Luca Grella, Daniele Lunghi

Software Engineering 2 project

RASD

TrackMe

[1. INTRODUCTION 2](#_Toc529742702)

[1A. Purpose 2](#_Toc529742703)

[1B. Scope 2](#_Toc529742704)

[1C. Definitions & Acronymous 3](#_Toc529742705)

[1D. Revision history 3](#_Toc529742706)

[1E. Reference Documents 3](#_Toc529742707)

[1F. Document Structure 3](#_Toc529742708)

[2. OVERALL DESCRIPTION 4](#_Toc529742709)

[2A. Product perspective 4](#_Toc529742710)

[Class diagram 4](#_Toc529742711)

[Statechart diagram 5](#_Toc529742712)

[2B. Product functions 6](#_Toc529742713)

[Goals 6](#_Toc529742714)

[2C. User characteristics 6](#_Toc529742715)

[2D. Assumptions, dependencies and constraints 6](#_Toc529742716)

[Domain Assumptions 6](#_Toc529742717)

[Functional requirements 6](#_Toc529742718)

[3. SPECIFIC REQUIREMENTS 8](#_Toc529742719)

[3A. External interface requirements 8](#_Toc529742720)

[3A1. User Interfaces 8](#_Toc529742721)

[3A2. Hardware Interfaces 12](#_Toc529742722)

[3A3. Software Interfaces 12](#_Toc529742723)

[3A4. Communication Interfaces 13](#_Toc529742724)

[3B. Functional Requirements 14](#_Toc529742725)

[Use case descriptions 14](#_Toc529742726)

[Use Case Diagram 26](#_Toc529742727)

[3C. SCENARIOS 27](#_Toc529742728)

[Scenario 1 27](#_Toc529742729)

[Scenario 2 27](#_Toc529742730)

[Scenario 3 27](#_Toc529742731)

[Scenario 4 27](#_Toc529742732)

[Sequence Diagrams 28](#_Toc529742733)

[3D. Performance Requirements 30](#_Toc529742734)

[3E. Design Constraints 30](#_Toc529742735)

[3F. Software System Attributes 30](#_Toc529742736)

[3F1. Reliability 30](#_Toc529742737)

[3F2. Availability 30](#_Toc529742738)

[3F3. Security 30](#_Toc529742739)

[3F4. Mantenailability 30](#_Toc529742740)

[3F5. Portability 30](#_Toc529742741)

[4. FORMAL ANALYSIS USING ALLOY 31](#_Toc529742742)

[5. EFFORT SPENT 36](#_Toc529742743)

[Luca Grella 36](#_Toc529742744)

[Daniele Lunghi 36](#_Toc529742745)

[6. REFERENCES 36](#_Toc529742746)

# 1. INTRODUCTION

## 1A. Purpose

This document is meant to provide a detailed and complete description of Data4Help and AutomatedSOS systems.  
It provides a complete description of its purpose, goals, and of the requirements and the assumptions though which they will be achieved.  
This document has been written in order to be used by the users, by the the system and requirements analysts, the project managers, software developers, testers and all the others who’ll take part in the realization of this project.

## 1B. Scope

Service Description:

Data4Help is a software-based service which goal is to save the location and the health state to the users and to allow third parties to monitor it.

The system supports the registration of two kind of users: normal users (NU), who have access to their own data and agree by registering that TrackMe acquires them too, and third parties (TPs), who pay a subscription in order to keep their account open and have access to some special features:  
1) Access the data of some specific individuals by sending them a request.   
The user can decide whether they want to accept the request from the TP, and if they do it, the TP will have the access to their data.  
Naturally the users can remove the TPs from the list of people with access to their data any time

2) Access to anonymized data of groups of individuals filtered by some characteristics.

AutomatedSOS is a service built on top of Data4Help, designed in order to help users with severe health problems.

AutomatedSOS monitors the health status of the subscribed customers and, when such parameters are below thresholds, sends to the customer’s location an ambulance, guaranteeing a reaction time of less than 5 seconds from when the parameters are below the thresholds.

Payment Management:

**Data4Help**

As already mentioned, TPs have to pay a fee in order to maintain their account open.  
This fee has to be payed every month and is handled by some external Payment System, which will guarantee a high level of security and reliability.

**AutomatedSOS**

The access to AutomatedSOS is completely free, in fact it’s a feature added in order to attract new customers, whose data are then sold to the TPs.

## 1C. Definitions & Acronyms

SOS: An SOS is a signal which indicates to other people that you are in danger and need help quickly.

RASD: Requirements Analysis and Specification Document

GPS: Global Positioning System

SSN: Social Security Number

TPU: Third Part User

API: Application Programming Interface

NU: Normal User

## 1D. Revision history

V1.0

V2.0 During the making of the DD, we analysed in a more detailed way some parts, that brought us to the decision to modify minor parts of the RASD.  
First of all, we had to rename one of the Sequence Diagrams: we substituted the title “Access the data of specific users” with the more accurate “Request Handling”.  
We also eliminated the possibility for the Normal Users to block particular Third Part Users for two reasons: it wasn’t necessary and there was a risk for the NUs to accidentally block a TPU.

Naturally because of these modifications we had to change both the Class Diagram and the Alloy World in order to maintain the coherence.

We counted the hours spent during this operation as spent on the DD, because they were mainly caused by the necessity to integrate the two parts of the project.

## 1E. Reference Documents

- “Mandatory Project Assignment AY 2018-2019.pdf”.

