



POLITECNICO DI MILANO

SOFTWARE ENGINEERING 2 PROJECT  
A.Y. 2018-19

**Data4Help, AutomatedSOS**  
**Requirements Analysis and Specifications**  
**Document**  
Version 1.0

Comolli Federico, 920258  
Corda Francesco, 920912

Referent professor: Di Nitto Elisabetta

November 11, 2018

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.1.1	Goals . . . . .	4
1.2	Scope . . . . .	4
1.2.1	Analysis of the phenomena . . . . .	4
1.2.2	Stakeholders . . . . .	5
1.3	Definitions, Acronyms, Abbreviations . . . . .	6
1.3.1	Definitions . . . . .	6
1.3.2	Acronyms . . . . .	6
1.3.3	Abbreviation . . . . .	6
1.4	Revision history . . . . .	7
1.5	Reference Documents . . . . .	7
1.6	Document Structure . . . . .	7
<b>2</b>	<b>Overall Description</b>	<b>8</b>
2.1	Product perspective . . . . .	8
2.1.1	Data State Diagram . . . . .	8
2.1.2	UML Class Diagram . . . . .	8
2.2	Product functions . . . . .	11
2.2.1	Data4Help . . . . .	11
2.2.2	AutomatedSOS . . . . .	11
2.3	User characteristics . . . . .	12
2.3.1	Actors . . . . .	12
2.4	Assumptions, dependencies and constraints . . . . .	13
2.4.1	Domain Assumptions . . . . .	13
<b>3</b>	<b>Specific Requirements</b>	<b>14</b>
3.1	External Interface Requirement . . . . .	14
3.1.1	User Interfaces . . . . .	14
3.1.2	Hardware Interfaces . . . . .	18
3.1.3	Software Interfaces . . . . .	18
3.1.4	Communication Interfaces . . . . .	18
3.2	Scenarios . . . . .	18
3.2.1	Scenario 1 . . . . .	18

## TABLE OF CONTENTS

---

3.2.2	Scenario 2 . . . . .	18
3.2.3	Scenario 3 . . . . .	19
3.2.4	Scenario 4 . . . . .	19
3.2.5	Scenario 5 . . . . .	20
3.2.6	Scenario 6 . . . . .	20
3.3	Functional Requirements . . . . .	21
3.4	Use Case Diagrams . . . . .	25
3.5	Sequence Diagrams . . . . .	34
3.6	Performance Requirements . . . . .	39
3.7	Design Constraints . . . . .	39
3.7.1	Standards compliance . . . . .	39
3.7.2	Hardware limitations . . . . .	39
3.7.3	Any other constraint . . . . .	39
3.8	Software System Attributes . . . . .	40
3.8.1	Reliability . . . . .	40
3.8.2	Availability . . . . .	40
3.8.3	Security . . . . .	40
3.8.4	Maintainability . . . . .	41
3.8.5	Portability . . . . .	41
<b>4</b>	<b>Formal Analysis Using Alloy</b>	<b>42</b>
4.1	Alloy Model . . . . .	42
4.2	Proof of Consistency . . . . .	46
4.3	Generated World . . . . .	47
<b>5</b>	<b>Effort Spent</b>	<b>48</b>
5.1	Comolli Federico . . . . .	48
5.2	Corda Francesco . . . . .	49
<b>6</b>	<b>References</b>	<b>50</b>

# Section 1

## Introduction

### 1.1 Purpose

TrackMe is a company that wants to create a software-based system that allows third parties to monitor the location and health status of individuals.

This service, which from now on will be referred to as Data4Help, supports the registration of individuals who, by registering, agree to the collection of their data. Data acquisition can occur through smart watches or similar electronic devices.

Third parties can request access to the data of some specific individuals of which they know a form of identification (such as the SSN or the Fiscal Code). In this case, TrackMe forwards the request to the user who can accept or refuse it.

Another service offered to third parties by Data4Help is to access to anonymous data of classes of individuals, grouped by different criteria (for example geographical area, age, gender, etc.). Anonymity is a crucial value for TrackMe, so it approves these requests only if third parties cannot go back to users' identity; for this reason requests referred to groups composed by less than 1000 individuals will be denied.

After a request has been accepted, third parties can subscribe to the reception of updated data as soon as they are produced.

Moreover, in order to exploit the features offered by Data4Help, TrackMe wants to develop a customized SOS service to elderly people. This system, called AutomatedSOS, uses the data collected by Data4Help to monitor the health status of the subscribed users. Through the real time screening, the system is able to react to the variation of the parameters by sending an ambulance to the location of the customer in case of emergency.

### 1.1.1 Goals

Data4Help:

- [G1] Allow a visitor to become registered user after providing credentials.
- [G2] Monitor the location of users through electronic devices.
- [G3] Monitor the health status of users through electronic devices.
- [G4] Allow a user to accept or refuse a request to access to personal data.
- [G5] Allow third parties to register to the system.
- [G6] Allow third parties to request an access to users' data.
  - [G6.1] Allow third parties to request access to data of some specific individuals by providing SSN or Fiscal Code.
  - [G6.2] Allow third parties to request access to anonymized data of group of people.
- [G7] Send the demanded data to third parties whose request has been approved.
- [G8] Accept request for data of group of people only if the system can guarantee the anonymity of the people.
- [G9] Allow third parties to subscribe to new data and to receive them as soon as they are produced.

AutomatedSOS:

- [G10] Allow a visitor to become registered user after providing credentials.
- [G11] Monitor the location of users through electronic devices.
- [G12] Monitor the health status of users through electronic devices.
- [G13] Send to the location of the customer very quickly an ambulance when such parameters are below certain thresholds.

## 1.2 Scope

### 1.2.1 Analysis of the phenomena

Data4Help application can be used after the registration through the interface on the electronic devices. People who are for different reasons interested in services have to provide their name, surname, gender, birth date, SSN or Fiscal Code.

Users of Data4Help have to wear some kind of electronic devices that record information about their position and their health status. Data4Help is an

application-to-be that is projected to be installed on different wearable electronic devices (smart-watches, smart-bands, etc.) and eventually can interact with other health tools such as heart rate bands, smart scales or similar.

To identify the health status, Data4Help searches for different parameters provided by sensors on smart-watches or health tools (such as heart rate, skin temperature, blood glucose level, weight, etc.). Also, Data4Help acquires the position of its users through GPS technology.

Data acquired by Data4Help may be matter of interest for third parties that can make a request using the functionality provided by the application. If a request is referred to a single user it's necessary to provide his/her SSN or Fiscal Code. In addition it's possible to request data referred to group of people (associated by age, gender, location, etc.). Data4Help has to accept or decline these requests using a defined privacy criteria.

Users who receive a request for their personal information can accept or decline it using the provided functionality.

Data4Help offers the possibility to subscribe to data in order to receive the updated ones as soon they are produced.

Furthermore, TrackMe wants to build a new application on top of Data4Help, with the purpose of exploiting its functionality to provide an emergency service. As for Data4Help, registration is necessary to use AutomatedSOS.

AutomatedSOS analyzes real-time data provided by Data4Help technology. If the software detects an anomaly in the parameters it notifies an emergency to the Operations Center of the National Health Service within 5 seconds. From now on the National Health Service is responsible for the first aid and it manages the request accordingly to its own protocol. The intervention is carried on determining the emergency level and sending an ambulance to the location of the user.

### 1.2.2 Stakeholders

In this paragraph we analyze all the entities involved in the project of Data4Help and AutomatedSOS.

TrackMe is the main stakeholder of the project being the one who commissioned the project and will pay for it. Data provided by users are the core of the two applications. Of course, this amount of data is a source of interest for third parties that wants to access it.

The National Health Service plays a fundamental role in the realization of AutomatedSOS taking care of the emergency intervention notified by the software-to-be.

## 1.3 Definitions, Acronyms, Abbreviations

### 1.3.1 Definitions

- Users: people who use the services provided by TrackMe.
- Third Parties: entity that are interested to data provided by TrackMe.
- National Health Service: the national institution that provides health care to citizens.
- Social Security Number: a nine-digit-number that identifies uniquely a citizen.
- Fiscal Code: synonym to Social Security Number for Italian people.
- Credentials: username, password, SSN or Fiscal Code.
- Anomaly: when the parameters are below a defined threshold.
- Emergency: it occurs when an anomaly is detected.
- System: defines the set of software components that implement the required functionalities.
- Real-time acquisition: the interval of time between two different acquisitions is less than 2 seconds.
- Operations Center: the public authority that coordinates and manages the first aid operations.

### 1.3.2 Acronyms

- RASD: Requirements Analysis and Specification Document.
- SSN: Social Security Number.
- GPS: Global Position System.
- API: Application Programming Interface.
- NHS: National Health Service.
- OC: Operations Center.
- TP: Third Party

### 1.3.3 Abbreviation

- [Gn]: n-th goal
- [Rn]: n-th functional requirement
- [Dn]: n-th domain assumption

## 1.4 Revision history

Version 1.0 delivered on date 11/11/2018.

## 1.5 Reference Documents

- Specification Document: “Mandatory Project Assignment AY 2018-19”.
- 29148-2011 - ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes –Requirements engineering

## 1.6 Document Structure

Section 1 gives an introduction to the problem and describes the purpose of the applications Data4Help and AutomatedSOS and states their goals. The scope of the two applications is defined.

Section 2 presents the overall description of the project. The product perspective includes details on the shared phenomena and the domain models. The class diagram describes the domain model used, and the state diagram analyzes the process of the collection and the access to data by third parties. Here the majority of functions of the system are more precisely specified, with respect to the already mentioned goals of the system. In the user characteristics section the types of actors that can use the application are described.

Section 3 contains the interfaces provided by the system: user interfaces, hardware interfaces, communication interfaces. The section contains also some scenarios that describe specific usage situations of the applications. Furthermore, the functional requirements are defined by using use cases and sequence diagrams. The non-functional requirements are defined through performance requirements, design constraints and software system attributes.

Section 4 includes the Alloy model and the discussion of its purpose. Also, few worlds generated by it are shown.

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

Section 6 includes the reference documents used to compose the project analysis.



