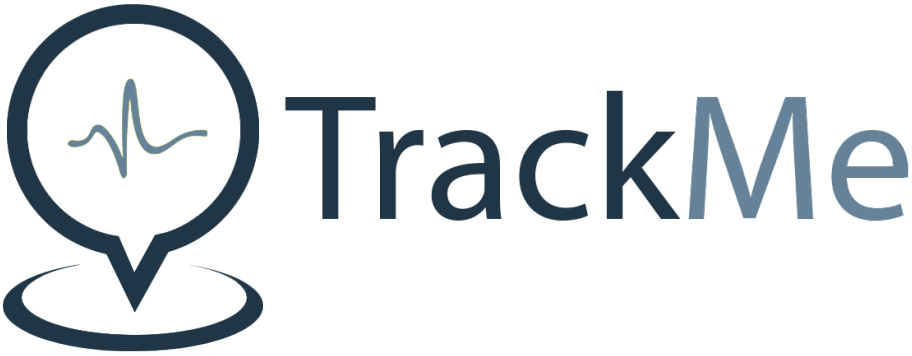
POLITECNICO
MILANO 1863

Computer Science and Engineering
Software Engineering 2 Project




# Requirement Analysis and Specification Document

Gargano Jacopo Pio, Giannetti Cristian, Haag Federico

11 November 2018

GitHub Repository: https://github.com/federicohaag/GarganoGiannettiHaag

Version 1.0

# Contents

# Chapter 1

# Introduction

## 1.1 Purpose

TrackMe wants to offer a service named "Data4Help" on top of which will be built two services named "AutomatedSOS" and "Track4Run".

**Data4Help:** the basic idea behind Data4Help is to acquire the location and health data of *Users* through *Smart wearables* connected to a smartphone. Moreover, data can be directly sent to *Third Party* customers who pay for the service. In order to analyze *Users data*, these need to obtain *User* authorization. Furthermore, they can request anonymized data of a group of *Users*.

**AutomatedSOS:** a service offered only to subscribed customers that constantly monitors their health status. Its purpose is to identify when a *User* is in need of immediate assistance and send an ambulance to their location.

**Track4Run:** a service used to track runners participating in running competitions. *Organizers* will be able to define a path for the run, *Participants* will share their position and health data and *Spectators* may watch the competition on their smart devices.

## 1.2 Scope

TrackMe offers its services in a world where technology and health are taking huge strides forward every day and innovation is commonplace.
Nowadays, people use smart devices such as smartphones and Smart wearables more than any other object that they own. This means that any activity they perform already is or can be integrated with these devices.
TrackMe, with the introduction of Data4Help, offers the possibility to monitor users' location and health data and allows third parties to register in the system

to acquire these data.

When it comes to personal data acquisition, privacy is a fundamental issue that TrackMe needs to consider. Privacy is, in fact, regulated by several laws: there are many restrictions on how user's data is acquired and stored. Therefore, TrackMe is concerned with users' consent to transferring data to TrackMe itself and to third parties for individual specific analysis. Moreover, TrackMe guarantees that anonymized data of groups of individuals are properly anonymized by checking specific constraints.

Over the course of their daily routine, users perform several actions during which their data can be analyzed to provide them with insights. For instance, they might want to monitor their heart rate while sleeping or to keep track of the distance they have walked during their day and the places they have been to.

People with a potential need for immediate assistance have always been a huge concern for their relatives and for technology makers. These may include old people with limited movement and a high chance to need urgent assistance, anyone who has a specific disease, but also a healthy individual who can suffer from a sudden heart failure. Until now, the only practical way to receive help has been to call for help, either by using a cell phone or by pushing an SOS button on a dedicated device. TrackMe proposes to automatize the step of calling for help through AutomatedSOS. In fact, when determined health values will no more be considered as normal, the system will automatically send a request for help.

Furthermore, nowadays when it comes to sports and working out, having the possibility of collecting and sharing athletes' data is a disruptive innovation. In fact, giving anybody the possibility of having on their smartphone an accurate analysis of their health while performing a work out session is a breakthrough. A sport that is practiced and loved by many is running. Organizing a run requires several steps to be taken such as defining a path, getting athletes to participate and spectators to watch it. TrackMe proposes to simplify the organization of a run, by introducing Track4Run. This service will allow the definition of a path, easy enrollment for participants and a real-time tracking of each runner's position on a map.

### 1.2.1 Analysis of Shared Phenomena

1. *User* moves.
2. *User* can have health problems.
3. *Smart wearable* sensors acquire data.
4. *Smart wearables* communicate.

5. *Smart wearable* sensors break.
6. *Third parties* collect data from Data4Help.
7. *Third parties* register to Data4Help.
8. *User* grants direct usage of personal data.
9. *User* adds a new *Service*.
10. *Organizers* define a path for a *Run*.
11. *Participant* enrolls in a *Run*.
12. *Spectators* of a *Run* see on a map the position of the runners.

## 1.3 Goals

**Data4Help**

$G_1$ Collect *User Data* through *Smart Wearables*.

$G_2$ Send specific *User Data* to *Third Parties* only if *User* consent was given after *Third Party* access request.

$G_3$ Send anonimyzed requested *Group Data* to *Third Parties* if the group it refers to is made up of 1000 or more *Users*.

$G_4$ Send *Users Data* and *Group Data* to subscribed authorized *Third Parties* as soon as they are produced.

$G_5$ Allow *Users* to manage their subscription to *Services* and to Data4Help.

**AutomatedSOS**

$G_6$ Analyze *User data* to check whether or not a *User* is a *User in need*.

$G_7$ Send an ambulance to the last position of a *User in need*.

**Track4Run**

$G_8$ Allow *Organizers* to create a *Run*, defining a path.

$G_9$ Allow *Users* to enroll in a *Run* as *Participants*.

$G_{10}$ Allow *Spectators* to watch a *Run*.

## 1.4 Definitions, Acronyms, Abbreviations

### 1.4.1 Definitions

**User:** registered individual of Data4Help who agreed on the acquisition and processing of their data (see *User Data*).

**User Data:** *User*'s health data and location acquired by Data4Help

**Third party:** a company that is willing to access *User data* stored in TrackMe's database.

**Service:** application available for some Data4Help *Users*, generally offered by a Third party.

**Group Data:** set of *Users data* acquired by Data4Help. The set of *Users* is determined by specific characteristics and constraints defined by the *Third Party* requesting the data. When sent to the *Third Party*, this data is anonymized.

**Smart wearable:** smart devices that can be worn on the body as accessories. These devices are required to have specific sensors for data acquisition, to be compatible with the system to be. The adjective 'smart' refers to the possibility of connecting them to an external device, such as a smartphone, and to the ability of operating autonomously even if not connected.

**Anomalous data:** health data that is outside certain intervals; these intervals identify a *User* normal health condition.