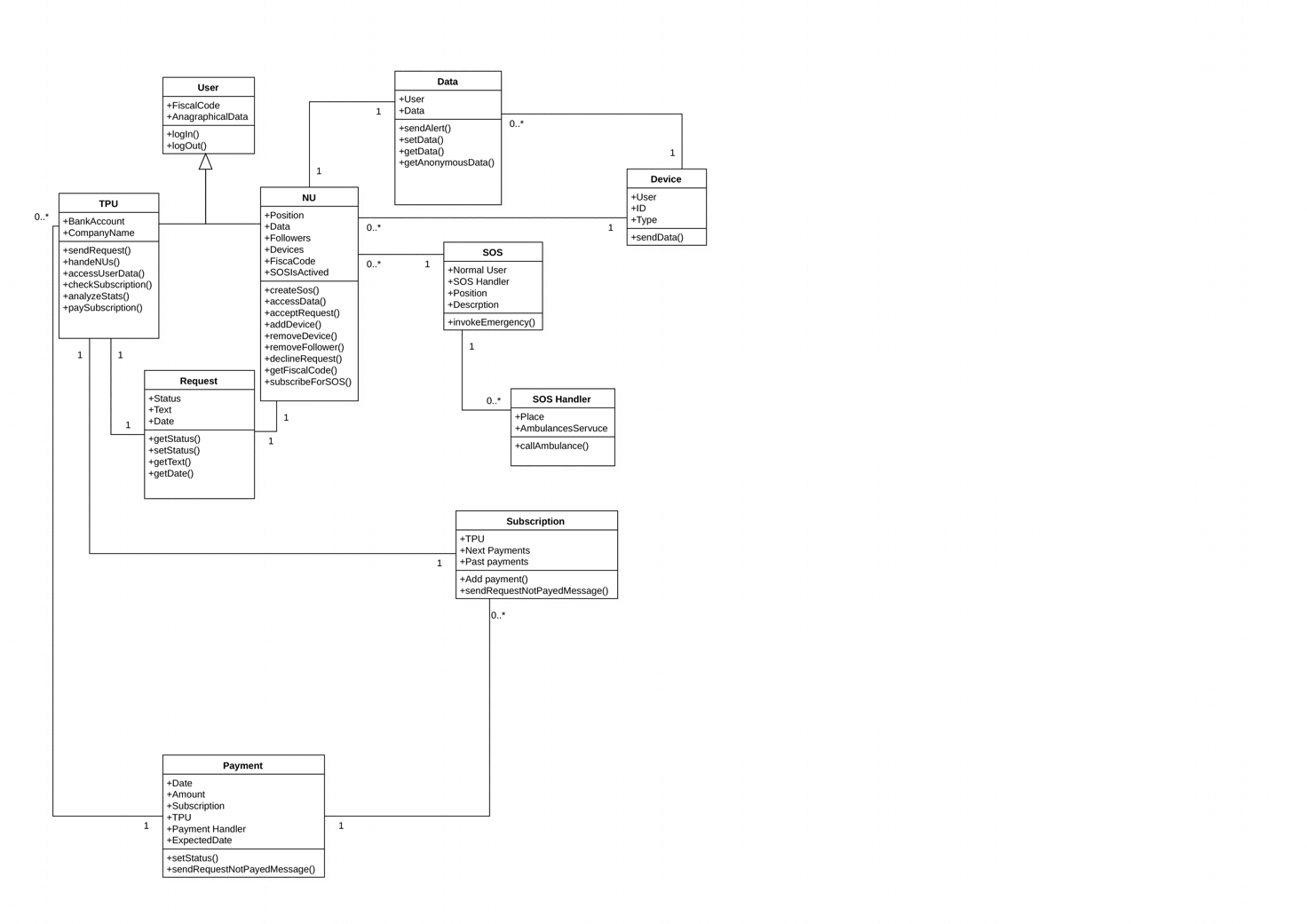
## 1F. Document Structure

This document is divided into 6 macro-chapters. The first part contains the introduction, the objectives of the project and the indications to better read the document. The second part contains the class diagrams and state diagrams, the most important requirements and domain assumptions. The third part contains specific requirements, such as mockups, UML modeling (use case diagrams, sequence diagrams) and performance requirements. The fourth part contains the Alloy model and the formal analysis. The fifth part includes the division of working hours among group members. Finally, the last part cites the references used for the realization of the project.