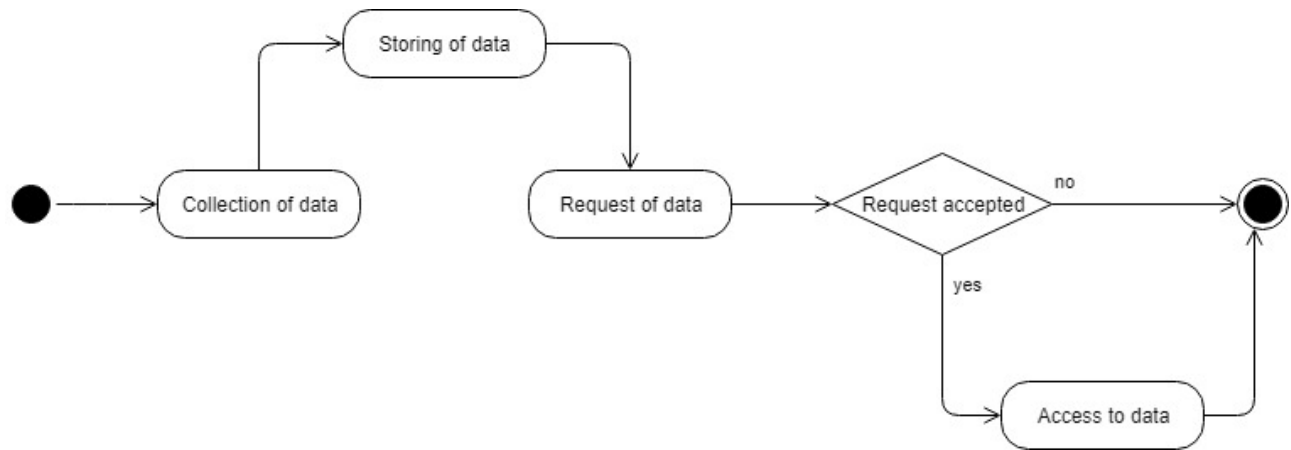
## Section 2

# Overall Description

### 2.1 Product perspective

#### 2.1.1 Data State Diagram

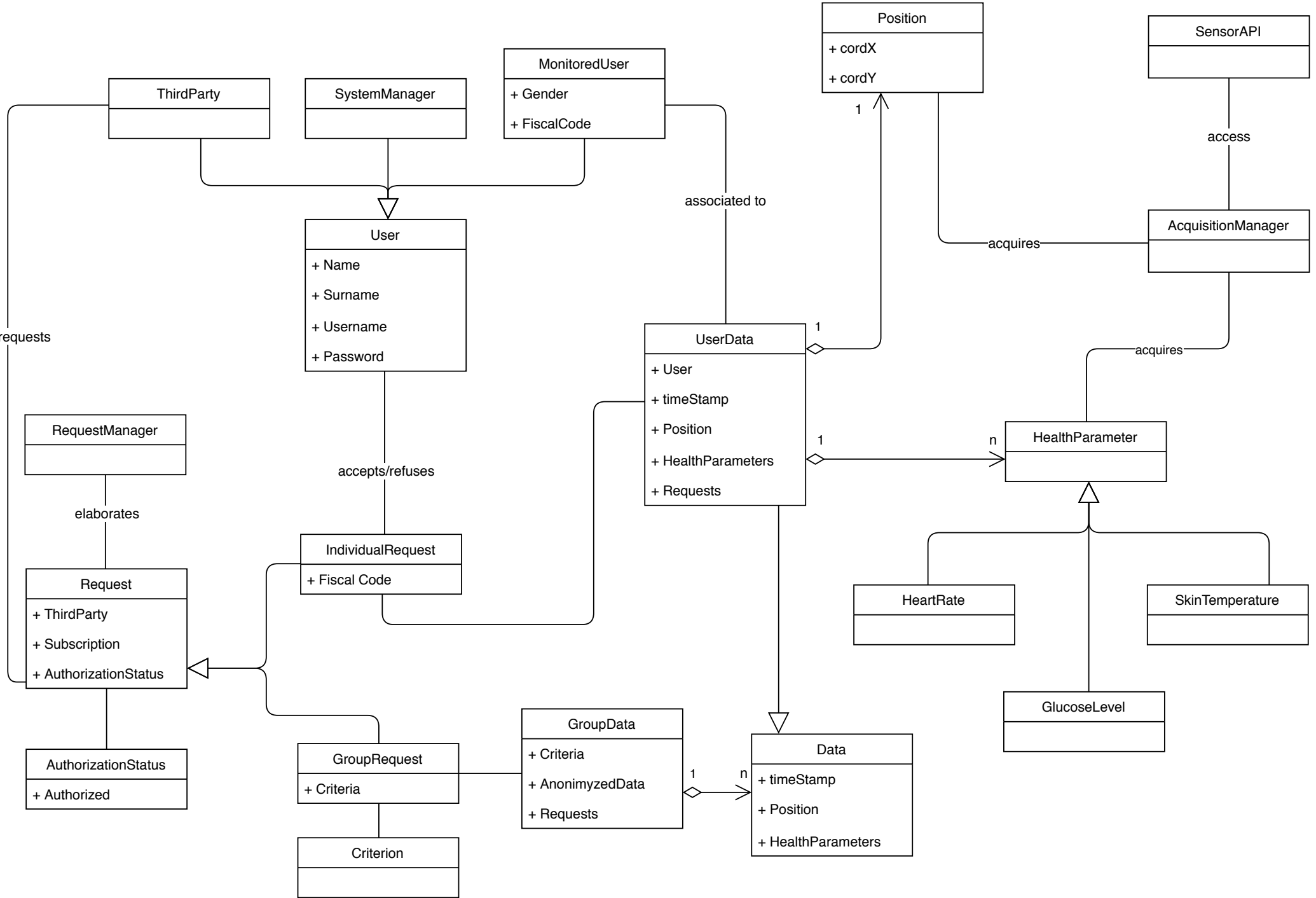
The collected data will certainly be the nucleus of the proposed system. The following state diagram displays the main actions that Data4Help will perform on the data, from the collection to an eventual access by a third party.



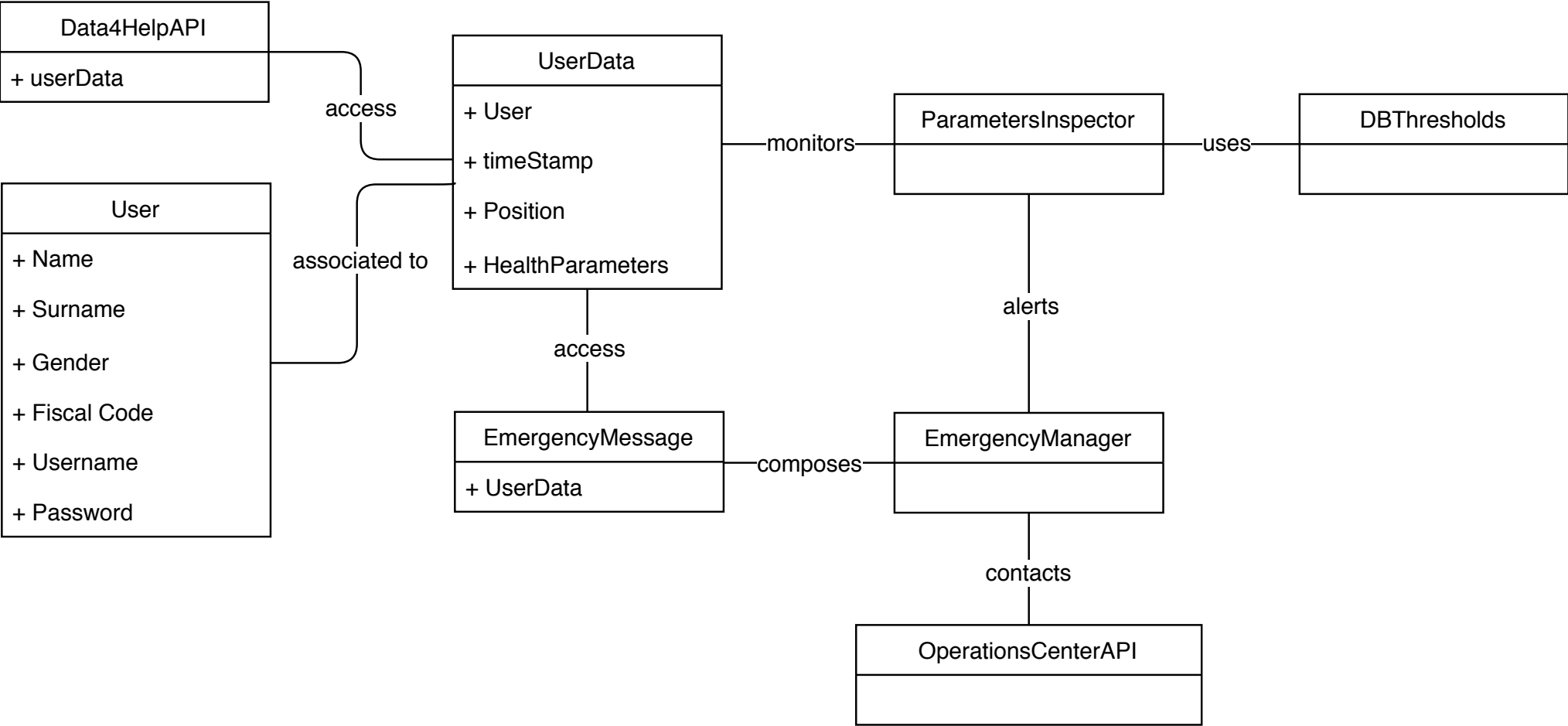
(a) General data flow from the user to a requiring third party.

#### 2.1.2 UML Class Diagram

The following class diagrams describe a possible model for the Data4Help and AutomatedSOS services:



AutomatedSOS



The diagrams showed above are intended as an high level representation of the essential features of the system. A detailed description of the communication infrastructure and the client applications will be given in the design and implementation phases.

## 2.2 Product functions

The following section lists the main functions offered by the system, focusing both on the end users and on the third parties. Of course, all the goals identified in the previous section will be offered as functions of the system-to-be.

### 2.2.1 Data4Help

#### Data collection

The core function offered by Data4Help will be the real-time acquisition of the location and the health status of the users. The system will collect parameters such as GPS position, hearth rate and temperature through the sensors' interfaces in the device OS. All the data is permanently stored by the service and will be available to those who request them.

#### Third parties access to data

The system will let registered third parties to request an access to Data4Help data. This function can be divided in two categories: access to individual data and access to anonymized group of data. Requests for individual data require to insert the SSN or Fiscal Code of the corresponding individual.

The system will offer the possibility to customize a request for group data according to different criteria (age, gender, location, etc.). Third parties can also select the option of subscribing to the data, in order to receive them as soon as they are collected by Data4Help.

#### Accept or Refuse a request for data

When a third party requests the access to individual data, the system will notify the corresponding user and ask him/her to accept or decline the request. The notification will show to the users the identity of the third party who requested their data.

### 2.2.2 AutomatedSOS

#### Users health monitoring

Using Data4Help technology, AutomatedSOS will be able to monitor in real-time the health status of its registered users. The system will compare the parameters provided by Data4Help with predefined thresholds to detect the current condition of the users.

### **Emergency intervention**

If the system detects an anomaly in the user's parameters, an emergency request will be sent to the Operations Center. AutomatedSOS will compose a message containing all the user's personal information, his/her current location and the health parameters list. The ambulance intervention won't be managed by the system, that can only guarantee to contact the NHS within 5 seconds from the anomaly detection.

## **2.3 User characteristics**

In general, users of our system are not expected to be particularly tech-savvy. It is assumed that the users are comfortable to interact with a basic application on a mobile device. On the other side, third parties are expected to have a broader technological back-ground, since they will interact with the data sent by the service. In particular, third parties who requests data of a group of people have to know how to manage the huge amount of data received by the system.

### **2.3.1 Actors**

- Visitor: a person that is not registered yet. The only action he/she can perform is enter the registration process.
- User: a person who has registered to the service and, after the login phase, can exploit all the functionalities provided by the system.
- System Manager: an employee of TrackMe in charge of maintaining and updating the system. Possible updates concerning Data4Help are changes in the privacy criteria for group requests and additions of new research categories. Further updates are modifications of parameters' thresholds in AutomatedSOS.
- Third Party: an entity who is registered to the service with the purpose of accessing to data stored by the system. After the login phase it can request data of Data4Help users by filling the apposite form: it has to select some criteria for group requests and to insert a Fiscal Code if it wants to receive the data of a single user.
- Operations Center employee: the NHS operator working at the Operations Center at the moment in which an emergency request arrives from AutomatedSOS. This actor establishes the emergency level and coordinates the ambulance intervention.

## 2.4 Assumptions, dependencies and constraints

### 2.4.1 Domain Assumptions

Data4Help:

- [D1] Users insert credentials that correspond to their identity.
- [D2] When a new registered user sends his/her credentials to the system, the message will be surely received.
- [D3] The communication channel doesn't corrupt the data sent by the user's device to the system and vice versa.
- [D4] The electronic device on which the system is installed is equipped with the GPS sensor.
- [D5] The electronic device on which the system is installed is equipped with the sensors related to the parameters tracked by Data4Help.
- [D6] The system retrieves the data from the sensors through the interfaces available on the electronic device.
- [D7] Position information provided by GPS is sufficiently accurate[1].
- [D8] The sensors related to the health status collect the data with a reasonable precision.
- [D9] It's sufficient that the number of people involved in a group query is greater than 1000 people to ensure the users' privacy.

AutomatedSOS:

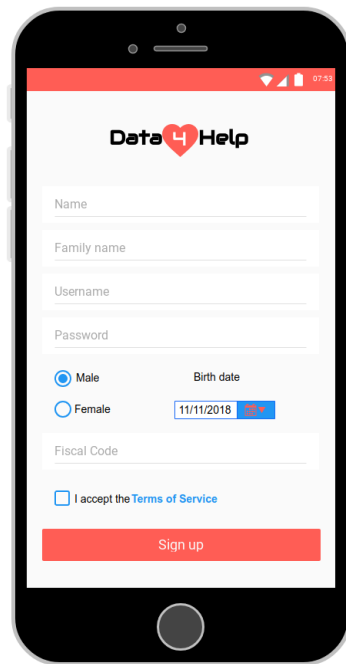
- [D10] Data provided by Data4Help API are surely received and they are not corrupted.
- [D11] When an emergency request is sent to National Health Service, it surely receive it.
- [D12] When the National Health Service receives an emergency request, at least an ambulance is available.
- [D13] When parameters are below certain thresholds, it means that the user actually needs first aid.
- [D14] Operations Centers of the NHS are up 24/7.
- [D15] When the Operations Center gather a request from AutomatedSOS, it sends at least an ambulance and at least one arrives to the location of the emergency.
- [D16] NHS provides an API that allows third parties to send emergency requests in the form of an instant message received in their platform.

