POLITECNICO
MILANO 1863

SOFTWARE ENGINEERING 2 PROJECT

TRACKME

Requirements Analysis and Specifications Document

Authors:
Matteo VELATI
Niccolò ZANGRANDO

Professor:
Matteo ROSSI

Delivery date:
11/09/2018

1 Introduction

1.1 Purpose of this document

The purpose of a Requirement Analysis and Specifications Document is the process of discovering the purpose for which a software system was intended, by identifying stakeholders and their needs, and documenting these in a form that is amenable to analysis, communication, and subsequent implementation. It is also concerned with the relationship of software's factors such as goals, functions and constrains to precise specifications of software behavior, and to their evolution over time and across software families.

1.2 Scope

1.2.1 Description of the given problem

TrackMe is a software-based service that allows third parties to monitor the health and positions of individuals. The software must allow third parties to make requests for access to data in two different ways. The first way is to request data for a group of individuals, the second way is to request specific data for a single individual. The group requests will be accepted if they guarantee the anonymity, instead the individual requests must be accepted by the individual user. In addition to this service, called Data4Help, TrackMe also offers a service for individuals: AutomatedSOS. This service constantly monitors the vital parameters of individuals and, when it detects an anomaly in the parameters, it must send an help request. The reaction time for sending the request must not exceed 5 seconds.

1.2.2 Goals

[G1] Allow an individual to become registered user after providing credentials

[G2] Allow third parties to become registered user after providing credentials

[G3] Allow third parties to monitor the location and health status of individuals

[G4] Allow third parties to monitor the location and health status of groups of individuals

[G5] A specific individual can accept or refuse the request of processing his/her personal data from a third party

[G6] When health parameters are below certain threshold, the system sends to the location of the customer an ambulance

[G7] The AutomatedSOS must guarantee a reaction time of less than 5 seconds in dispatching an ambulance

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

Company user: a third party registered user

Individual user: an individual registered user

Single request: a request made by a Company user to a specific Individual user

Group request: a request made by a Company user for a group of people

Health data: the vital parameters of an individual

Location data: the coordinates of the position of an individual

Warning: a message sent by the AutomatedSOS service to an Individual user to advise him that an ambulance is arriving

System: the TrackMe software we are to develop

1.3.2 Acronyms

RASD: Requirements Analysis and Specifications Document

API: Application Programming Interface

GPS: Global Position System

1.3.3 Abbreviations

[Gn]: n-th goal

[Dn]: n-th domain assumption

[Rn]: n-th requirement

i.e.: id est

1.4 Revision History

- **v1.0** November 11th, 2018

1.5 Document Structure

According to IEEE standard, this document is structured in six chapters:

1. *Introduction:* provides an overview of this entire document and product goals, describing the purpose of the application.

2. *Overall description:* this chapter describes general factors that affect the product providing the background for system requirements.

3. *Specific requirements:* contains all system's functional and non-functional requirements, some scenarios describing specific situations and use case diagrams.

4. *Alloy model:* contains a brief description of the formal analysis made with Alloy and, in the end, the World Generated by it.

5. *Effort spent:* reports a summary of the effort spent for each section by each team member.

6. *References:* includes the references of this document.

2. Overall Description

2.1 Product Perspective

The TrackMe system is designed to be completely new software application. It doesn't need any API except for the AutomatedSOS service which uses the AmbulanceExternalSystem API for help requests.

Regarding the TrackMe service itself the following diagram is provided:

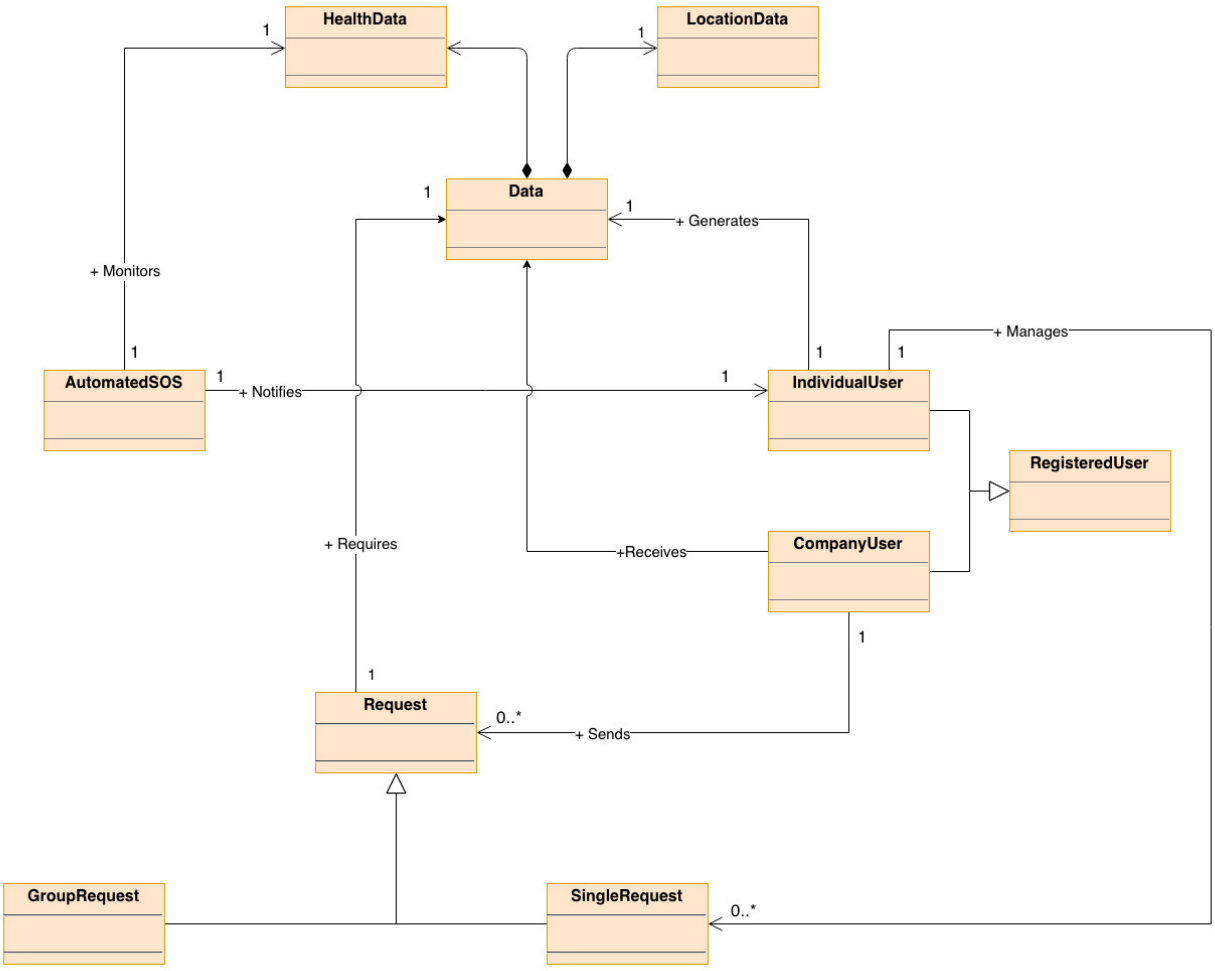

Figure 1: Class diagram

As we can see the Requests are the focal point of the system. Also there is the AutomatedSOS section that is included only for the individual users.

In the next state diagram is analyzed a new request made by a CompanyUser:

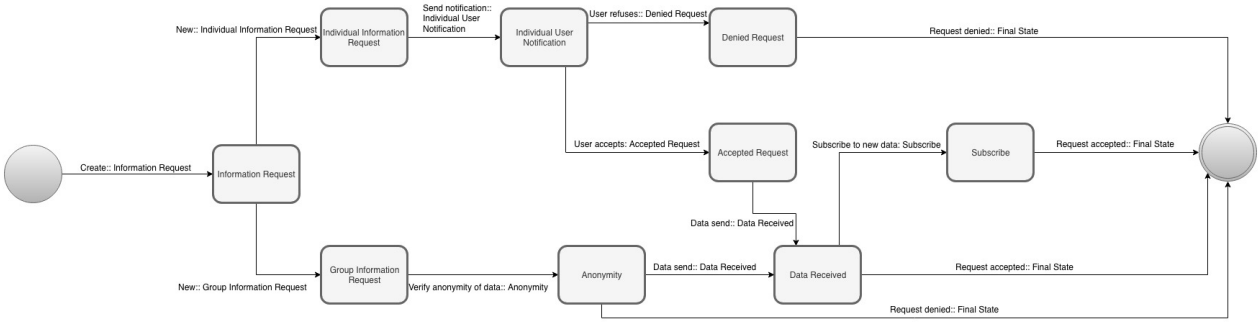

Figure 2: CompanyUser request state diagram

As we see, a Company user can create a SingleRequest or a GroupRequest and in both cases his request can be accepted or refused by an individual (if it's a SingleRequest) or by the system(if it's a GroupRequest).