# 2. OVERALL DESCRIPTION

## 2A. Product perspective

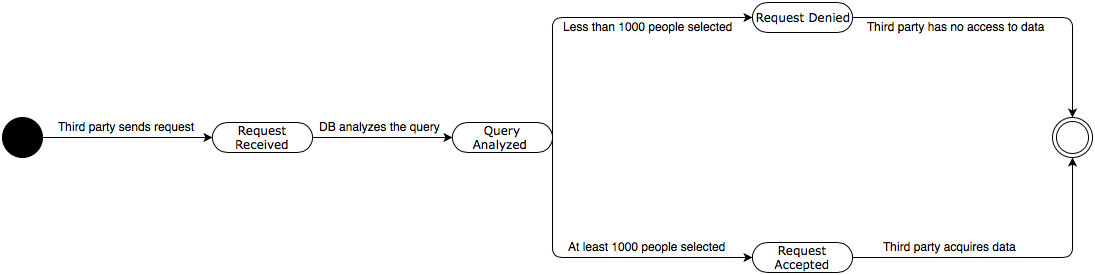
### Class diagram



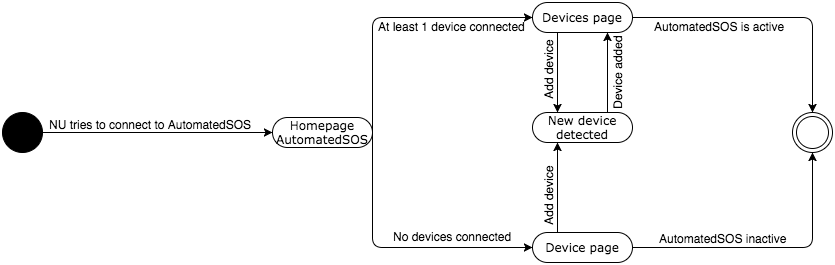
### Statechart diagram



This state diagram represents a TPU sending a request to a NU.



This state diagram represents a TPU searching for anonymous data for statistical analysis.



This state diagram represents a NU who want to activate his AutomatedSOS system.

## 2B. Product functions

### Goals

G1. NU’s account correct handling.

G2. TPU’s account correct handling.

G3. Users data registration.

G4. Allow TPU to access the data of NU, upon acceptance.

G5. Allow TPU to access anonymous data of a group of at least one thousand people.

G6. Notify ill if health values ​​are below threshold for more than 5 seconds.

## 2C. User characteristics

NU: person that is successfully registered to TrackMe, whose data are stored on the database, can access his own data.

TPU: person that is successfully registered to TrackMe, he can access the data of anonymous groups of people or of a single followed NU.

Administrator: An employee of TrackMe that is maintaining and updating the system. The administrator does not have to register, since he is added during system’s installation process.

Bank Service: Takes care of the payment from the party third user to the company.

## 2D. Assumptions, dependencies and constraints

### Domain Assumptions

D1.A secure channel is available for transporting data.

D2. TPU uses an external service payment

D3. The smartwatch is connected to the Internet

D4. The national sanitary system owns and provides correct data

D5. The database is handled in a secure way, so that personal data are not lost or stolen

D6. The external banking service is safe and reliable

D7. The NU owns a smartwatch or a similar device

D8. Handling personal data and their usage is compatible with local laws

D9. The AutomatedSOS service is always available and the call is answered on time

D10. The identifier of the NU is unique for the local sanitary service

D11. It is possible to set a threshold value for variables that matter the most with respect to the general data, such as the age, the body weight

### Functional requirements

F1. The Ambulance receives a notification that contains the geographical location of the NU that is in danger, that allows it to arrive there

F2. The smartwatch is connected to the smartphone and it can interpret data that are sent and process them

F3. The smartphone can be connected to the smartwatch

F4. The NU is identified with a username and a password

F5. It is possible to restore a password

F6. Personal data are eliminated once the user decide to erase the account

F7. Constant update of data from SSN

F8. User can access his own data

F9. TPUs are allowed to register to the application

F10. Identification possible with username + password

F11. Possibility to restore a password

F12. Block of the account if there is no payment

F13. It is possible for the users to sign up

F14. The smartphone receives data from a smartwatch and forwards them to the server

F15. All the data received are saved and associated to the time when they are registered

F16. Allow TPU to access NU's data, upon acceptance

F17. Possibility to send the request to a user of whom it knows the id to receive his data

F18. Possibility for the user to decide if he wants to accept the request

F19. If the user accept, the third part continue to receive the user data (follow)

F20. At any time, the NU is allowed to remove a follower

F21. TPUs are allowed to access data of a group, but it must contain at least 1000 people

F22. A group of anonym users can be selected through a query, that allows to see data without personal information about the NU, but the result is shown only if more than 1000 people are in the group

F23. If data fall outside the thresholds for more than 5 seconds, the illness is notified in less than 5 seconds

