system has to send all the statistics already produced.

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

R.12 Allow third parties to subscribe to groups or individuals in order to receive live data.

R.13 Provide to subscribed third parties raw 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.14 Allow third parties to insert the fiscal code of the user he wants to track.

R.15 Deny third parties to receive single mode information about users that have not accepted the second part of the privacy policy.

R.8 Collect all the useful data retrieved by Data4Help that are produced by the interested users.

R.10 Send all the collected information to request applicant.

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

R.16 Allow third parties to insert search area and attributes in which they are interested to restrict their field of search.

R.17 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.8 Collect all the useful data retrieved by Data4Help that are produced by the interested users.

R.10 Send all the collected information to request applicant.

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

R.15 Deny third parties to receive single mode information about users that have not accepted the second part of the privacy policy.

R.17 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.18 Allow users to be tracked from AutomatedSOS filling up the registration and agreeing only to first part of 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.19 The application has to retrieve users' health status every 2 seconds in order to guarantee a reaction time of 5 seconds.

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

- R.19 The application has to retrieve users' health status every 2 seconds in order to guarantee a reaction time of 5 seconds.
- R.20 The application sends to Data4Help service all the data retrieved in live acquisition.
- R.21 The application gets from Data4Help service all the historical data about the user.
- R.22 Allow the user to set personal threshold values.

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

- R.23 The application has to control health status with data retrieved in local to immediately realize whether certain parameters are critical.
- R.24 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.25 Supply to the hospital the user's location and all the useful information to provide efficient first aid.
- R.26 In the case no answer arrives from the hospital the software must repeat another time the request until an answer in reached.
- R.27 As soon as the acknowledgement message is received a warning message is displayed on the user's smartwatch.
- D.11 The ambulance successfully reach the location of the individual.

- Track4Run

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

- 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.
- D.4 Devices used to monitor individuals always report correct values.
- R.28 The application has to interact with Smartwatch/Smartphone APIs in order to retrieve GPS location with a resolution of 10 seconds when the user is in the run .

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

- R.29 Allow users to create a run once all the general information are inserted.
- 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.

R.32 Allow promoters to specify maximum number of athletes that can participate.

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

R.33 Allow the user to see all the runs generated (which he is invited or not).

R.34 Allow user to enroll in a run.

R.35 Deny user to enroll in a run if maximum number of participants is already reached.

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.**

R.33 Allow the user to see all the runs generated (which he is invited or not).

D.17 All athletes have their tracking devices with them and the application is enabled for the entire duration of the run.

R.36 Allow user to select a run to be viewed.

R.37 The application requests to Data4Help the position of all the other athletes involved.