**User in need:** registered user of AutomatedSOS in need of assistance since their health data is *amomalous*.

**Run:** running competition registered on Track4Run.

**Organizer:** company or private person organizing a *Run*.

**Spectator:** person participating as spectator of a *Run*.

**Participant:** *User* subscribed to Track4Run participating in a *Run*.

**Username:** *User*'s email address.

### 1.4.2 Acronyms

**GPS:** Global Positioning Service

### 1.4.3 Abbreviations

**$G_n$:** $n^{th}$ goal

**$D_n$:** $n^{th}$ domain assumption

**$R_n$:** $n^{th}$ requirement

**$S_n$:** $n^{th}$ scenario

**$PR_n$:** $n^{th}$ performance requirement

## 1.5 Revision History

1. Version 1.0 - $11^{th}$ November 2018

## 1.6 Reference Documents

TO DO DURING THE WRITING OF THIS DOCUMENT

## 1.7 Document Structure

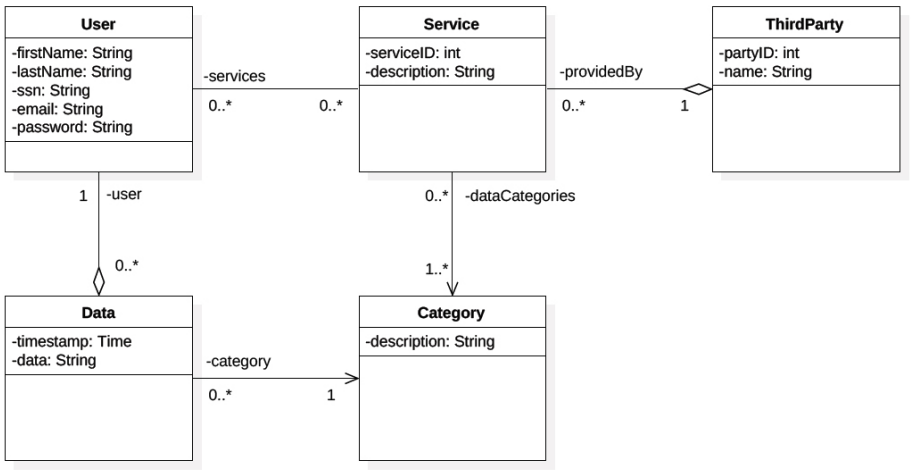
WORK IN PROGRESS

# Chapter 2

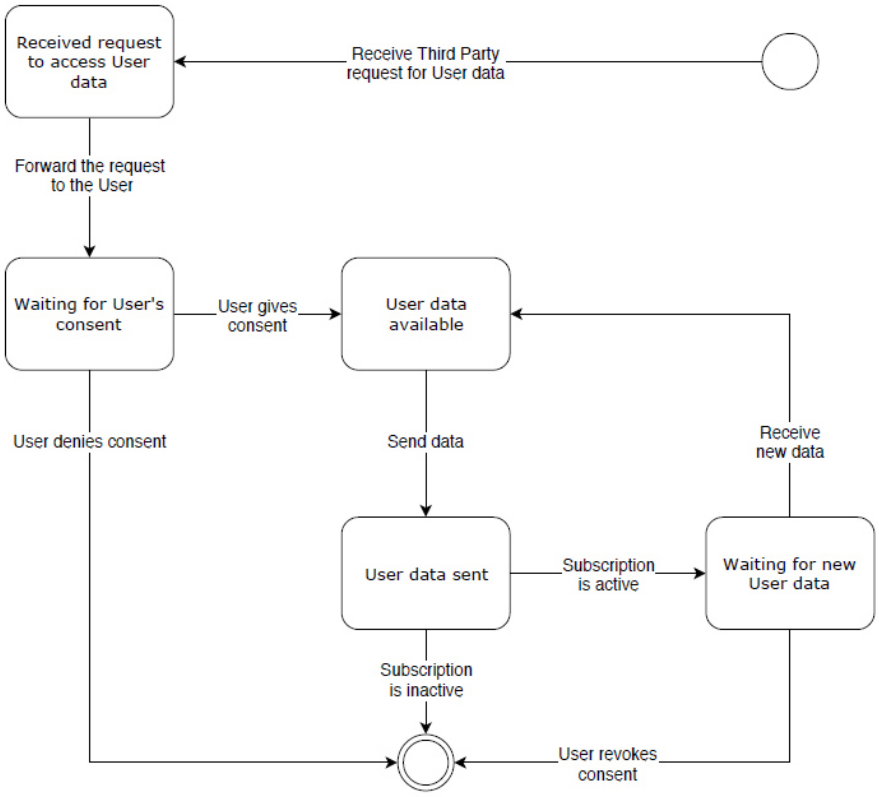
# Overall Description

## 2.1 Product perspective

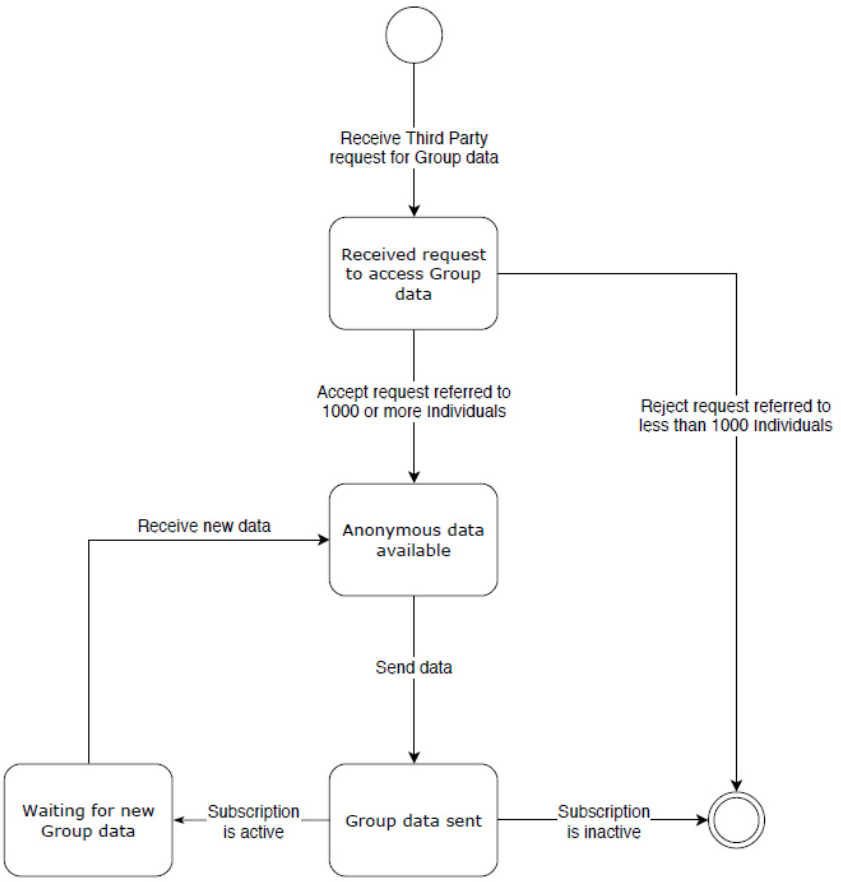
**Data4Help**




Data4Help class diagram



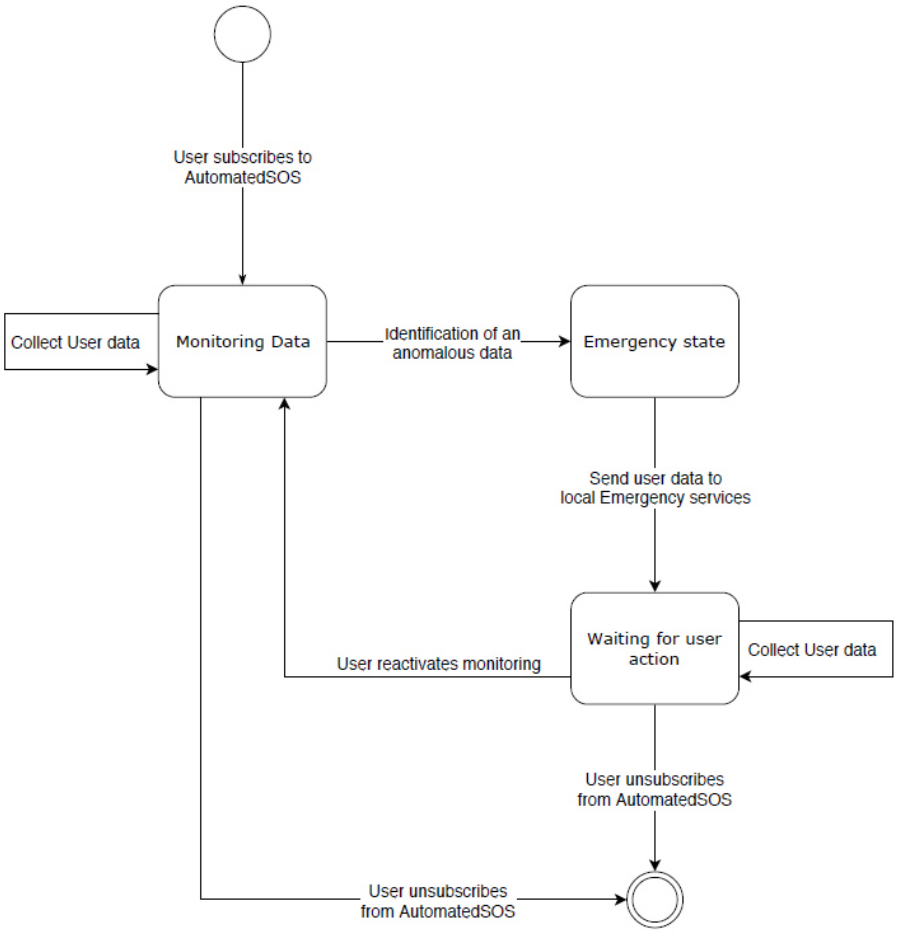

Data4Help state chart referred to *User data* request




Data4Help state chart referred to *Group data* request

- *User* subscribes to Data4Help
- *User* log in to Data4Help account on the app
- *User* adds a new *Service* on personal Data4Help account granting to it the direct access of data
- *User* unsubscribes from Data4Help
- *Smart wearables* send data to Data4Help system

**AutomatedSOS**

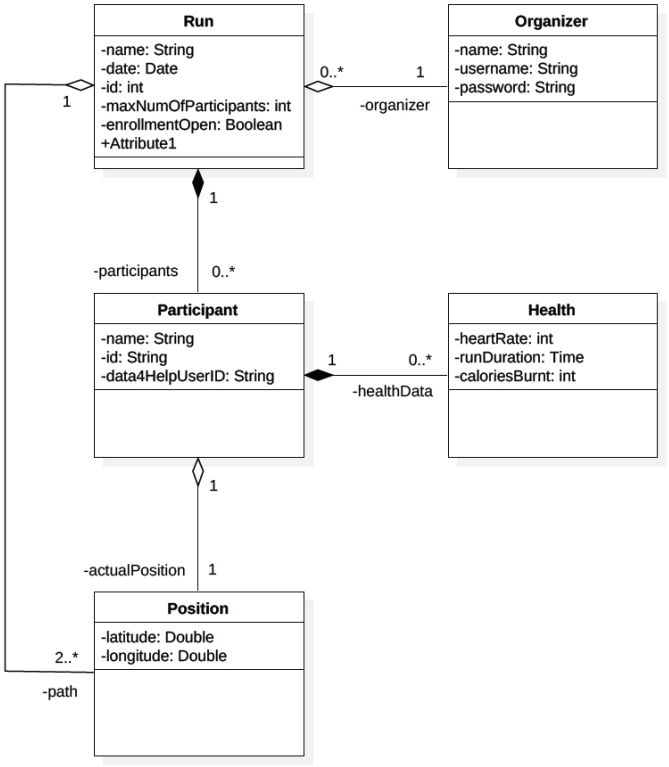



AutomatedSOS state chart

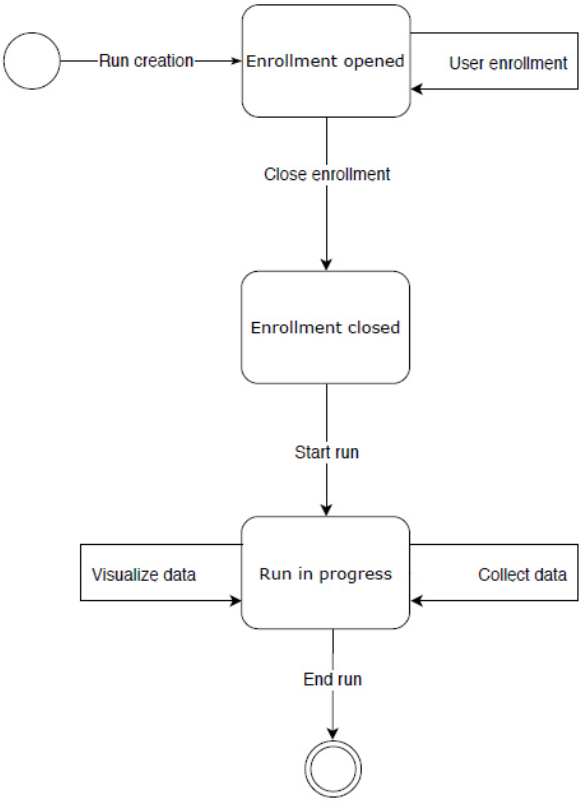
The product, because it is built on top of Data4Help, inherites from it the same shared phenomena.

- *User* becomes a *User in need*
- The system calls an ambulance

**Track4Run**




Track4Run class diagram




Track4Run state chart

The product, because it is built on top of Data4Help, inherites from it the same shared phenomena.

- *Organizer* sets a path for a running competition
- *User* enrolls to a running competition
- *Spectators* watches the *Participants*' tracking map

## 2.2 Product functions

### 2.2.1 Data4Help

#### User Registration

Data4Help will allow individuals to register. These will register by entering all the required information (see $R_2$). When registering to Data4Help, an individual will first declare to have read the privacy statement and secondly they

will have to accept the terms and conditions, which specifically include their consent to the acquisition and processing of their data, including sensitive ones, by TrackMe.
The *User* registration process will be carried out on the Data4Help application (see Section 3.1.1).