F24. The Administrator can see all the data and is allowed to modify the tables

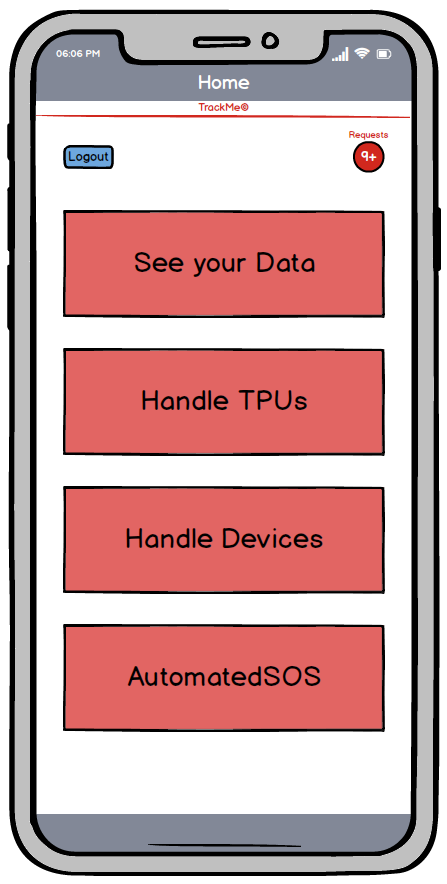
# 3. SPECIFIC REQUIREMENTS

## 3A. External interface requirements

### 3A1. User Interfaces

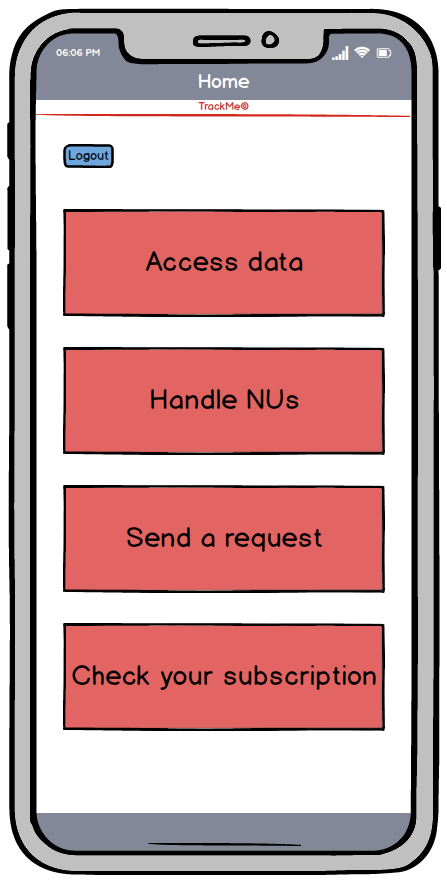
The following mockups represent a basic idea of what the mobile app will look like in the first release.

**Normal User Home Page**



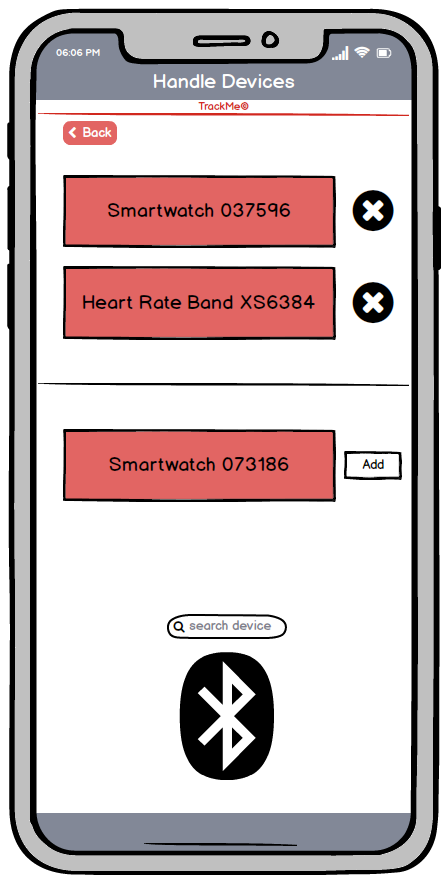
This is the main screen that can be viewed by a normal registered user. It can be reached after login (it can be done if you have an account created with the appropriate registration form reachable at the start of the app). From the following page you can log out by clicking on the appropriate logout button, you can reach the handle requests page, clicking on the red button (notifying pending requests), or you can reach the page of your data acquired by Data4Help, the TPUs or devices handle pages and the activation page of AutomatedSOS.

**Third Party User Home Page**



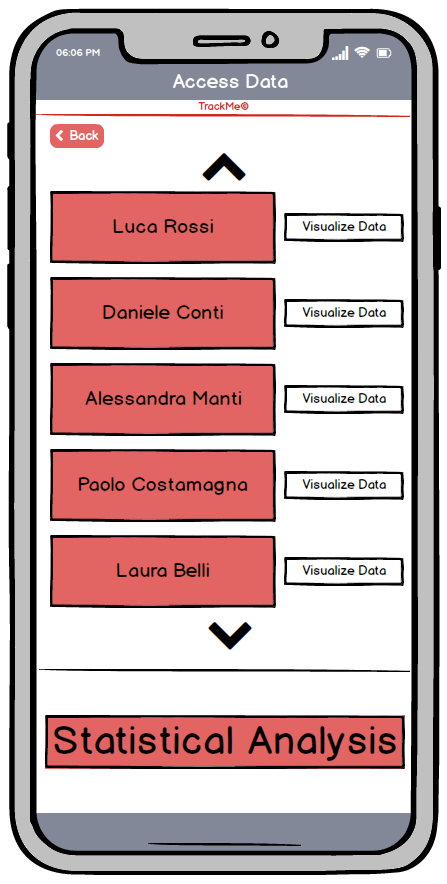
This is the main screen that can be viewed by a third part user. It can be reached after login (it can be done if you have an account created with the appropriate registration form reachable at the start of the app). From the following page you can log out by clicking on the appropriate logout button or you can access the page of users followed to view their data, or to collect a statistical sample (receiving anonymous data depending on the research performed). You can manage the users you follow, send a request or check the status of your subscription.

**Handle Devices Normal User Screen**



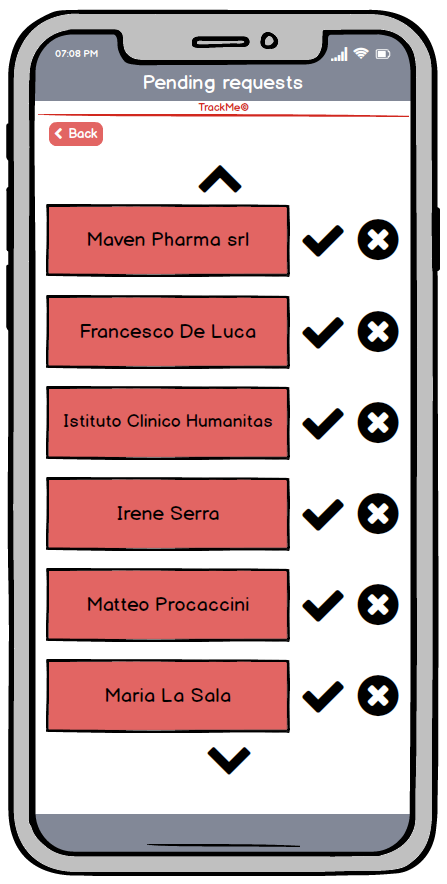
This screen is reachable by a NU who wants to handle his own devices, he can remove the already connected devices by clicking on the "X" next to the device or he can search for a device nearby (via Bluetooth) or add (with the proper button) an already detected device. NU can reach the previous page by clicking on “back” button on the top.

