POLITECNICO
MILANO 1863

Politecnico di Milano
AA 2018-2019
Computer Science and Engineering
Software Engineering 2 Project

Dalle Rive Fabio - 920082
Di Giacomantonio Marco - 846515

Table of Contents

1. Introduction
    1.1. Purpose
    1.2. Scope
        1.2.1.. Description of the given problem
        1.2.2.. Goals
    1.3. Definitions, Acronyms, Abbreviations
        1.3.1.. Definitions
        1.3.2.. Acronyms
        1.3.3.. Abbreviations
    1.4. Document structure
2. Overall Description
    2.1. Product perspective
    2.2. Product functions
        2.2.1.. Acquisition of users' data
        2.2.2.. Company access to users' data
        2.2.3.. Forwarding of following requests directly to the users and data anonymization
        2.2.4.. AutomatedSOS
    2.3. User characteristics
    2.4. Assumptions, dependencies and constraints
        2.4.1.. Domain assumptions
3. Specific requirements
    3.1. External interface requirements
        3.1.1.. Users interfaces
        3.1.2.. Hardware interfaces
        3.1.3.. Software interfaces
        3.1.4.. Communication interfaces
    3.2. Scenarios
        3.2.1.. Scenario 1
        3.2.2.. Scenario 2
        3.2.3.. Scenario 3
    3.3. Functional requirements
        3.3.1.. Use case diagram
        3.3.2.. Sequence diagram

3.4. Performance requirements
3.5. Design Constraints
3.5.1.. Standard compliance
3.5.2.. Hardware limitation
3.5.3.. Other constraint
3.6. Software system attributes
3.6.1.. Reliability
3.6.2.. Security
3.6.3.. Maintainability
3.6.4.. Compatibility
4. Formal Analysis Using Alloy
4.1. Alloy model
4.2. World generated
4.3. Alloy results
5. Effort Spent
5.1. Dalle Rive Fabio
5.2. Di Giacomantonio Marco
6. Resources
7. Revision History

1 Introduction

1.1 Purpose

The purpose of this project is to build a system, called Data4Help, that allows third parties to monitor the position and health status of users. The data are collected by TrackMe, the company that wants to develop Data4Help, and are shared with other companies which are interested in those data. Furthermore, TrackMe wants to develop AutomatedSOS, a system build on top of Data4Help. AutomatedSOS is a service designed for elderly people, it is able to intervene by calling an ambulance if the health parameters of the user are below some fixed thresholds.

1.2 Scope

1.2.1 Description of the given problem

TrackMe is a company that wants to develop a software-based service allowing third parties to monitor the location and health status of users. This service is called Data4Help. The service supports the registration of the visitors who, by registering, allow TrackMe to acquire their data. Also it supports the registration of third parties. After registration, these third parties can request:

- Access to the data of some specific user.
- Access to anonymized data of groups of users.

TrackMe also wants to develop a non-intrusive SOS service for elderly people, called AutomatedSOS. AutomatedSOS is build on top of Data4Help. This service is designed to monitor health status of users and to send an ambulance to the location of the user if some parameters are below some specified thresholds.

1.2.2 Goals

- [G1] Visitor can become User after providing credentials.
- [G2] User can accept or reject the request of access to his data formulated by companies.
- [G3] If user's parameters are below specified thresholds, an ambulance is called within 5 seconds.
  - [G3.1] Ambulance is required at current user's location.
- [G4] Company can sign up as Company to Data4Help and AutomatedSOS.
- [G5] Company can be recognized providing a password and vat number.
- [G6] Company can formulate a request to see anonymized data of a group of users.

- [G7] Company can formulate a request to see data of a specific user providing his SSN.
- [G8] Company can see anonymized data of a group of users.
- [G9] Company can see data of a specific user providing his SSN.
- [G10] Company can subscribe to users' new data.
- [G11] Data4Help can forward companies' requests to users.
- [G12] A user of Data4Help becomes automatically a user of AutomateSOS if he is older than *Age*.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- **Visitor:** a person who still has to register to Data4Help and AutomatedSOS.
- **User:** a person who is registered to Data4Help and AutomatedSOS.
- **Third Parties / Companies:** company.
- **Data:** User's monitored data: location + heart rate + calories burned + time spent exercising + step walked
- **Threshold:** Flexible value related to the heart rate data acquired by the smart watch in which the system-to-be is installed. This value is computed by well-known equation that operate with user's data. Among the others it also depends of the kind of activity that a user is doing and his age.
- **Subscribe to data:** A company which is subscribed to user's data or to users group's data, receives the requested data as soon as they are produced.
- **Pendent user:** A user that a company want to follow but has not answered to the following request yet.
- **Single user request / Following request:** A formal request issued by a company to see the data of a user.
- **Group of users request:** A formal request issued by a company to see the data of a group of users.
- **Dandling request:** A single user request that has not already been approved by a user.
- **Age:** age over which a user is subscribed to AutomateSOS.

- **Ambulance Service:** service which is offered by external components. AutomatedSOS will contact this service sharing the current location if a user if the user's heart rate is below the *Threshold*.
- **Call Ambulance**: this term is used through the document has a short cut to: "AutomatedSOS issue a formal request for an ambulance at the current user location to the Ambulance Service".

1.3.2 Acronyms

- RASD - Required Analysis and Specification Document
- GPS - Global Positioning System
- SSN - Social Security Number

1.3.3 Abbreviations

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

1.4 Document structure

Introduction gives an introduction to the problem and describe the purpose of Data4Help and AutomatedSOS. It also contains the goals that these systems-to-be must be able to deliver to users and third party companies.

Overall Description gives an overview of the functions that the systems-to-be are able to deliver to users and third parties. In this section, users of the systems-to-be, are better identified, i.e. the kind of users that interact with Data4Help and AutomatedSOS. Assumptions used by Data4Help and AutomatedSOS are presented in this section.