#### Third Party Registration

A *Third party* may register to Data4Help through the *Third party* dedicated website, including all required information (see $R_3$). Once terms and conditions have been accepted by the *Third party*, it will be successfully registered to the service.

#### User Data Acquisition

Data4Help will acquire *User data* through *Smart wearables*.
*Users* must give consent to the acquisition of their data when registering to Data4Help.
Data acquisition frequency can be changed according to *Third party* needs. For instance, if a *Third party* would like to track more accurately the position of a *User*, a higher location acquisition frequency can be requested.

#### Third Party Data Request

Once a *Third party* is registered to Data4Help, it can request access to *Users data* acquired through Data4Help and stored by TrackMe. *Third parties* may request data that refers either to a specific individual - *User data* - or to a group of *Users* identified by common characteristics - *Group data*.
Consent to individual data access is left to the specific *User*, who can either give or deny it to a *Third party* request.
*Group data* will be shared with *Third parties* as long as TrackMe will be able to anonymize it properly (see $R_{23}$).

#### Data Management and Privacy

All data acquired through Data4Help will be stored on a database accessible only by TrackMe. Each piece of *Users data* will have a list of *Services* offered by *Third parties* to whom access was granted by the *User*. At any time, a *User* will be able to revoke the previously given consent to any *Third party* or to TrackMe. Moreover, a *User* may exercise their right to data portability, which means that TrackMe will have to provide them with all the collected data regarding them (see $R_{32}$ and $R_{34}$). Finally, *Users* may ask the deletion of all their data stored by TrackMe (see $R_{33}$ and $R_{35}$).
By guaranteeing these functions, Data4Help will respect existing general regulations on privacy (e.g. EU GDPR).

### 2.2.2 AutomatedSOS

#### User Subscription

All Data4Help *Users* may subscribe to AutomatedSOS through Data4Help application (see Section 3.1.1).

#### Health Status Monitoring

The service will constantly monitor *User*'s health data to verify if it is *Anomalous*. Data acquisition frequency can be tweaked only by Data4Help, however AutomatedSOS may request a different value according to *User* needs.

#### Calling an Ambulance

In case the health status of a subscribed *User* is considered not to be good, AutomatedSOS will make a call to local emergency services within 5 seconds and send an ambulance to the last registered location of the *User*.

### 2.2.3 Track4Run

#### User Registration

Track4Run will be a service used by three different kinds of *Users*: *Organizers*, *Participants* and *Spectators*.*Organizers* will register to Track4Run by filling in all required information in the organizers registration form (see $R_{39}$). *Participants* will enroll in the *Run* through the Data4Help application. *Spectators* may navigate to the *Spectators* dedicated website and select the *Run* they wish to watch from the list. If they are not able to find it, they just need to know the *Run* name or identifier and insert it in the search box at the top of the page (see Section 3.1.1).

#### Run Creation and Path Definition

*Organizers* have the ability of creating a *Run*. They will be able to give the *Run* a name, set a date and time the *Run* is going to be held on and define a path for it. Moreover, they may limit the number of *Participants* or decide when to close enrollment (see $R_{41}$).

#### Display Runners on Map

Track4Run will display a map with the real time position of all the *Participants* during a *Run*. *Spectators* may watch a *Run* by inserting its name or identifier. Real time statistics of *Participants* will be shown (e.g. heart rate, rankings).

## 2.3 User characteristics

### 2.3.1 Data4Help

**Users:** People having at least a device with a sensor connected to the Internet, willing to share their data (see *User data* in Section 1.4.1) with TrackMe so as to use the *Services* built on top of Data4Help.

**Third parties:** Companies or private persons willing to collect bulk data. This data is mainly used for building *Services* on top of Data4Help; for many of these *Services* it is very important that data is transferred in real time. Otherwise data may be used for statistical analysis. In both cases, *Third parties* need that collected data is correct and accurate.

### 2.3.2 AutomatedSOS

**Users:** People with a high probability of needing immediate assistance. AutomatedSOS users are willing to monitor their health parameters and GPS location to prevent finding themselves alone when in need. These are mainly elderly people, especially those living by themselves. However, all categories of people may want to use AutomatedSOS, specifically those suffering from a disease that may strike any moment.

### 2.3.3 Track4Run

**Organizers:** Companies or private persons organizing *Runs* willing to better engage *Spectators* giving them the possibility to track in real-time the position of all *Participants*. They need to provide this *Service* easily in order to ensure *Spectators* and *Participants* are not prevented from using it.

**Participants:** People participating in a *Run*. They need to have a small device with no required interaction during the *Run* so as to avoid distractions.

**Spectators:** People participating as spectators of *Runs*. They are willing to enjoy the event by tracking *Participants* during all the *Run*. Watching a *Run* must be easy: no need of particular devices or installed applications.

## 2.4 Assumptions, Dependencies and Constraints

### 2.4.1 Domain Assumptions

$D_1$ Personal data inserted by the *User* at sign up corresponds to their real data.

$D_2$ *User data* collected at a certain instant corresponds to the actual status (GPS position and health data) of the *User* at that precise moment.

$D_3$ The maps in use accurately represent the world.

$D_4$ A *Third Party* can receive consent to *User data* access only through a *Service* it offers and can use the data only for that specific *Service*.

$D_5$ AutomatedSOS and Track4Run are *Services* developed by TrackMe.

$D_6$ AutomatedSOS and Track4Run are subscribed to new data.

$D_7$ When AutomatedSOS needs to send an ambulance to a *User in need* it forwards the request to local emergency services, which eventually dispatch an ambulance.

$D_8$ *Smart Wearables* are correctly worn by *Users*.

$D_9$ Data4Help and all *Services*, including AutomatedSOS and Track4Run, are always online.

$D_{10}$ *Users* own a working smartphone which is always connected to the Internet.

$D_{11}$ *Users* own a working *Smart wearable* which is always connected to the *User*'s smartphone.

$D_{12}$ Either smartphones or *Smart wearable* have a working and active GPS.

### 2.4.2 Dependencies

DA FARE

### 2.4.3 Constraints

**Data4Help**

- Smartphones must always be online.
- Smartphones or *Smart wearables* must have GPS activated.
- Smartphones must be able to communicate with *Smart wearables* (e.g. via Bluetooth).
- Smartphones must have enough free space for downloading and installing the Data4Help application.

**AutomatedSOS**

- *User* must be wearing *Smart wearable* at all times to allow for *Anomalous data* detection.
- AutomatedSOS must always be able to call local emergency services.

**Track4Run**

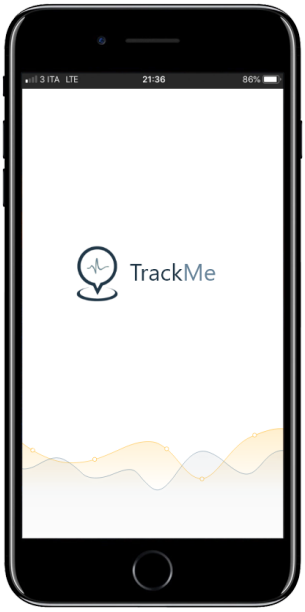
- *Organizers* and *Spectators* must have a modern browser installed on a device connected to the Internet in order to access respectively the *Organizers* administrator panel and the *Spectators* dedicated website.
- Only *Users* may enroll in a *Run*: individuals not registered to Data4Help cannot enroll in a *Run*.