## Section 3

# Specific Requirements

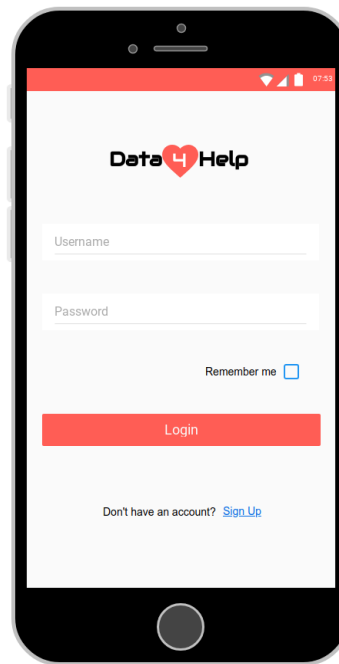
### 3.1 External Interface Requirement

#### 3.1.1 User Interfaces



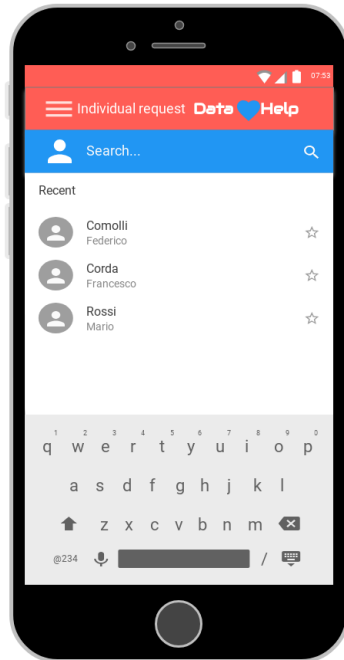
The Data4Help Sign Up screen features a red header bar with the app's logo. Below the header, there are input fields for Name, Family name, Username, and Password. A section for gender selection includes radio buttons for Male and Female, with the Male option selected. Next to it is a birth date field showing 11/11/2018 and a flag icon. Below these is a Fiscal Code field. At the bottom, there is a checkbox for accepting the Terms of Service and a red Sign up button.

(a) Data4Help Sign Up.

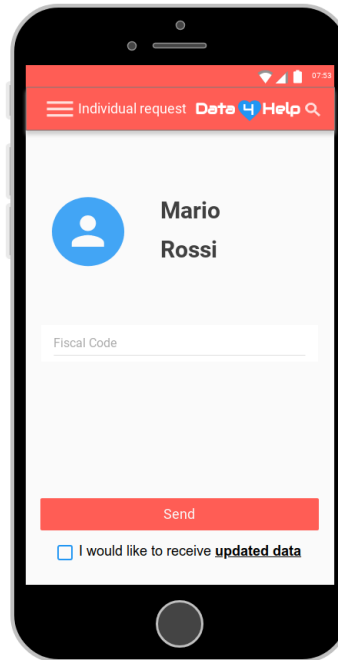


The Data4Help Login screen has a red header bar with the app's logo. It contains input fields for Username and Password. Below the password field is a Remember me checkbox. A red Login button is positioned below the input fields. At the bottom, there is a link for users who don't have an account, labeled Sign Up.

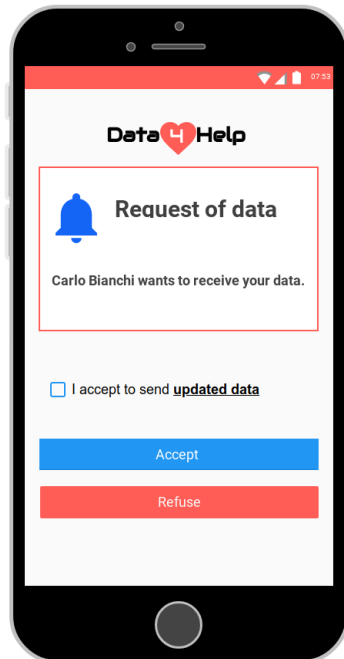
(b) Data4Help Login.



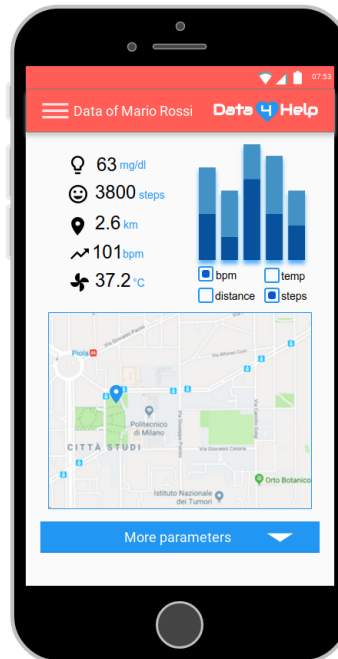
(c) TP researches a user.



(d) TP inserts a user's Fiscal Code.

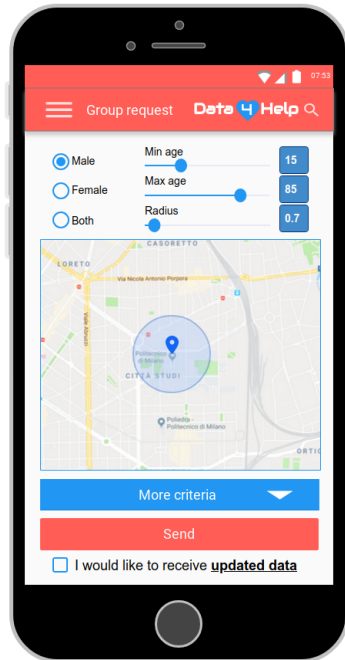


(e) User accepts a request.

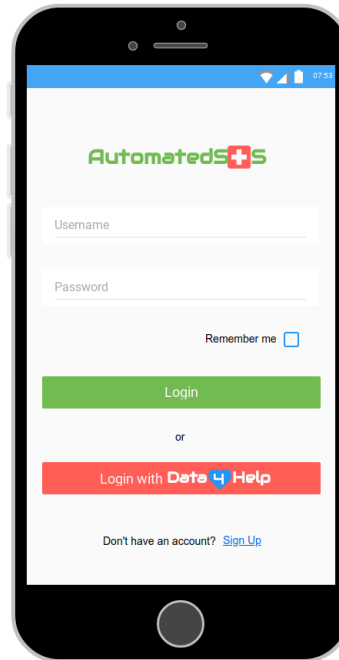


(f) TP accesses to data of a user.





(g) TP creates a group request.



(h) AutomatedSOS Login.



(i) User accesses to his/her own data.



(j) User receives the First Aid notification.

Request for group of data

Carlo Bianchi

Data4Help

Radius

LORETO CASORETTO

Via Nicola Antonio Porpora

Viale Abruzzi

Politecnico di Milano

CITTA' STUDI

Poliedra - Politecnico di Milano

ORTIC

Male

Female

Both

Min age

Max age

More criteria

☐ I would like to receive updated data

Send

(k) TP creates a group request.

Third Party

Carlo Bianchi

Data4Help

Group request #1

▼ Distance[km]	▼ Blood glucose lev[mg/dl]	▼ Heart rate[bpm]
3.8	63	76
5.6	48	120
4.7	112	89
1.9	84	65
0.7	129	101
12	172	76
2.9	84	111
6	101	92
2.1	93.7	88
1.4	122	66.6
0.1	167	87
3.7	88	118

Latest update 11/11/2018 h 23:59 UTC

Download .xls

Download .sql

(l) TP accesses to an anonymized group of data.

### **3.1.2 Hardware Interfaces**

Data4Help and AutomatedSOS are not designed to offer hardware interfaces to interact with external services.

### **3.1.3 Software Interfaces**

Data4Help will offer an API to let third parties access to users' data.

### **3.1.4 Communication Interfaces**

The system will perform its communications using the HTTPS and TCP/IP protocols.

## **3.2 Scenarios**

### **3.2.1 Scenario 1**

Nike is a worldwide brand of sportswear mainly focused on shoes. The company wants to introduce a new model of shoes addressed to younger people that live in metropolis. Carl, the research manager of the company, wants to understand the habits concerning walking of the young people who are the target for this new model of shoes. Fortunately, he has just read on a famous daily newspaper about a new service called Data4Help, which collects data of a huge amount of people all around the world.

After Carl has been registered to the service, he can make a request for data by filling the apposite form. Data4Help offers the possibility to customize a request by selecting a range for some categories. He selects as parameters of the request “people living in city with more than one million inhabitants” and “people aged between 15 and 25 years”.

TrackMe approves the request since it refers to more than 1000 people and it sends to Carl all the demanded data. Now Carl can carry on his project since he know the target's habits.

### **3.2.2 Scenario 2**

Anna is really worried since her mother's latest accident. Her mother is elderly and from now on, also due to the broken foot, has to rest to bed for some period. Martha, a friend of Anna, recommends her a new system of monitoring health status focused on elderly people.

After a short research on the net, she downloads AutomatedSOS on her mobile phone and buys a specific electronic device compatible with the app to be supplied to her mother. Anna registers her mother to the service by filling the gaps (username, password and Fiscal Code) in the interface provided by AutomatedSOS. After the confirmation message, it starts to acquire health parameters (such as position, heart rate, skin temperature, blood glucose level and so on)

and it sends them to AutomatedSOS every 2 seconds. Anna can breathe a sigh of relief: whenever her mother will be in danger, AutomatedSOS will detect it by comparing the parameters provided by the electronic device with specific thresholds.

If an emergency occurs, AutomatedSOS will send a request to the Operations Center of the NHS for an ambulance within 5 seconds. The request will include the parameters and her position, so that the ambulance can estimate the alert level, it can reach Anna's mother and can provide her first aid.

Hope it will never occur!

### 3.2.3 Scenario 3

Bianca is very caring for her son Mark, a 15 years old teenager very lively. Mark's passion is the bike and, unfortunately, he has no friends with whom he could share this sport. At least twice a week Mark, after school, takes a bike ride with his mountain-bike both for the roads and for the wood. He is always alone while he is riding so Bianca is worried about Mark safeness. Bianca's hair stylist tell her about a system that monitor real-time position of its users.

She obliges Mark to download and register to Data4Help application on his smart phone and, meanwhile, also Bianca does it. After the login, Bianca makes a request for Mark position by inserting in the apposite form his Fiscal Code. She also subscribes to receive the updated position as soon it is available.

Mark has only to accept the request of Bianca by pressing the specific button in the Data4Help interface and doing so, he can riding carelessly of his mother...she always knows his position and she is more quiet. At most every 2 seconds, the GPS on Mark's smart phone collects and sends the latest position to Data4Help system. Once the system receives a new position, it forwards it within 2 seconds to Bianca's smart-phone. Bianca now can find Mark's actual position by looking it into the Data4Help interface. Well done!

### 3.2.4 Scenario 4

Clara's life is a bit difficult since she was diagnosed with diabetes 5 years ago. Up to now she used the traditional method consisting of a blood injection three times a day, every day, to monitor her glucose level. During her last periodical check, the doctor told her about a new smart-watch for monitoring blood glucose level in a non intrusive way, detecting it through the sweat.

This smart-watch is equipped with Data4Help, an innovative system able to collect health parameters of the users. Clara must simply download Data4Help app and register with her personal information (name, surname, age, gender, Fiscal Code or SSN).

Clara's life is simpler from now on, she has just to look the parameters on her smart-watch to know in real time her health status. Another functionality of Data4Help allows Clara's doctor to monitor in real-time her patient. After registering to Data4Help as a third party, the doctor makes a request for individual data inserting Clara's name, surname and Fiscal Code. In order to receive a

continuum stream of data, the doctor makes a subscription request to Clara's parameters. Clara has to accept these requests through the functionality provided by Data4Help. The doctor has all Clara's parameters in real time and can monitor her health status. Next check up will be much faster.

### 3.2.5 Scenario 5

Tom is an avid smoker from many years and suffers of respiratory problems. For this reason he often goes to the doctor to monitor his lungs status. Tom's last visit showed a critical health status, so his doctor prescribed him to download AutomatedSOS app on his smart-watch. Tom must register to AutomatedSOS by inserting his credentials (name, surname, age, gender, Fiscal Code or SSN) in order to be monitored in real-time.

AutomatedSOS is built on top of Data4Help, a service that acquires position and health parameters of its users and makes them available to third parties. Tom's doctor uses Data4Help to monitor his patient's health status by requesting a subscription to his parameters through the apposite function. Tom can accept the request using AutomatedSOS app, since it includes all the Data4Help's functionalities.