R.38 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

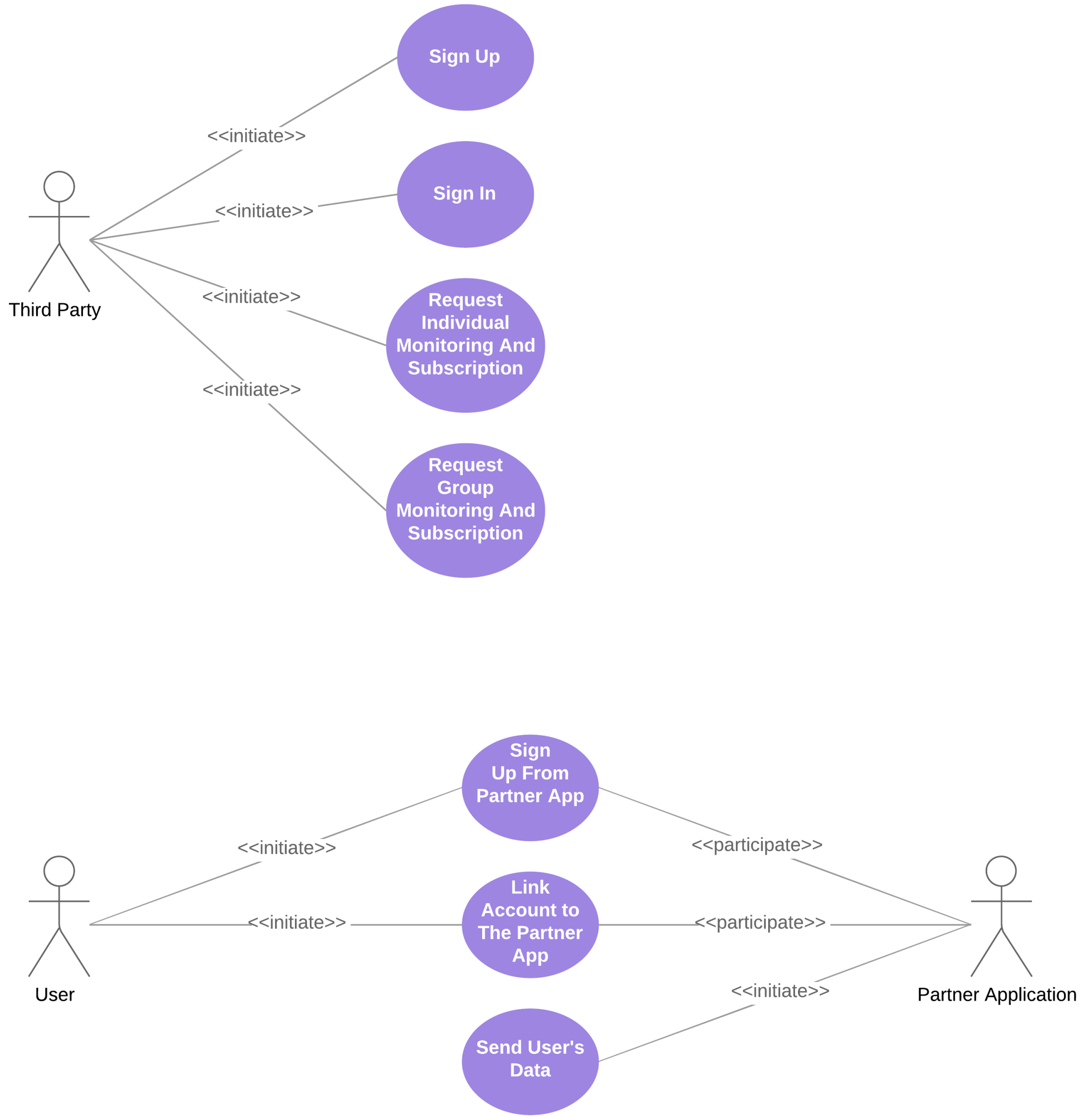


Figure 22: Data4Help Use Case Diagrams

- AutomatedSOS

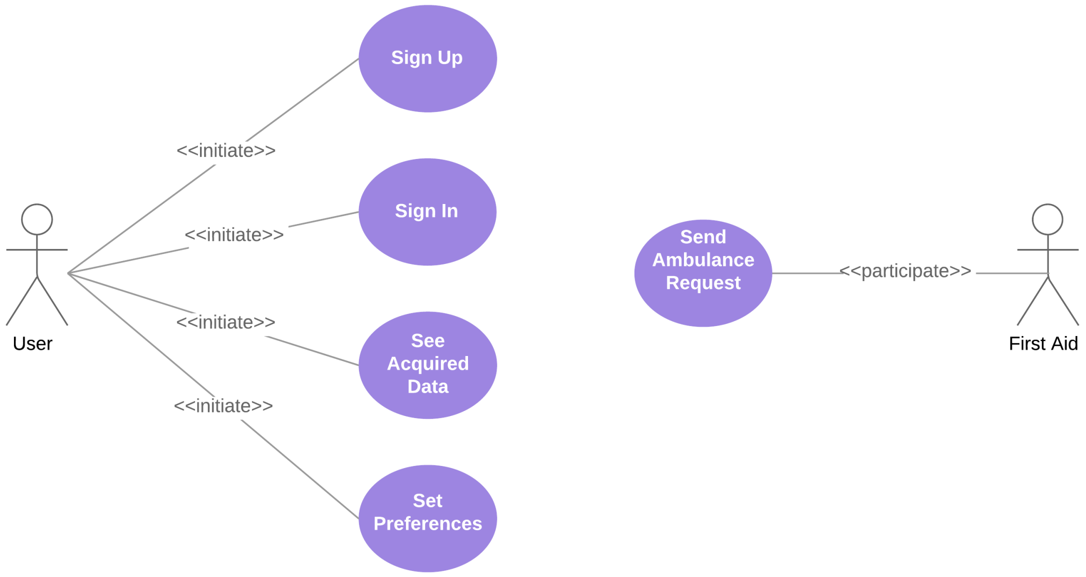


Figure 23: AutomatedSOS Use Case Diagram

- Track4Run

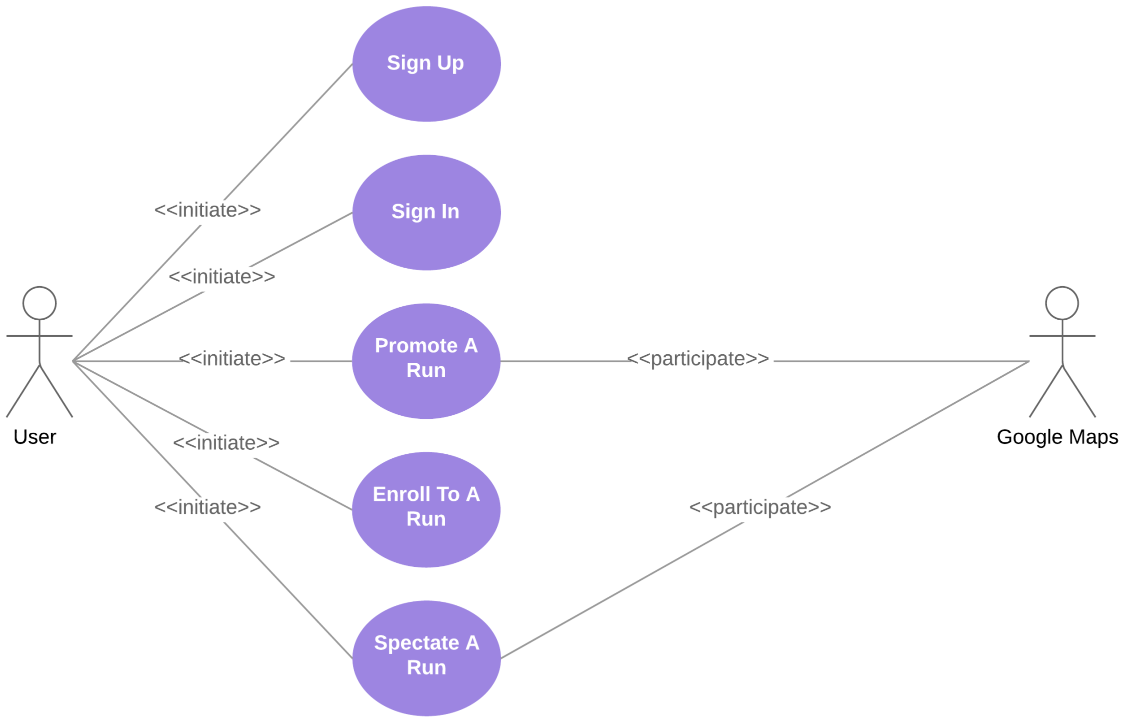


Figure 24: Track4Run Use Case Diagram

3.3.2 Use Cases

- Data4Help Use Cases

Name	Sign Up
Actors	Third Party
Entry Conditions	TRUE
Event Flow	(a) The Third Party enters the sign up section. (b) The system shows to the third party all the attributes fields needed for the registration. (c) The Third Party fills all the attribute fields. (d) The Third Party confirms all the attributes inserted stating he wants to register. (e) The system creates and saves the third party's account.
Exit Condition	The third party's account has been created and the third party is now registered.
Exceptions	• 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.