In the next state diagram is analyzed the function of the AutomatedSOS:

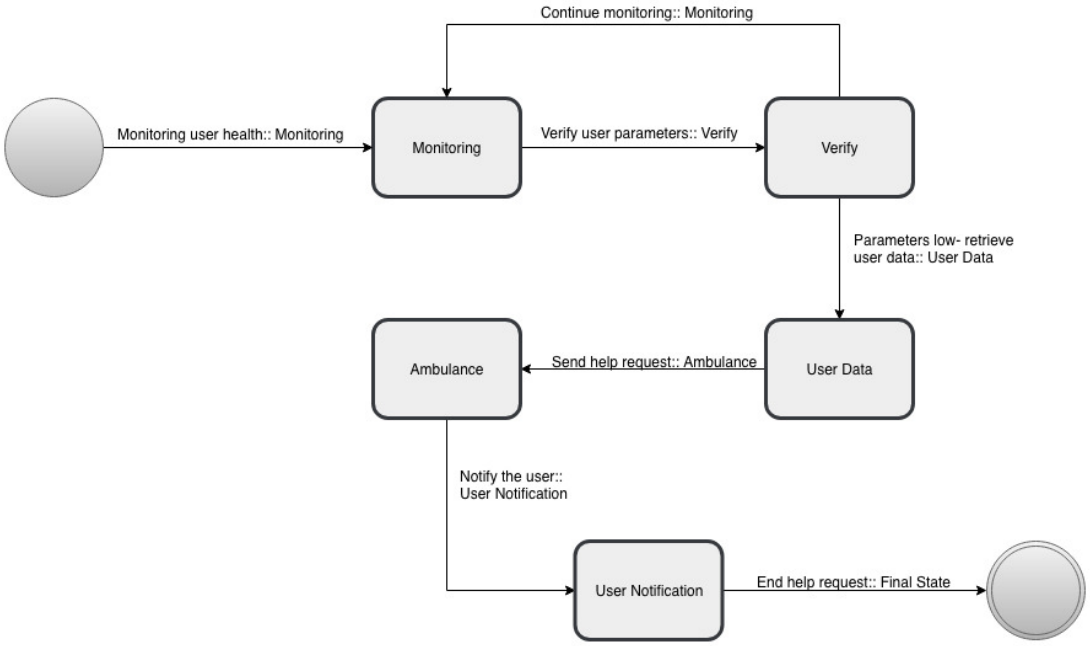

Figure 3: AutomatedSOS state diagram

As we see, the AutomatedSOS monitors the user parameters and in case of necessity send a help request for the individual.

2.2 Product Functions

2.1.1 Group Request management

This function allows a CompanyUser to create a new Group Request and to specify the parameters of the data research. The system, after receiving the request, has to verify if the required data are anonymous. If they are, the request is accepted, if not the request is refused.

2.1.2 Single Request management

This function allows a CompanyUser to create a new Single Request and to specify the individual of whom he wants the data. After creating the request, this is sent to the individual who must be accepted or refused it. In the time in which he decides, the status of the request is set “Pending”.

2.1.3 AutomatedSOS management

This function is only for Individual User. The system monitors the health parameters of the individual and is always connected with the AmbulanceExternalSystem. If the individual parameters are under a threshold, the AutomatedSOS sends a help request for the individual and notifies him that an ambulance is arriving.

2.3 User Characteristics

Users can use our system when they want to share their data with third parties in order to be analyzed.
Necessary conditions for the user in order to use the system are:

- He must have a smartphone and be able to use it
- He must be in the age of majority to accept Terms and Conditions

The actors involved in use the application are divided into two types:

- IndividualUser:
  An ordinary person who is successfully registered to TrackMe in order to share its data
- CompanyUser:
  A person who, by registration through his company, can requests access to those data

2.4 Assumptions, Dependencies and Constraints

2.4.1 Domain Assumptions

[D1] The registration informations provided by the user are correctly encoded

[D2] Accurate individuals' positions are known by GPS

[D3] Health data are retrieved by a wearable device (i.e. smartwatch)

[D4] Every information sent to third parties is surely received by them

[D5] The data sent to third parties are anonymous

[D6] Any request from third parties is certainly received by the individual

[D7] The ambulance will surely receive the help request from AutomatedSOS system

2.5 The World and The Machine

The world and the machine is a way of thinking about problems and phenomena relevant in the reality of the software system we are to develop. The world contains the phenomena that happens in reality but are not observable by our system; the machine instead contains the phenomena which take place inside the system and are not observable from outside of it. The intersection between these two sets of phenomena is the so called set of shared phenomena, which contains the phenomena that are controlled by the world and observed by the machine or controlled by the machine and observed by the world.

Goals are prescriptive assertions formulated in terms of world phenomena (not necessarily shared); domain properties/assumptions are descriptive assertions assumed to hold in the world; requirements are prescriptive assertions formulated in terms of shared phenomena.

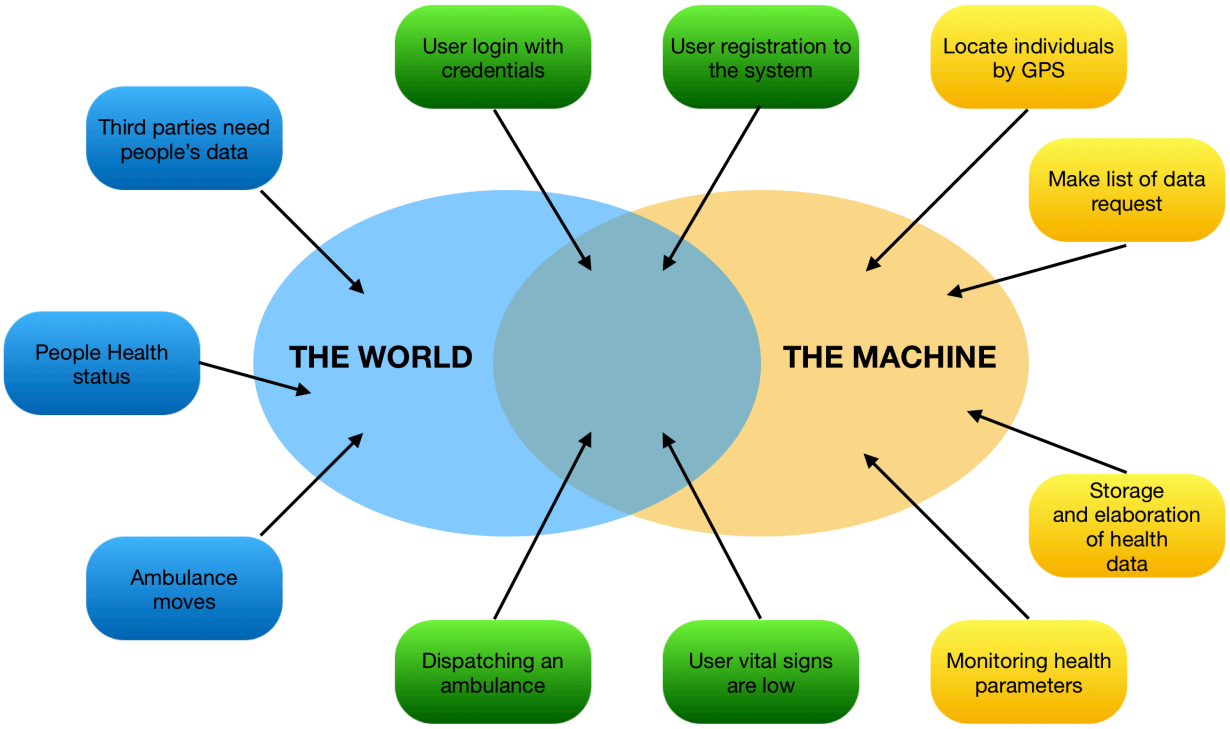

Figure 4: The World and the Machine phenomena

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The following mockups represents a basic idea of what the mobile app will look in the first release