One day, while he is on his way to work, he starts not feeling good. AutomatedSOS system detects an anomaly in the parameters and immediately finds out that that an emergency is occurring. Within 5 seconds from the moment that parameters are received, the system sends a request to the OC of the NHS.

The request includes Tom health parameters and his position so that NHS can estimates the alert level and provide him first aid. As soon as the request arrives at the OC, the situation is clear: Tom is having an heart attack. An ambulance is promptly sent to Tom's location and it arrives within 10 minutes from the moment of the illness. Few minutes more would have been fatal.

Great job AutomatedSOS!

### 3.2.6 Scenario 6

Oliver is the responsible of a sports center which includes also an athletic track. At the moment the track is not in the best condition, since it was built 10 years ago, when the sports center opened. Oliver would like to restore the track but he wants to be sure that his costumers use it frequently and that the average of the training's length is more than 5 km.

Oliver is already a user of Data4Help and knows all the functionalities provided by that service. He asks to all the customers of the sports center that use the athletic track to download and register to Data4Help application on their devices. If they don't own one, Oliver would provide them with a smart-watch. Of course, during the whole training the customers must bring with them a device equipped with Data4Help and the GPS. Now Oliver has just to make a request to access to data of a group of people.

Through the functionality provided by the app, Oliver fills the form by selecting the following criteria: "People that do physical activities from 8 AM to 10 PM"

(the opening hours of the sports center) and “People that do physical activities within 1 km far for Elizabeth road, London, UK” (the location of the sports center). Unfortunately, only 438 people corresponds to those criteria so the system denies the request made by Oliver. A negative message appears on his smart-phone showing the reason for the declined data. Oliver has to find a new way to discover if the investment is a good idea. Try again, you’ll be luckier.

### 3.3 Functional Requirements

#### **Data4Help:**

**[G1] Allow a visitor to become registered user after providing credentials.**

- [D1] Users insert credentials that correspond to their identity.
- [D2] When a new registered user sends his/her credentials to the system, the message will be surely received.
- [R1] The system should provide a registration form offering the following mandatory fields: name, surname, SSN or Fiscal Code.
- [R2] The system must guarantee that the credentials are unique.

**[G2] Monitor the location of users through electronic devices.**

- [D3] The communication channel doesn’t corrupt the data sent by the user’s device to the system and vice versa.
- [D4] The electronic device on which the system is installed is equipped with the GPS sensor.
- [D6] The system retrieves the data from the sensors through the interfaces available on the electronic device.
- [D7] Position information provided by GPS is sufficiently accurate.
- [R3] The system must acquire real-time users’ positions from the GPS installed on the user’s device.
- [R4] The system must collect users’ data in specific databases.

**[G3] Monitor the health status of users through electronic devices.**

- [D3] The communication channel doesn’t corrupt the data sent by the user’s device to the system and vice versa.
- [D5] The electronic device on which the system is installed is equipped with the sensors related to the parameters tracked by Data4Help.
- [D6] The system retrieves the data from the sensors through the interfaces available on the electronic device.

- [D8] The sensors related to the health status collect the data with a reasonable precision.
- [R5] The system must acquire real-time users' health parameters from the sensors installed on the user's device.
- [R4] The system must collect users' data in specific databases.
- [G4] **Allow a user to accept or refuse a request to access to personal data.**
  - [R6] Every time that a third party inserts a request for data relative to a Fiscal Code, the system must ask the authorization to the corresponding user.
  - [R7] The system must provide a function to allow users to accept or refuse a request for personal data.
- [G5] **Allow third parties to register to the system.**
  - [D1] Users insert credentials that correspond to their identity.
  - [D1] When a new registered user sends his/her credentials to the system, the message will be surely received.
  - [R8] The system should provide a registration form to third parties.
  - [R2] The system must guarantee that the credentials are unique.
- [G6] **Allow third parties to request an access to users' data.**
  - [G6.1] **Allow third parties to request access to data of some specific individuals by providing SSN or Fiscal Code.**
    - [R9] The system must provide a function to allow third parties to request the access to individuals data. The form must require the filling of the SSN or Fiscal Code of the corresponding user.
  - [G6.2] **Allow third parties to request access to anonymized data of group of people.**
    - [R10] The system must provide a function to allow third parties to request the access to data of a group of people. The form must offer some selection criteria such as geographical, time, health parameters, movement habits, ecc.
- [G7] **Send the demanded data to third parties whose request has been approved.**
  - [D3] The communication channel doesn't corrupt the data sent by the user's device to the system and vice versa.
  - [R11] As soon as a request has been approved, the system must forward the most recent data to the third party applying for them.
  - [R12] Data4Help must provide a function to let third parties download the received data.

**[G8] Accept request for data of group of people only if the system can guarantee the anonymity of the people.**

[D9] It's sufficient that the number of people involved in a group query is greater than 1000 people to ensure the users' privacy.

[R13] The system accepts a request for anonymous data only if the group composing the data is formed by more than 1000 people.

**[G9] Allow third parties to subscribe to new data and to receive them as soon as they are produced.**

[R3] The system must acquire real-time users' positions from the GPS installed on the user's device.

[R5] The system must acquire real-time users' health parameters from the sensors installed on the user's device.

[R14] The system must provide a function that allows to customize the requests for subscription to updated data. The requests must include how long the permission is still valid.

[R15] The system must provide a function to the corresponding user that allows him/her to accept or refuse the subscription request.

[R16] The system must forward in real-time the most recent data to the third parties that subscribed to them.

**AutomatedSOS:**

**[G10] Allow a visitor to become registered user after providing credentials.**

[D1] Users insert credentials that correspond to their identity.

[D2] When a new registered user sends his/her credentials to the system, the message will be surely received.

[R1] The system should provide a registration form offering the following mandatory fields: name, surname, SSN or Fiscal Code.

[R2] The system must guarantee that the credentials are unique.

**[G11] Monitor the location of users through electronic devices.**

[D3] The communication channel doesn't corrupt the data sent by the user's device to the system and vice versa.

[D4] The electronic device on which the system is installed is equipped with the GPS sensor.

[D7] Position information provided by GPS is sufficiently accurate.

[D11] Data provided by Data4Help API are surely received and they are not corrupted.



[R17] The system must acquire real-time users' position from their device through the Data4Help API.

[R18] The system saves the most recent user's position in a database.

**[G12] Monitor the health status of users through electronic devices.**

[D3] The communication channel doesn't corrupt the data sent by the user's device to the system and vice versa.

[D5] The electronic device on which the system is installed is equipped with the sensors related to the parameters tracked by Data4Help.

[D6] The system retrieves the data from the sensors through the interfaces available on the electronic device.

[D8] The sensors related to the health status collect the data with a reasonable precision.

[D11] Data provided by Data4Help API are surely received and they are not corrupted.

[R19] The system must acquire real-time users' health parameters from their device through the Data4Help API.

[R20] The system must elaborate users' data comparing each of them with a defined threshold.

[R21] Elaboration must be completed within 5 seconds from the instant in which the data are received.

**[G13] Send to the location of the customer very quickly an ambulance when such parameters are below certain thresholds.**

[D12] When an emergency request is sent to National Health Service, it surely receive it.

[D13] When National Health Service receives an emergency request, at least an ambulance is available.

[D14] When parameters are below certain thresholds, it means that the user actually needs first aid.

[D15] Operations Center of the NHS is up 24/7.

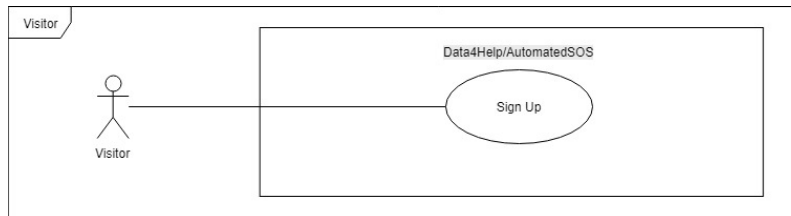
[D16] When Operations Center gather a request from AutomatedSOS, it sends at least an ambulance and at least one of them arrives to the location of the emergency.

[D17] NHS provides an API that allows third parties to send emergency requests in the form of a instant message received in their platform.

[R22] As soon as an anomaly in the parameters is detected, the system must organize the data in a human-readable request for first aid. The form for request must contain these mandatory fields: name, surname, age, gender, SSN or Fiscal Code, latest available position, list of the latest health parameters.

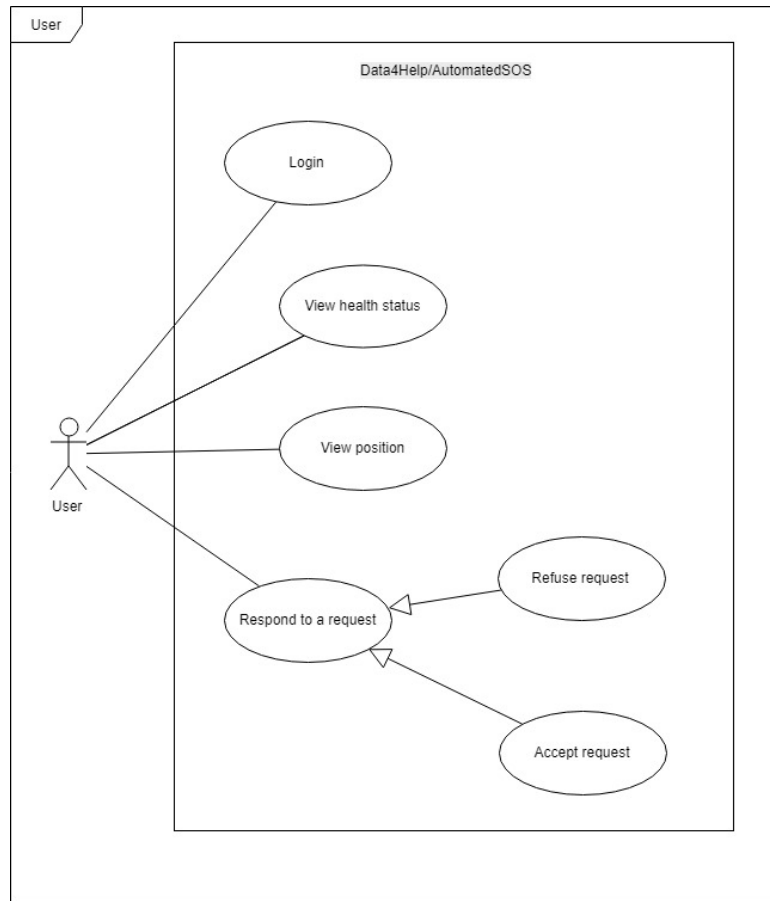
- [R23] AutomatedSOS must send a request for first aid to the Operations Center within 5 seconds from the moment that the emergency is detected.

### 3.4 Use Case Diagrams



(a) Use case-Visitor.

Name	Sign Up
Actor	Visitor
Entry Conditions	The visitor has installed the application on his/her device.
Event Flows	<ol style="list-style-type: none"> <li>1. Click on the “Sign up” button.</li> <li>2. Complete the mandatory fields providing the necessary informations.</li> <li>3. Click on “Confirm” button.</li> <li>4. The application checks that the provided data are complete and correct and saves them in the system.</li> </ol>
Exit Conditions	The user has completed the registration and can proceed to use the functionalities of the system.
Exceptions	<ul style="list-style-type: none"> <li>- The user is already registered.</li> <li>- The information provided by the user are not complete/correct.</li> <li>- The username is already taken.</li> </ul> <p>The exceptions are handled by notifying the user and taking him back to the sign up activity.</p>



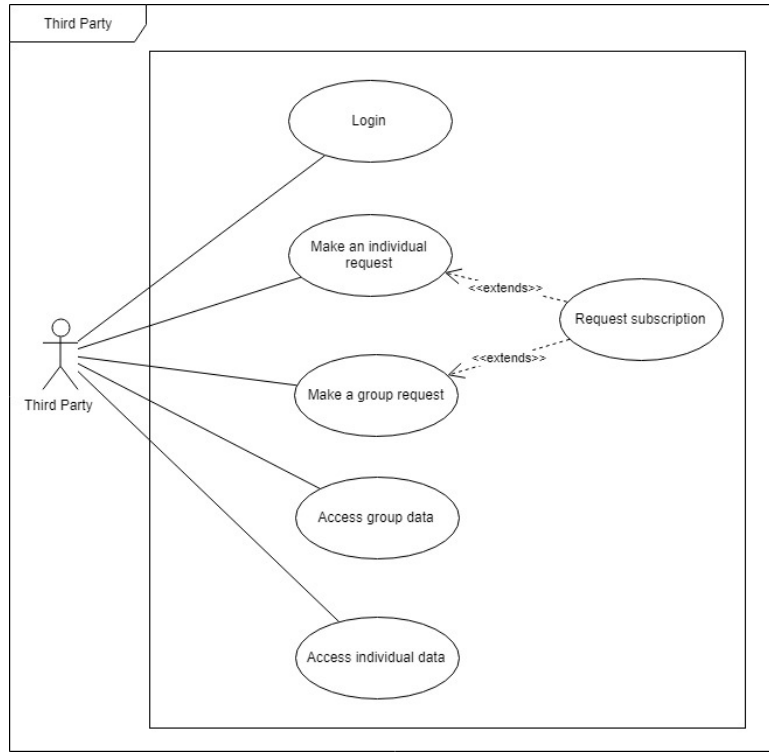
(b) Use case-User.