**Access Data Third Party User Screen**



This screen is reachable by a TPU who wants to access followed NU’s data. TPU can visualize the data of each followed NU by clicking on "Visualize Data". TPU can reach a page in which he can search for anonymous data for statistical analysis, by clicking on the "Statistical Analysis" button. TPU can reach the previous page by clicking on “back” button on the top.

**Pending Request Screen**



This screen is reachable by a NU who wants to accept (clicking on “V”) or refuse (clicking on “X”) TPU’s requests. NU can reach the previous page by clicking on “back” button on the top.

### 3A2. Hardware Interfaces

Specific hardware interfaces are not required for the implementation and operation of this

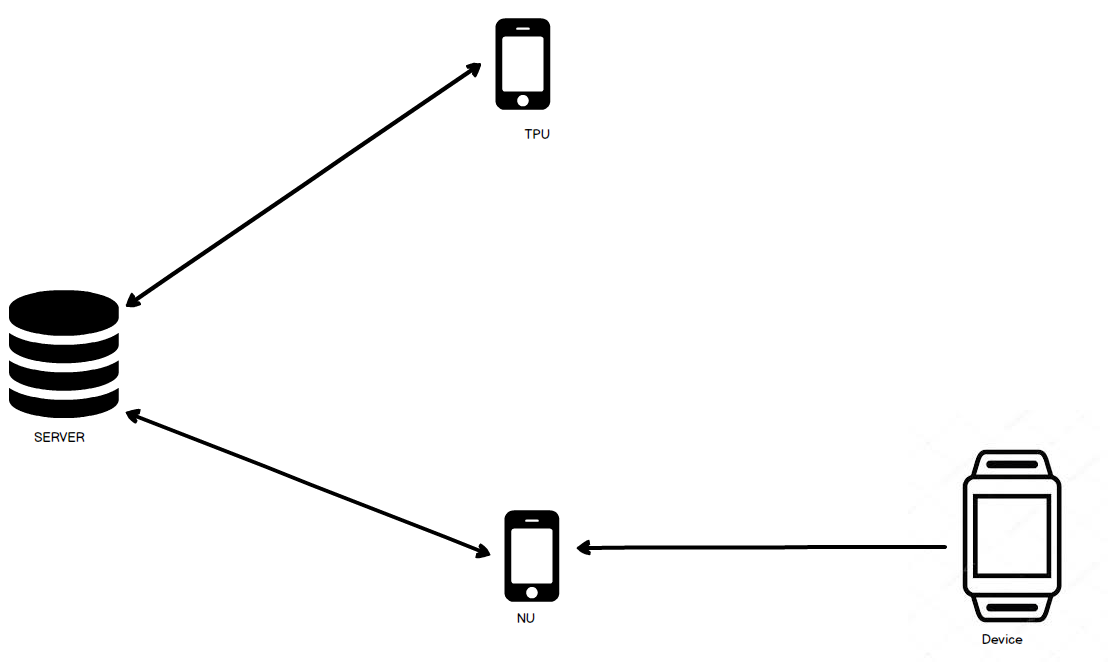
system.

### 3A3. Software Interfaces

The system should be able to communicate with the main external APIs with which it interacts frequently, especially the ones of the devices connected to the system,

Moreover, our system should support mobile devices with iOS 10 or above, Android 5.0 or above and Windows Phone.

### 3A4. Communication Interfaces



## 3B. Functional Requirements

### Use case descriptions

|  |  |
| --- | --- |
| Use Case | Sign Up |
| GOALS | G1, G2, G3 |
| Actor | 1. NU  2. TPU |
| Entry conditions | The user has installed the application on his device. |
| Events flow | 1. Click on “Sign up” button.  2. Fill all the mandatory fields and provide the necessary information.  3. Click on “Confirm” button.  4. The system saves the data. |
| Exit conditions | The user has successfully registered and now he’s able to use the application. |
| Exceptions | 1. The user is already signed up.  2. The user didn’t fill all of the mandatory fields with valid data.  3. The username is already taken.  4. The e-mail is already registered.  5. All the exceptions are handled by notifying the user and taking him back  to the sign-up activity. |

|  |  |
| --- | --- |
| Use Case | Log in |
| GOALS | G1, G2 |
| Actor | 1. NU  2. TPU |
| Entry conditions | 1. The user has an application installed on his device.  2. The user has an existing account with a password and data. |
| Events flow | 1. The user opens the application on his device.  2. He enters his credentials in the “Username” and “Password” fields of  the home page of “TrackMe”.  3. The user clicks on the “Log in” button.  4. The user is successfully logged in his “TrackMe” and the  system automatically redirects him to the home page |
| Exit conditions | The user is successfully redirected to home page |
| Exceptions | 1. The user enters invalid Username.  2. The user enters invalid Password.  3. All the exceptions are handled by notifying the user and taking  him back to the login activity. |