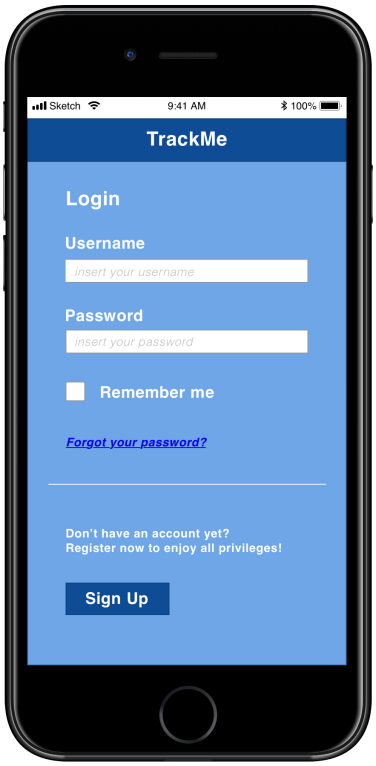

Figure 5: Login mockup

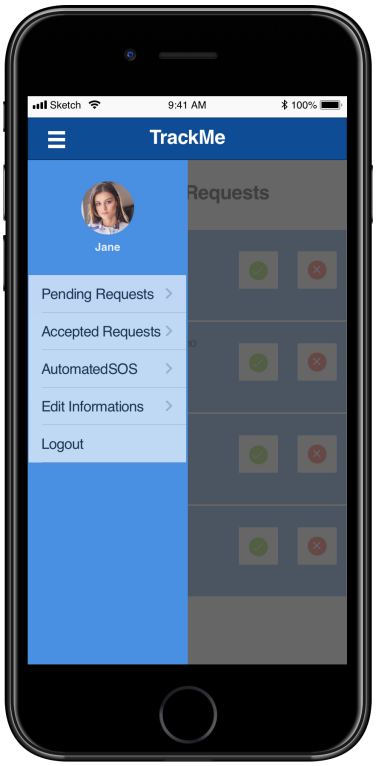

Figure 6: IndividualUser menu interface mockup

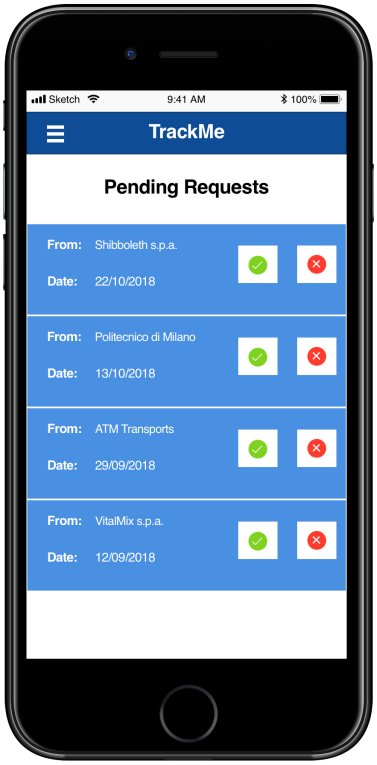

Figure 7: IndividualUser pending requests mockup

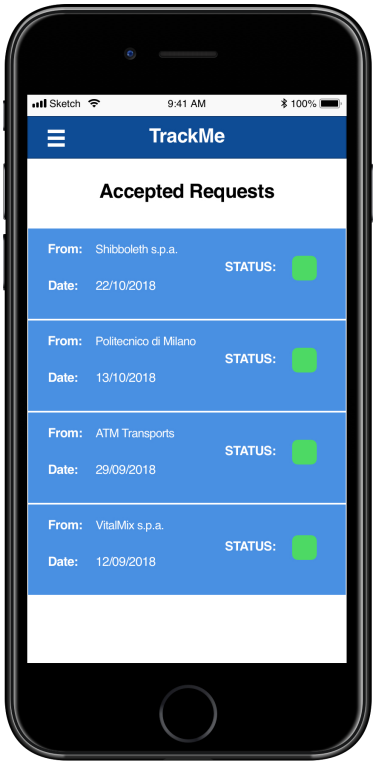

Figure 8: IndividualUser accepted requests mockup

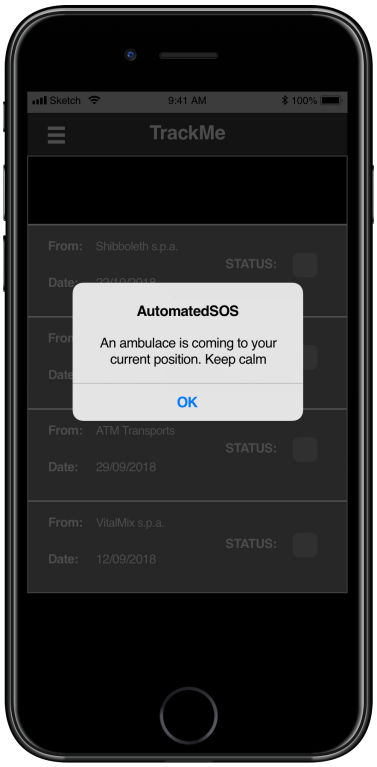

Figure 9: IndividualUser notification by AutomatedSOS

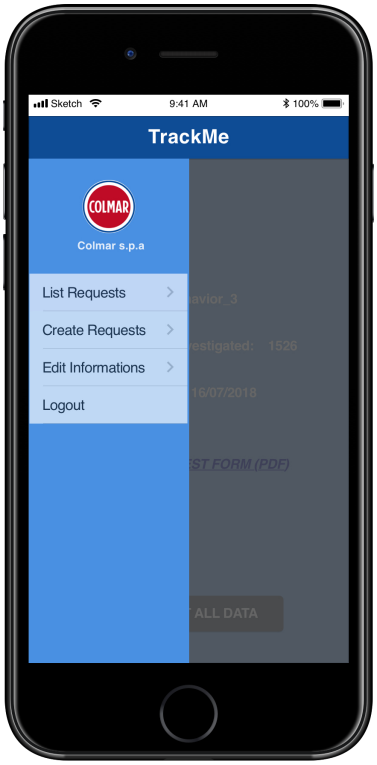

Figure 10: CompanyUser menu interface mockup

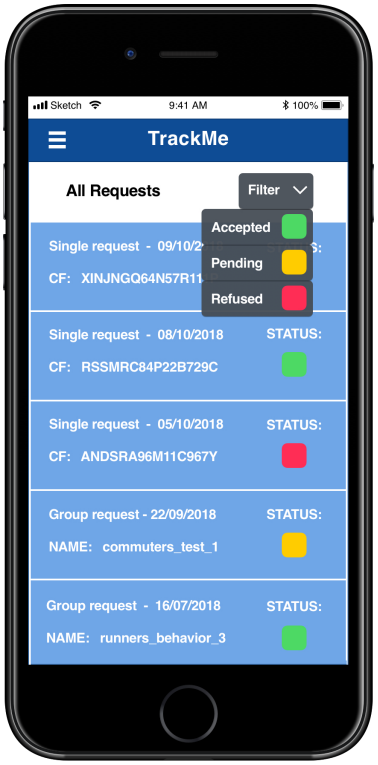

Figure 11: CompanyUser list of all requests made

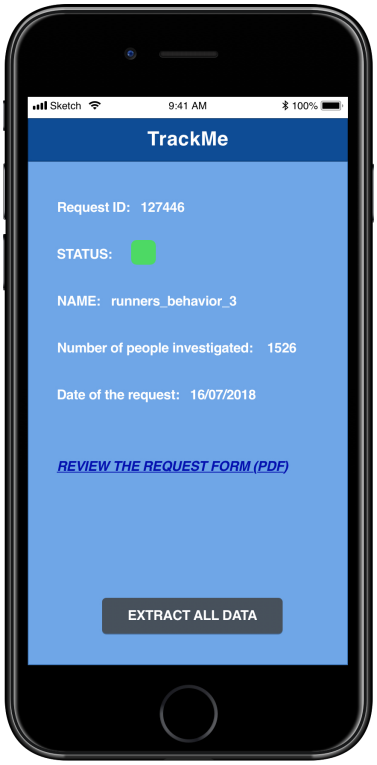

Figure 12: CompanyUser detailed request info

3.1.2 Hardware Interfaces

The application allows registered users to share and retrieve data stored in TrackMe databases. Therefore, individual users must have a wearable device that can calculate and send health data, as well as a recent smartphone capable of using GPS services to obtain user location data

3.1.3 Software Interfaces

The application uses external services to not weigh down the system's architecture. these are:

- Ambulance services
  For proper functioning of AutomatedSOS, the application must be able to communicate with the National Health Service. In this way the position data related to the person who feels bad will be sent, so that it can be reached by an ambulance as soon as possible.

  Furthermore, sending personal data speeds up the patient identification process and gives a lot of useful information on what the symptoms might be.

  This is made possible thanks to the AmbulanceExternalSystem API that guarantee the receipt of a distress call and the immediate deployment of an ambulance.

- Health calculator services
  The data captured by the wearable device are managed and calculated by an external service able to recognize if the vital parameters are respected or are below the standard thresholds (automatically set based on the data entered by the user, such as gender, height, weight , etc…)

3.1.4 Communication Interfaces

The application uses HTTP protocol to send and receive data.
For AutomatedSOS service, we assume that the ExternalAmbulanceSystem has APIs to be able to receive requests for help directly by AutomatedSOS service.

3.2 Scenarios

Here are some scenarios that describe the usage of the system.

3.2.1 Scenario 1

The municipality of Milan is conducting a research on the number of commuter students. the group of researchers in charge decides to use TrackMe to quickly get the data they are interested in. after registering as a third party, they start a new request for a group of people: they set the Central Milan Station as the place and the age of the individuals between 16 and 21 years old, then they send the request to the system.