<b>Name</b>	Log-in
<b>Actor</b>	User
<b>Entry Conditions</b>	The user is successfully signed up.
<b>Event Flows</b>	<ul style="list-style-type: none"> <li>-The user opens the application on his/her device.</li> <li>-The user inserts his/her credentials(username and password) in the apposite fields on the start page of Data4Help app.</li> <li>-The user clicks on the “Log-in” button.</li> </ul>
<b>Exit Conditions</b>	The user is redirected to the view page of his/her position and his/her health parameters.
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>-The user inserts a wrong username.</li> <li>-The user inserts a wrong password.</li> </ul> All the exceptions are handled by showing a notification on the user’s screen and taking him/her back to the log-in activity.

<b>Name</b>	View health status
<b>Actor</b>	User
<b>Entry Conditions</b>	-The user is successfully logged in and he/she is on the Data4Help home-page.
<b>Event Flows</b>	-The user clicks on the button “activities” and a list of the possible activities appears on the user’s screen. -The user choose the activity “view health status” from the list.
<b>Exit Conditions</b>	The user is redirected to the view page in which he can observe his/her health parameters.
<b>Exceptions</b>	-No parameters are available. The exception is handled by showing a notification on the user’s screen and he/she is redirected to the list of the possible activities.

<b>Name</b>	View position
<b>Actor</b>	User
<b>Entry Conditions</b>	-The user is successfully logged in and he/she is on the Data4Help home-page.
<b>Event Flows</b>	-The user clicks on the button “activities” and a list of the possible activities appears on the user’s screen. -The user choose the activity “view position” from the list.
<b>Exit Conditions</b>	The user is redirected to the view page in which he/she can observe his/her position on a maps or by coordinates.
<b>Exceptions</b>	-The position is not available. The exception is handled by showing a notification on the user’s screen and he/she is redirected to the list of the possible activities.

<b>Name</b>	Accept/refuse a request for personal data
<b>Actor</b>	User
<b>Entry Conditions</b>	-The user is successfully logged in. -A third party sends a request to the user to access to his/her personal data.
<b>Event Flows</b>	-A notification appears on the user’s screen. -The user clicks on the “accept” or on the “refuse” button.
<b>Exit Conditions</b>	The user is redirected to the page he/she was surfing at the moment that the notification arrived.
<b>Exceptions</b>	No possible exceptions.



(c) Use case-Third Party.

<b>Name</b>	Log-in
<b>Actor</b>	Third Party
<b>Entry Conditions</b>	The user is successfully signed up.
<b>Event Flows</b>	<ul style="list-style-type: none"> <li>-The third party opens the application on his/her device or he/she opens the Data4Help web-page.</li> <li>-The third party inserts his/her credentials (username and password) in the apposite fields on the start page of Data4Help app or on the web-page.</li> <li>-The third party clicks on the “Log-in” button.</li> </ul>
<b>Exit Conditions</b>	The third party is redirected to the home page of the Data4Help app or web-page
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>-The third party inserts a wrong username.</li> <li>-The third party inserts a wrong password.</li> </ul> <p>All the exceptions are handled by showing a notification on the third party’s screen and taking him/her back to the log-in activity.</p>

<b>Name</b>	Send a request for group of anonymized data
<b>Actor</b>	Third party
<b>Entry Conditions</b>	The third party is successfully logged in and it is on the home page of Data4Help web-site.
<b>Event Flows</b>	<ul style="list-style-type: none"> <li>-The third party clicks on the button “Create a group request”.</li> <li>-The third party customizes some of the searching criteria.</li> <li>-The third party clicks on the “send” button.</li> </ul>
<b>Exit Conditions</b>	A notification showing the outcome of the operation appears on the third party’s screen and he/she is redirected to the home page of the Data4Help web-site.
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>-No searching criteria are selected by the third party.</li> <li>-Conflicting criteria are selected by the third party.</li> <li>-The searching criteria corresponds to less than 1000 people. All the exceptions are handled by showing a notification on the third party’s screen and taking him/her back to the home page of the Data4Help web-site.</li> </ul>

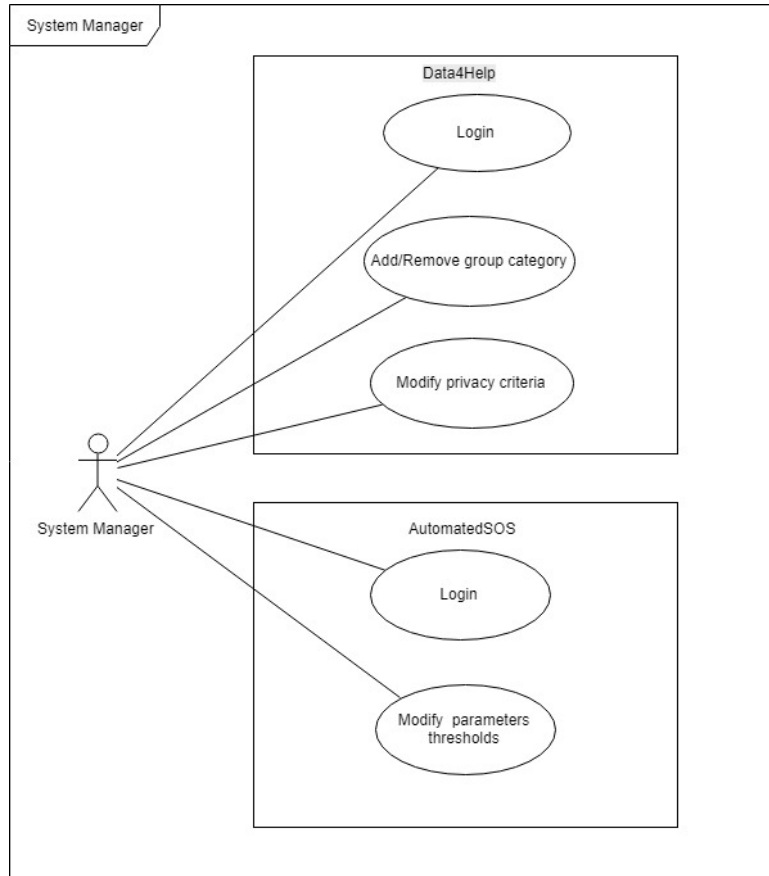
<b>Name</b>	Access to group of data
<b>Actor</b>	Third party
<b>Entry Conditions</b>	-The third party is successfully logged in and he/she is on the Data4Help home-page of the mobile application or on the web-site.
<b>Event Flows</b>	<ul style="list-style-type: none"> <li>-The third party clicks on the button “activities” and a list of the possible activities appears on the third party’s screen.</li> <li>-The third party chooses the activity “access to a group of data” from the list.</li> <li>-A list of all the available group of data is shown on the third party’s screen.</li> <li>-The third party chooses a group of data he/she wants to access.</li> </ul>
<b>Exit Conditions</b>	The third party is redirected to the view page in which he can observe all the data available of that group.
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>-No group of data are available.</li> <li>-No data for the selected group are available.</li> </ul> <p>The exception is handled by showing a notification on the third party’s screen and he/she is redirected to the list of the possible activities.</p>

<b>Name</b>	Send a request for individual data
<b>Actor</b>	Third party
<b>Entry Conditions</b>	The third party is successfully logged in and it is on the home page of Data4Help mobile application or on the web-site.
<b>Event Flows</b>	<ul style="list-style-type: none"><li>-The third party clicks on the button “Create an individual request”.</li><li>-The third party fills the following mandatory fields: Name, Surname, Fiscal Code or SSN.</li><li>-The third party clicks on the “send” button.</li></ul>
<b>Exit Conditions</b>	A notification showing the outcome of the operation appears on the third party’s screen and he/she is redirected to the home page of the Data4Help web-site. When the user accepts or refuse the request, a notification will appear on the third party’s screen.
<b>Exceptions</b>	<ul style="list-style-type: none"><li>-none is selected by the third party.</li><li>-an unexisting Fiscal Code or SSN is inserted by the third party.</li><li>-a wrong Fiscal Code or SSN is inserted by the third party.</li></ul> All the exceptions are handled by showing a notification on the third party’s screen and taking him/her back to the home page of the Data4Help web-site.

<b>Name</b>	Access to individual data
<b>Actor</b>	Third party
<b>Entry Conditions</b>	-The third party is successfully logged in and he/she is on the Data4Help home-page of the mobile application or on the web-site.
<b>Event Flows</b>	-The third party clicks on the button “Activities” and a list of the possible activities appears on the third party’s screen. -The third party chooses the activity ”access to individual data” from the list. -A list of all the individual’s data to which the third party is authorized to access, is shown on the third his/her screen. -The third party selects a user among the available data.
<b>Exit Conditions</b>	The third party is redirected to the view page in which he can observe all the data available of that user.
<b>Exceptions</b>	-No individual data are available. -No data for the selected user are available. The exception is handled by showing a notification on the third party’s screen and he/she is redirected to the list of the possible activities.

<b>Name</b>	Request Subscription to Data
<b>Actor</b>	Third party
<b>Entry Conditions</b>	-The third party is successfully logged in and he/she is in the process of making a request for group or individual data.
<b>Event Flows</b>	-After filling the request procedure with the required criteria, the third party can click on the button ”Request Subscription” to receive updated data as soon as they are created. -The request is sent to the system.
<b>Exit Conditions</b>	A notification showing the outcome of the operation appears on the third party’s screen and he/she is redirected to the home page of the Data4Help web-site.
<b>Exceptions</b>	-This use case is an extension of the “Data Request” use cases and inherits all its parent’s exceptions.





(d) Use case-System Manager.