|  |  |
| --- | --- |
| Use Case | Access personal data |
| GOALS | G1, G3 |
| Actor | NU |
| Entry conditions | The NU is logged in. |
| Events flow | 1. The user clicks “See your data”  2. The user sees the most recent data  3. The user is provided with an option to check his previous data pressing the “See older data” button, the user can choose which data he wants to see.  4. When he is done, he presses the “go back” to return the home page. |
| Exit conditions | The user is able to access his own data |
| Exceptions | 1. NoDataException  2. NoConnectionException |

|  |  |
| --- | --- |
| Use Case | Confirm Request |
| GOALS | G1 |
| Actor | NU |
| Entry conditions | NU successfully logged in and he received a request from a TPU to access his data. |
| Events flow | 1. NU clicks on “Requests” red button.  2. He checks the pending requests.  3. He provided with the option of accepting the request and let the TPU access his data pressing the “V” button, or denying the request pressing the “X” button. |
| Exit conditions | Request is now handled, and the third part user receives the answer. |
| Exceptions | 1. RequestNotFoundException  2. GenericException |

|  |  |
| --- | --- |
| Use Case | Add a device |
| GOALS | G1 |
| Actor | NU |
| Entry conditions | NU successfully logged in and he has to add a compatible device already communicating with his mobile phone. |
| Events flow | 1. NU clicks on “Handle devices” button.  2. NU provided with a list of compatible devices.  3. User presses the “Add” button on the right of the device to connect. |
| Exit conditions | The selected device is now sending data to the sever |
| Exceptions | 1)TimeOutException  2)DeviceNotCompatibleException  3)GenericException |

|  |  |
| --- | --- |
| Use Case | Remove Device |
| GOALS | G1 |
| Actor | NU |
| Entry conditions | 1. NU successfully logged in  2. NU has a device already “connected” which he wants to remove |
| Events flow | 1. NU clicks on “Handle devices” button.  2. NU provided with a list of connected devices.  3. NU presses the “X” button on the right side of the device he wants to remove |
| Exit conditions | The device doesn’t send data to the server anymore. |
| Exceptions | 1. DeviceNotFoundException |

|  |  |
| --- | --- |
| Use Case | Remove Accepted TPU |
| GOALS | G1 |
| Actor | NU |
| Entry conditions | 1. NU successfully logged in.  2. NU has already accepted a request from a TPU. |
| Events flow | 1. NU clicks on “Handle TPUs” button.  2. NU provided with a list of TPUs allowed to see your data.  3. NU presses the “X” button on the right side of the TPU he wants to remove. |
| Exit conditions | TrackMe doesn’t send NU’s data to the removed TPU anymore. |
| Exceptions | 1. GenericException |

|  |  |
| --- | --- |
| Use Case | Access NU’s data |
| GOALS | G2, G3 |
| Actor | TPU |
| Entry conditions | 1. TPU has successfully logged in.  2. TPU has sent a request to a NU.  3. The request has been accepted. |
| Events flow | 1. TPU presses the “Access data” button.  2. TPU is provided with a list of the followed NUs.  3. TPU chooses a NUs.  4. TPU presses the “Visualize data” button on the right side of the chosen NU’s name. |
| Exit conditions | TPU is provided with user’s data. |
| Exceptions | 1. NoData exception  2. NoConnection exception  3. GenericException |

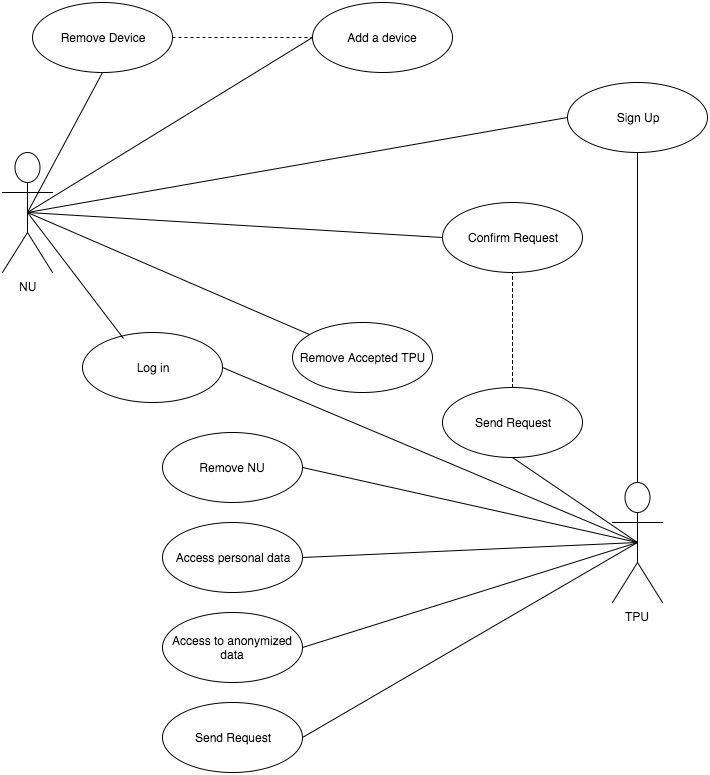
|  |  |
| --- | --- |
| Use Case | Remove NU |
| GOALS | G2 |
| Actor | TPU |
| Entry conditions | 1. TPU has successfully logged in  2. TPU has sent a request to a user  3. The request has been accepted |
| Events flow | 1. TPU presses the “Handle NUs” button.  2. TPU is provided with a list of the followed users.  3. TPU presses the “X” button on the right side of the NU he wants to remove. |
| Exit conditions | TPU hasn’t access to user’s data anymore. |
| Exceptions | 1. NoDataException  2. NoConnectionException  3. GenericException |