The system then sends a notification to the researchers' account, notifying them that their request has been accepted and that the data is ready to be exported. At this point the researchers download all the data to be analyzed.

3.2.2 Scenario 2

Mario Rossi every morning use to jogging in Sempione Park in the center of Milan.

Last week he bought a new smartwatch and downloaded the TrackMe app on his mobile phone and signed up to that. During his morning activity, Mario Rossi starts to fell bad, indeed the AutomatedSOS service, after analyzing his actual health conditions thanks to the smartwatch, recognizes that he is at risk of hearth attack. Immediately, the AutomatedSOS connects with the AmbulanceExternalSystem and sends to that the help request with the actual Mario Rossi’s location data and health conditions. After this time (that longs no more than 5 seconds), Mario Rossi receives a notification on his mobile phone by TrackMe that warnings that an ambulance is coming for rescue him.

3.2.3 Scenario 3

Shibboleth s.p.a. is a pharmaceutical company that has just entered the market.

John, a research and development officer, discovers through online advertising the new TrackMe application that allows to keep track of the health data of those participating in the initiative. He is developing a new drug for people with low blood pressure and, in order to analyze the long-term data of the patients to whom the drug will be submitted, he decides to download TrackMe on his mobile device.

John then opens the application and creates multiple individual requests for data to be sent to his patients, searching for them using their tax code.

Once the request has been sent, the patients accept TrackMe's data processing, which will then provide real-time feedback to Shibboleth, without them being physically in the presence of John but simply wearing a wearable device.

3.2.4 Scenario 4

Michael D. is the owner of a small appliance company based in Berlin.

He wants to have the opportunity to monitor the position of their sellers when they spend more than one day away from the office, for example on abroad secondment.

Michael then decides to register for the services offered by TrackMe and asks his sellers to do the same. He then sends a request for tracking location data to all those who perform this function within the company.

Alex, one of the sellers, opens the application that he has previously installed and receives a new notification from his company: he then accepts to share his personal data, retrieved from his smartphone, so that the owner always knows that he is doing his job properly.

3.3 Functional Requirements

[G1] Allow an individual to become registered user after providing credentials

[D1] The registration informations provided by the user are correctly encoded
[R1] The username is unique for each user
[R2] A user must be able to begin the registration process, in which the system will ask to fill out the form with his/her credentials
[R3] The user must agree that TrackMe acquires his/her data
[R4] The user must accept the terms and conditions of TrackMe company

[G2] Allow third parties to become registered user after providing credentials

[D1] The registration informations provided by the user are correctly encoded
[R1] The username is unique for each user
[R2] A user must be able to begin the registration process, in which the system will ask to fill out the form with his/her credentials
[R4] The user must accept the terms and conditions of TrackMe company

[G3] Allow third parties to monitor the location and health status of individuals

[D2] Accurate individuals' positions are known by GPS
[D3] Health data are retrieved by a wearable device (i.e. smartwatch)
[D4] Every information sent to third parties is surely received by them

[R5] A registered user must be able to login the application using his/her credentials

[R6] The system must allow third parties to submit a request for access to individual data

[R7] As soon as a request for data is approved, the system makes the previously saved data available to the third party

[R8] The system must allow third parties to subscribe to new data and to receive them as soon as they are produced after the approval of the request

[G4] Allow third parties to monitor the location and health status of groups of individuals

[D2] Accurate individuals' positions are known by GPS

[D3] Health data are retrieved by a wearable device (i.e. smartwatch)

[D4] Every information sent to third parties is surely received by them

[D5] The data sent to third parties are anonymous

[R5] A registered user must be able to login the application using his/her credentials

[R7] As soon as a request for data is approved, the system makes the previously saved data available to the third party

[R8] The system must allow third parties to subscribe to new data and to receive them as soon as they are produced after the approval of the request

[R9] The system must allow third parties to submit a request for access to data of groups of individuals

[R10] The system must guarantee the anonymity of the data of the groups of people, accepting only a request of at least 1000 individuals

[G5] A specific individual can accept or refuse the request of processing his/her personal data from a third party

[D6] Any request from third parties is certainly received by the individual
[R5] A registered user must be able to login the application using his/her credentials
[R6] The system must allow third parties to submit a request for access to individual data

[G6] When health parameters are below certain threshold, the system sends to the location of the customer an ambulance

[D2] Accurate individuals' positions are known by GPS
[D3] Health data are retrieved by a wearable device (i.e. smartwatch)
[D7] The ambulance will surely receive the help request from AutomatedSOS system
[R11] The user must be logged in the application
[R12] Health parameters must be sent to the system every second
[R13] When the system detects that the user's vital signs are low, an ambulance is sent to the last available position according to GPS information

[G7] The AutomatedSOS must guarantee a reaction time of less than 5 seconds in dispatching an ambulance

[D2] Accurate individuals' positions are known by GPS
[D7] The ambulance will surely receive the help request from AutomatedSOS system
[R14] The processing of the distress request must take less than 5 seconds
[R15] The user will be notified through the application that an ambulance is arriving at his/her current position

3.3.1 Use Case Diagram

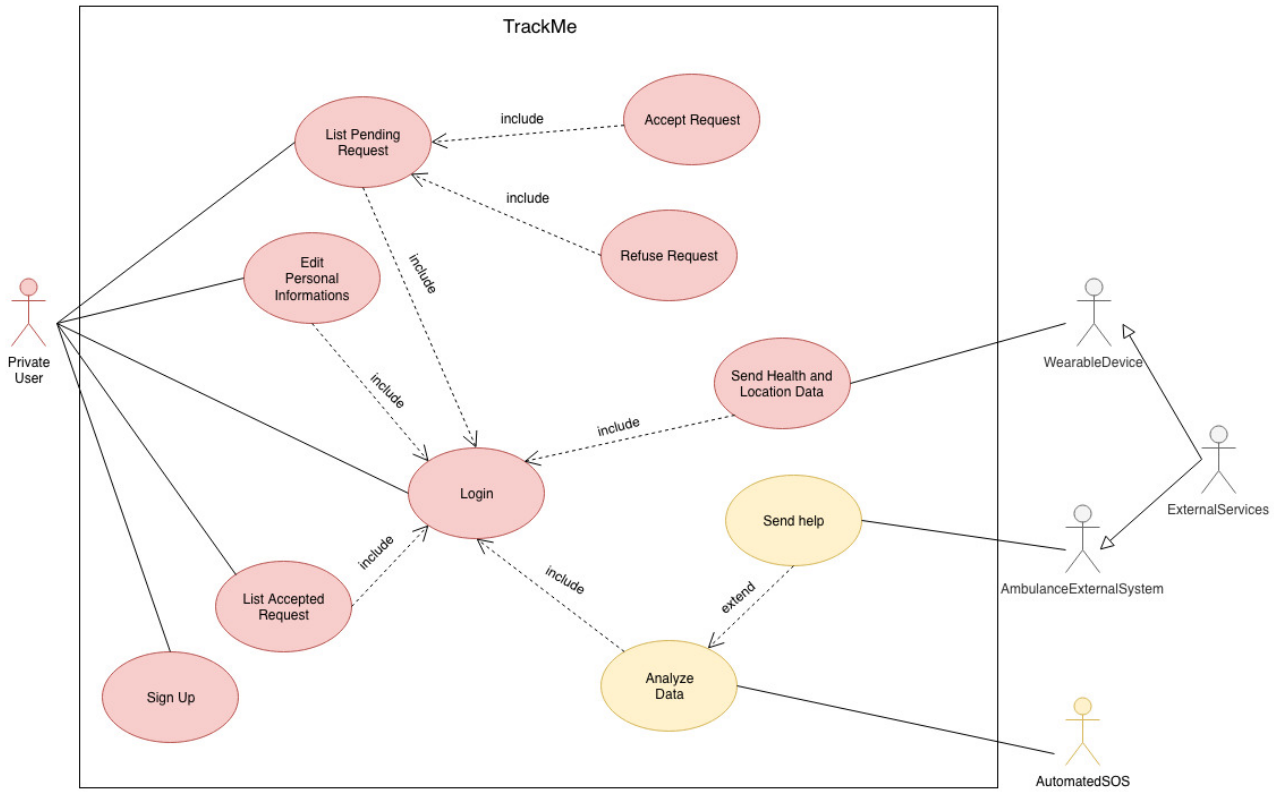

Figure 13: IndividualUser use case diagram

Name	Sign up
Actor	IndividualUser
Entry conditions	The user has already installed the application on his/her device
Events flow	1. The user asks the system to register to its services 2. The system provides the form of registration 3. The user checks the box ”Private user” 4. The user fill out the form with his/her personal data 5. The user read and accept terms and conditions of TrackMe 6. The user accept to share it’s data with TrackMe 7. The user click on ”Sign Up” button to confirm registration 8. The system checks the unicity of email and username inserted 9. The system sends him/her a confirmation email with a unique link to verify the email address 10. The user clicks on the link received and he’s shown that he’s successfully registered to TrackMe
Exit conditions	The user is able to authenticate to the system as registered user with its own credentials
Exceptions	1. If he username inserted is already taken, the system displays an error message asking to insert another username 2. If the email inserted is already used by another user, the system displays an error message asking to insert another valid email