Table 4: Sign Up Use Case

Name	Sign In
Actors	Third Party
Entry Conditions	TRUE
Event Flow	(a) The Third Party enters the sign in section. (b) The system shows to the third party all the credentials fields needed for the log in. (c) The Third Party fills all credentials fields and confirms he wants to log in. (d) The system accepts the log in request.
Exit Condition	The third party is now logged in.
Exceptions	• If the third party inserts invalid log in credentials a warning is generated saying the credentials are invalid.

Table 5: Sign In Use Case

Name	Request Individual Monitoring And Subscription
Actors	Third Party
Entry Conditions	The third party is logged in.
Event Flow	(a) The Third Party enters the Individual Request section. (b) The system shows to the third party all the information fields needed for the identification of the individual. (c) The Third Party can ask to subscribe to new data as soon as they are produced (live acquisition). (d) The Third Party fills all the attribute fields and confirms he wants to track that specific individual. (e) The system shows all the individual's data that have been collected until the moment of the request along with some statistics. (f) The system subscribe the Third Party to the group if it was asked.
Exit Condition	The request's outcome is shown and third party is subscribed if it was asked.
Exceptions	• If the inserted attributes are not linked to any user account then a warning message is displayed saying that the individual is not registered. • 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.

Table 6: Request Individual Monitoring and Subscription Use Case

Name	Request Group Monitoring And Subscription
Actors	Third Party
Entry Conditions	The third party is logged in.
Event Flow	(a) The Third Party enters the Group Request section. (b) The system shows to the third party all the information fields needed for defining the group. (c) The Third Party insert the search area. (d) The Third Party inserts all the attributes. (e) The Third Party can ask to subscribe to new data as soon as they are produced (live acquisition). (f) The system accepts the request. (g) The system shows all the group's data that have been collected until the moment of the request along with some statistics. (h) The system subscribe the Third Party to the group if it was asked.
Exit Condition	The request's outcome is shown and third party is subscribed if it was asked.
Exceptions	• If the group request get rejected by the system a warning message will be displayed saying the request is rejected.
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.

Table 7: Request Group Monitoring and Subscription Use Case

Name	Sign Up From Partner App
Actors	User, Partner Application
Entry Conditions	The user accepted the treatment of data policy.
Event Flow	(a) The user starts the sign up function on the partner app. (b) The Partner Application shows to the user all the attributes fields needed for the registration. (c) The User fills all the attribute fields. (d) The Partner Application sends to the system the attributes inserted by the user. (e) The system receives by the partner application all the attributes inserted by the user. (f) The system creates the user's account and saves the received data.
Exit Condition	The system registered the user.
Exceptions	• 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.

Table 8: Sign Up From Partner App Use Case

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	(a) The user starts the account linking function on the partner app. (b) The Partner Application shows to the user all the credential fields needed for the linking process. (c) The User fills all the credential fields. (d) The Partner Application sends to the system the credentials inserted by the user. (e) The system receives by the partner application all the credentials inserted by the user. (f) The system sends back to the partner application the outcome of the operation.
Exit Condition	The system registered the user.
Exceptions	• 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.

Table 9: Link Account To The Partner App Use Case

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	(a) The Partner Application collects user's data. (b) The Partner Application sends the user's data to the the system. (c) The system receives and saves the user's data sent by the partner application.
Exit Condition	The system saved the user's data.
Exceptions	None

Table 10: Send User's Data Use Case

- AutomatedSOS Use Cases

Name	Sign Up
Actors	User
Entry Conditions	The User has AutomatedSOS installed on his smartwatch.
Event Flow	(a) The User enters the sign up section of the app. (b) The User accepts the treatment of data policy. (c) The system shows to the user all the attributes fields needed for the registration. (d) The User fills all the attribute fields. (e) The User confirms all the attributes inserted stating he wants to register. (f) The system creates and saves the user's account.
Exit Condition	The user's account has been created and the user is now registered.
Exceptions	• 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. • 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.

Table 11: Sign Up to AutomatedSOS Use Case

Name	Sign In
Actors	User
Entry Conditions	The User has AutomatedSOS application installed on his smartwatch.
Event Flow	(a) The User enters the sign in section of the app. (b) The system shows to the user all the credentials fields needed for the log in. (c) The User fills all credentials fields and confirms he wants to log in. (d) The system accepts the log in request.
Exit Condition	The User user is now logged in.
Exceptions	• If the user inserts invalid log in credentials a warning is generated saying the credentials are invalid.

Table 12: Sign In to AutomatedSOS Use Case

Name	See Acquired Data
Actors	User
Entry Conditions	The User is logged in.
Event Flow	(a) The User enters the Acquired Info section of the app. (b) The system gets all the user's information that have been retrieved by the application until that moment. (c) The system displays the user's information.
Exit Condition	All the information retrieved by the system are shown on the app.
Exceptions	• 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.

Table 13: See Acquired Data Use Case

Name	Set Preferences
Actors	User
Entry Conditions	The User is logged in.
Event Flow	(a) The User enters the Preferences section of the application. (b) The User can add or remove certain health parameters in order to personalize the monitoring profile. (c) The User can also change certain parameters threshold. (d) The User confirms all the changes done. (e) The system saves all the changes made by the user.
Exit Condition	The parameters are correctly updated as the user wants them to be.
Exceptions	• If the user does not confirm the changes then all parameters remain the same as before.

Table 14: Set Preferences Use Case

Name	Send Ambulance Request
Actors	AutomatedSOS, First Aid
Entry Conditions	A critical health parameter value is below the threshold.
Event Flow	(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. (b) First Aid immediately sends an ambulance to the user's location. (c) First Aid sends an acknowledge message to AutomatedSOS. (d) AutomatedSOS displays on the app a warning message saying that an ambulance is currently heading to the user's location.
Exit Condition	A warning message is shown saying that an ambulance is currently heading to the user's location.
Exceptions	• If no acknowledge message is received by AutomatedSOS after the form has been sent, as soon as a certain time out expires AutomatedSOS resend the form with updated information.
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.