Specific Requirements gives and overview of which functional requirements are needed to fulfill goals we already presented. Moreover it gives an overview of non-functional requirements.

Formal Analysis Using Alloy includes the Alloy model and the discussion of its purpose. Also, a world generated by this model is shown.

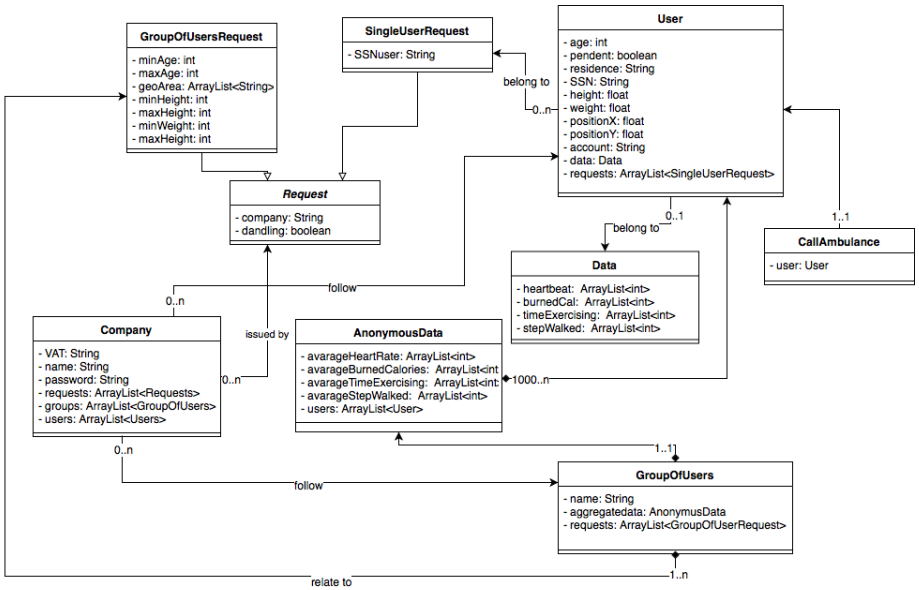
Effort Spent shows the effort spent by each group member while working on this project.

Resources includes the reference documents.

2 Overall Description

2.1 Product perspective

Data4Help and AutomatedSOS are intended as a background application that are installed inside a smart watch. The systems-to-be acquire data of the users through the smart watch. A user register himself for Data4Help accessing the website page(https://www.data4help.com) that TrackMe provide for this application. Once a user is in the web page he can register himself to Data4Help typing the required information. During the registration process, the user will be required to type his Google or Apple account. This is meant to allow TrackMe to know which device a user has; and among those, let the user picks the one from which data will be acquired. Once a user subscribe to Data4Help, if he has an age that is greater or equal to *Age* he will be automatically enrolled in AutomatedSOS.



2.2 Product functions

Considering all the goals previously presented, we can sum up into four categories the main functions of the product:

2.2.1 Acquisition of users' data

The system-to-be runs on smart watches, assumed to be able to communicate the data directly to TrackMe's servers. It is also assumed that these smart watches have GPS integrated in their hardware. It means that this product works only with smart watches that have a direct internet connection(they have an integrated SIM card). Future updates will enable to use also smart watch that use Bluetooth to communicate data to a smart phone that then send data towards TrackMe servers.

2.2.2 Company access to users' data

The system-to-be enables companies to access data of a specific user or of groups of users. Companies access to these data through the web page(https//www.data4help.com). Data are shown in a user-friendly way. Some samples of the web page will be shown in the User Interface section.

2.2.3 Forwarding of following requests directly to the users and data anonymization

Companies are able to send following requests, in order to access user data. The system-to-be allows companies to issue request to users. Every time that a company request to access some data, a request is created and stored inside TrackMe's servers. Analizing the request, TrackMe's servers will decide how to proceed:

- If the request concerns a specific user, TrackMe sends an email to the user asking to accept or reject it.
- If the request concerns a group of users(with more than 1000 user), create a group and anonymizes the corresponding data.

2.2.4 AutomatedSOS

While registering Data4Help, the user is asked to submit his age. Basing on some creteria chosen from the managment, a value *Age* for the age is chosen. If the user's age is greater than *Age* AutomatedSOS is activated automatically. AutomatedSOS is a service that monitors the heartbeat of the user and if it's lower than *Threshold*, calls an ambulance addressed to the user's position within 5 seconds. This is achieved by sharing the current user location(acquire through GPS integrated in the smart watch) with the external *Ambulance Service*.

2.3 User characteristics

- *Visitor*: a person not registered. The only thing he can do is proceeding with sing up.

- *User*: a person passed through a successful registration process. His data can be tracked, he can accept or reject compaines' following request.
- *Company*: an entity that can be recognised as a company. It can see data of a specific user or anonimous data of a group of users.
- *TrackMe*: the whole hardware and software interfaces over which Data4Help and AutomatedSOS run.

2.4 Assumption, dependencies and constraints

2.4.1 Domain assumptions

- [D1] The user's email is already known by TrackMe.
- [D2] Only real companies can sing up as companies.
- [D3] At most 5 seconds are necessary to send user location and call an ambulance when parameters are below the threshold.
- [D4] Smart watch in which the system-to-be is installed have: *GPS, Accelerometer, LTE, Photoplethysmography.*
- [D5] When the system shows the position of a user it means that the user is actually there.
- [D6] During the registration process the user inserts all the requested data.
- [D7] The user has the physical characteristics that he inserted in the system.
- [D8] Companies have a name and a vat number. VAT number is unique.
- [D9] Companies know the SSN of the user.
- [D10] Current user's residence is the CAP of user's residence.