Name	Login
Actor	IndividualUser
Entry conditions	The user has already registered
Events flow	1. The user open the application on his/her device 2. The user inserts his username and password in the appropriate form 3. The user checks the box ”Remember me” so that he will not be asked to enter the credentials in future accesses 4. The user clicks on ”Login” button
Exit conditions	If the credential validation is successful the user is redirect to TrackMe homepage
Exceptions	1. If the credential validation failed an error message is displayed

Name	List pending requests
Actor	IndividualUser
Entry conditions	The user has already logged in
Events flow	1. The user clicks on the dropdown menu button 2. The user select the voice ”List pending requests” 3. The system shows a list of pending requests ordered by addition date
Exit conditions	The user can accept or deny each single request
Exceptions	

Name	List accepted requests
Actor	IndividualUser
Entry conditions	The user has already logged in
Events flow	1. The user clicks on the dropdown menu button 2. The user select the voice ”List accepted requests” 3. The system shows a list of accepted request with the date and the sender
Exit conditions	The user is shown the history of his requests
Exceptions	

Name	Edit personal informations
Actor	IndividualUser
Entry conditions	The user has already logged in
Events flow	1. The user clicks of the dropdown menu button 2. The user select ”Edit information” voice 3. The user is shown his personal information 4. The user is allowed to modify his personal information 5. The user clicks on ”Save” button
Exit conditions	User’s data have been correctly modified
Exceptions	1. If the inserted information are not valid, an error message is displayed

Name	Accept request
Actor	IndividualUser
Entry conditions	1. The user has already logged in 2. The user has already received at least one request
Events flow	1. The user clicks on the dropdown menu button 2. The user select the voice ”List pending requests” 3. The system shows a list of pending request ordered by addition date 4. The user taps on ☑ to accept sharing its data with the selected company
Exit conditions	From now on the user is sharing its data with the selected company and the request is added to the accepted requests list
Exceptions	1. If the request is expired, a message is displayed and the request is automatically deleted

Name	Refuse request
Actor	IndividualUser
Entry conditions	1. The user has already logged in 2. The user has already received at least one request
Events flow	1. The user clicks on the dropdown menu button 2. The user select the voice ”List pending requests” 3. The system shows a list of pending request ordered by addition date 4. The user taps on ⊠ to refuse sharing its data with the selected company
Exit conditions	The refused request is deleted from the list
Exceptions	1. If the request is expired, a message is displayed and the request is automatically deleted

Name	Send health and location data
Actor	Data4Help
Entry conditions	The user has already logged in
Events flow	1. The system acquires the data through the wearable device of the user 2. The system stores the health and location data into the database
Exit conditions	The user’s data are correctly stored into the database of TrackMe
Exceptions	

Name	Analyze data
Actor	AutomatedSOS
Entry conditions	The user has already logged in
Events flow	1. The system gets the health data of the user 2. The system analyze the user's health data 3. The system verifies that threshold values are respected
Exit conditions	1. The system is now able to communicate with AmbulanceExternalSystem
Exceptions	1. If the data received are insufficient to establish if they are under the fixed thresholds, a warning message is displayed

Name	Send help
Actor	AutomatedSOS
Entry conditions	The health values of the user don't respect the thresholds
Events flow	1. The system retrieves the health and location data of the user 2. The system contacts the AmbulanceExternalSystem 3. The system sends all the user data to the AmbulanceExternalSystem 4. The system notifies the user that an ambulance is coming to his rescue
Exit conditions	The user is notified of the incoming ambulance
Exceptions	

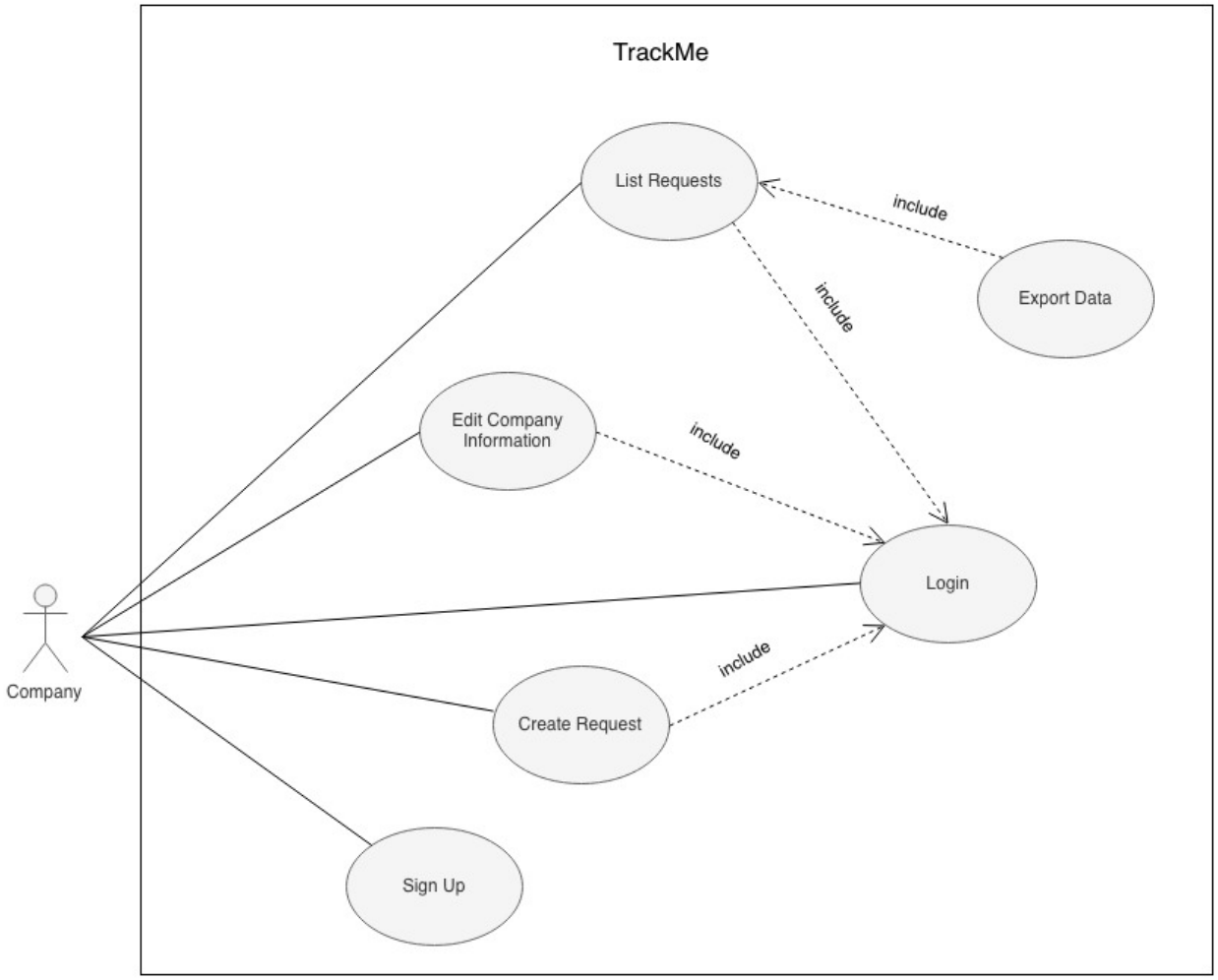

Figure 14: CompanyUser use case diagram

Name	Sign up
Actor	CompanyUser
Entry conditions	The user has already installed the application on his/her device
Events flow	1. The user asks the system to register to its services 2. The system provides the form of registration 3. The user checks the box ”Company user” 4. The user fill out the form with his/her company data 5. The user read and accept terms and conditions of TrackMe 6. The user click on ”Sign Up” button to confirm registration 7. The system checks the unicity of email and username inserted 8. The system sends him/her a confirmation email with a unique link to verify the email address 9. The user clicks on the link received and he’s shown that he’s successfully registered to TrackMe
Exit conditions	The user is able to authenticate to the system as registered user with its own credentials
Exceptions	1. If he username inserted is already taken, the system displays an error message asking to insert another username 2. If the email inserted is already used by another user, the system displays an error message asking to insert another valid email

Name	Login
Actor	CompanyUser
Entry conditions	The user has already registered
Events flow	1. The user open the application on his/her device 2. The user inserts his username and password in the appropriate form 3. The user checks the box ”Remember me” so that he will not be asked to enter the credentials in future accesses 4. The user clicks on ”Login” button
Exit conditions	If the credential validation is successful the user is redirect to TrackMe homepage
Exceptions	1. If the credential validation failed an error message is displayed