Table 15: Send Ambulance Request Use Case

- Track4Run Use Cases

Name	Sign Up
Actors	User
Entry Conditions	The User has Track4Run application installed on his device.
Event Flow	(a) The User enters the sign up section of the app. (b) The User accepts the treatment of data policy. (c) The system shows to the user all the attributes fields needed for the registration. (d) The User fills all the attribute fields. (e) The User confirms all the attributes inserted stating he wants to register. (f) The system creates and saves the user's account.
Exit Condition	The user's account has been created and the user is now registered.
Exceptions	• 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. • 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.

Table 16: Sign Up to Track4Run Use Case

Name	Sign In
Actors	User
Entry Conditions	The User has Track4Run application installed on his smartwatch.
Event Flow	(a) The User enters the sign in section of the app. (b) The system shows to the user all the credentials fields needed for the log in. (c) The User fills all credentials fields and confirms he wants to log in. (d) The system accepts the log in in request.
Exit Condition	The User user is now logged in.
Exceptions	• If the user inserts invalid log in credentials a warning is generated saying the credentials are invalid.

Table 17: Sign In to Track4Run Use Case

Name	Promote A Run
Actors	User
Entry Conditions	The User is logged in.
Event Flow	(a) The User enters the Promote a Run section of the app. (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. (c) The system creates and saves the run’s information. (d) The system automatically sends notifications to all athletes specified by the promoter asking them if they want to participate to the run.
Exit Condition	The run event has been created and added to the list of promoted runs.
Exceptions	• 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.

Table 18: Promote a Run Use Case

Name	Enroll To A Run
Actors	User
Entry Conditions	The User is logged in.
Event Flow	(a) The User enters to the "Enroll to a Run" section of the app. (b) The system shows to the user a list containing all invites received and also a second list containing all created runs taking place in the future. (c) The User can filter the runs with some attributes. (d) The User chooses the run he wants to participate to. (e) The system enrolls the user to the run.
Exit Condition	The user is now enrolled to the chosen run.
Exceptions	• If the chosen run has already capped the maximum amount of athletes then a warning message is displayed saying that no more athletes are allowed to participate to the run.

Table 19: Enroll to a Run Use Case

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

Table 20: Spectate a Run Use Case

3.4 Sequence Diagram

- Data4Help

  Individual request with on-demand acquisition performance and automatic data update inside Data4Help.

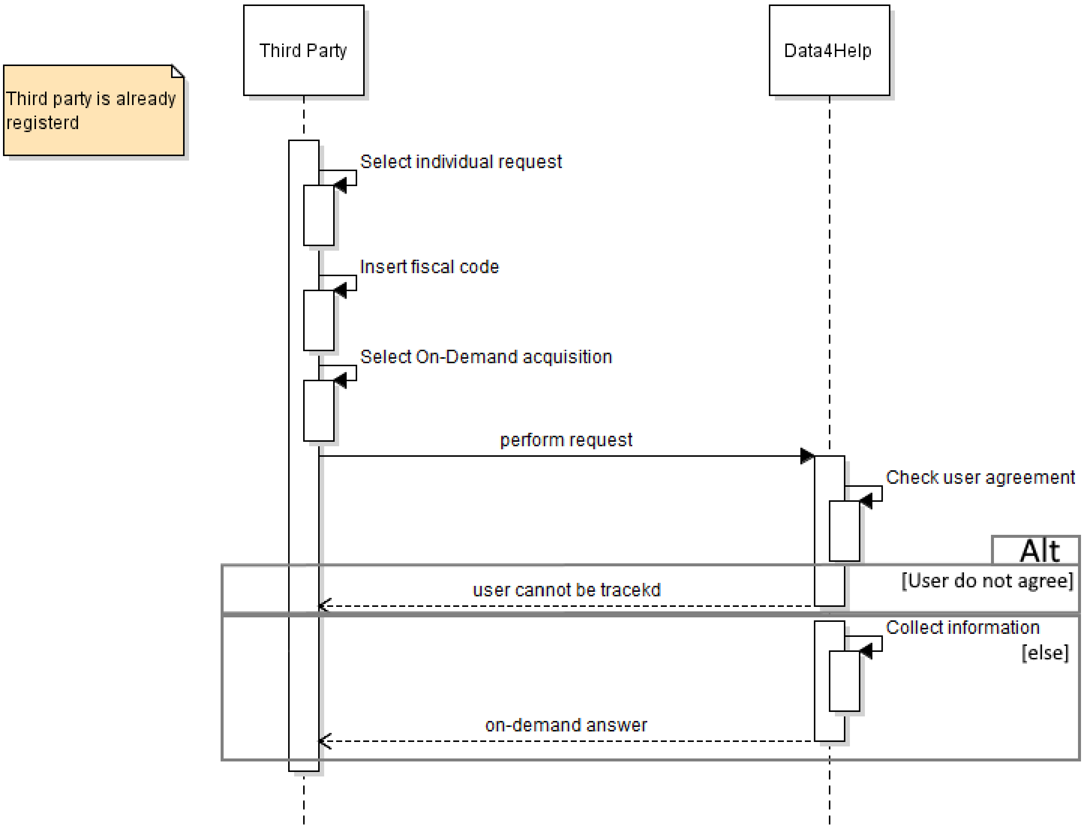


Figure 25: On-demand Acquisition Sequence Diagram

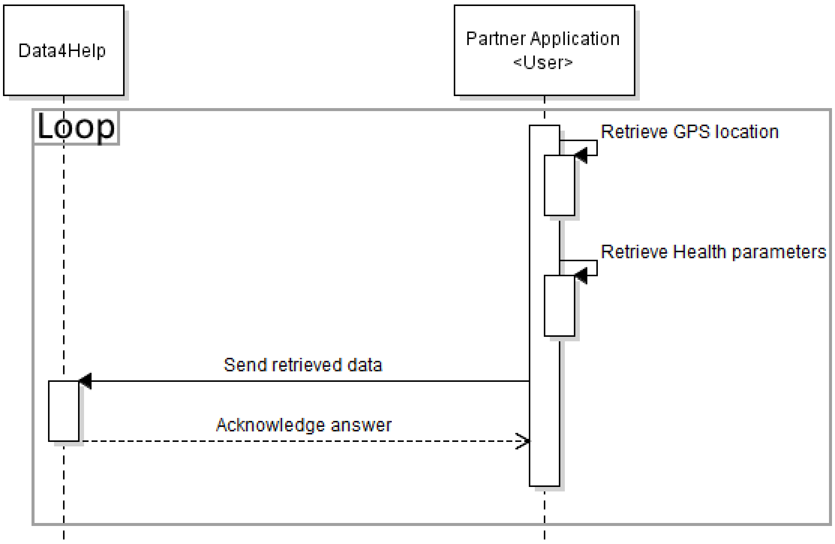


Figure 26: Collecting Data from Partner Application Sequence Diagram

Group request with live acquisition and user registration performances.