|  |  |
| --- | --- |
| Use Case | Access to anonymized data |
| Goals | G5 |
| Actor | TPU |
| Entry conditions | 1. TPU has successfully logged in.  2. The request satisfies the constraints. |
| Events flow | 1. TPU presses the “Access data” button.  2. TPU presses the “Statistical Analysis” button.  3. TPU fills the research filter fields.  4. TPU clicks “Search” button. |
| Exit conditions | TPU has access to anonymized data. |
| Exceptions | 1. NoDataException  2. GenericException |

|  |  |
| --- | --- |
| Use Case | Log in for AutomatedSOS |
| Goal | G6 |
| Actor | NU |
| Entry Condition | 1. NU has successfully logged in.  2. At least one device is connected. |
| Event flow | 1. NU clicks “AutomatedSOS” button.  2. NU clicks on “Switch on” button. |
| Exit condition | AutomatedSOS system is on and it monitors NU’s vital activities |
| Exception | 1. DeviceNotFoundException  2. 2. NoConnectionException |

|  |  |
| --- | --- |
| Use Case | Send Request |
| Goal | G4 |
| Actor | TPU |
| Entry Condition | 1. TPU has successfully logged in.  2. TPU knows SSN or the fiscal code of the NU |
| Event flow | 1 TPU sends a request clicking the “Send a request” button  2 TPU can add a text attached to the request  3 TPU presses the “Send” button |
| Exit condition | NU receives the request from the TPU |
| Exception | 1. UserNotFoundException 2. GenericException |

### Use Case Diagram



## 3C. SCENARIOS

### Scenario 1

Edgardo is a doctor and he is complaining that sometimes he doesn’t have enough information about his patients to make a correct diagnosis.

His colleague Mefistofele asks him “Did you try TrackMe? It’s an online application wich allows you to receive all the data from your patients who have a smartwatch or a similar device”  
“I will try” answers Edgardo.  
Three months later he is very happy about the results.

### Scenario 2

Aldo wants to open a surgery in Milan, but he doesn’t know which zone would be more suitable.  
He needs to collect some data in order to take this decision.  
He searches on the Internet and he finds out about TrackMe application, and he decides that it’s perfect for him.4

He creates an account and searches the data of Milan inhabitants, grouped by the zone.

The result is clear, Loreto is the best choice.

### Scenario 3

Martina is worried for her mother, she is very old, and she might have a heart attack any moment.

She wishes she could know exactly if her heartbeat is ok, in order to call the ambulance if needed.

Alessandra, a friend of her, suggests that she could use AutomatedSOS service on TrackMe application.

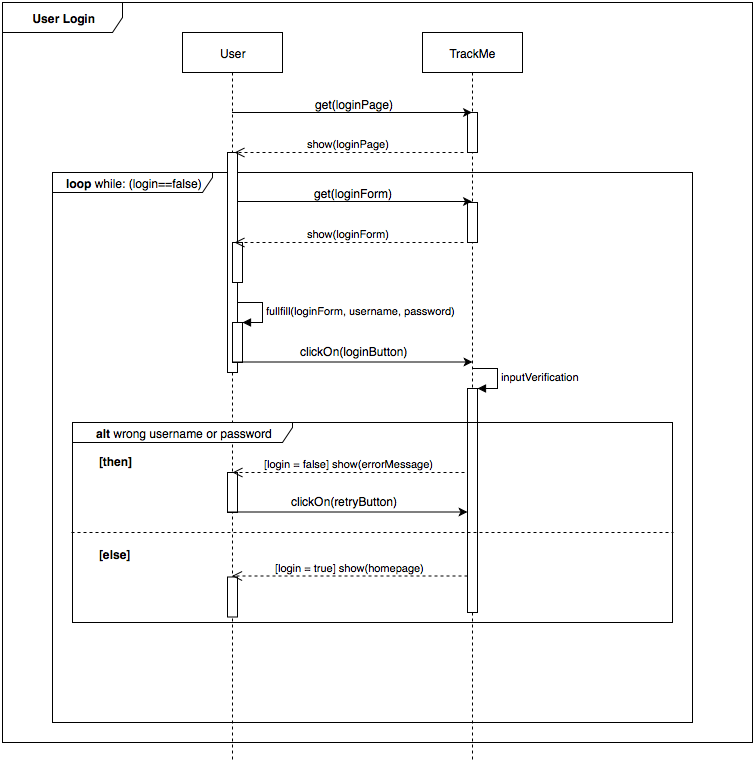
When they meet again after a few weeks Martina thanks Alessandra, she is now much safer about her mother’s situation.