Name	List requests
Actor	CompanyUser
Entry conditions	The user has already logged in
Events flow	1. The user clicks on the dropdown menu button 2. The user select the voice ”List requests” 3. The system shows a list of all requests made by the company ordered by addition date 4. The user can filter the requests by pending requests, accepted requests or refused requests
Exit conditions	The user is shown the history of his data requests
Exceptions	

Name	Export data
Actor	CompanyUser
Entry conditions	1. The user has already logged in 2. There are accepted requests
Events flow	1. The user clicks on the dropdown menu button 2. The user select the voice ”List requests” 3. The system shows a list of all requests made by the company ordered by addition date 4. The user taps on an accepted request 5. The user is shown the main information about the request 6. The user taps on ”Export” button to store the all data available on his device or cloud storage
Exit conditions	The user can now access offline the data collected
Exceptions	1. If there isn’t enough capacity to store the data for offline mode, the user is notified with an error message

Name	Edit company informations
Actor	CompanyUser
Entry conditions	The user has already logged in
Events flow	1. The user clicks of the dropdown menu button 2. The user select ”Edit information” voice 3. The user is shown his personal informations 4. The user is allowed to modify his personal information 5. The user clicks on ”Save” button
Exit conditions	User’s data have been correctly modified
Exceptions	1. If the inserted information are not valid, an error message is displayed

Name	Create request
Actor	CompanyUser
Entry conditions	The user has already logged in
Events flow	1. The user clicks on the dropdown menu button 2. The user select the voice ”Create request” from the menu 3. The user is asked if he wants to create a single or group request 4. The user chooses the type of the request 5. The user is shown a form with the data to be asked 6. The user fill out the form 7. The user taps on ”Send” to send the request
Exit conditions	The request has been sent and the user can see the status of the request in the list requests
Exceptions	1. If the data inserted are not valid, an error message is displayed