# Chapter 3

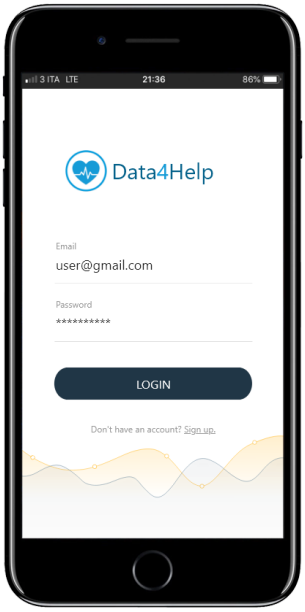
# Specific Requirements

## 3.1 External Interface Requirements

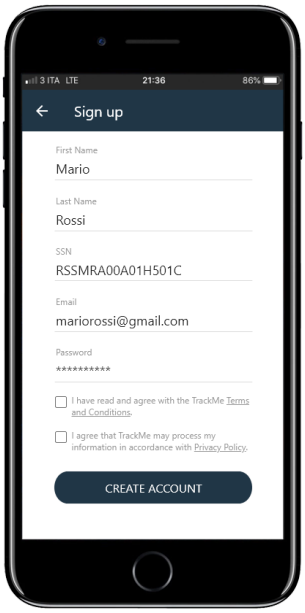
### 3.1.1 User Interfaces



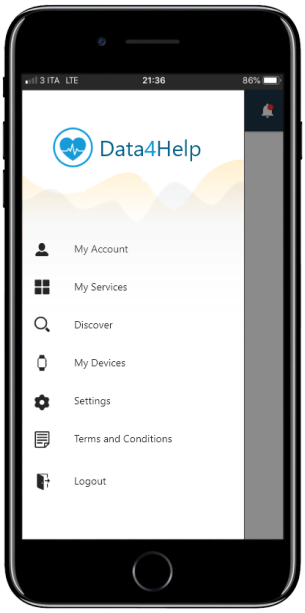

Data4Help Welcome Page




Data4Help Login Page

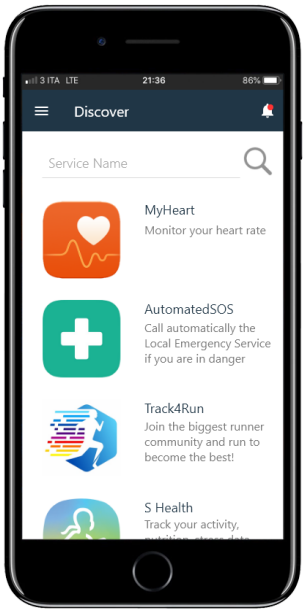



Data4Help Sign Up Page

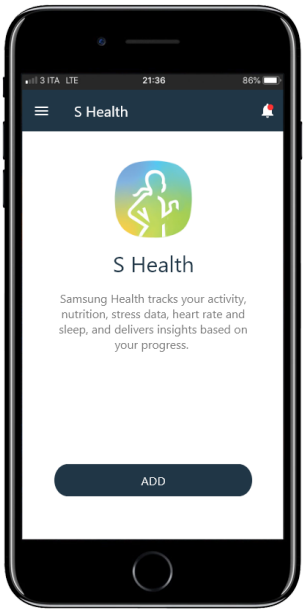



Data4Help Menu

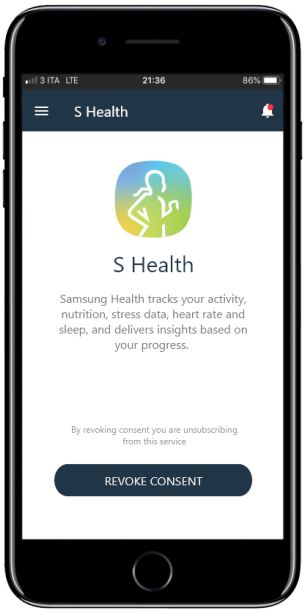

User Services Page



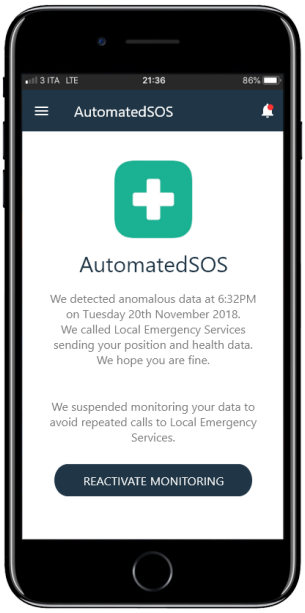

Services Discovery Page



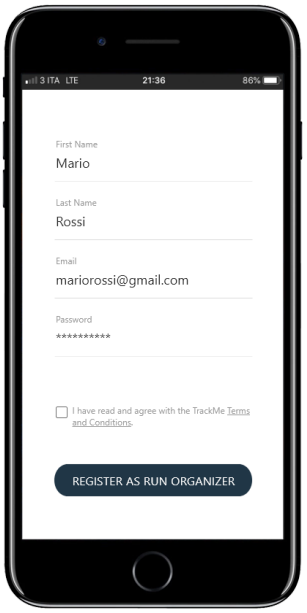

Add Service Page



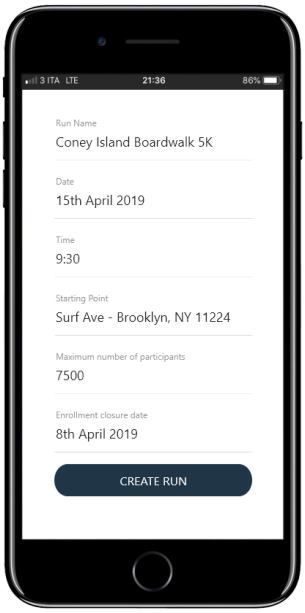

User Subscribed Service Page



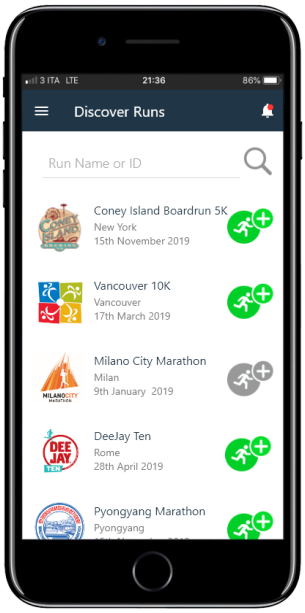

AutomatedSOS Page



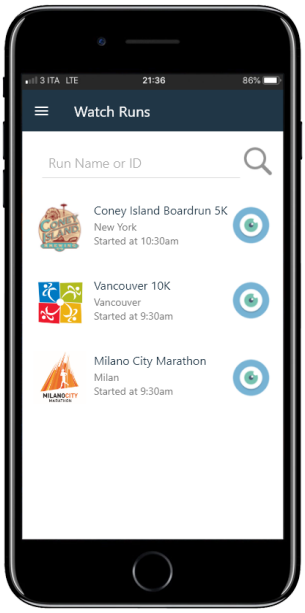

Organizer Registration Page




New Run Creation Page




Runs Discovery Page




Spectators Run Discovery Page


Watch Real Time Run Page

### 3.1.2 Hardware Interfaces

Data4Help is a service based on native applications for smartphones and on a centralized backend handling the retrieval of data from users and their storage. Data4Help acts only as an intermediary between the sources of data and the parties that want to get access to those. Due to this, Data4Help has no hardware interfaces.
Obviously also the services built on top of it, being pure software adds-on working on top of basic API's of Data4Help, don't have hardware interfaces.

### 3.1.3 Software Interfaces

All the following requests contain a parameter named *serviceID* that is the ID of the service that is performing the action. In case of requests of data, the data retrieved is relative to the service corresponding to the given serviceID. Moreover, each request contains the authentication digest (add reference to HMAC standard in 3.5.3 security).

- GET - */user?username=...&serviceID=...*
  Retrieves all records available for the given *username*
- GET - */user?username=...&quantity=...&serviceID=...*
  Retrieves last n records for the given *username*. The quantity parameter must be an integer.
- GET - */group?filter...&serviceID=...*
  Retrieves all records available for all people using the given service filtered according to ....
  Decidere quali sono i modi possibili di filtrare i dati collettivi
- PUT - */user?username=...&serviceID=...*
  Request the specified user authorization for subscribing to data updates.
- DELETE - */user?username=...&serviceID=...*
  Unsubscribes the service from the data updates of the given user.

### 3.1.4 Communication Interfaces

- smart wearables
- google maps
- bluetooth
- internet

## 3.2 Functional Requirements

**Data4Help**

$R_1$ Unregistered individuals and companies must not be able to use Data4Help.

$R_2$ At sign up, *User* must provide: first name, last name, SSN, email and password.

$R_3$ At sign up, *Third Party* must provide a company name.

$R_4$ At sign up, *User* must accept terms and conditions, including the privacy statement.

$R_5$ At sign up, *Third Party* must accept terms and conditions.

$R_6$ Identify a *User* by their identifier.

$R_7$ Query the database for a *User* by their identifier.

$R_8$ Receive *User Data*.

$R_9$ Validate *User Data*.

$R_{10}$ Authenticate *User Data*.

$R_{11}$ Store collected *User Data* in a database.

$R_{12}$ Retrieve specific *User Data* by database querying based on *User* identification.

$R_{13}$ Receive *Third Party* data access request.

$R_{14}$ Validate *Third Party* data access request.

$R_{15}$ Authenticate *Third Party* data access request.

$R_{16}$ Forward *User Data* access request to the specific *User*.

$R_{17}$ Receive *User* consent approval or denial.

$R_{18}$ Check if a specific *User* gave consent to a specific *Service*.

$R_{19}$ Send specific *User Data* to the requesting *Third Party*.

$R_{20}$ Not send specific *User Data* to the requesting *Third Party* if the specific *User* denied consent.

$R_{21}$ *Third Party* must be able to set specific constraints to define a group of *Users*.

$R_{22}$ Check how many *Users* requested *Group Data* refers to.

$R_{23}$ Properly anonymize *Group Data*.

$R_{24}$ Send *Group Data* to the requesting *Third Party*.

$R_{25}$ Not send *Group Data* if the group it refers to is made up of less than 1000 *Users*.

$R_{26}$ Receive *Third Party* subscription request.

$R_{27}$ Validate *Third Party* subscription request.

$R_{28}$ Authenticate *Third Party* subscription request.

$R_{29}$ Automatically send new data to subscribed authorized *Third Parties* as soon as they are produced.

$R_{30}$ Allow *Users* to subscribe to *Services*.

$R_{31}$ Allow *Users* to unsubscribe from *Services*.

$R_{32}$ Send a specific *User* all their data stored, if requested by them.

$R_{33}$ Delete a *User* specific data, if requested by them.

$R_{34}$ Allow *Users* to request all their data stored by TrackMe at any time.

$R_{35}$ Allow *Users* to request the deletion of all their data stored by TrackMe at any time.

**AutomatedSOS**

$R_{36}$ Compare *User Data* against certain thresholds.

$R_{37}$ Call local emergency services providing necessary *User Data* of *User in need*.

$R_{38}$ *User* must be able to reactivate AutomatedSOS monitoring.

**Track4Run**

$R_{39}$ At sign up, *Organizers* must provide: first name, last name, email and password.

$R_{40}$ At sign up, *Organizers* must accept terms and conditions.

$R_{41}$ Allow *Organizers* to create a *Run*, defining: name, path, date, maximum number of *Participants* and enrollment closure date.

$R_{42}$ Provide *Users* enrollment for an existing *Run*.

$R_{43}$ Prevent a *User* from enrolling in a *Run* if the maximum number of *Participants* was already reached.

$R_{44}$ Prevent a *User* from enrolling in a *Run* if it already started or finished.

$R_{45}$ Prevent a *User* from enrolling in a *Run* if enrollment is closed.

$R_{46}$ Show a *Run* by displaying the position of *Participants* on a map.

$R_{47}$ Identify a *Run* by its identifier.

$R_{48}$ Query the database for a *Run* given its identifier.

### 3.2.1 Satisfying Goals

**Data4Help**

$G_1$ Collect *User Data* through *Smart Wearables*.

- $R_4$ At sign up, *User* must accept terms and conditions, including the privacy statement.
- $R_6$ Identify a *User* by their identifier.
- $R_8$ Receive *User Data*.
- $R_9$ Validate *User Data*.
- $R_{10}$ Authenticate *User Data*.
- $R_{11}$ Store collected *User Data* in a database.

- $D_2$ *User data* collected at a certain instant corresponds to the actual status (GPS position and health data) of the *User* at that precise moment.
- $D_3$ The maps in use accurately represent the world.
- $D_8$ *Smart Wearables* are correctly worn by *Users*.
- $D_9$ Data4Help and all *Services*, including AutomatedSOS and Track4Run, are always online.
- $D_{10}$ *Users* own a working smartphone which is always connected to the Internet.
- $D_{11}$ *Users* own a working *Smart Wearable* which is always connected to the *User*'s smartphone.
- $D_{12}$ Either smartphones or *Smart wearable* have a working and active GPS.

- $U_1$ *User Sign Up*
- $U_3$ *User Login*

$G_2$ Send specific *User Data* to *Third Parties* only if *User* consent was given after *Third Party* access request.

- $R_1$ Unregistered individuals and companies must not be able to use Data4Help.

$R_6$ Identify a *User* by their identifier.

$R_7$ Query the database for a *User* by their identifier.

$R_{11}$ Store collected *User Data* in a database.