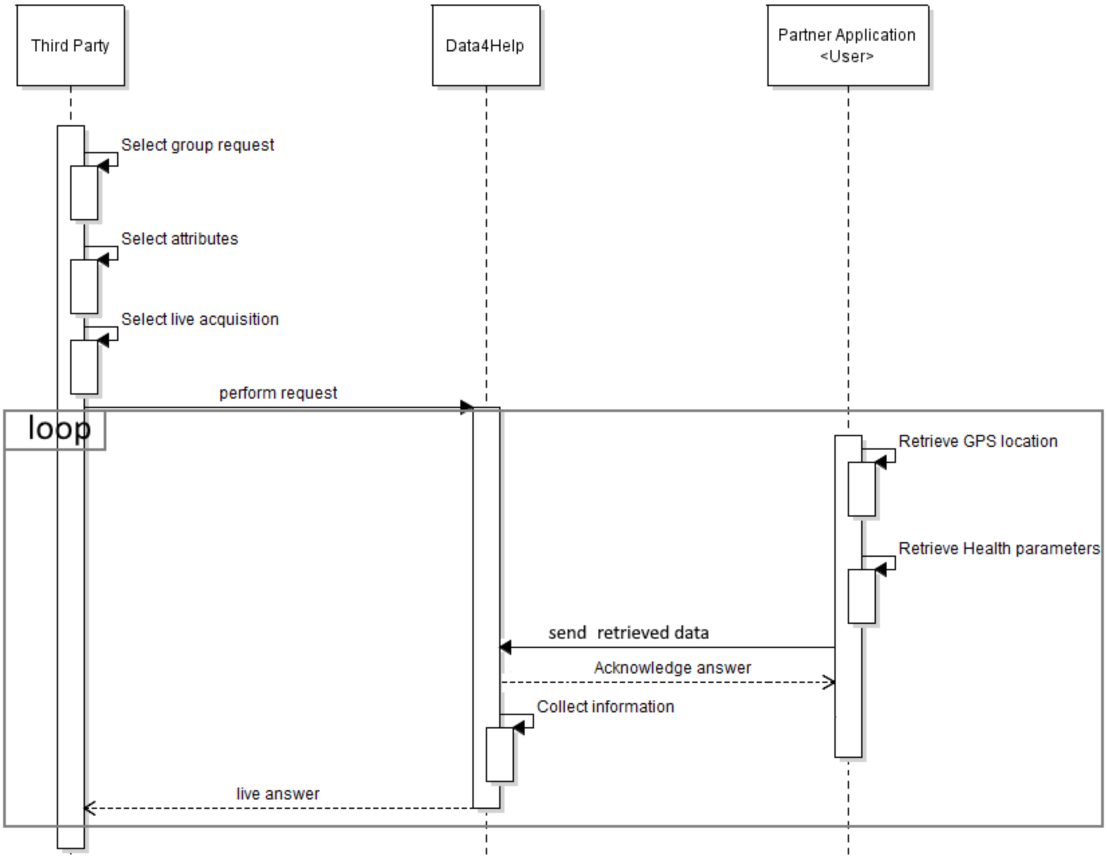


Figure 27: Group Monitoring Request with Live Acquisition Sequence Diagram

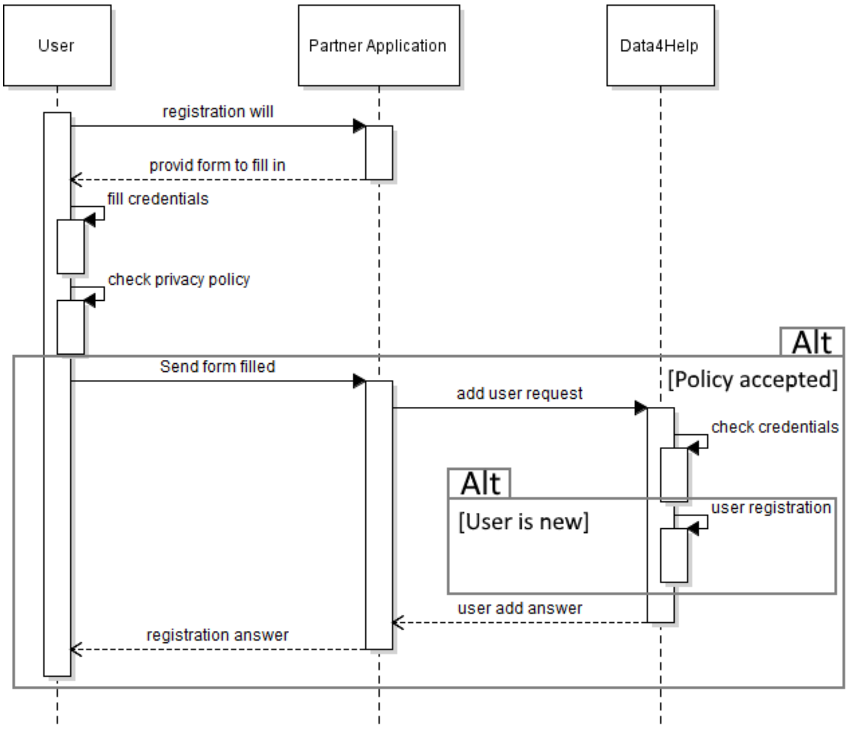


Figure 28: User Registration Sequence Diagram

- AutomatedSOS AutomatedSOS monitor and ambulance caller services.