3.3.2 Sequence Diagram

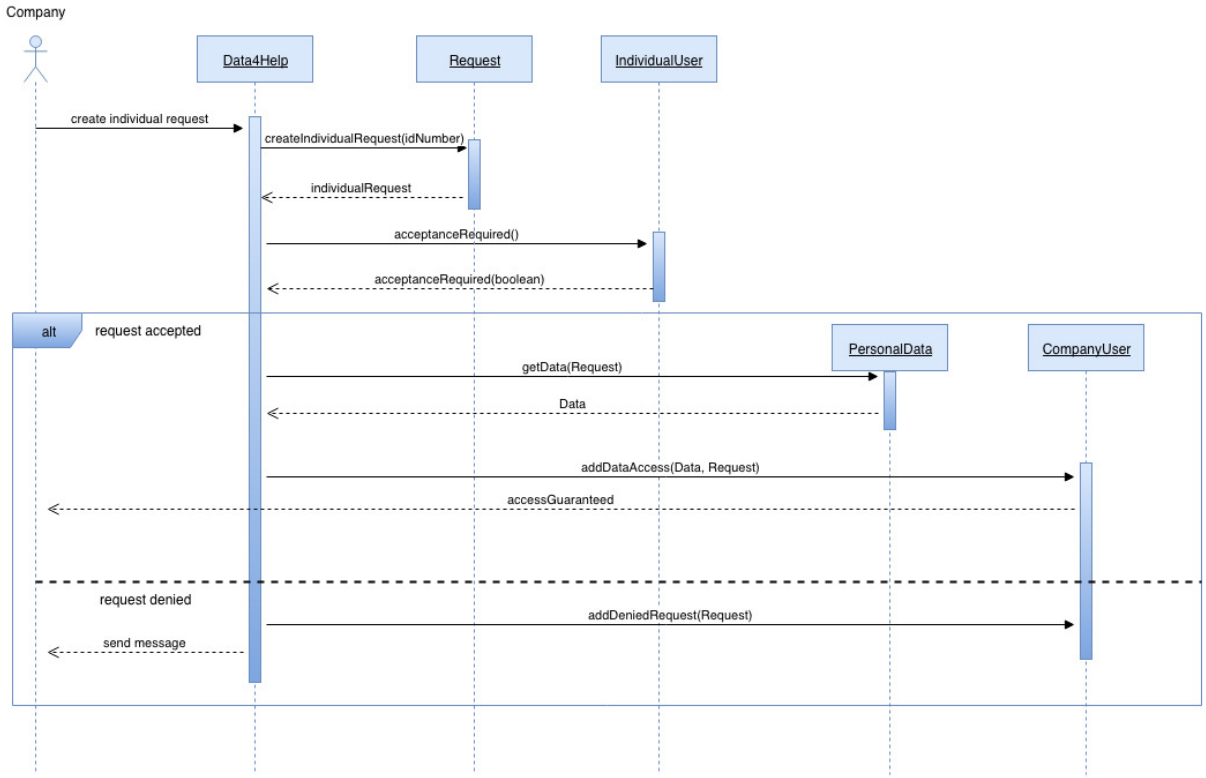

Figure 15: CompanyUser creates a single request

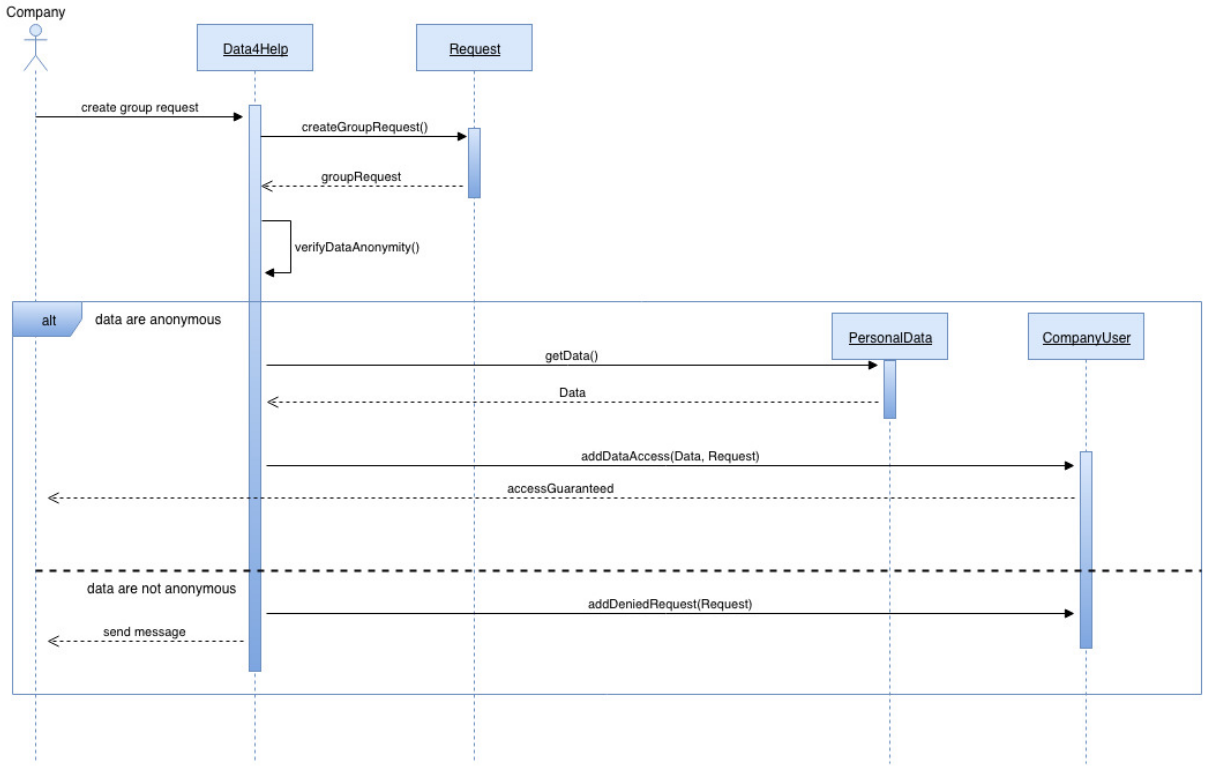

Figure 16: CompanyUser creates a group request

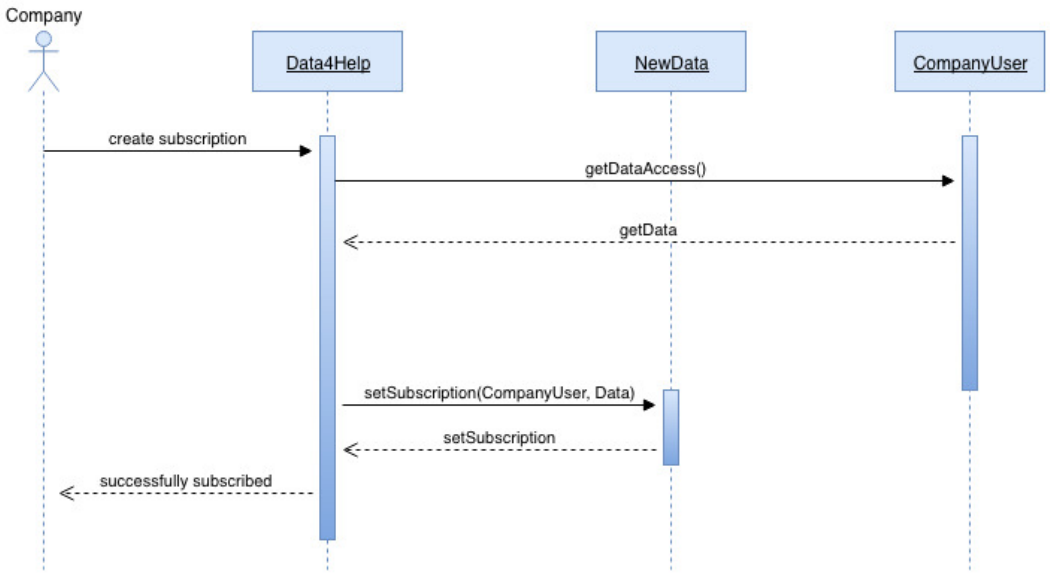

Figure 17: CompanyUser makes a subscription to new data

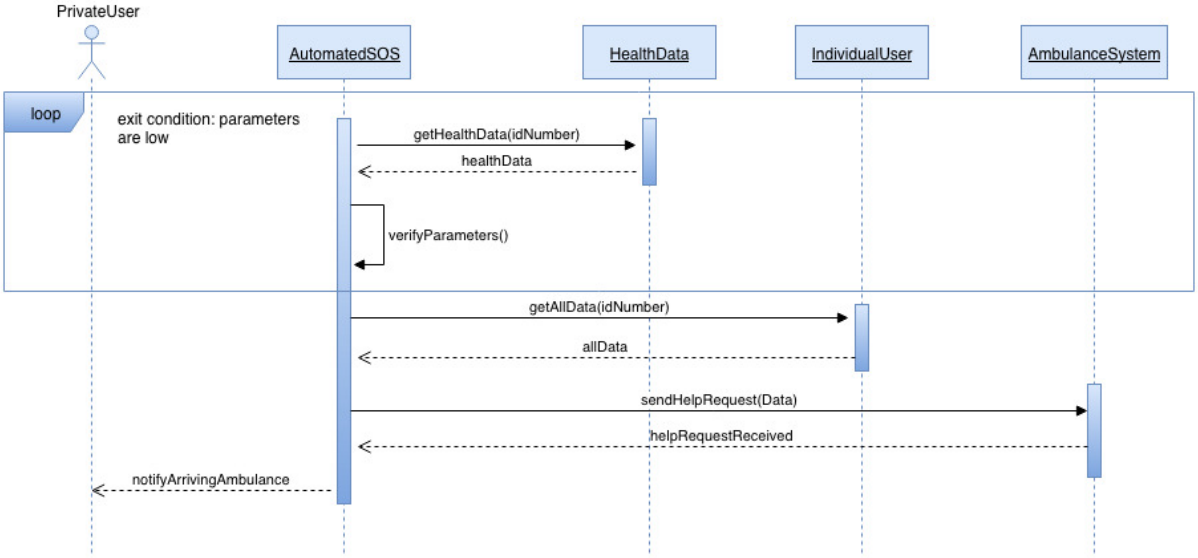

Figure 18: AutomatedSOS sends an help request

3.4 Performance Requirements

The system must manage thousands of data requests per minute without any data loss.
Furthermore, AutomatedSOS service must guarantee a reaction time of less than 5 seconds in contacting the external Ambulance Service when it finds out that health parameters of the monitored person are below the established thresholds.

3.5 Design Constraints

3.5.1 Standard Compliance

The application supports both landscape and portrait orientations with same features and action, but minor changes in visualization are acceptable.

The application supports multitasking features: when it's put in the background, the system memorizes the current status and any pending transition. Once it is brought back to the forefront, the system shows itself to the user as it was left.

3.5.2 Hardware Limitations

For a correct functioning of all the services offered by the application, the user must have a wearable device always with him (i.e. smartwatch) able to connect to a smartphone with iOS or Android operating system.
A 2G / 3G / 4G internet connection and a GPS system are also required.

3.5.3 Other Constraints

The system asks the user to read carefully and accept Terms and Conditions of TrackMe services, in which is explained the Privacy Policy on data processing.

3.6 Software System Attributes

3.6.1 Reliability

The application must be available 24/7 and databases in which data are stored must always be online and accessible.

3.6.2 Availability

The system must be available 99,9% of the time. The system should be accessible 24 hours per day.

3.6.3 Security

Users personal informations are encrypted and must be protected during transmission. Restricted access APIs must check that who tries to use them is actually allowed to do so.

3.6.4 Maintainability

During the development of the application, the possibility of adding new features to those already present will be taken into account; for this purpose the implementation will use some of the most common design patterns and the code will be well commented, so as to facilitate any maintenance interventions carried out also by external.

3.6.5 Portability

The system must be also accessible by the most common mobile platforms (iOS and Android devices).

4. Formal Analysis using Alloy

The Alloy model is focused on the main product functions. It explains in more details the Requests that a CompanyUser can do and the AutomatedSOS system:

- Every SingleRequest must have a unique sender, a unique recipient, specific data that are own by the IndividualUser, and a Status.

- Every GroupRequest must have a unique sender, specific data and a status.

- A GroupRequest is accepted if and only if the request data are anonymous.

- A CompanyUser can access the data only if the status of his Request is Accepted.

- The AutomatedSOS system is unique for every IndividualUser who can decide to have it or not.

- The AutomatedSOS monitors the healthData of the IndividualUser, in particular his vitalParameters and if they are lower or equal than 4, sends a warning to the IndividualUser.

It's important to notice that different GroupRequest can have the same data (in fact there can be two different CompanyUser who made two different Request for the same data).

In addition, a GroupRequest to be anonymous must use data of at least 1000 people. to simplify the model we considered 4 as a threshold.