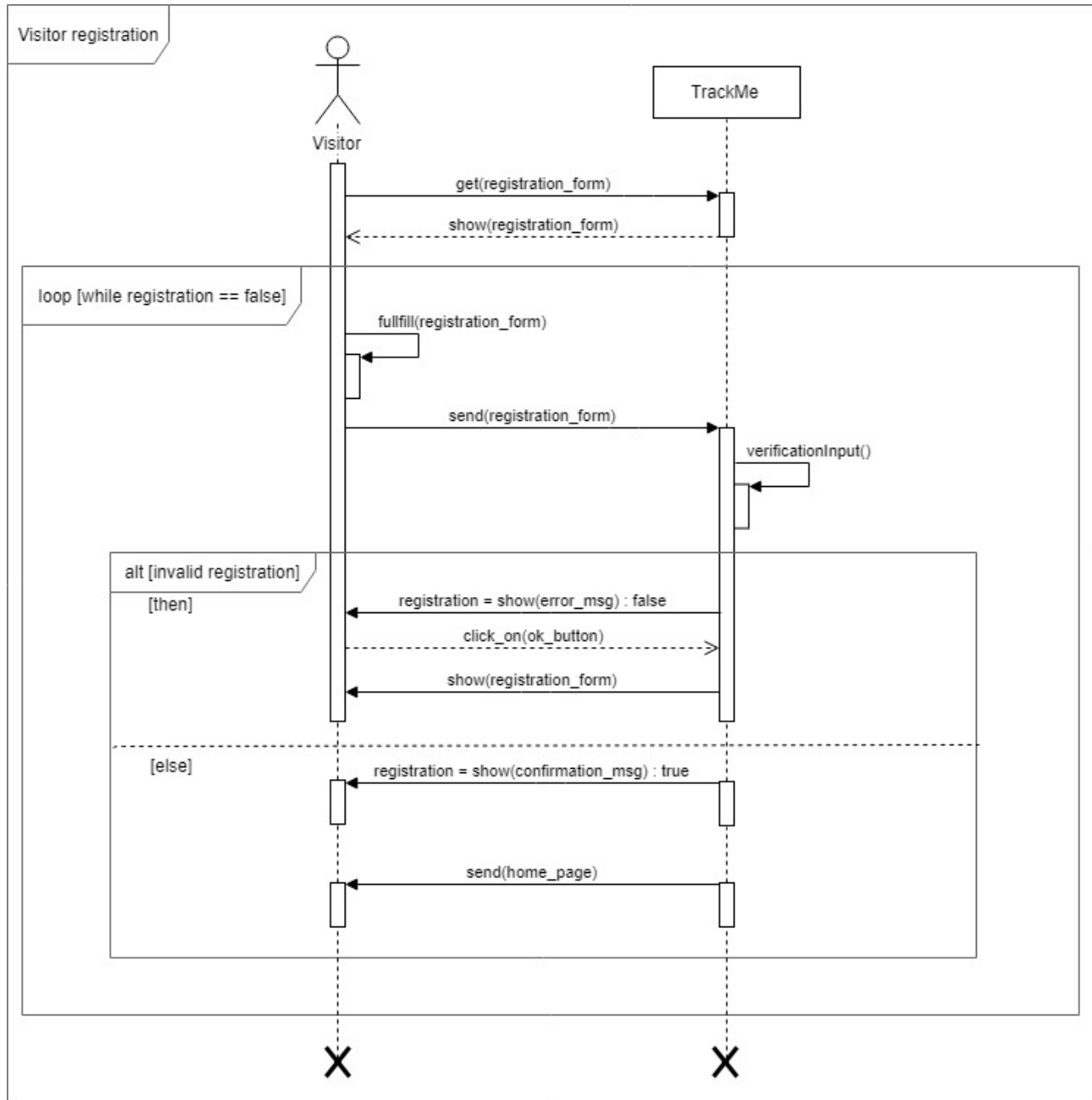
<b>Name</b>	Login
<b>Actor</b>	System Manager
<b>Entry Conditions</b>	-No entry conditions.
<b>Event Flows</b>	-The system manager access Data4Help or AutomatedSOS back-end system on his/her device. -The system manager inserts his/her credentials (username and password) in the apposite fields of the back-end system.
<b>Exit Conditions</b>	The third party is redirected to the control panel of the back-end system.
<b>Exceptions</b>	-The system manager inserts a wrong username. -The system manager inserts a wrong password. All the exceptions are handled by showing a notification on the system manager's screen and taking him/her back to the log-in activity.

<b>Name</b>	Add/Remove Group Category
<b>Actor</b>	System Manager
<b>Entry Conditions</b>	The system manager is logged in Data4Help back-end system successfully.
<b>Event Flows</b>	-The system manager updates Data4Help architecture adding or removing a group category.
<b>Exit Conditions</b>	The system manager is redirected to the control panel of the back-end system.
<b>Exceptions</b>	No exceptions.

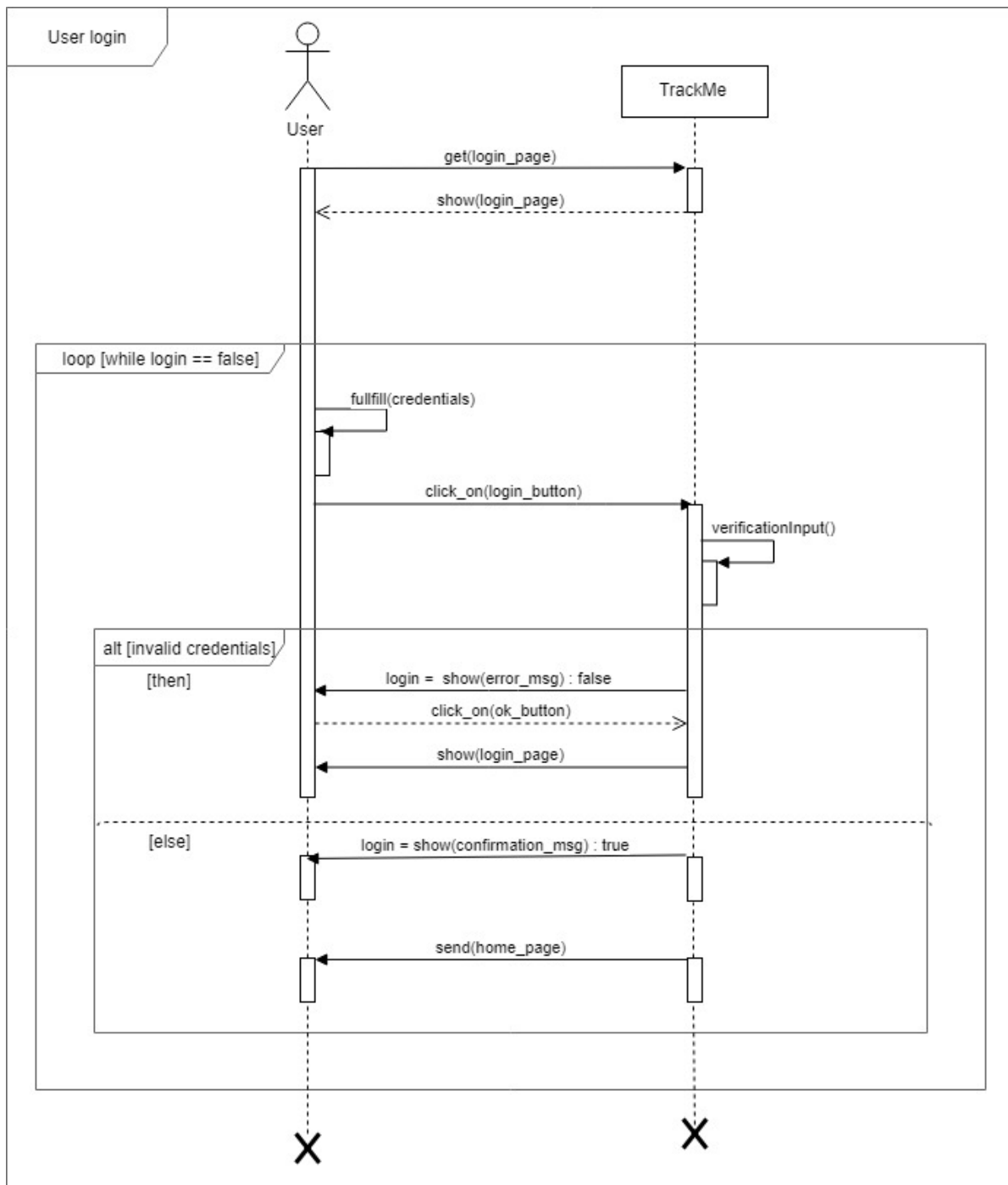
<b>Name</b>	Modify Privacy Criteria
<b>Actor</b>	System Manager
<b>Entry Conditions</b>	The system manager is logged in Data4Help back-end system successfully.
<b>Event Flows</b>	-The system manager updates Data4Help architecture modifying the privacy criteria.
<b>Exit Conditions</b>	The system manager is redirected to the control panel of the back-end system.
<b>Exceptions</b>	No exceptions.

<b>Name</b>	Modify Health Parameters Thresholds
<b>Actor</b>	System Manager
<b>Entry Conditions</b>	The system manager is successfully logged in AutomatedSOS back-end system.
<b>Event Flows</b>	-The system manager updates the database containing the health parameters thresholds.
<b>Exit Conditions</b>	The system manager is redirected to the control panel of the back-end system.
<b>Exceptions</b>	No exceptions.

### 3.5 Sequence Diagrams



(a) Visitor registration.



(b) User login.

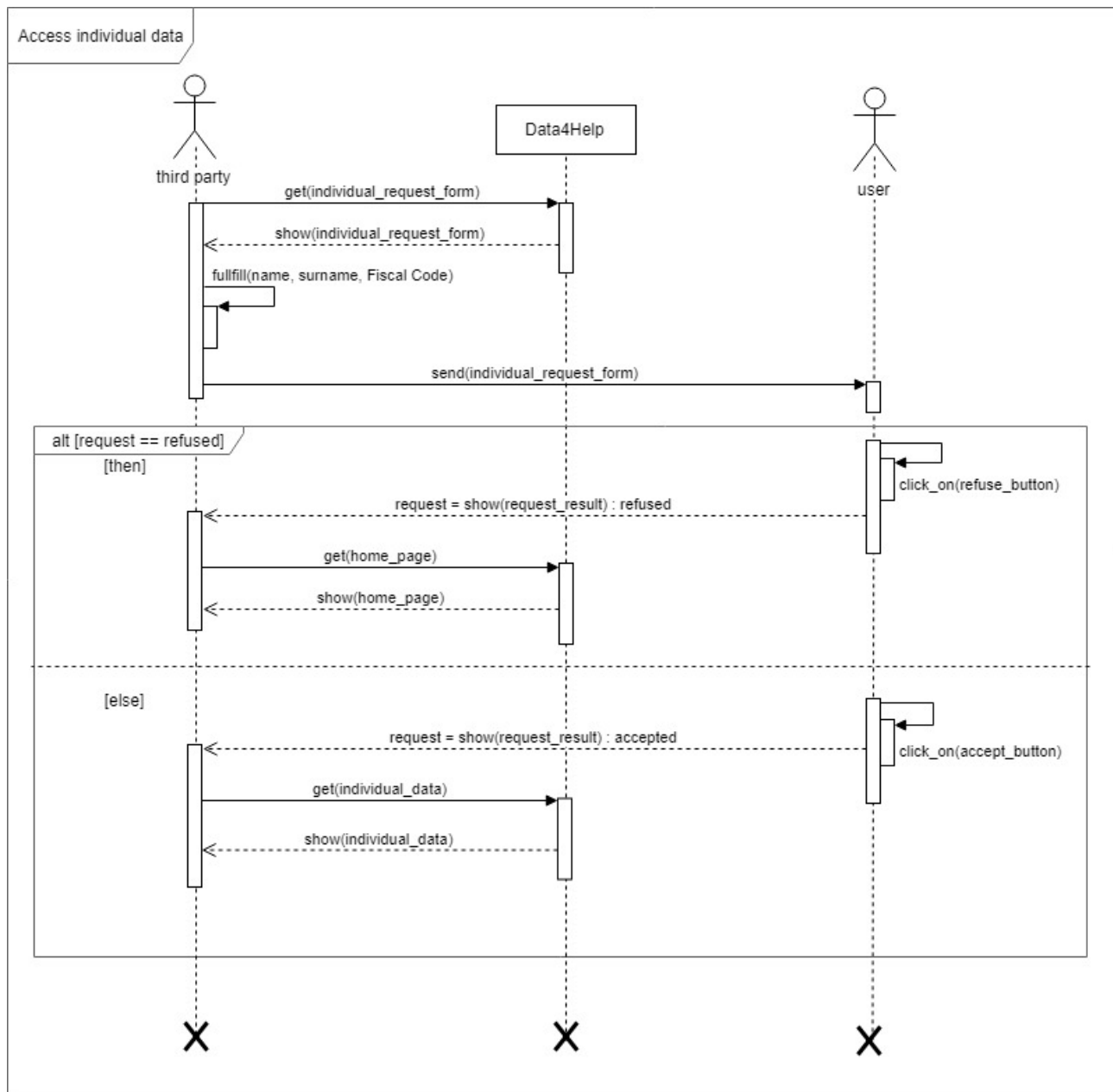


Figure c: Third party accesses to the data of a single user.