$R_{12}$ Retrieve specific *User Data* by database querying based on *User* identification.

$R_{13}$ Receive *Third Party* data access request.

$R_{14}$ Validate *Third Party* data access request.

$R_{15}$ Authenticate *Third Party* data access request.

$R_{16}$ Forward *User Data* access request to the specific *User*.

$R_{17}$ Receive *User* consent approval or denial.

$R_{18}$ Check if a specific *User* gave consent to a specific *Service*.

$R_{19}$ Send specific *User Data* to the requesting *Third Party*.

$R_{20}$ Not send specific *User Data* to the requesting *Third Party* if the specific *User* denied consent.

$U_2$ *Third Party Sign Up*

$U_4$ *Third Party Login*

$U_5$ *Third Party requests User Data*

$U_{10}$ *User revokes consent to a Service*

$G_3$ Send anonymized requested *Group Data* to *Third Parties* if the group it refers to is made up of 1000 or more *Users*.

$R_{11}$ Store collected *User Data* in a database.

$R_{12}$ Retrieve specific *User Data* by database querying based on *User* identification.

$R_{13}$ Receive *Third Party* data access request.

$R_{14}$ Validate *Third Party* data access request.

$R_{15}$ Authenticate *Third Party* data access request.

$R_{21}$ *Third Party* must be able to set specific constraints to define a group of *Users*.

$R_{22}$ Check how many *Users* requested *Group Data* refers to.

$R_{23}$ Properly anonymize *Group Data*.

$R_{24}$ Send *Group Data* to the requesting *Third Party*.

$R_{25}$ Not send *Group Data* if the group it refers to is made up of less than 1000 *Users*.

$U_2$ *Third Party Sign Up*

$U_4$ *Third Party Login*

$U_6$ *Third Party requests Group Data*

$G_4$ Send *Users Data* and *Group Data* to subscribed authorized *Third Parties* as soon as they are produced.

$R_7$ Query the database for a *User* by their identifier.

$R_8$ Receive *User Data*.

$R_9$ Validate *User Data*.

$R_{10}$ Authenticate *User Data*.

$R_{18}$ Check if a specific *User* gave consent to a specific *Service*.

$R_{19}$ Send specific *User Data* to the requesting *Third Party*.

$R_{20}$ Not send specific *User Data* to the requesting *Third Party* if the specific *User* denied consent.

$R_{21}$ *Third Party* must be able to set specific constraints to define a group of *Users*.

$R_{22}$ Check how many *Users* requested *Group Data* refers to.

$R_{23}$ Properly anonymize *Group Data*.

$R_{24}$ Send *Group Data* to the requesting *Third Party*.

$R_{25}$ Not send *Group Data* if the group it refers to is made up of less than 1000 *Users*.

$R_{26}$ Receive *Third Party* subscription request.

$R_{27}$ Validate *Third Party* subscription request.

$R_{28}$ Authenticate *Third Party* subscription request.

$R_{29}$ Automatically send new data to subscribed authorized *Third Parties* as soon as they are produced.

$D_9$ Data4Help and all *Services*, including AutomatedSOS and Track4Run, are always online.

$D_{10}$ *Users* own a working smartphone which is always connected to the Internet.

$D_{11}$ *Users* own a working *Smart Wearable* which is always connected to the *User*'s smartphone.

$U_7$ *Third Party subscribes to New User Data*

$U_8$ *Third Party subscribes to New Group Data*

$U_{10}$ *User revokes consent to a Service*

$G_5$ Allow *Users* to manage their subscription to *Services* and to Data4Help.

$R_4$ At sign up, *User* must accept terms and conditions, including the privacy statement.

$R_6$ Identify a *User* by their identifier.

$R_7$ Query the database for a *User* by their identifier.

$R_{12}$ Retrieve specific *User Data* by database querying based on *User* identification.

$R_{30}$ Allow *Users* to subscribe to *Services*.

$R_{31}$ Allow *Users* to unsubscribe from *Services*.

$R_{32}$ Send a specific *User* all their data stored, if requested by them.

$R_{33}$ Delete a *User* specific data, if requested by them.

$R_{34}$ Allow *Users* to request all their data stored by TrackMe at any time.

$R_{35}$ Allow *Users* to request the deletion of all their data stored by TrackMe at any time.

$U_9$ *User subscribes to a Service*

$U_{10}$ *User revokes consent to a Service*

**AutomatedSOS**

$G_6$ Analyze *User data* to check whether or not a *User* is a *User in need*.

$R_6$ Identify a *User* by their identifier.

$R_7$ Query the database for a *User* by their identifier.

$R_8$ Receive *User Data*.

$R_9$ Validate *User Data*.

$R_{10}$ Authenticate *User Data*.

$R_{29}$ Automatically send new data to subscribed authorized *Third Parties* as soon as they are produced.

$R_{36}$ Compare *User Data* against certain thresholds.

$R_{38}$ *User* must be able to reactivate AutomatedSOS monitoring.

$D_2$ *User data* collected at a certain instant corresponds to the actual status (GPS position and health data) of the *User* at that precise moment.

$D_6$ AutomatedSOS and Track4Run are subscribed to new data.

$D_9$ Data4Help and all *Services*, including AutomatedSOS and Track4Run, are always online.

$D_{10}$ *Users* own a working smartphone which is always connected to the Internet.

$D_{11}$ *Users* own a working *Smart Wearable* which is always connected to the *User*'s smartphone.

$D_{12}$ Either smartphones or *Smart wearable* have a working and active GPS.

$U_{11}$ User in need assisted by AutomatedSOS

$G_7$ Send an ambulance to the last position of a *User in need*.

$R_6$ Identify a *User* by their identifier.

$R_7$ Query the database for a *User* by their identifier.

$R_{11}$ Store collected *User Data* in a database.

$R_{12}$ Retrieve specific *User Data* by database querying based on *User* identification.

$R_{37}$ Call local emergency services providing necessary *User Data* of *User in need*.

$D_3$ The maps in use accurately represent the world.

$D_6$ AutomatedSOS and Track4Run are subscribed to new data.

$D_7$ When AutomatedSOS needs to send an ambulance to a *User in need* it forwards the request to local emergency services, which eventually dispatch an ambulance.

$U_{11}$ User in need assisted by AutomatedSOS

**Track4Run**

$G_8$ Allow *Organizers* to create a *Run*, defining a path.

$R_{39}$ At sign up, *Organizers* must provide: first name, last name, email and password.

$R_{40}$ At sign up, *Organizers* must accept terms and conditions.

$R_{41}$ Allow *Organizers* to create a *Run*, defining: name, path, date, maximum number of *Participants* and enrollment closure date.

$U_{12}$ Organizer Sign Up

$U_{13}$ Organizer creates a Run

$G_9$ Allow *Users* to enroll in a *Run* as *Participants*.

$R_{42}$ Provide *Users* enrollment for an existing *Run*.

$R_{43}$ Prevent a *User* from enrolling in a *Run* if the maximum number of *Participants* was already reached.

$R_{44}$ Prevent a *User* from enrolling in a *Run* if it already started or finished.

$R_{45}$ Prevent a *User* from enrolling in a *Run* if enrollment is closed.

$R_{47}$ Identify a *Run* by its identifier.

$R_{48}$ Query the database for a *Run* given its identifier.

$U_{14}$ User enrolls in a Run

$G_{10}$ Allow *Spectators* to watch a *Run*.

$R_{46}$ Show a *Run* by displaying the position of *Participants* on a map.

$R_{47}$ Identify a *Run* by its identifier.

$R_{48}$ Query the database for a *Run* given its identifier.

$D_3$ The maps in use accurately represent the world.

$D_6$ AutomatedSOS and Track4Run are subscribed to new data.

$D_8$ *Smart Wearables* are correctly worn by *Users*.

$D_9$ Data4Help and all *Services*, including AutomatedSOS and Track4Run, are always online.

$D_{10}$ *Users* own a working smartphone which is always connected to the Internet.

$D_{11}$ *Users* own a working *Smart Wearable* which is always connected to the *User*'s smartphone.

$D_{12}$ Either smartphones or *Smart wearable* have a working and active GPS.

$U_{15}$ Spectator watches a Run

### 3.2.2 Scenarios

**Data4Help**

$S_1$ Dante is an individual who would like to keep track of his GPS position and health data. For this purpose he decides to use Data4Help. He downloads the Data4Help application on his smartphone and proceeds to sign up. He inserts all required information, which include his name, his social security number and date of birth. He is asked to insert an email that will later be his username and a password. Dante inserts his name as his password and the system tells him that the inserted password is shorter than 8 characters, so he tries again with a new one. Eventually he inserts a valid password, accepts the terms and conditions and taps on "Create an Account". He is successfully signed up, after receiving a

confirmation email by TrackMe. He tries to Loginto the application by inserting the newly created username and password. The system accepts the credentials and Dante is in.

$S_2$ YourHealth is a company that analyzes individuals' health data to provide users with insights on their well-being. It decides to offer its *Service* also on Data4Help so as to have a greater pool of users. The person in charge navigates to the *Third Party* dedicated website and clicks on "Sign Up". They fill in all required information about their company, insert an email that will be used as username and a valid password. They then accept the terms and conditions by clicking the specific checkbox, and finally click on "Create an Account". YourHealth receives an email confirming the account creation: now YourHealth *Services* are available to all Data4Help *Users*.

$S_3$ Dante, a Data4Help *User*, needs to monitor his heart rate through the day. He navigates to the "Discover" page inside of his Data4Help application and scrolls through the available *Services*. He finds MyHeart, a *Service* developed by YourHealth, a *Third Party* registered to Data4Help. The description of the service seems to suit his need, so he adds MyHeart to his *Services*. In order to finalize the subscription, Dante will have to accept that his data will be sent to YourHealth for analysis. He does so. After a while, in the specific MyHeart *Service* page, the "Analyze" button appears. Dante taps on it and promptly he sees a personalized graph showing his heart rate levels throughout the day, starting from the first day he registered to Data4Help.

$S_4$ LocalStats is a company that performs intensive statistics on individuals' positions in some cities of Switzerland. It decides to acquire individuals' GPS positions data from Data4Help to enlarge its database. LocalStats registers as a Data4Help *Third Party*. Once registration is complete, the first request it makes to Data4Help refers to all female *Users* between 30 and 35 years old living in Lausanne. Unfortunately, the number of *Users* with the requested characteristics is less than 1000, which does not guarantee proper data anonymization. Therefore, Data4Help rejects the *Group Data* request. LocalStats tries again changing the interval of interest to 25-35 years old. This time the request refers to more than 1000 *Users* and finally Data4Help can send the requested *Group Data* to LocalStats.