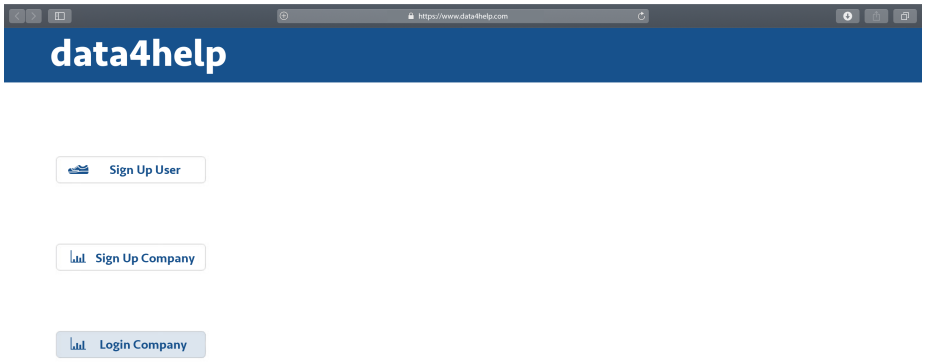
3 Specific Requirements

3.1 External interface requirements

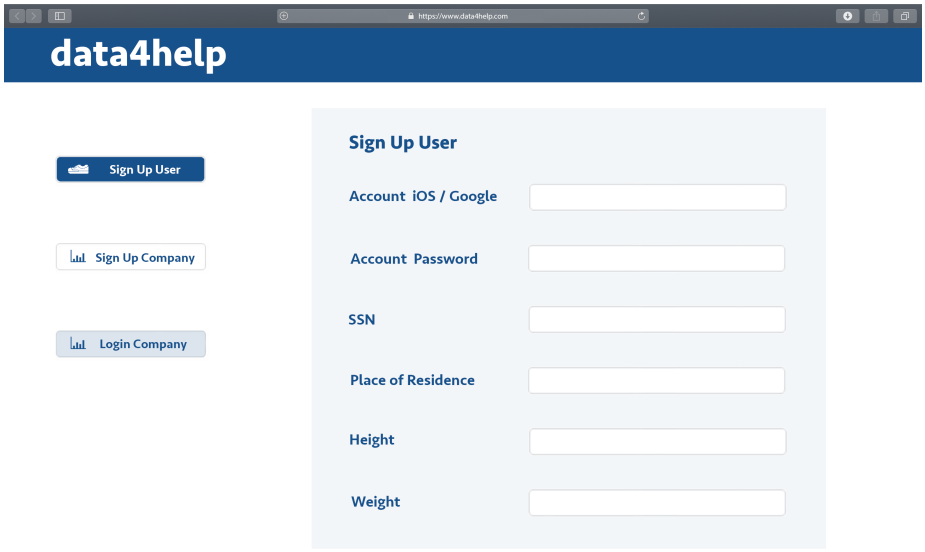
3.1.1 User interfaces

In this section are presented some pictures that show how the web page https://www.data4help.com should look like after the deployment.

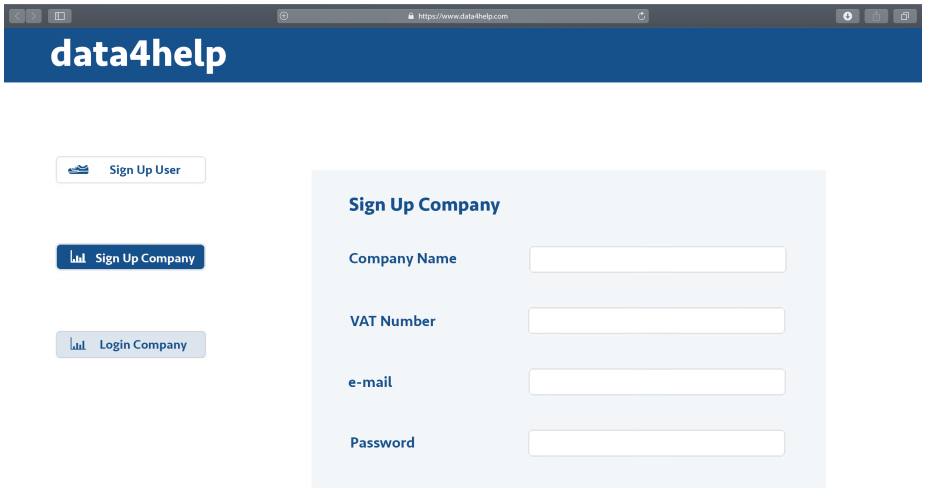
Homepage.



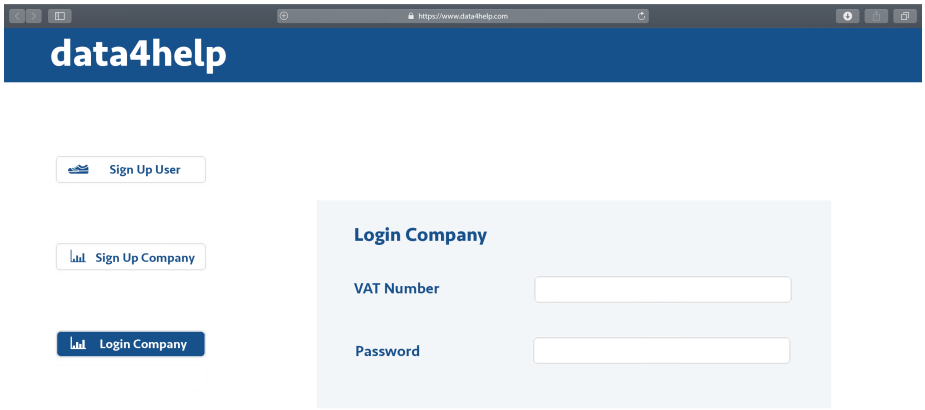
User sign up.