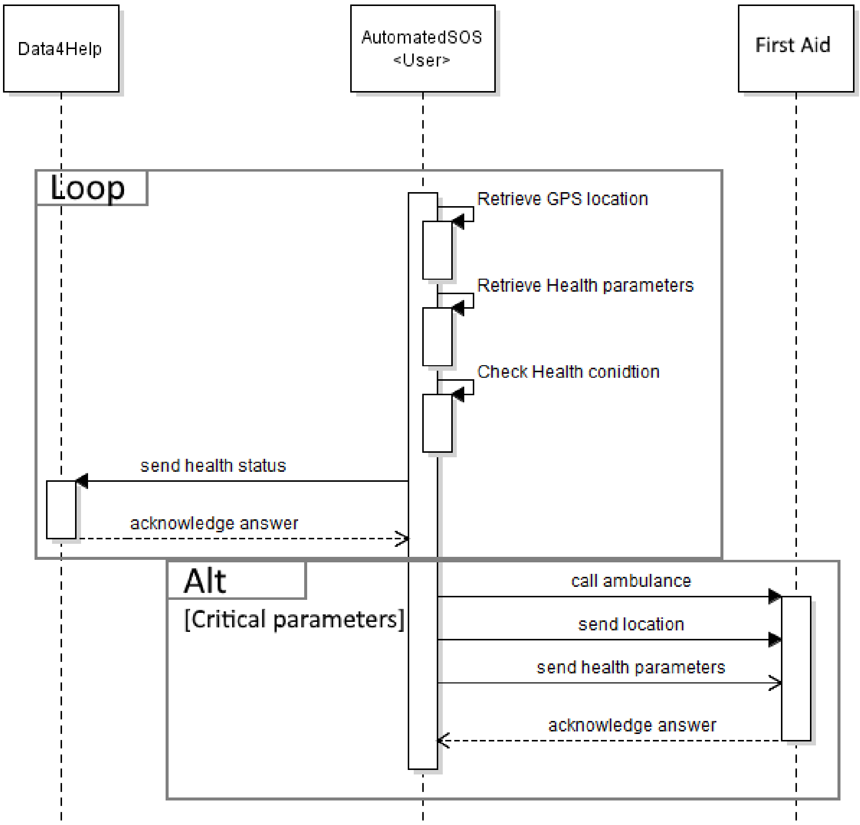


Figure 29: AutomatedSOS Sequence Diagram

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

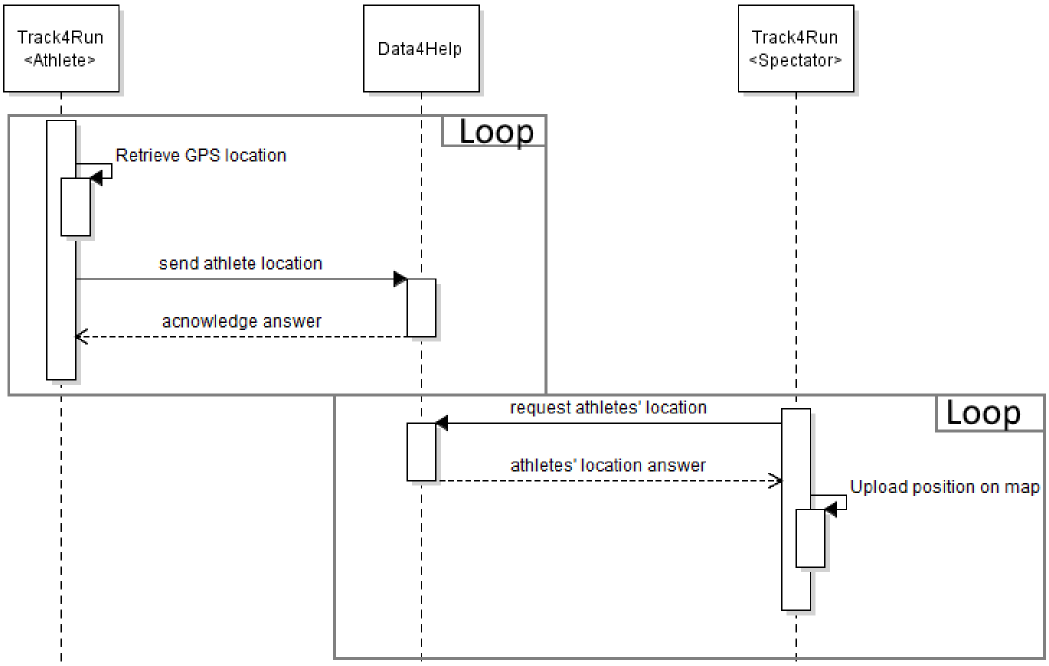


Figure 30: Track4Run Sequence Diagram

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.

- Data4Help unsubscribes third parties after one month from live acquisition because it is a very expensive process.

- 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 guarantee 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 in 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

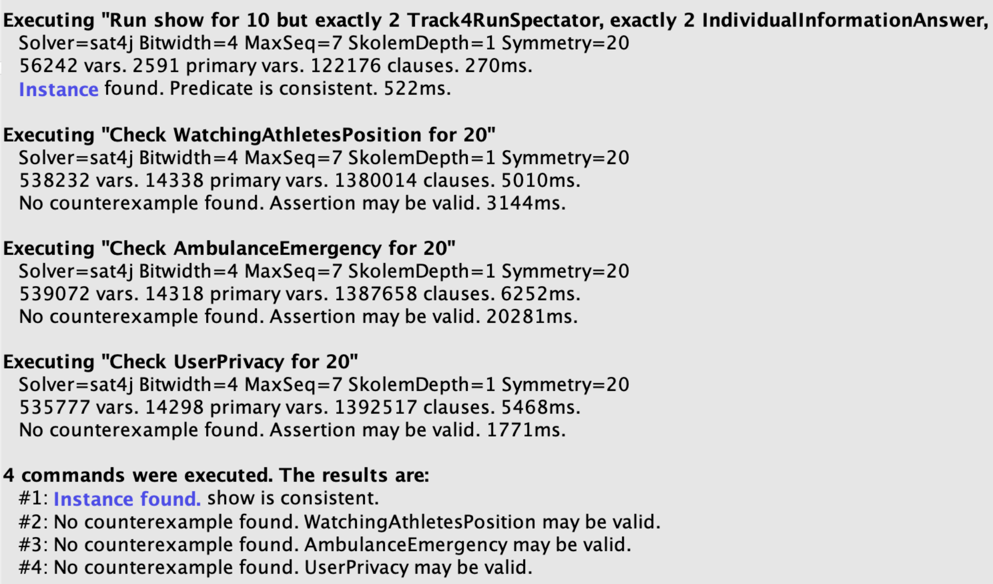
```
open util/integer

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

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

sig SecurityNumber {}

sig Time {}

sig Location {
        time: one 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 {
        securityNumber: one SecurityNumber,
        policy: one IndividualPrivacyPolicy, --Status of policy acceptance
        credentials: one UserAttributes, --User's credentials
        retrievedData: lone AcquisitionSetData --User's set of acquisition acquired
}

sig ThirdParty {}

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