$S_5$ Dante, from scenario $S_3$, would like to keep MyHeart active day by day. To do so, he taps on "Analyze Daily", which is a function offered by MyHeart. YourHealth, which developed and manages MyHeart, requests subscription to Dante's new data. Data4Help registers that anytime Dante's *User data* is collected, it needs to send it to YourHealth for analysis. Starting from the following day, Dante does not need anymore to tap on "Analyze" every day: new analysis is provided to him as soon as it is available from MyHeart.

$S_6$ Dante, who subscribed to Data4Help and used its *Third Party Services* for a while, decides that he does not want to use one of them, TrackKer, anymore. Therefore, he navigates to the "My Services" page and taps on TrackKer. The *Service* page shows up and he taps on the "Revoke Consent" button at the bottom of the page. From now on, Data4Help will stop sending Dante's data to the *Third Party* managing TrackKer.

**AutomatedSOS**

$S_7$ GianVito is a 57 years old man subscribed to AutomatedSOS. After getting very angry at work, he drives home, but as soon as he gets there he feels dizzy and falls on the ground. He is alone and cannot call for help. Fortunately, AutomatedSOS notices that his heart rate is below a certain threshold and identifies him as *User in need*. AutomatedSOS calls local emergency services and sends them GianVito's position and health data. When the local emergency services dispatch an ambulance and GianVito is being taken care of, AutomatedSOS waits for GianVito's action, still collecting his data, without possibly identifying it as *Anomalous*. Finally, GianVito opens up Data4Help application and taps on "Reactivate Monitoring". AutomatedSOS has fulfilled his need for immediate assistance and starts monitoring his health data again.

**Track4Run**

$S_8$ Charity4All is a Swedish charity association that organizes a running competition every year to raise money for their causes. The person in charge decides to use Track4Run to manage the run. They navigate to the *Run* dedicated website and sign up as an *Organizer*, inserting an email and a password for registration. Once sign up is complete, they click on "Create Run" and the *Run* creation page shows up. They give the *Run* a name - Run4Char - they define a path around Gothenburg and set the date and time the competition will take place on. They do not want to limit the number of participants, so they click on "Create Run" and obtain a *Run* identifier back from Track4Run. They will distribute this identifier to all viewers who wish to enjoy the *Run* on their devices.

$S_9$ Hannah lives in Gothenburg and she loves running. In fact, she is subscribed to Track4Run. While browsing the available *Runs* in her city, she finds Run4Char - from $S_8$. She enrolls in the run right away and Track4Me records her registration. Hannah is now a *Participant* of the *Run*.

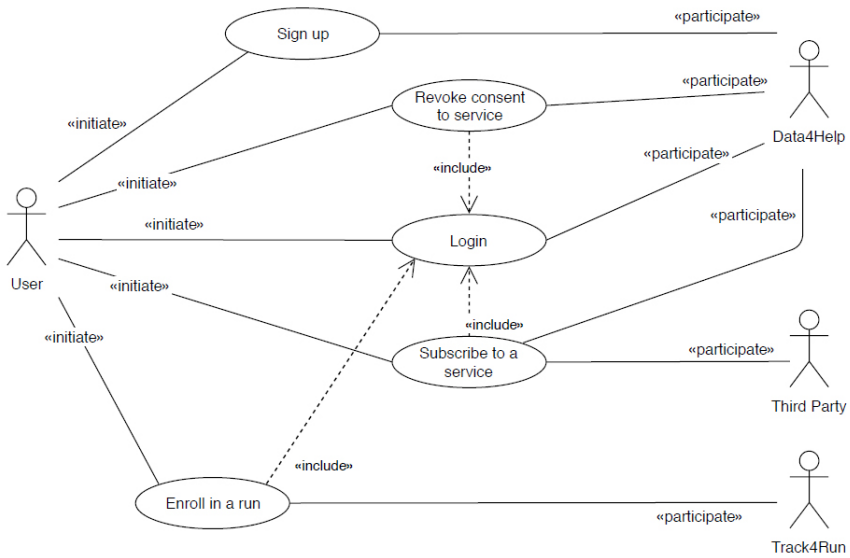
$S_{10}$ George enjoys sports a lot, however he is very old now and cannot participate in competitions anymore. He still likes watching sports event, especially when it comes to running. Since he is also into helping others, he is subscribed to Charity4All - from $S_8$ - newsletter. He reads that they are organizing a *Run* and writes down the *Run* identifier. On the day of the *Run*, he navigates to the *Spectators* dedicated website and inserts the

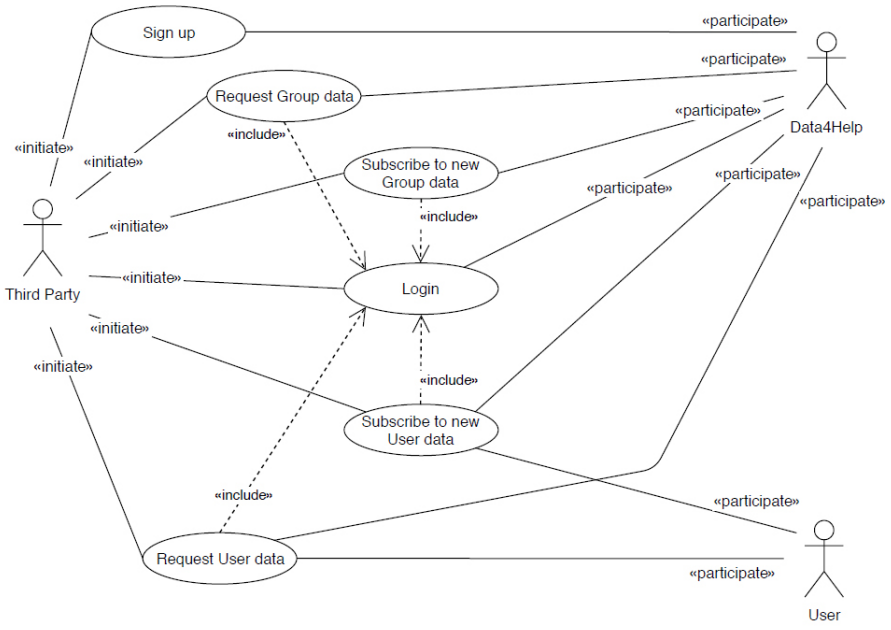
*Run* identifier. As soon as the *Run* starts, he enjoys it by watching the position of the *Participants* on the map right on his device, comfortably in his house.

### 3.2.3 Use Cases

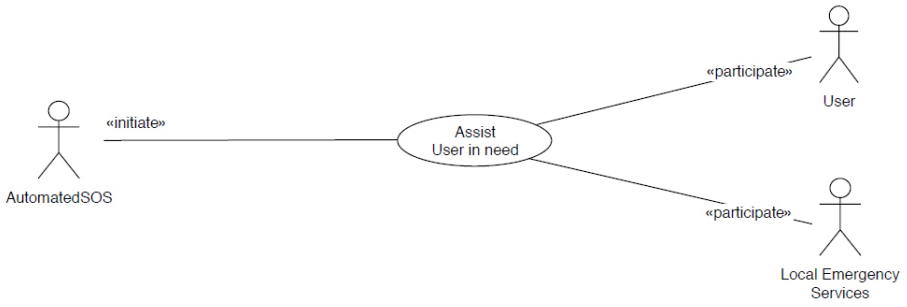
#### Use Case Diagrams



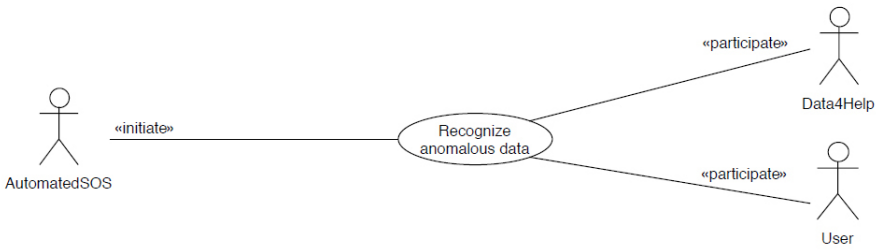

User Use Case Diagram



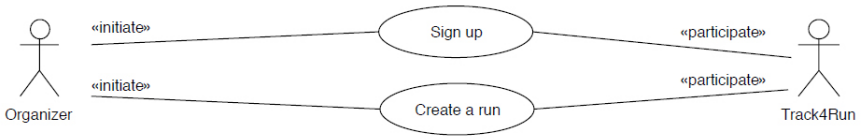

Third Party Use Case Diagram




AutomatedSOS Use Case Diagram




AutomatedSOS Use Case Diagram




Track4Run Organizer Use Case Diagram


Track4Run Spectator Use Case Diagram

**Use Case Analysis**

**Data4Help**

| **Name** | **User Sign Up** |
|---|---|
| Actors | *User*, Data4Help |
| Entry Conditions | *User* successfully installed Data4Help application on their smartphone. |

| | |
|---|---|
| Events Flow | 1. *User* taps on "Sign Up" button.<br>2. *User* fills in all required fields for *User* registration, including username and password.<br>3. *User* checks the "Accept terms and conditions" checkbox.<br>4. *User* taps on "Create an Account" button.<br>5. Data4Help saves *User* information. |
| Exit Condition | *User* successfully registered by Data4Help. |
| Exceptions | 1. Inserted email already registered for another *User*.<br>2. Inserted password is not valid.<br>3. Not all required fields are filled in.<br>4. "Accept terms and conditions" checkbox not checked.<br>5. *User* already signed up.<br>*User* is invited to try again signing up, reporting which error(s) they have committed. |

$U_1$

| **Name** | **Third Party Sign Up** |
|---|---|
| Actors | *Third Party*, Data4Help |
| Entry Conditions | *Third Party* is connected to the *Third Party* dedicated website. |

| | |
|---|---|
| Events Flow | 1. *Third Party* clicks on "Sign Up" button.<br>2. *Third Party* fills in all required fields for *Third Party* registration, including username and password.<br>3. *Third Party* checks the "Accept terms and conditions" checkbox.<br>4. *Third Party* clicks on "Create an Account" button.<br>5. Data4Help saves *Third Party* information. |
| Exit Condition | *Third Party* successfully registered by Data4Help. |
| Exceptions | 1. Inserted email already registered for another *Third Party*.<br>2. Inserted password is not valid.<br>3. Not all required fields are filled in.<br>4. "Accept terms and conditions" checkbox not checked.<br>5. *Third Party* already signed up.<br>*Third Party* is invited to try again signing up, reporting which error(s) it has committed. |