4.1 Alloy Model

```
open util/boolean
open util/integer


sig AutomatedSOS{
    user: one IndividualUser,
    sendWarning: one Bool
}
{(user.data.healthData.vitalParameters <=4 implies sendWarning = True) and
(sendWarning = True implies user.data.healthData.vitalParameters <=4)}

-------------------------------------------------------------------------------------

sig Data{
    healthData: one HealthData,
    locationData: one LocationData,
}
sig HealthData{
    vitalParameters:  Int
}{vitalParameters >= 0 and vitalParameters <=7}
sig LocationData{}

-------------------------------------------------------------------------------------

//Requests Status
abstract sig Status{}
sig Accepted extends Status{}
sig Pending extends Status{}
sig Refused extends Status{}

-------------------------------------------------------------------------------------

abstract sig Request{
    sender: one CompanyUser,
    status: one Status,
    data: one Data
}

sig SingleRequest extends Request{
    recipient: one IndividualUser,
}
//The number of people in a GroupRequest must be greater or equal
//than 4 to be anonymous variable equal to True
sig GroupRequest extends Request{
    anonymous: one Bool,
    numberPeople: one Int
}
{(numberPeople>0) and
(anonymous = False implies status = Refused) and
(status = Refused implies anonymous=False) and
(anonymous = True implies status = Accepted) and
(status = Accepted implies anonymous=True) and
(anonymous = True implies numberPeople >=4) and
(numberPeople >=4 implies anonymous=True)}

-------------------------------------------------------------------------------------
```

```
abstract sig RegisteredUser{}
sig IndividualUser extends RegisteredUser{
    receivedRequests: set SingleRequest,
    data: one Data,
    automatedSOS: lone AutomatedSOS
}
sig CompanyUser extends RegisteredUser{
    sentRequests: set Request,
    availableData: set Data
}

-----------------------------------------------------------------------------------

//Every AutomatedSOS instance has its IndividualUser instance
fact AutomatedSOSIndividualUserConnection{
    all u: IndividualUser | all a: AutomatedSOS |
    (a in u.automatedSOS implies a.user=u) and (u in a.user implies u.automatedSOS=a)
}

//Every Request made by a CompanyUser implies that the CompanyUser
//is the sender of the Request
fact RequestCompanyUserConnection{
    all u:CompanyUser | (all r: Request |
    (r in u.sentRequests implies r.sender=u) and
    (u in r.sender implies u.sentRequests=r))
}

//Every CompanyUser can access to the AvailableData only if
//has already sent a request and the request has been accepted
fact AvailableData{
    all u: CompanyUser | all d: Data | (d in u.availableData implies
    (some r: Request | (r in u.sentRequests and r.status=Accepted and r.data=d)))
    and (all r: Request | (r in u.sentRequests and r.status=Accepted and r.data=d)
    implies d in u.availableData)
}

//Every data in a SingleRequest must be the same in the IndividualUser
//who is the recipient of the Request
fact DataIndividualUserDataRequestConnection{
    all r: SingleRequest | (all u: IndividualUser |
    (u in r.recipient implies r.data = u.data))
}


//Every Request recieved by an IndividualUser implies that the IndividualUser
//is the recipient of the Request
fact SigleRequestIndividualUserConnection{
    all u: IndividualUser | (all r: SingleRequest |
    (r in u.receivedRequests implies r.recipient = u) and
    (u in r.recipient implies u.receivedRequests = r))
}

//Every IndividualUser has his own Data
fact UniqueIndividualData{
    no disjoint u1,u2: IndividualUser | u1.data = u2.data
}

//The Data of a GroupRequest must be different from the Data of an IndividualUser
fact NoCommonDataGroupRequestIndividualUser{
    all u: IndividualUser | all r: GroupRequest | u.data !=r.data
}
```

```
//The Data must be unique
fact UniqueData{
    no disjoint d1,d2: Data | d1.healthData = d2.healthData or
    d1.locationData = d2.locationData
}

-----------------------------------------------------------------------------------------

pred createSingleRequest(u: CompanyUser, u2: IndividualUser, r: SingleRequest, d: Data){
    r.sender = u
    r.recipient = u2
    r.data = d
}

pred createGroupRequest(u: CompanyUser, r: GroupRequest, d: Data){
    r.sender = u
    r.data = d
}

pred acceptedRequest(u: IndividualUser, r: SingleRequest){
    r.status= Accepted
    u.receivedRequests = r
}

pred refusedRequest(u: IndividualUser, r: SingleRequest){
    r.status= Refused
    u.receivedRequests = r
}

pred sendWarning(a: AutomatedSOS, u: IndividualUser){
    a.user = u
    u.data.healthData.vitalParameters <= 4
}

pred show{}

//A CompanyUser has any availableData if he has already sent a request
//and this has been already accepted
assert availableData{
    all u: CompanyUser | (all r: Request | all d: Data |
    ( r in u.sentRequests and d in r.data and r.status=Accepted)
    implies u.availableData=d)
}
//A group request must be refused if the number of people investigated is less than 4
assert anonymityGroupRequest{
    all r: GroupRequest | (r.numberPeople < 4 implies r.status=Refused)
    and (r.status=Refused implies r.numberPeople < 4)
}

-----------------------------------------------------------------------------------------

run show for 10 but  exactly 2 Status, exactly 2 Data,
    exactly 2 LocationData, exactly 2 HealthData, exactly 1 AutomatedSOS,
    exactly 1 IndividualUser, exactly 1 SingleRequest,
    exactly 1 GroupRequest, exactly 2 CompanyUser
run sendWarning
run refusedRequest
run acceptedRequest
run createSingleRequest
run createGroupRequest
check availableData
check anonymityGroupRequest
```

4.2 World Generated

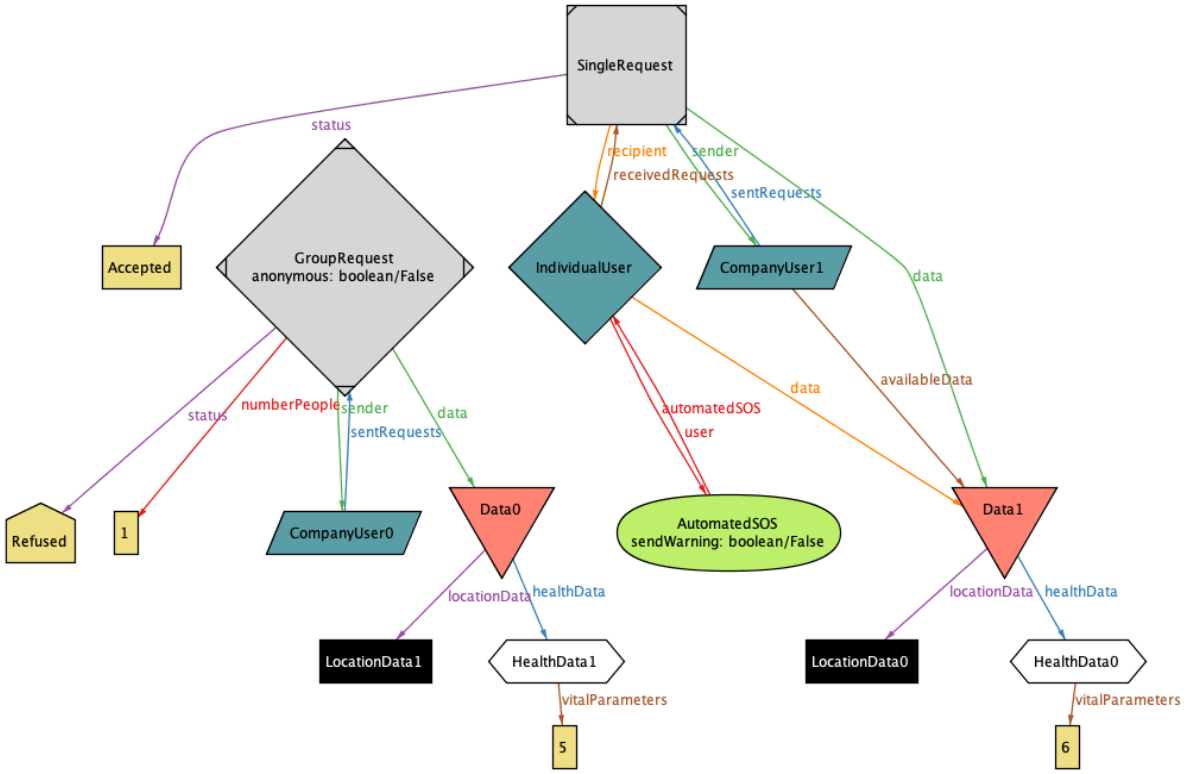

Figure 19: World generated by Alloy

4.3 Alloy Results

These are the results of the code shown above

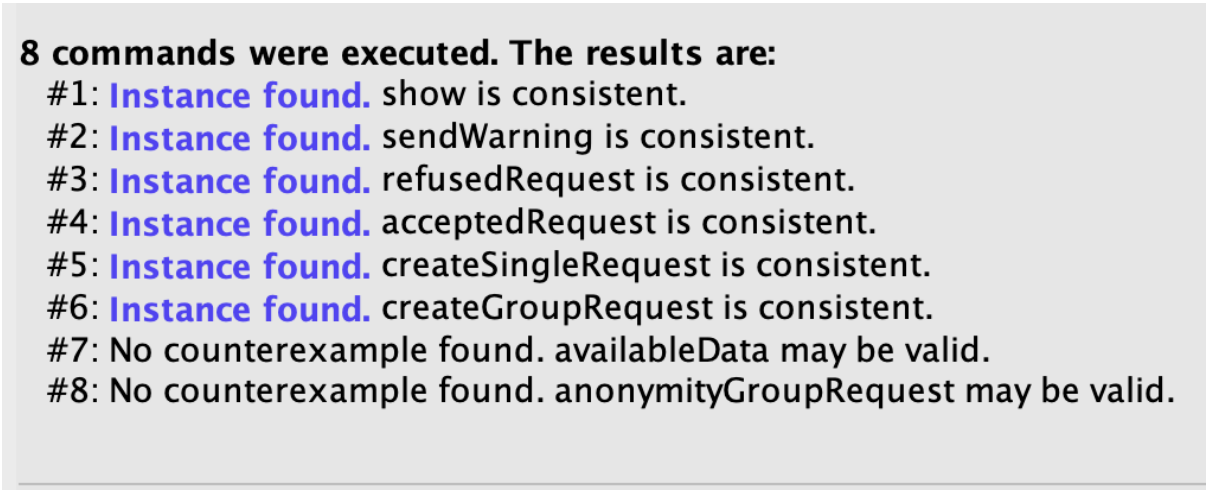

Figure 20: Alloy results

5. Effort spent

5.1 Matteo Velati

TASK	HOURS
Introduction	6
Product functions and perspective	5
Assumptions	3
External interface requirements	3
Scenarios	2
Functional requirements	7
Alloy	4
Various	3
RASD	4

Total: about 37 hours

5.2 Niccolò Zangrando

TASK	HOURS
Introduction	6
Product functions and perspective	5
Assumptions	3
External interface requirements	3
Scenarios	2
Functional requirements	7
Alloy	6
Various	3
RASD	2

Total: about 37 hours

6. References

- Specific documentation ”AY 2018-2019 Project Assignment”
- Alloy documentation
- Slides from Software Engineering 2 [AA 2018/2019] - Politecnico di Milano
- IEEE Recommended Practice for Software Requirements Specification