```
}

sig IndividualRequest extends InformationRequest {
        securityNumber: one SecurityNumber --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,
}

sig Run {
        athletes: some Track4RunUser,
        map: one Map
} {#athletes > 1}

sig Track4RunUser extends User {}

sig Track4RunSpectator extends Track4RunUser {
        watchingRun: one Run
}

-------------------- FACTS ---------------------

--------------- DATA4HELP FACTS ---------------

fact NoAloneBool {
        all b: Bool | some i: IndividualPrivacyPolicy | b in i.IndividualMonitoring
}


fact HealthStatusInAcquisitionSetData {
        all h: HealthStatus | one 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 GroupAnswerRightUserLocatin {
        all a: AcquisitionSetData | all g: GroupInformationAnswer | one u: User | (a in g
            ↪ .acquisitionData) implies ( a in u.retrievedData and
        g.groupRequest.groupAttributes.area = u.credentials.address)
}

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.securityNumber ≠ u2.securityNumber
}
fact AnswerToRequestWithRightFiscalCode {
        all i: IndividualInformationAnswer | i.user.securityNumber = i.individualRequest.
            ↪ securityNumber
}

fact RightAcquisitionDataAnswerToRequest {
        all i: IndividualInformationAnswer | one u: User | (i.user = u) and (u.
            ↪ retrievedData = i.acquisitionData)
}

fact NoMoreThanOneHealthStatusWithSameTimeForTheSameIndividual {
        all h1, h2: HealthStatus | all u: User | (h1 in u.retrievedData.
            ↪ healthStatusAcquisition and h2 in u.retrievedData.healthStatusAcquisition
        and h1.time = h2.time) implies h1 = h2
}

fact NoMoreThanOneLocationWithSameTimeForTheSameIndividual {
        all l1,l2: Location | all u: User | (l1 in u.retrievedData.locationAcquisition
            ↪ and l2 in u.retrievedData.locationAcquisition
        and l1.time = l2.time) implies l1 = l2
}

------------- AUTOMATEDSOS FACTS -------------


fact NoAloneFirstAid {
        all f: FirstAid | some r: Report | f in r.receiver
}

fact NoAloneReport {
        all r: Report | one a : AmbulanceRequest | r in a.report
}

fact AmbulanceRequestSentThereIsParamDown {
         all a: AmbulanceRequest | some h: HealthStatus | some p: Parameter |
                        (h in a.report.user.retrievedData.healthStatusAcquisition and a.
                            ↪ time = h.time and p in h.parameter and p.value < p.
                            ↪ threshold)
}

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


--------------- TRACK4RUN FACTS ---------------

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

fact NoLocationInMoreMaps {
        all l: Location | all m1, m2: Map | (l in m1.athletesLocation and l in m2.
            ↪ athletesLocation) implies m1 = m2
}


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

fact NoMapWithoutRun {
        all m: Map | one r: Run | r.map = m
}

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


fact MapLocationsAreRunnersLocations {
        all t : Track4RunUser | all r: Run | some l: Location |  t in r.athletes implies
            ↪  (l in r.map.athletesLocation and l in t.retrievedData.locationAcquisition
            ↪ )
}

fact MapLocationsAreRunnersLocations2 {
         all r: Run | all l: Location | some t : Track4RunUser | l in r.map.
             ↪ athletesLocation implies (t in r.athletes and l in t.retrievedData.
             ↪ locationAcquisition)
}

---------------- PREDICATES ---------------------

------------- 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 | one a: AmbulanceRequest |
                        ((h.parameter.value < h.parameter.threshold)) implies
                                        ((a.time = h.time) and (h in a.report.user.
                                            ↪ retrievedData.healthStatusAcquisition))
}


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

pred MapLocationsAreRunnersLocations {
        all t : Track4RunUser | all r: Run | some l: Location |  t in r.athletes implies
            ↪  (l in r.map.athletesLocation and l in t.retrievedData.locationAcquisition
            ↪ )

}

----------------- ASSERTIONS -------------------

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

assert UserPrivacy {
        IndividualAnswerRegardOnlyPolicyAgreedUsers and MinimumGroupMembers
}

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

assert AmbulanceEmergency {
        AmbulanceRequestSent
}

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

assert WatchingAthletesPosition {
        MapLocationsAreRunnersLocations
}

------------------------------------------------
pred show {
        IndividualAnswerRegardOnlyPolicyAgreedUsers and MinimumGroupMembers and
            ↪ AmbulanceRequestSent
        and MapLocationsAreRunnersLocations
}

run show for 10 but exactly 2 Track4RunSpectator,
exactly 2 IndividualInformationAnswer, exactly 1 GroupRequest,
exactly 1 GroupInformationAnswer, exactly 1 AmbulanceRequest,
2 IndividualRequest

check WatchingAthletesPosition for 20
check AmbulanceEmergency for 20
check UserPrivacy for 20
```

4.2 Results



```
Executing "Run show for 10 but exactly 2 Track4RunSpectator, exactly 2 IndividualInformationAnswer,
  Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
  56242 vars. 2591 primary vars. 122176 clauses. 270ms.
  Instance found. Predicate is consistent. 522ms.