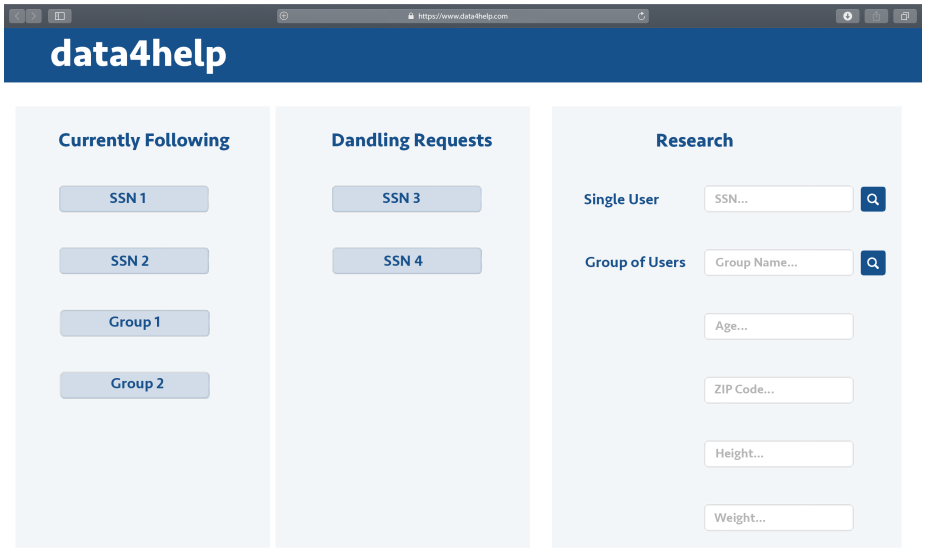
Company sing up.



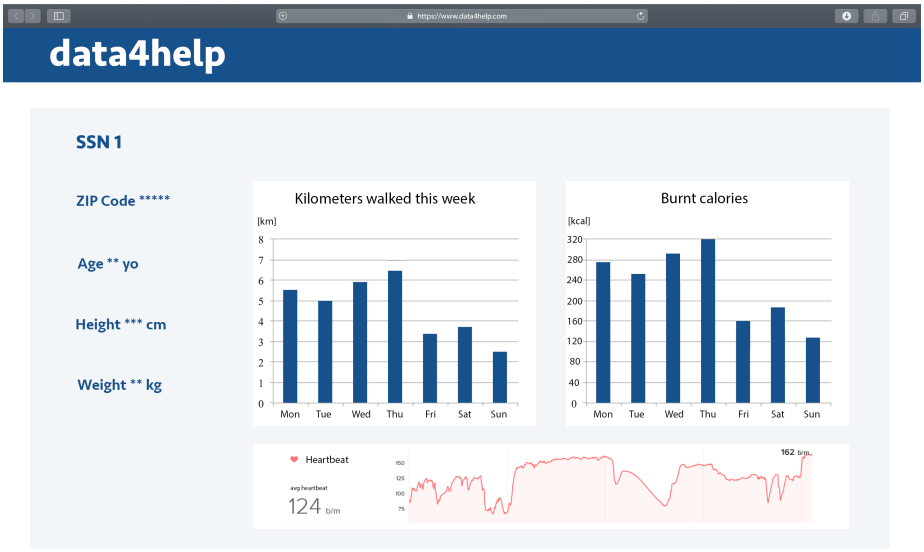
Company login.



Requests.



Single user's data.



Group users' data.

3.1.2 Hardware interfaces

- GPS
- Internet Connection(LTE)
- Photoplethysmography (PPG): integrated in the smart watch to monitor the heart rate of the user.
- Accelerometer: integrated in the smart watch and used to acquire data.

3.1.3 Software interfaces

- Email Service: external interface used to forward the single user request from company to user and the other way around.
- Ambulance Service: external interface with which AutomatedSOS will interact.

3.1.4 Communication interfaces

The network-connected background app uses LTE to send data to TrackMe server. Companies access https://www.data4help.com through the HTTPS protocol. This must be guaranteed because of the sensible data that are present in the web site page.

3.2 Scenarios

3.2.1 Scenario 1

Accenture wants to monitor the health status of its managers and decides to rely on Data4Help, thus signs up on Data4Help website. Accenture's manager decided to try Data4Help only on one manager, Paolo, before starting with all the others. Paolo already has an Apple Watch, he signs up on Data4Help website. In order to monitor Paolo's data, Accenture logs into the website and performs a request for data of a single user by inserting Paolo's SSN, that they know already. Data4Help sends Paolo an email, notifying him that Accenture wants to see his data. Paolo accepts Accenture's request by clicking on the corresponding link. Once that Paolo accepted, Data4Help redirects Accenture on Paolo's profile page, and shows his main personal data: age, city, weight, height, kilometers walked during the current week, calories burnt during the current week and his average heartbeat during the day. In the end Data4Help asks Accenture if is interested in being kept updated with Paolo's new data, Accenture answers yes.

3.2.2 Scenario 2

Allianz want to create a new insurance policy for people living in Milan, whose cost changes based on the lifestyle of the customer. In orded to make some

statistical analisys Allianz relys on Data4Help. Allianz subsrcibes on Data4Help website and performs a request for a group of users, searching for all the users living in Milan with age between 25 and 65 years old. Data4Help performs the research and the outcome number is higher than 1000 users, thus data can be considered as anonymous and are shown to Allianz.

3.2.3 Scenario 3

San Raffaele hospital wants to create a new service in order to monitor its patients that previously had a stroke. They rely on AutomatedSOS. Each patient is given a smart watch and is subscribed to Data4Help website. Happens that Elena, one of the patients involved in this new service, has a stroke. In 3.7 seconds the hospital receives her current position due to an ambulance request.

3.3 Functional requirements

3.3.1 [G1] Visitor can become User after providing credentials.

- [R1] The system must allow a visitor to begin the registration process. During the process the system will ask him to provide credentials.
- [D6] During the registration process the user inserts all the requested data.
- [D7] The user has the physical characteristics that he inserted in the system.

3.3.2 [G2] User can accept or reject the request of access to his data formulated by companies.