$U_2$

| **Name** | **User Login** |
|---|---|
| Actors | *User*, Data4Help |
| Entry Conditions | *User* successfully registered to Data4Help and installed Data4Help application on their smartphone. |
| Events Flow | 1. *User* enters username.<br>2. *User* enters password.<br>3. *User* taps on "Login" button.<br>4. Data4Help checks *User* credentials. |

| | |
|---|---|
| Exit Condition | *User* is successfully logged in. |
| Exceptions | 1. Inserted username is not valid.<br>2. Inserted password is not correct.<br>*User* is invited to try again logging in. |

$U_3$

| **Name** | **Third Party Login** |
|---|---|
| Actors | *Third Party*, Data4Help |
| Entry Conditions | *Third Party* successfully registered to Data4Help and is connected to the *Third Party* dedicated website. |
| Events Flow | 1. *Third Party* enters username.<br>2. *Third Party* enters password.<br>3. *Third Party* clicks on "Login" button.<br>4. Data4Help checks *Third Party* credentials. |
| Exit Condition | *Third Party* is successfully logged in. |
| Exceptions | 1. Inserted username is not valid.<br>2. Inserted password is not correct.<br>*Third Party* is invited to try again logging in . |

$U_4$

| **Name** | **Third Party requests User Data** |
|---|---|
| Actors | *Third Party*, Data4Help, *User* |
| Entry Conditions | *Third Party* and *User* successfully registered to Data4Help. |

| | |
|---|---|
| Events Flow | 1. *Third Party* requests access to specific *User data*.<br>2. Data4Help forwards the request to the specific *User* unless the consent was already given.<br>3. *User* gives consent to the requesting *Third Party* to access their data. |
| Exit Condition | Data4Help sends *User data* to the *Third Party*. |
| Exceptions | 1. *User* denies consent to their data access by the requesting *Third Party*. |

$U_5$

| **Name** | **Third Party requests Group Data** |
|---|---|
| Actors | *Third Party*, Data4Help |
| Entry Conditions | *Third Party* successfully registered to Data4Help. |
| Events Flow | 1. *Third Party* requests access to *Group data*.<br>2. Data4Help checks if the requested data refers to minimum 1000 *Users*. |
| Exit Condition | Data4Help sends *Group data* to the *Third Party*. |
| Exceptions | 1. *Group data* refers to less than 1000 *Users*.<br>Data4Help denies *Group data* access to the *Third Party*. |

$U_6$

| **Name** | **Third Party subscribes to New User Data** |
|---|---|
| Actors | *Third Party*, *User*, Data4Help |
| Entry Conditions | *Third Party* successfully registered to Data4Help and obtained access to *User data*. |

| | |
|---|---|
| Events Flow | 1. *Third Party* requests subscription to *User data*.<br>2. *User* gives consent. |
| Exit Condition | Data4Help registers the *Third Party* subscription to new data. Each time new data is produced, it is sent to the *Third Party*. |
| Exceptions | No Exceptions |

$U_7$

| **Name** | **Third Party subscribes to New Group Data** |
|---|---|
| Actors | *Third Party*, Data4Help |
| Entry Conditions | *Third Party* successfully registered to Data4Help and obtained access to *Group data*. |
| Events Flow | 1. *Third Party* requests subscription to *Group data*. |
| Exit Condition | Data4Help registers the *Third Party* subscription to new data. Each time new data is produced, it is sent to the *Third Party*. |
| Exceptions | No Exceptions |

$U_8$

| **Name** | **User subscribes to a Service** |
|---|---|
| Actors | *Third Party*, *User*, Data4Help |
| Entry Conditions | *User* is successfully registered to Data4Help and installed Data4Help application on their smartphone. |
| Events Flow | 1. *User* navigates to "Discover" page.<br>2. *User* chooses which *Service* they would like to subscribe to.<br>3. *User* taps on "Add" button.<br>4. *User* gives consent to sharing their data with the specific *Third Party*. |
| Exit Condition | Data4Help registers the new *Service* for the *User*. |

| | |
|---|---|
| Exceptions | 1. *User* does not give consent to sharing their data.<br>The *Service* is not added and the *User* is invited to try again adding it. |

$U_9$

| **Name** | **User revokes consent to a Service** |
|---|---|
| Actors | *User*, Data4Help |
| Entry Conditions | *User* gave consent to sharing their data with a *Third Party*. |
| Events Flow | 1. *User* navigates to "My Services" page.<br>2. *User* chooses which service they would like to revoke consent.<br>3. *User* navigates to the *Service* dedicated page by tapping on its name.<br>4. *User* taps on "Revoke consent" button. |
| Exit Condition | Data4Help stops sharing the data of the *User* with the specific *Third Party*. |
| Exceptions | No Exceptions. |

$U_{10}$

? ? ? Collect data –¿ probably it is not a use case, but regular product function ? ? ? User manages data (gdpr)

**AutomatedSOS**

| **Name** | **User in need assisted by AutomatedSOS** |
|---|---|
| Actors | *User*, Data4Help, AutomatedSOS, Local emergency services |
| Entry Conditions | *User* is subscribed to AutomatedSOS and installed Data4Help application on their smartphone. |

| | |
|---|---|
| Events Flow | 1. Data4Help sends *User Data* to AutomatedSOS as soon as it is produced.<br>2. AutomatedSOS identifies User data as *anomalous data*.<br>3. *User* is identified as *User in need*.<br>4. AutomatedSOS calls local emergency services requesting an ambulance.<br>5. AutomatedSOS sends *User data* including GPS location and health data to local emergency services.<br>6. Local emergency services send an ambulance to the location of the *User in need*. |
| Exit Condition | *User in need* taps on "Reactivate Monitoring" button. |
| Exceptions | 1. Local emergency services don't answer the call.<br>2. The call to local emergency services is answered but the communication fails before giving all the necessary details.<br>AutomatedSOS repeats the call. |

$U_{11}$

**Track4Run**

| **Name** | **Organizer Sign Up** |
|---|---|
| Actors | *Organizer*, Track4Run |
| Entry Conditions | *Organizer* is connected to the Track4Run website dedicated to organizers. |

| | |
|---|---|
| Events Flow | 1. *Organizer* taps on "Sign Up" button.<br>2. *Organizer* fills in all required fields for *Organizer* registration, including username and password.<br>3. *Organizer* checks the "Accept terms and conditions" checkbox.<br>4. *Organizer* taps on "Create an Account" button.<br>5. Track4Run saves *Organizer* information. |
| Exit Condition | *Organizer* successfully registered by Track4Run. |
| Exceptions | 1. Inserted email already registered for another *Organizer*.<br>2. Inserted password is not valid.<br>3. Not all required fields are filled in.<br>4. "Accept terms and conditions" checkbox not checked.<br>5. *Organizer* already signed up.<br>*Organizer* is invited to try again signing up, reporting which error(s) they have committed. |

$U_{12}$

| **Name** | **Organizer creates a Run** |
|---|---|
| Actors | *Organizer*, Track4Run |
| Entry Conditions | *Organizer* is logged in and connected to the *Run* dedicated website. |
| Events Flow | 1. *Organizer* clicks on "Create Run" button.<br>2. *Organizer* fills in all required fields for *Run* creation.<br>3. *Organizer* clicks on "Confirm" button. |

| | |
|---|---|
| Exit Condition | Track4Run creates the *Run* defined by the *Organizer*. |
| Exceptions | 1. Not all required fields are filled in.<br>*Organizer* is invited to try again creating the *Run*. |

$U_{13}$

| **Name** | **User enrolls in a Run** |
|---|---|
| Actors | *User*, Track4Run |
| Entry Conditions | *User* is subscribed to Track4Run and installed Data4Help application on their smartphone. |
| Events Flow | 1. *User* navigates to "My Services" page.<br>2. *User* taps on Track4Run.<br>3. *User* taps on "Discover Runs".<br>4. *User* chooses the *Run* they wish to enroll in from the list or inserts the *Run* identifier or name in the search box.<br>5. *User* taps on "+" button to enroll in the *Run*. |
| Exit Condition | Track4Run registers the *User* as a *Participant* of the *Run*. |
| Exceptions | 1. No *Runs* are listed.<br>2. The selected *Run* reached the maximum number of participants.<br>3. The selected *Run* is closed to enrollment.<br>4. There are no *Runs* associated to the inserted run identifier or name.<br>*Participant* is invited to try again later enrolling in a *Run*. |

$U_{14}$

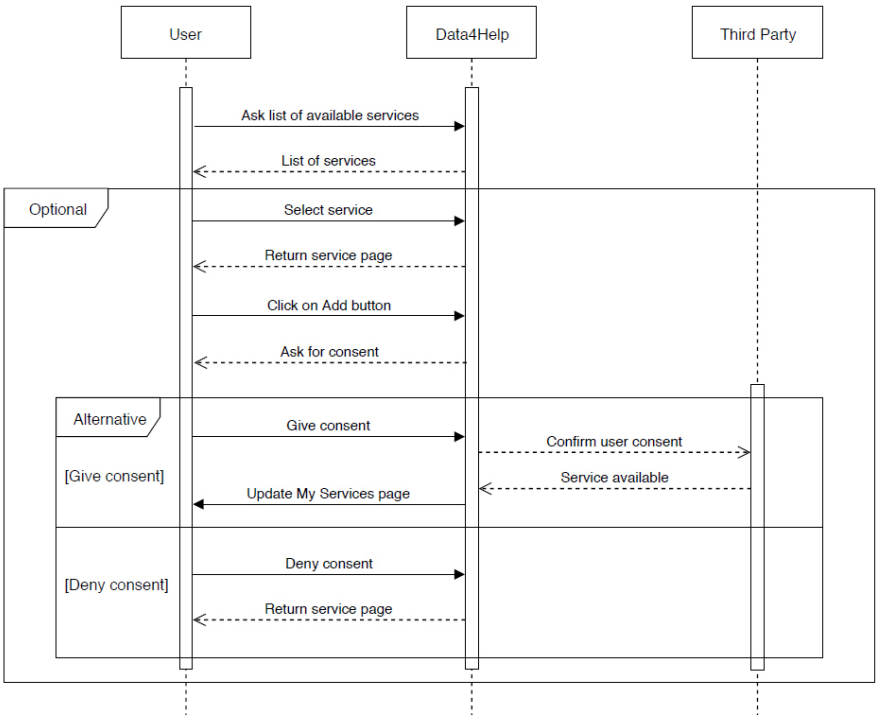
| **Name** | **Spectator watches a Run** |
|---|---|

| Actors | Individual |
|---|---|
| Entry Conditions | Individual is connected to the *Spectators* dedicated website. |
| Events Flow | 1. *Individual* clicks on the *Run* they would like to watch or inserts the run identifier or name.<br>2. *Individual* watches the *Run* as a *Spectator*. |
| Exit Condition | The *Run* is over. |
| Exceptions | 1. No *Runs* are listed.<br>2. There is no *Run* associated to the inserted run identifier or name.<br>3. The *Spectator* disconnects from the *Spectators* dedicated website.<br>Exception 1: *Spectator* is invited to try again later watching a *Run*.<br>Exception 2: no action is taken. |