Executing "Check WatchingAthletesPosition for 20"
  Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
  538232 vars. 14338 primary vars. 1380014 clauses. 5010ms.
  No counterexample found. Assertion may be valid. 3144ms.

Executing "Check AmbulanceEmergency for 20"
  Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
  539072 vars. 14318 primary vars. 1387658 clauses. 6252ms.
  No counterexample found. Assertion may be valid. 20281ms.

Executing "Check UserPrivacy for 20"
  Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
  535777 vars. 14298 primary vars. 1392517 clauses. 5468ms.
  No counterexample found. Assertion may be valid. 1771ms.

4 commands were executed. The results are:
  #1: Instance found. show is consistent.
  #2: No counterexample found. WatchingAthletesPosition may be valid.
  #3: No counterexample found. AmbulanceEmergency may be valid.
  #4: No counterexample found. UserPrivacy may be valid.
```

Figure 31: Alloy Results

4.3 Generated World

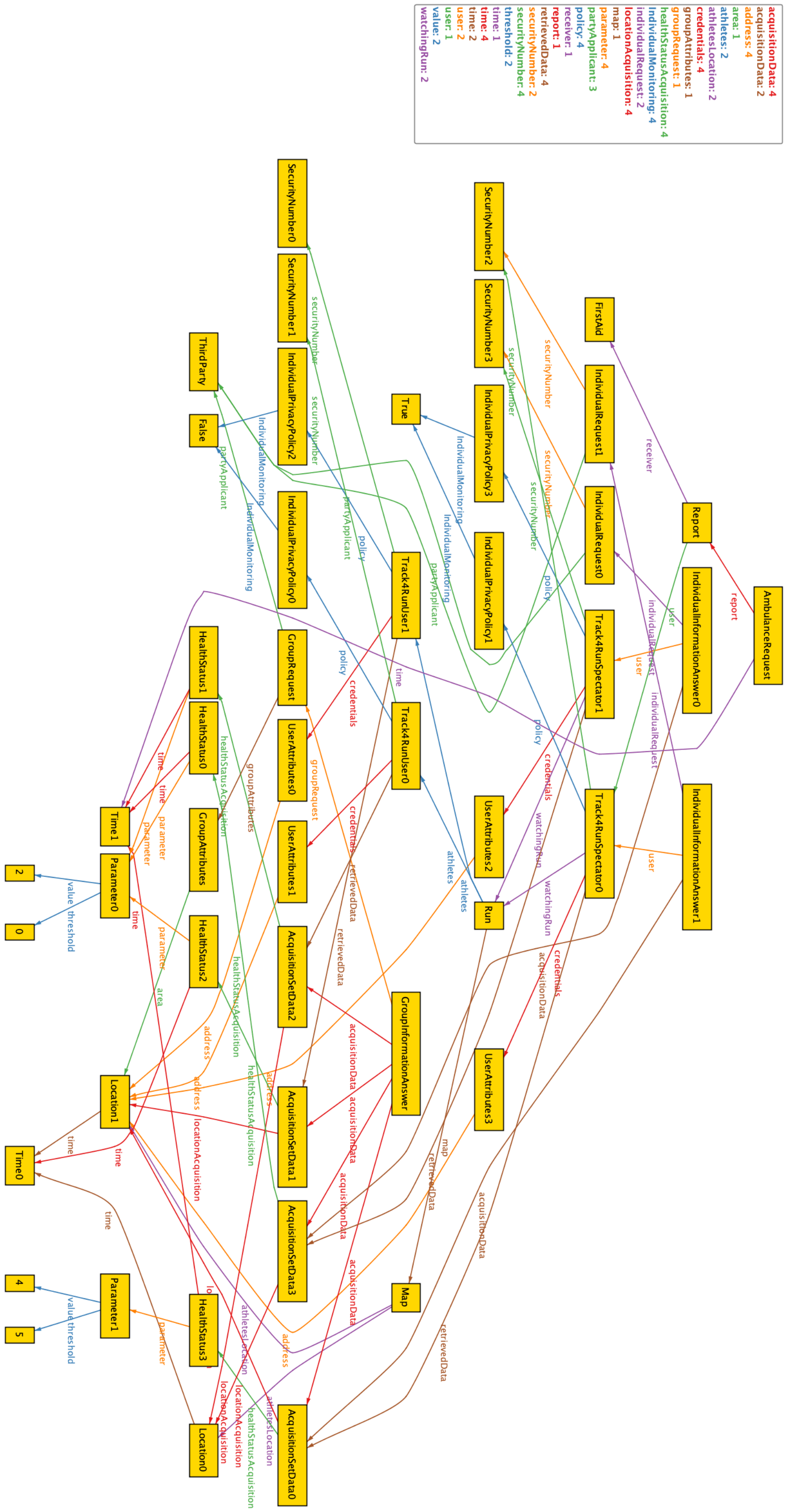


Figure 32: Sample of world generated by Alloy

5 Effort Spent

This section contains information about how many hours each group member has 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

Table 21: Effort Spent Luca Alessandrelli

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 Revisiong		8
Total		46

Table 22: Effort Spent Andrea Caraffa

5.0.3 Andrea Bionda

<table>
<tr><th>/</th><th>Task</th><th>Hours</th></tr>
<tr><td colspan="2">Text Assumptions</td><td>3</td></tr>
<tr><td colspan="2">Goals and Introduction</td><td>6</td></tr>
<tr><td colspan="2">Domain Assumptions</td><td>3</td></tr>
<tr><td colspan="2">Functional requirements</td><td>13</td></tr>
<tr><td colspan="2">Class Diagram</td><td>5</td></tr>
<tr><td colspan="2">Sequence Diagram</td><td>5</td></tr>
<tr><td colspan="2">Performance requirements and Constraints</td><td>3</td></tr>
<tr><td colspan="2">Alloy</td><td>7</td></tr>
<tr><td colspan="2">Total</td><td>45</td></tr>
</table>

Table 23: Effort Spent Andrea Bionda

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