- [D1] The user's email is already known by TrackMe.
- [R2] Each time that an accepting or rejecting notification is sent to Data4Help, the system must mark a *Pendent User* as not pendent anymore.
- [R3] Each time that a user changes state from pendent to non-pendent, the corresponding single user request in the company list moves from dandling to accepted.

3.3.3 [G3] If user's parameters are below specified thresholds, an ambulance is called within 5 seconds. [G3.1] Ambulance is required at current user's location.

- [R4] The system must be able to monitor the user heart rate through hardware interfaces that are installed in the smart watch.
- [D4] Smart watch in which the system-to-be is installed have: *GPS, Accelerometer, LTE, Photoplethysmography.*
- [D5] When the system shows the position of a user it means that the user is actually there.

- [D3] At most 5 seconds are necessary to send user location and call an ambulance when parameters are below the threshold.

3.3.4 [G4] Company can sign up as Company to Data4Help and AutomatedSOS.

- [R5] The system must allow a company to begin the registration process. During the process the system will ask to provide credentials.
- [D2] Only real companies can sing up as compaines.
- [R6] At the end of the registration process, the system must provide to the company a password. The tuple $(VAT_number, password)$ is stored in the DB as unique.
- [D8] Companies have a name and a vat number. VAT number is unique.

3.3.5 [G5] Company can be recognized providing a password and vat number.

- [D8] Companies have a name and a vat number. VAT number is unique.
- [R7] The company can log in to the website by providing the combination of a name and a password that match its account.

3.3.6 [G6] Company can formulate a request to see anonymized data of a group of users.

- [R8] The system must allow the company to formulate a group of users request and fill it with drop down menu.
- [R9] The system must allow the company to see its own request and their status(dandling, accepted or rejected).

3.3.7 [G7] Company can formulate a request to see data of a specific user providing his SSN.

- [R10] The system must allow the company to formulate a single user request and fill it with a valid SSN of a user.
- [R9] The system must allow the company to see its own request and their status(dandling, accepted or rejected).

3.3.8 [G8] Company can see anonymized data of a group of users.

- [R11] The system must create a group if and only if more than 1000 user exists.

- [R12] The system must be able to anonymize data. The system must be able to access the DB where all the data are stored, perform researches based on the type of filter present in the group user request and once these information are retrieved fetch all the data that are related to a specific user(meaning all the information through which data can be related to a specific user).
- [R13] The system must allow companies to see the data belonging to a group and for which exist the group request.

3.3.9 [G9] Company can see data of a specific user providing his SSN.

- [D9] Companies know the SSN of the user.
- [R14] The system must allow companies to see data of a single user if the single user request exists and if the user approved that request.

3.3.10 [G10] Company can subscribe to users' new data.

- [R15] The system must allow companies to see users' data as soon as they are produced.

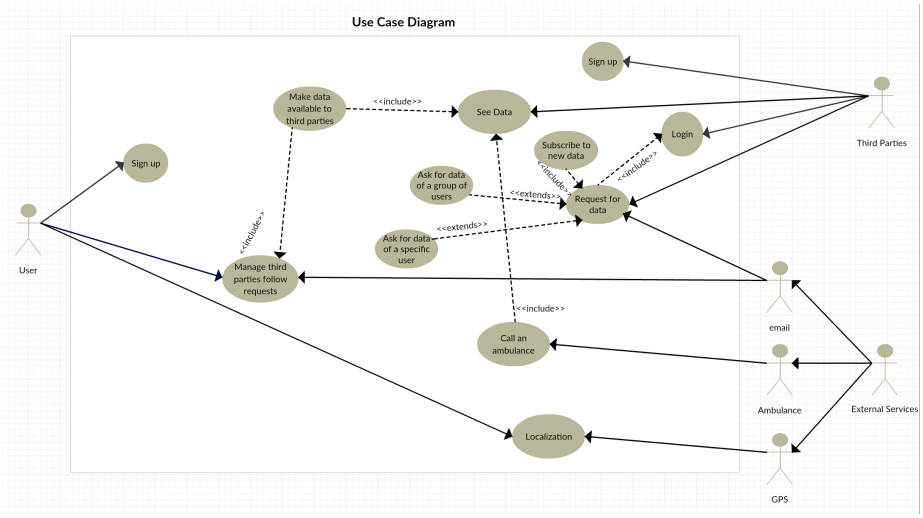
3.3.11 [G11] Data4Help can forward companies' requests to users.

- [D1] The user's email is already known by TrackMe.
- [R9] The system must allow the company to see its own request and their status(dandling, accepted or rejected).
- [R16] The system must allow TrackMe to send an email to a user for which a company issue a single user request.

3.3.12 [G12] A user of Data4Help becomes automatically a user of AutomateSOS if he is older than Age

- [R17] Data4Help allow AutomatedSOS to see data of a user.
- [R18] If a user insert as age a value grater or equal than *Age* the system must automatically enroll this user to AutomatedSOS.
- [R19] If the age inserted by the user + the current data ¿= than *Age* the system must automatically enroll this user to AutomatedSOS.

3.3.13 Use case diagram



Name	Visitor sing up.
Actors	User
Goals	[G1]
Input Conditions	There are no entry conditions.
Event Flow	1. The user on the web home page clicks on the sign in button to start the registration process. 2. The user insert his Google account or Apple account. 3. The user fills all the mandatory fills. 4. The user clicks on the confirm button. 5. The system saves the data.
Output Conditions	The user successfully ends the registration process. From now on he can use Data4Help and AutomatedSOS(if age is greater or equal than *Age*).
Exceptions	1. The user is already registered. 2. The user inserts no valid information in one or more mandatory fills. 3. The user chooses a Google account or Apple account that is already used. All the exceptions are handled notifying the issues to the user and taking back the event flow to point 2.