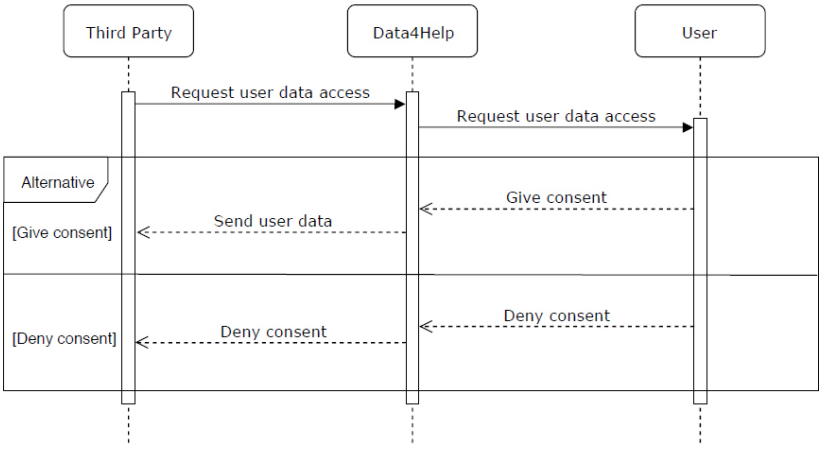
$U_{15}$

??? participant runs in a run ???

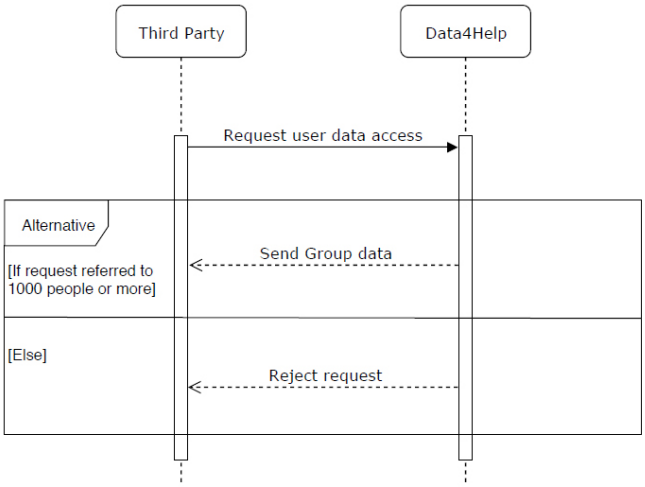
### 3.2.4 Sequence Diagrams



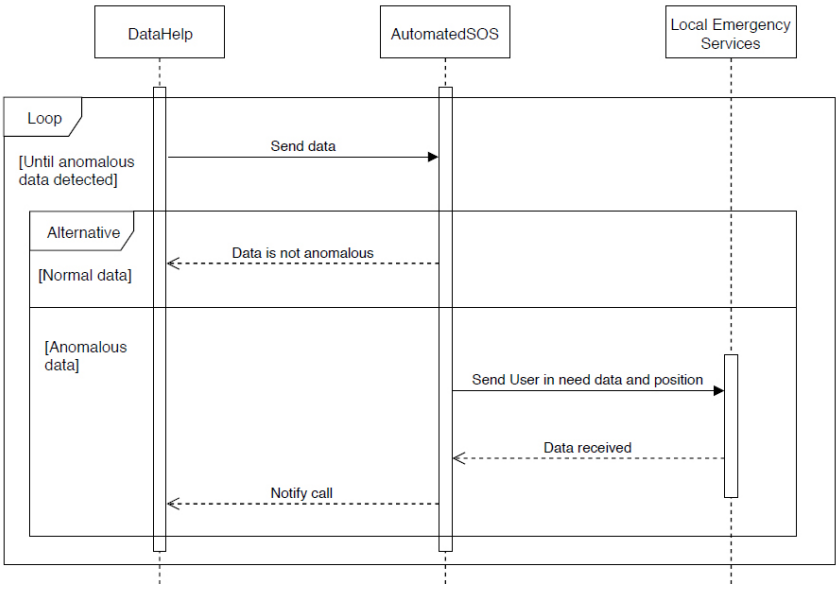

Add New Service Sequence Diagram



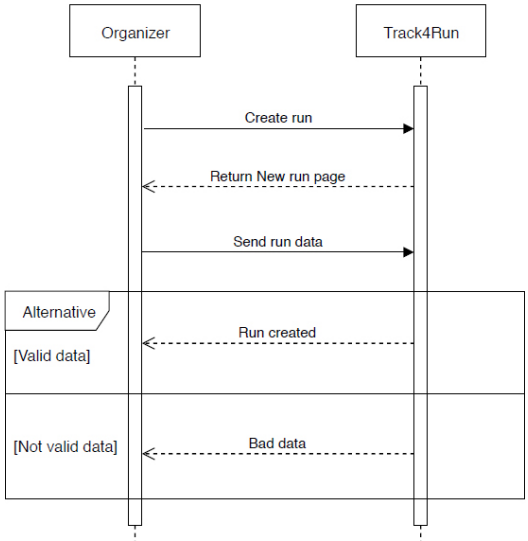

User Data Request Sequence Diagram



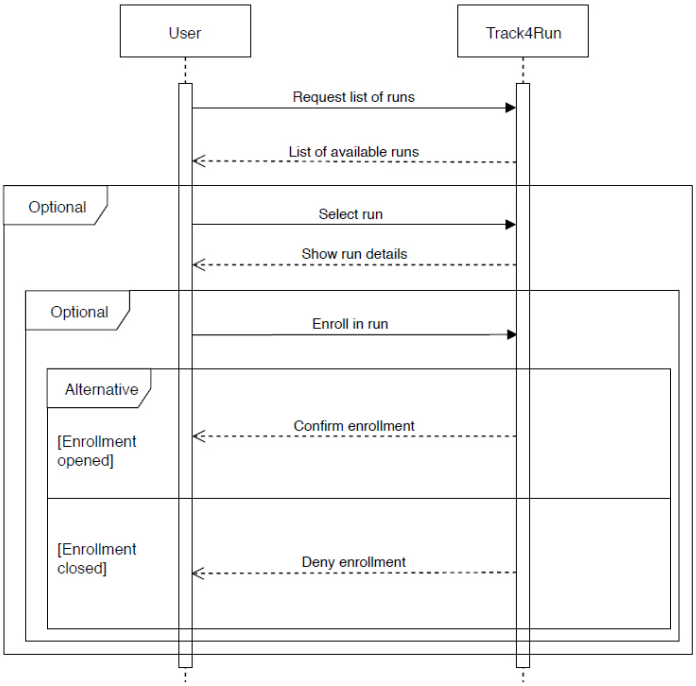

Group Data Request Sequence Diagram




User in Need Assistance Sequence Diagram




Run Organization Sequence Diagram




Enroll in a Run Sequence Diagram

## 3.3 Performance Requirements

$PR_1$ AutomatedSOS must call local emergency services within 5 seconds from the moment it identified a *User* as *User in need*.

$PR_2$ Data4Help, AutomatedSOS and Track4Run must have a high speed Internet connection.

Requirement $PR_1$ refers to the reaction time of the system to be. Being able to respect this requirement can mean saving a person's life, besides making the system to be reliable. This is why AutomatedSOS must always be ready to call local emergency services as soon as needed.

Having a high speed Internet connection - $PR_2$ - is fundamental for guaranteeing a high quality service. For each of the three services that build up the system to be, this requirement has a specific purpose:

**Data4Help:** reducing delay between data production on *User Smart wearable* and data collection and sharing with *Third parties*. In particular, AutomatedSOS subscribed *User* data collection and sharing with AutomatedSOS is crucial for satisfying $PR_1$.

**AutomatedSOS:** reducing delay between data collection by Data4Help and analysis for possible identification of a *User in need*.

**Track4Run:** increase real time accuracy while displaying *Users* positions during a *Run* for *Spectators* to watch.

The system to be depends on external parties and pieces of hardware. These have a great impact on the performance and on the functions of the system to be. It is possible to put constraints on the quality of external pieces of hardware compliant with the system to be (e.g. sensors quality and accuracy, *Smart wearable* connection to smartphone). By doing this, the system will be able to send accurate and precise data to *Services*, positively contributing to their performance level.

## 3.4 Design Constraints

### 3.4.1 Standards Compliance

- GDPR? - API REST? -

### 3.4.2 Hardware Limitations

- track4run potrebbe non funzionare se ci sono tante persone connesse alla stessa cella o se la run si svolge in un luogo ostico - velocità trx dati limita funzionalità e performance di ASOS - sensori potrebbero non essere accurate e quindi rilevare dati non precisi

### 3.4.3 Other

WORK IN PROGRESS

## 3.5 Software System Attributes

### 3.5.1 Reliability

WORK IN PROGRESS

### 3.5.2 Availability

The software must offer the maximum availability, granting its service 24/7. The lack of service must be minimal.

**AutomatedSOS** AutomatedSOS must be active 24/7. The lack of service is acceptable only if it is due to maintenance. AutomatedSOS users must have received a warning fourty-eight hours before, and they must be noticed again one hour before the service disabling.
Even in this case, the lack of service must be kept to a minimum.

### 3.5.3 Security

WORK IN PROGRESS

### 3.5.4 Maintainability

WORK IN PROGRESS

### 3.5.5 Portability

Portability of *User data* from a device to another is possible by entering personal login data, also for devices with different operating systems. Personal data and settings are stored in a database and they are downloaded when a new device is connected.

# Chapter 4

# Formal Analysis using Alloy

# Chapter 5

# Effort Spent

WORK IN PROGRESS

# Chapter 6

# References

WORK IN PROGRESS