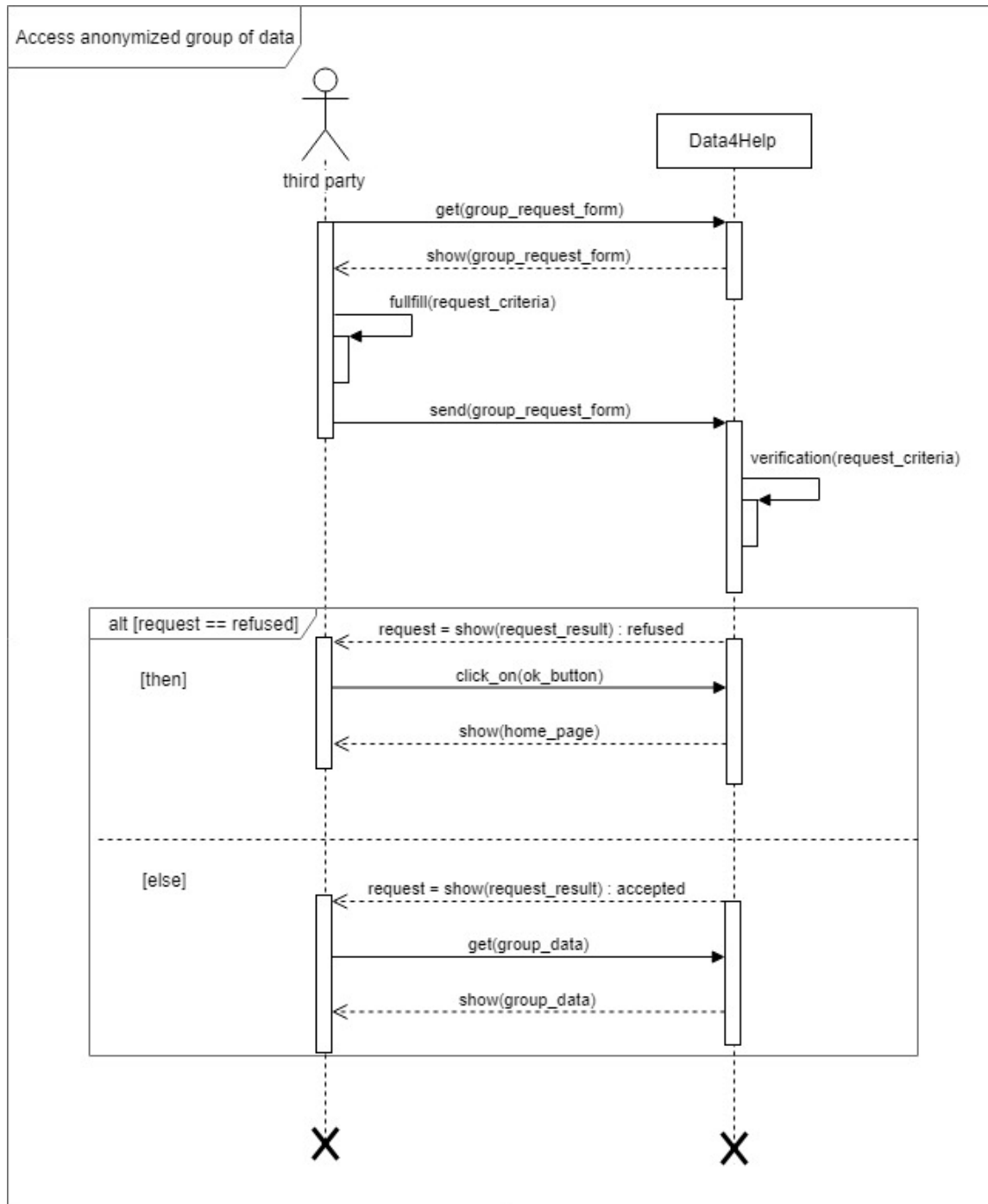


Figure d: Third party accesses to a group of anonymized data.

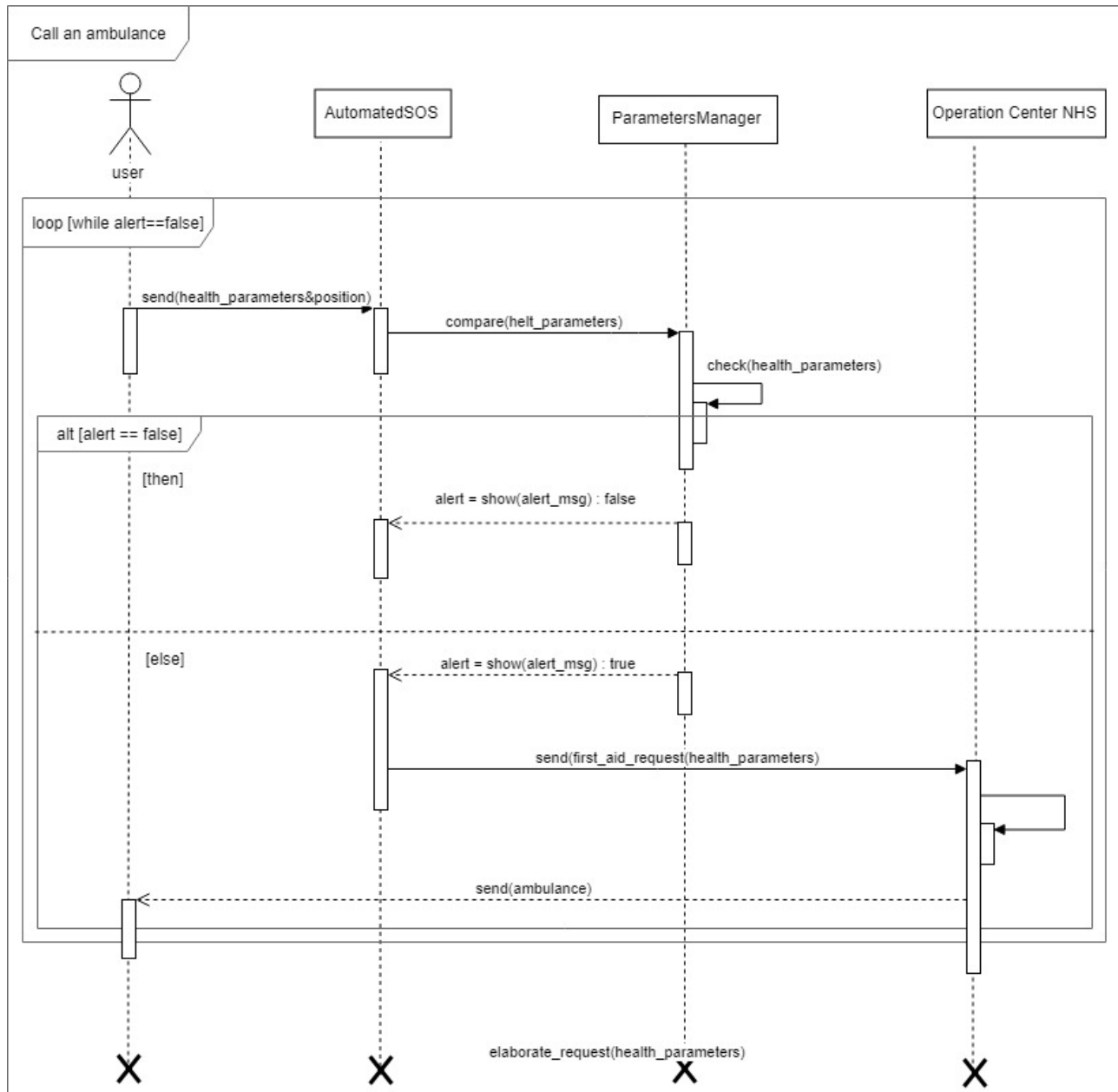


Figure e: AutomatedSOS sends an ambulance.

## 3.6 Performance Requirements

The system has to be able to respond to a huge number of transactions simultaneously. The service should be able to scale depending on the number of active users. Speed and performance are critical especially for AutomatedSOS service, since it must guarantee to contact the Operations Center within 5 seconds from the time an anomaly is detected.

## 3.7 Design Constraints

### 3.7.1 Standards compliance

- Health parameters are stored according the international units of measure.
- Position is stored according geographic coordinates system.
- SSN or Fiscal Code format changes according to the country of usage.

### 3.7.2 Hardware limitations

- Mobile App
  - Apple or Android smartphone
  - Apple, Android, Fitbit or Garmin wearables
  - 3G/4G/WiFi connection
  - Bluetooth
  - GPS
  - Heart rate, temperature, blood glucose level, accelerometer sensors
- Web App
  - Devices that supports modern browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari, etc.

### 3.7.3 Any other constraint

#### Regulatory Policies

The system will be subject to the privacy requirements enforced by the national law. In order to respect the minimum constraints, the system will ask for users' permission to store their data and to let third parties access to their personal informations.



### **Interfaces**

Sensors interact with Data4Help through an apposite API (healthKit for iOS[2], Sensors API for Android[3]). AutomatedSOS interacts with the Operations Center through an interface provided by the NHS. Data4Help uses Google Maps API to show the position of a user on a mobile or web app. Google Maps API is also used to transform a position on a map into an address while composing a group request.

### **Level of Criticality**

Since an unexpected fault in Data4Help acquisition could implies very important consequences on the safeness of AutomatedSOS users, it is critical that the system is almost always available.

### **Parallel operations**

The system has to support multiple concurrent requests. Individual data may be accessed concurrently by more than one third party. Also Data4Help will be able to record data from many different users at the same time.

## **3.8 Software System Attributes**

### **3.8.1 Reliability**

The system should offer a level of reliability such that core functions crash-down occur rarely and can promptly be restored. The application should aim to a 24/7 reliability, of course small variations from the ideal case are tolerable.

### **3.8.2 Availability**

The system is required to have an availability of 99.99%, meaning that the maximum downtime accepted every year is of 1 hour. To ensure this requirement the system will have to be implemented with redundancy of data in order to overcome potential malfunctioning.

### **3.8.3 Security**

It is important to ensure that the system is adequately protected against attacks to ensure the privacy and the safety of the users. The system must guarantee that only authorized third parties are able to access individuals data and that only the system managers can update the internal functionalities. All the users' data should be encrypted both during the client-server communication and on the server database. The data center should also be protected against natural disasters such as fire or floods.

#### **3.8.4 Maintainability**

The system should be easy to maintain and update. Services such as Data4Help and AutomatedSOS could need in the future some improvement in their functionalities (such as the addition of a new revolutionary sensor for Data4Help or the changes in the thresholds in AutomatesSOS), so it's crucial that they are designed in a flexible way. In order to observe this constraint, the system will be implemented following the most common software engineering best practices and will be detailed with a complete and specific documentation.

#### **3.8.5 Portability**

To ensure the portability of the core service, the database and the back-end system will be implemented avoiding platform dependent programming languages. Also the web client applications will be designed following the W3C standards, in order to ensure a correct rendering of the page in all major browsers.

## Section 4

# Formal Analysis Using Alloy

### 4.1 Alloy Model

This section contains a possible formalization of the proposed system using the Alloy language. The following alloy model is an attempt to represent the essential features of the system, in particular: the interactions between the users and the third parties for Data4Help and the generation and dispatch of an emergency notification for AutomatedSOS.

Due to the Alloy limitations, it was impossible to implement in the language the privacy condition that requires the presence of 1000 or more different users to accept a group data request. The limit was set to a symbolic value of 2 in the concerning predicate.

```
open util/integer
open util/time
enum Boolean {True, False}

//Signatures

sig Position {
    latitude: one Int,
    longitude: one Int
}

abstract sig User {
    name: one String,
    surname: one String,
    username: one String,
    password: one String
}

sig MonitoredUser extends User {
    gender: one String,
    fiscalCode: one String
}
```

```

sig ThirdParty extends User {}

sig HealthParameter {
  parameter: one Int,
  threshold: one Int
}

sig HeartRate extends HealthParameter {}

sig SkinTemperature extends HealthParameter {}

sig GlucoseLevel extends HealthParameter {}

sig Data {
  timeStamp: one Time,
  position: one Position,
  healthParameters: some HealthParameter
}

sig UserData extends Data {
  user: one User,
  requests: set IndividualRequest
}

sig Criterion { }

abstract sig Request {
  thirdParty: one ThirdParty,
  subscription: Boolean one -> Time,
  authorization: Boolean one -> Time
}

sig IndividualRequest extends Request {
  fiscalCode: one String
}

sig GroupRequest extends Request {
  criteria: some Criterion
}

sig GroupData {
  timeStamp: one Time,
  data: some Data,
  criteria: some Criterion,
  requests: set GroupRequest
}

sig EmergencyMessage {
  timeStamp: one Time,
  userData: one UserData
}

sig OperationsCenter {
  messages: set EmergencyMessage
}

//Facts

fact UniqueUsername {
  no disjoint u1, u2: User | u1.username = u2.username
  --every user has a unique username
}

fact UniqueFiscalCode {
  no disjoint u1, u2: MonitoredUser | u1.fiscalCode = u2.fiscalCode
  --every user has a unique Fiscal Code
}