Name	User accept or reject third party requests.
Actors	User
Goals	[G2]
Input Conditions	The user receives a request of following.
Event Flow	1. The user checks his email and open the following request. 2. The user clicks the link present in the email. 3. The user is redirected in the web page where I can accept or reject the request by clicking on one of the two button.
Output Conditions	The user is notified to have actually accepted/reject the request. The company that issued that request is notified that a request has been accepted/rejected. The request is moved from dandling to accepted/rejected. If the user was *pendent* just for this request, it is marked as *not pendent* anymore.
Exceptions	-

Name	Third part sign up.
Actors	Third part
Goals	[G4]
Input Conditions	There are no entry conditions.
Event Flow	1. The third part on the home page clicks on the sign in button to start the registration process. 2. The third part fills all the mandatory fills. 3. The third part clicks on the confirm button. 4. The system saves the data.
Output Conditions	The third part successfully ends the registration process. From now on it can log into the application, providing his credentials and start using Data4Help and AutomatedSOS.
Exceptions	1. The third part is already registered. 2. The third part inserts no valid information in one or more mandatory fills. 3. The third part chooses a vat number that has already been taken. All the exceptions are handled notifying the issues to the third part and taking back the event flow to point 2.

Name	Third part login.
Actors	Third part
Goals	[G5]
Input Conditions	There third part is already in the homepage.
Event Flow	1. The third part inserts its credentials in. 2. The third part clicks on the login button in order to access. 3. The system redirects the third part to his profile.
Output Conditions	The third part his successfully redirects to his profile.
Exceptions	1. The third part insert a not vat number. 2. The third part insert a not valid password. All the exceptions are handled notifying the issues to the user and taking back the event flow to point 1.

Name	Third part formulate a request to access anonymized data of a group of users.
Actors	Third part
Goals	[G6]
Input Conditions	The third part is already logged in into the system.
Event Flow	1. The third part clicks on the *group request*. 2. The request is filled using drop-down menu. Each drop-down menu is linked to a type of filter. By selecting a filter the company is able to better specify the composition of the group it is interested in. 3. The request is sent to TrackMe's servers by clicking on *send* button.
Output Conditions	The request is sent to TrackMe and a message for the correct sending of the request is presented to the company.
Exceptions	-

Name	Third part formulate a request to access data of a specific user through is SSN.
Actors	Third part
Goals	[G7]
Input Conditions	The third part is already logged in into the system.
Event Flow	1. The third part clicks on the *single user request*. 2. The request is filled with the user's SSN. 3. The request is sent to TrackMe by clicking on *send* button.
Output Conditions	The request is sent to TrackMe's servers and a message for the correct sending of the request is presented to the company.
Exceptions	-

Name	Third part can access anonymized data of a group of users.
Actors	Third part
Goals	[G8]
Input Conditions	There third part is already logged into the system and a group request for this group exists.
Event Flow	1. The third part accesses the approved group requests section. 2. The third part selects an approved group request. 3. The system redirects the third part to the request result.
Output Conditions	The data requested by the third part are shown.
Exceptions	1. The data requested aren't anonymous since the outcome of the request concerns less than 1000 users. All the exceptions are handled notifying the issues to the third part and taking back the event flow to point 1.

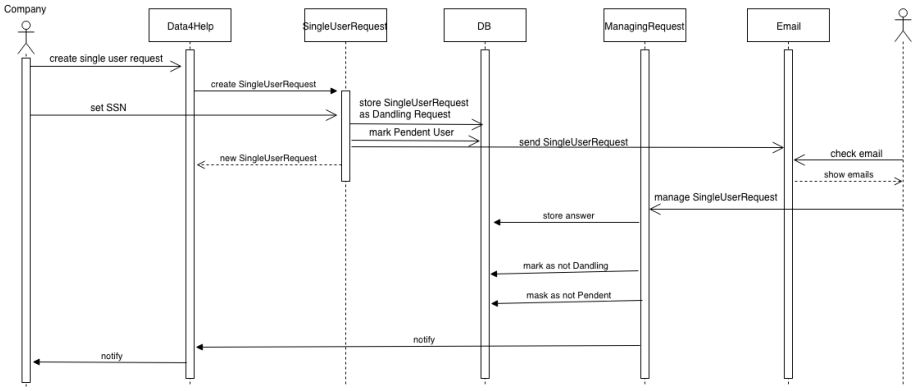
Name	Third part can access data of a specific user through his SSN.
Actors	Third part
Goals	[G9]
Input Conditions	There third part is already logged into the system.
Event Flow	1. The third part accesses the approved single user requests section. 2. The third part selects an approved single user request. 3. The system redirects the third part to the request result.
Output Conditions	The data requested by the third part are shown.
Exceptions	1. The user didn't approve the request. All the exceptions are handled notifying the issues to the third part and taking back the event flow to point 1.

Name	Data4Help can forward third parties requests to users.
Actors	Data4Help
Goals	[G11]
Input Conditions	Third part has requested data of a single user.
Event Flow	1. Data4Help forwards the request of being observed by a specific company to the user using the external email service. 2. Data4Help marks the user as *pendent user*.
Output Conditions	The request of being followed by a company is sent to the user.
Exceptions	-

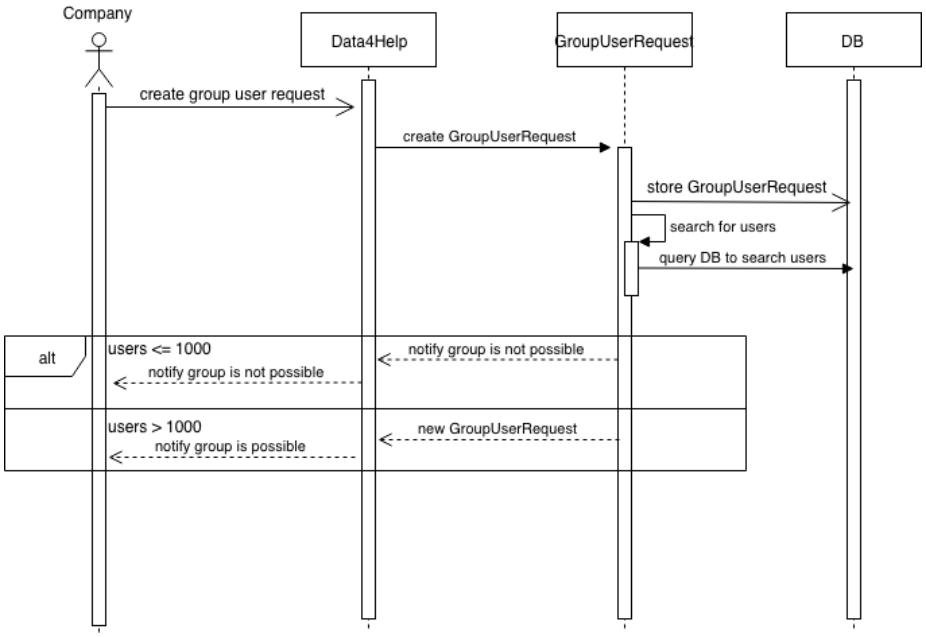
Name	A User of Data4Help becames automatically a user of AutomateSOS.
Actors	Data4Help
Goals	[G12]
Input Conditions	User sign up.
Event Flow	1. Data4Help checks the age of the user + current date. 2. Data4Help subscribes the user to AutomatedSOS.
Output Conditions	User is notified than he has been subscribed to AutomateSOS, through an email.
Exceptions	1. User's age + current date is less than *Age*. The user is not enrolled in AutomatedSOS.

3.3.14 Sequence diagram

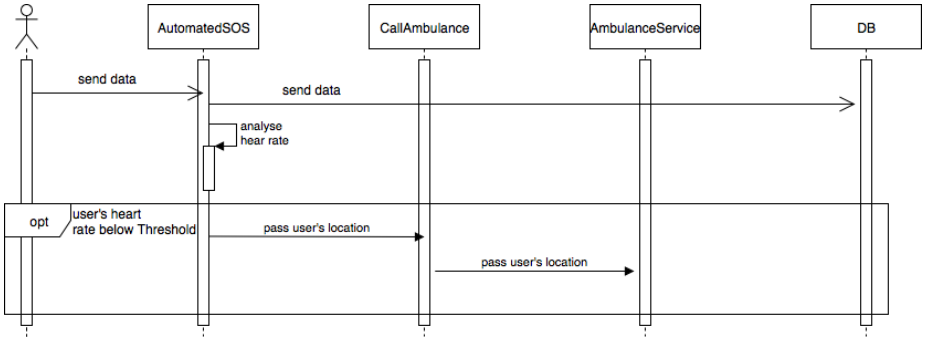
Company formulate single user request.



Company formulate group user request.



AutomatedSOS calls an ambulance.



3.4 Performance requirements

When hear rate of user is below the *Threshold*, AutomatedSOS calls an ambulance to the current user's position within 5 seconds. This is the most important *non-functional* constraint of the all system-to-be. Users will rely on the application in order to be sure, during their everyday activities, that they will be safe in case of dangerous heart dysfunction. During the test phase, this requirement must be tested with the high priority.

3.5 Design constraints

3.5.1 Standard compliance

- The app requires permission to access user current position.
- Personal data processing observes GDPR
- Users' data are stored in TrackMe's servers.
- Website access is guaranteed to be safe by using HTTPS

3.5.2 Hardware limitation

In order to work properly the system-to-be needs to be installed in a smart watch that has at least these hardware:

- a wearable device with watchOS or wearOS
- GPS
- Photoplethysmography (PPG)
- Accelerometer
- LTE

3.5.3 Other constraints

- Regulatory policies
- Usage is possible only for users with a Google or iOS account
- The smart watch must be logged in a Google or iOS account

3.6 Software system attributes

3.6.1 Reliability

The application must be available 24/7. Since AutomatedSOS is built on top of Data4Help, no concession is tolerated. Obviously it is not feasible to reach a 100% of availability, but a 99.999% of availability is required. This constraints

will result in a high cost of architecture design and will automatically bring to replication of core systems of the system-to-be. If an availability of 99.999% will result not feasible this must be indicated to the management and clearly stated in the legal contract of this product.

3.6.2 Security

Due to the sensitive nature of these data, security is an important issue. Web access will use HTTPS protocol in order to limit the probability of malicious attacks. Data will be store in DB using all the required techniques that helps to improve security of data. In order to accomplish the industry's standards, *ISO-14971* should be consulted.

3.6.3 Maintainability

Code is structured in such a way that possible future modification and functionality addition will be easy to implement, e.g. Track4Run, a service to track athletes participating in a run.

3.6.4 Compatibility

The website is compatible with Google Chrome, Safari and Firefox.
The background app is compatible only with smart watch running on Watch OS and Wear OS.

4 Formal Analysis Using Alloy

4.1 Alloy model

```
open util/integer
open util/boolean

sig User {
age: one Int,
height: one Int,
weight: one Int,
ssn: one String,
residence: one Int, –This is the user's CAP
datas: Data,
requests: set Request,
} {
age ≥ 0 and
height ≥ 0 and
weight ≥ 0
}

sig Group {
users: set User
} {
users ≥ 1001
}

sig Data {
heartrate: one Int,
calories: one Int,
km: one Int,
locations: Location
} {
heartrate ≥ 0 and
calories ≥ 0 and
km ≥ 0
}

sig Location{
cordX: one Int,
cordY: one Int
} {
cordX ≥ 0 and
cordY ≥ 0
}
```