```

```

fact equalsPositions {
    all p1, p2: Position | p1=p2 <=> p1.latitude=p2.latitude and p1.
        longitude=p2.longitude
    --2 positions are equal iff latitudine and longitude are equal
}

fact NoDuplicateTimeData {
    no disjoint ud1, ud2: UserData | ud1.user = ud2.user and ud1.
        timeStamp = ud2.timeStamp
    --all the data collected from a user at a specific time are contained
        in one UserData
}

fact RequestState {
    all r: Request | one t: Time | r.authorization.t = False
    --all the request are created with the authorization at "False"
    all r: Request, t: Time |
        (r.authorization.t = True implies (all t': Time | gte[t', t] implies
            (r.authorization.t' = True)))
    --once a request is accepted, it stay in this state forever
}

fact allEmergencyMessageAreSent {
    all em: EmergencyMessage | one oc: OperationsCenter |
        em in oc.messages
    --all the emegency messages are sent to one Operations Center
}

//Predicates

--Create an individual request
pred createIndividualRequest [r: IndividualRequest, tp: ThirdParty, u:
    MonitoredUser, t: Time] {
    //postconditions
    r.thirdParty = tp
    r.fiscalCode = u.fiscalCode
    r.authorization.t = False
    r.subscription.t = False
}

--Request a subscription to the requested data
pred requestSubscription [tp: ThirdParty, r: Request, t: Time] {
    r.authorization.t = False
    r.subscription.t = True
}

--Accept a request for personal data
pred acceptIndividualRequest [u: User, r: IndividualRequest, t: Time] {
    //postconditions
    r.authorization.t = True --the request is authorized
    r.subscription.t = False implies --if the third party
        did not requested the subscription
        (one d: UserData | d.user = u and r in d.requests and
            gte[d.timeStamp, t])
    --the third party will be able to access only one
        packet of data generated after the time of the
        authorization
    else
        (all d: UserData | d.user = u and gte[d.timeStamp, t]
            implies (r in d.requests))
    --the third party will be able to access all the user
        's data generated after the authorization
}

```

```

--Privacy criteria
pred isPrivacyRespected [g: GroupData] {
  //postconditions
  gt[#g.data.user, 2]
  --due to the alloy language constraints, the limit of 1000 different
  people composing a group of data has been reduced to a symbolic
  2 people
}

--Accept a request for group data
pred acceptGroupRequest [r: GroupRequest, t: Time] {
  //postconditions
  r.authorization.t = True --the request is authorized
  r.subscription.t = False implies --if the third party
  did not requested the subscription
  (one g: GroupData | isPrivacyRespected[g] and r in g.requests
  )
  --the third party will be able to access only one packet of
  data generated after the time of the authorization
  else
  (all g: GroupData | g.criteria = r.criteria and gte[g.
  timeStamp, t] implies (r in g.requests))
  --the third party will be able to access all the group data
  following the requested criteria generated after the
  authorization
}

--Refuse a request for personal data
pred refuseIndividualRequest [r: IndividualRequest, t: Time] {
  //postconditions
  r.authorization.t = False --the request is rejected
  r.subscription.t = False --the subscription is
  rejected
  no d: UserData | d.fiscalCode = r.fiscalCode and r in d.requests
  --there isn't any data associated with the specific user accessible
  by this request
}

--AutomatedSOS generates an emergcy
pred generateEmergency {
  //postconditions
  all d: UserData | some hp: HealthParameter | --for every user such
  that exists an health parameter
  (hp in d.healthParameters and gt[hp.parameter, hp.threshold])
  --the health parameter is over is corresponding threshold
  iff
  one em: EmergencyMessage, oc: OperationsCenter |
  em.userData = d and (em in oc.messages) and
  lte[em.timeStamp, d.timeStamp + 5]
  --an emergency message containing the user's data is
  generated and is sent to an Operations Center within 5
  seconds from the anomaly
}

pred show {
  all tp: ThirdParty | some ir: IndividualRequest | ir.thirdParty = tp
  all tp: ThirdParty | some ir: GroupRequest | ir.thirdParty = tp
  all u: User | some d:UserData | d.user = u
}

run show for 5
run createIndividualRequest for 5 but 3 Int, exactly 3 String
run requestSubscription for 5 but 3 Int, exactly 3 String
run acceptIndividualRequest for 5 but 3 Int, exactly 3 String
run isPrivacyRespected for 5 but 3 Int, exactly 3 String
run acceptGroupRequest for 5 but 3 Int, exactly 3 String
run refuseIndividualRequest for 5 but 3 Int, exactly 3 String
run generateEmergency for 5 but 3 Int, exactly 3 String

```

## 4.2 Proof of Consistency

### Executing "Run show for 5"

Solver=sat4j Bitwidth=4 MaxSeq=5 SkolemDepth=1 Symmetry=20  
6551 vars. 895 primary vars. 1113 clauses. 258ms.  
**Instance** found. Predicate is consistent. 125ms.

### Executing "Run createIndividualRequest for 5 but 3 int, exactly 3 String"

Solver=sat4j Bitwidth=3 MaxSeq=3 SkolemDepth=1 Symmetry=20  
6030 vars. 785 primary vars. 10009 clauses. 78ms.  
**Instance** found. Predicate is consistent. 78ms.

### Executing "Run requestSubscription for 5 but 3 int, exactly 3 String"

Solver=sat4j Bitwidth=3 MaxSeq=3 SkolemDepth=1 Symmetry=20  
5904 vars. 780 primary vars. 9679 clauses. 47ms.  
**Instance** found. Predicate is consistent. 63ms.

### Executing "Run acceptIndividualRequest for 5 but 3 int, exactly 3 String"

Solver=sat4j Bitwidth=3 MaxSeq=3 SkolemDepth=1 Symmetry=20  
6137 vars. 780 primary vars. 10554 clauses. 78ms.  
**Instance** found. Predicate is consistent. 47ms.

### Executing "Run isPrivacyRespected for 5 but 3 int, exactly 3 String"

Solver=sat4j Bitwidth=3 MaxSeq=3 SkolemDepth=1 Symmetry=20  
5841 vars. 770 primary vars. 9721 clauses. 46ms.  
**Instance** found. Predicate is consistent. 79ms.

### Executing "Run acceptGroupRequest for 5 but 3 int, exactly 3 String"

Solver=sat4j Bitwidth=3 MaxSeq=3 SkolemDepth=1 Symmetry=20  
6454 vars. 775 primary vars. 11424 clauses. 46ms.  
**Instance** found. Predicate is consistent. 282ms.

### Executing "Run refuseIndividualRequest for 5 but 3 int, exactly 3 String"

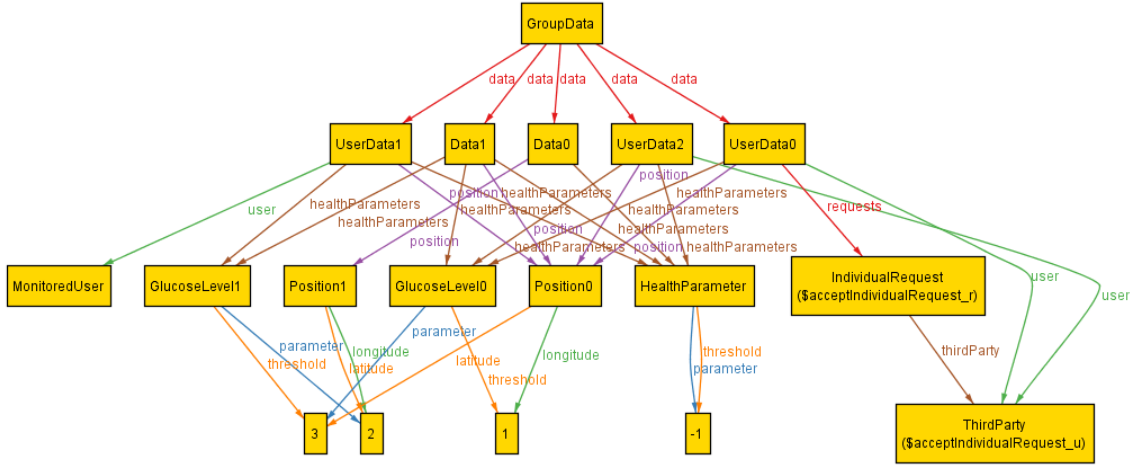
Solver=sat4j Bitwidth=3 MaxSeq=3 SkolemDepth=1 Symmetry=20  
5977 vars. 775 primary vars. 9810 clauses. 31ms.  
**Instance** found. Predicate is consistent. 63ms.

### Executing "Run generateEmergency for 5 but 3 int, exactly 3 String"

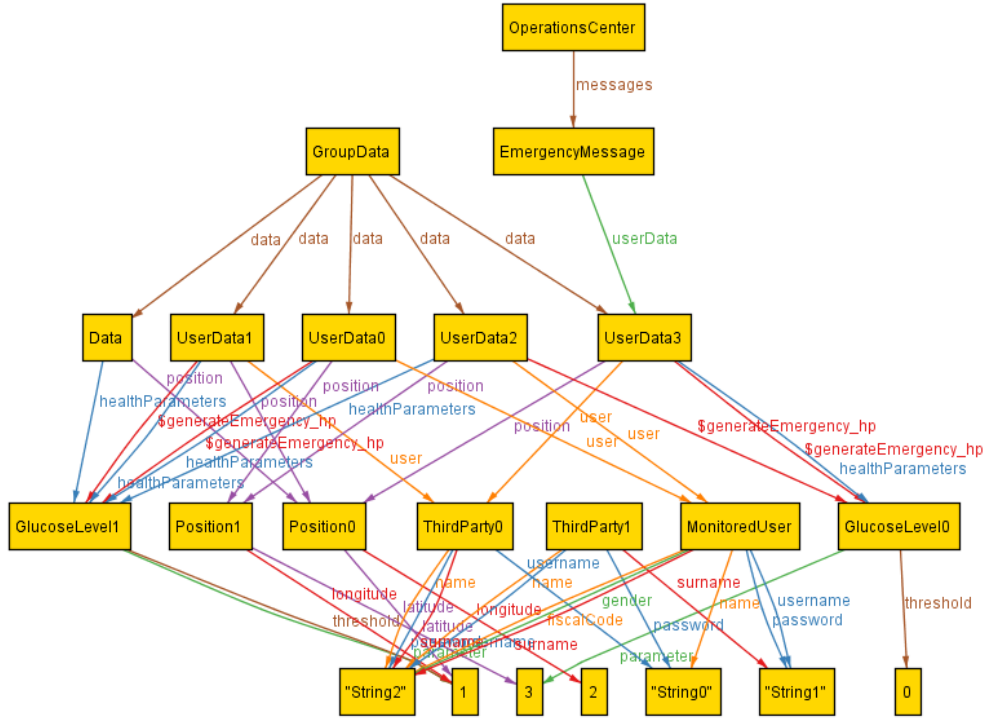
Solver=sat4j Bitwidth=3 MaxSeq=3 SkolemDepth=1 Symmetry=20  
7392 vars. 790 primary vars. 15083 clauses. 46ms.  
**Instance** found. Predicate is consistent. 45ms.

(a) Proof of Consistency

### 4.3 Generated World



(b) World 1



(c) World 2



## Section 5

# Effort Spent

### 5.1 Comolli Federico

Description of the task	Date	Hours
Goals identification	25/10/2018	3
Purpose	26/10/2018	3.5
Scope	27/10/2018	2.5
Section 1 revision	28/10/2018	3
UML Class Diagram	29/10/2018	2
Domain Assumptions	30/10/2018	2
Functional Requirements	31/10/2018	1.5
Functional Requirements	01/11/2018	1.5
Scenarios	01/11/2018	1
Scenarios	02/11/2018	2.5
User interface	02/11/2018	1
Product Functions and Users characteristics	03/11/2018	3
Section 3	04/11/2018	3.5
UML Class Diagram	05/11/2018	2
Use Cases	06/11/2018	3
UML Sequence Diagram and interface mock-up	08/11/2018	3
Document Revision	11/11/2018	3
<b>Total Effort Spent</b>		41

## 5.2 Corda Francesco

Description of the task	Date	Hours
Goals identification	25/10/2018	3
Purpose	26/10/2018	3.5
Scope	27/10/2018	2.5
Section 1 revision	28/10/2018	3
UML Class Diagram	29/10/2018	2
Domain Assumptions	30/10/2018	2
Functional Requirements	31/10/2018	1.5
Functional Requirements	01/11/2018	1.5
Scenarios	01/11/2018	1
Scenarios	02/11/2018	2.5
User interface	02/11/2018	1
Product Functions and Users characteristics	03/11/2018	3
Section 3	04/11/2018	3.5
UML Class Diagram	05/11/2018	2
Use Cases	06/11/2018	3
Alloy Model Analysis	08/11/2018	3
Document Revision	11/11/2018	3
<b>Total Effort Spent</b>		<b>41</b>

## Section 6

# References

- [1] U.S. government. Gps accuracy. Technical report, U.S. government, 2017.
- [2] Apple Inc. healthkit. Technical report, Apple Inc., 2018.
- [3] Google Inc. Sensors api. Technical report, Google Inc., 2018.