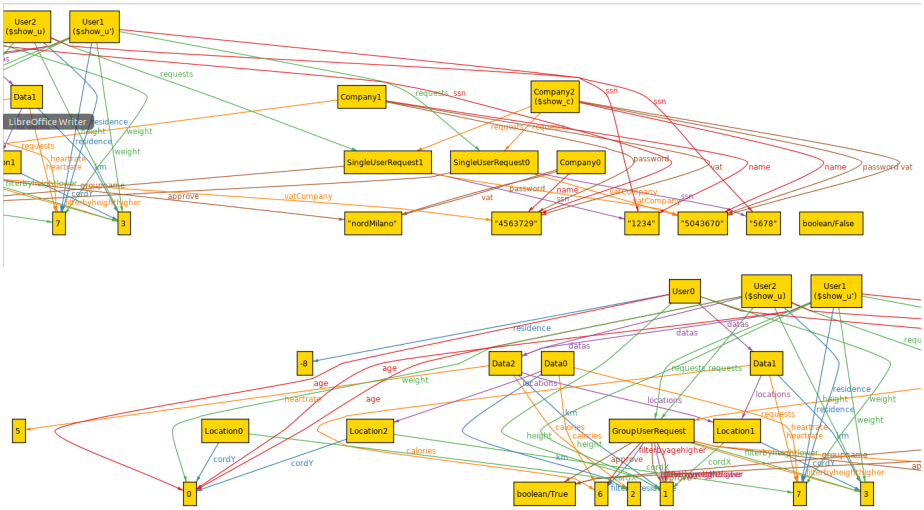
```
sig Company{
vat: one String,
name: one String,
password: one String,
requests: set Request
}

abstract sig Request{
vatCompany: one String,
approve: one Bool
}

sig SingleUserRequest extends Request{
ssn: one String
}

sig GroupUserRequest extends Request{
groupname: one String,
filterbyresidence: one Int,
filterbyagelower: one Int,
filterbyagehigher: one Int,
filterbyheightlower: one Int,
filterbyheighthigher: one Int,
filterbyweightlower: one Int,
filterbyweighthigher: one Int
}{
filterbyresidence ≥ 0 and
filterbyagelower ≥ 0 and
filterbyagehigher ≥ 0 and
filterbyheightlower ≥ 0 and
filterbyheighthigher ≥ 0 and
filterbyweightlower ≥ 0 and
filterbyweighthigher ≥ 0 and
(filterbyagelower ≥ filterbyagehigher) and
(filterbyheightlower ≥ filterbyheighthigher) and
(filterbyweightlower ≥ filterbyweighthigher)
}
```

– a signle request cannot exists on its own, or belong to more than one company

```
fact SingleRequestCompanyconnection {
all r:SingleUserRequest | all c:Company | (r in c.requests implies r.vatCompany
= c.vat) and (r.vatCompany = c.vat implies r in c.requests)
}

fact SingleRequestUserconnection {
```

```
all r:SingleUserRequest | all u:User | (r in u.requests implies u.ssn = r.ssn) and
(r.ssn = u.ssn implies r in u.requests)
}

– a group request cannot exists on its own, or belong to more than one company

fact GroupRequestCompanyconnection {
all r:GroupUserRequest | all c:Company | (r in c.requests implies r.vatCompany
= c.vat) and (r.vatCompany = c.vat implies r in c.requests)
}

fact GroupRequestUserconnection {
all g:GroupUserRequest | all u:User | (((g.filterbyagelower ≤ u.age and u.age ≤
g.filterbyagehigher) and
(g.filterbyheightlower ≤ u.height and u.height ≤ g.filterbyheighthigher) and
(g.filterbyweightlower ≤ u.weight and u.weight ≤ g.filterbyweighthigher) and
(g.filterbyresidence=u.residence))implies(g in u.requests))
}

– vat is unique

fact vatUnique{
no disjoint c1,c2 : Company | c1.vat = c2.vat
}

– ssn is unique

fact ssnUnique{
no disjoint u1,u2 : User | u1.ssn = u2.ssn
}

pred show (){
(some u:User | u.requests ≥ 2 and u.ssn = "1234") and
(some u:User | u.requests ≥ 2 and u.ssn = "5678") and
(some c:Company | c.requests ≥ 1 and c.vat ="5043670") and
(one s:SingleUserRequest | s.ssn = "1234" and s.vatCompany = "5043670") and
(one s:SingleUserRequest | s.ssn = "5678" and s.vatCompany = "5043670") and
(one g:GroupUserRequest | g.vatCompany = "4563729" and g.groupname =
"nordMilano")
}
run show
```

4.2 World generated



4.3 Alloy results

As we can see the model was able to find a world that satisfies all the facts.
The main point we wanted to highlight was the fact that our requests must be issued by a company and must be related to some user.
With our Alloy model we did that.

5 Effort Spent

5.1 Dalle Rive Fabio

Task Description	Hours
Purpose, Scope	2
Product Perspective, Product Functions	2
User Characteristics, Domain Assumptions	4
External Interfaces Requirements	5
Scenarios	1
Functional Requirements	10
Performance Requirements, Design Constraints, Software System Requirements	1
Formal Analysis Using Alloy	8

5.2 Di Giacomantonio Marco

Task Description	Hours
Purpose, Scope	2
Product Perspective, Product Functions	2
User Characteristics, Domain Assumptions	4
External Interfaces Requirements	5
Scenarios	1
Functional Requirements	10
Performance Requirements, Design Constraints, Software System Requirements	1
Formal Analysis Using Alloy	8

6 Resources

6.1 Used Tools

- Texmaker 5.0.3
- Adobe Photoshop
- Alloy Analyzer 4.2
- https://sequencediagram.org
- http://creately.com
- https://www.draw.io

6.2 References

- Specification document "Mandatory Project Assignment AY 2018-2019"
- Alloy documentation - http://alloy.mit.edu/alloy/documentation.html
- Software Engineering 2 Course Slides

7 History

- 1.1: Edit UML in section 2.1
- 1.2: Edit Sequence diagram in order to have coherence with DD
- 1.3: Delete D3,D5,D11,D12,D13,D14,D17 domain assumption, and change the document to deal with this change. Improve clarity of some paragraphs.