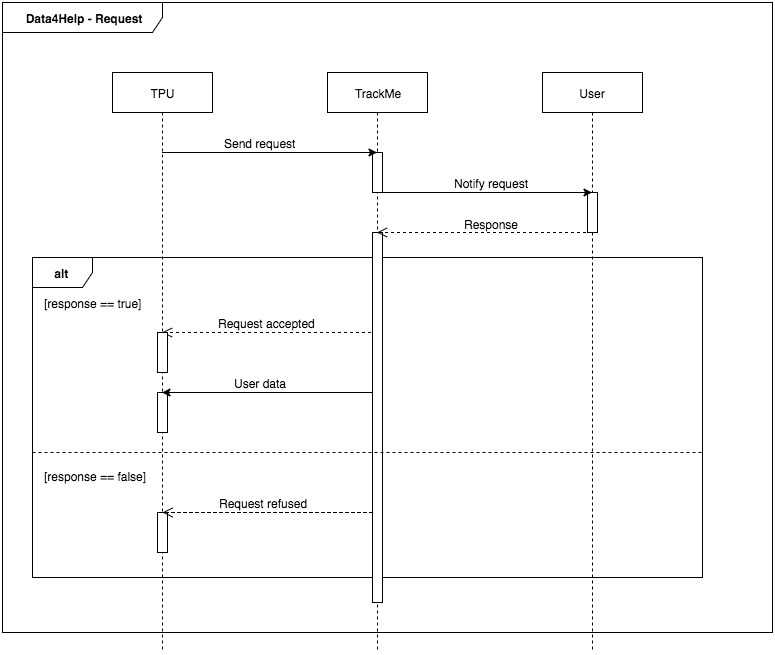
### Scenario 4

Arnoldo is a runner, but he’s getting slower in the last months.

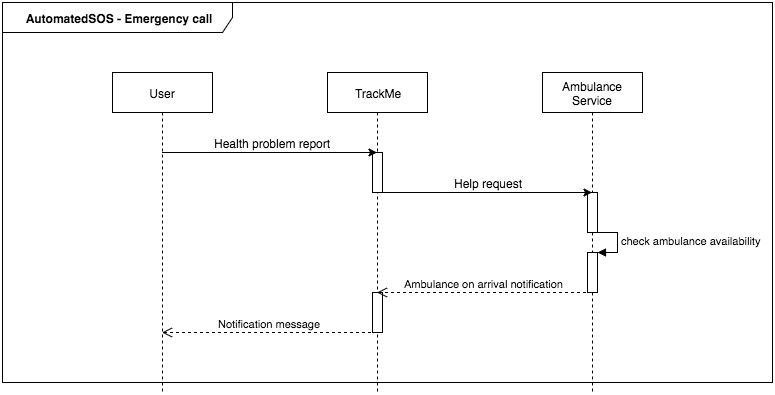
He is worried that he could have some health problems, so he checks his data on TrackMe and he finds his heartbeat had some irregularities.  
He goes to his doctor, and he finds out that he has an heart murmur.

### Sequence Diagrams

This sequence diagram represents the constrains and the sequence of action performed by the user (who can be NU or TPU) to login.



This sequence diagram represents the constrains and the sequence of action performed by the TPU to access specific individuals data.



This sequence diagram represents the help request sent to ambulance service, performed by AutomatedSOS.

## 3D. Performance Requirements

The system doesn’t have particularly strict constraints on the response time for the operations of signing up and logging in.

Personal and anonymous queries should be executed within fifteen seconds.

After a request has been accepted, the TPU should be able to access the user’s data within 2 minutes.

The system has to monitor simultaneously all the Individuals' heart beats, to be able to detect a value outside the thresholds and call an ambulance in less than 5 seconds from when the situation was defined as an emergency (that is to say after 5 seconds having).

## 3E. Design Constraints

Hardware limitations: The system must be provided with a server able to maintain the data of all the users, and of a mirror server in order to maintain the system always online and secure the data.

The system needs a high computational power in order to execute queries on large amounts of data

## 3F. Software System Attributes

### 3F1. Reliability

The server can’t afford to lose any data, because they might be.

### 3F2. Availability

The system must be able to support AutomatedSOS, which requires that the system is always online. In fact, it wouldn’t be able to handle all the request otherwise

### 3F3. Security

The system contains private and sensible data, so it must be extremely secure.

### 3F4. Maintainability

Functionalities must be isolated in a module, leading to the need to analyze and design the interfaces between the modules and the database.

### 3F5. Portability

The interface is simple compared to the rest of the system, which makes it easier to adapt to different devices

# 4. FORMAL ANALYSIS USING ALLOY

//signatures

abstract sig Bool{}

one sig True extends Bool{}

one sig False extends Bool{}

abstract sig User{

fiscalCode: one String,

age: one Int,

code: one Int,

sex: one Bool ,} //Sarebbe un boolean

sig NU extends User{ username: one String,

position: one Position,

follower: set TPU,

device: set Device,

sosIsActivated: one Int,

}

sig TPU extends User{

bankAccount: one String ,

companyName: lone String,

followedMan: set NU, //List of the followed normal users

}

sig Date{

dayNumber: one Int,

monthNumber: one Int,

yearNumber: one Int,

time: one Int, //Sarebbe un Time

}

sig Request {

status: one Int,

text: one String,

sender: one TPU,

receiver: one NU,

link: TPU -> NU,

date: one Date,

}

sig Position {

latitude: Int,

longitude: Int

}{

latitude >= 0 and longitude >= 0

}

sig Data {}

sig Device{

user: one NU,

data: lone String,

code: one Int,

}

sig DeviceHandler

{ device: set Device,}

sig SOS { normalUser: one NU,

position: one Position,

description: one String,}

sig SosHandler {

handledSOS: set SOS,

}

//FACT

// Data4Help

// I TPU del NU e i NU del TPU sono legati da tabelle date dalle request accettatte

fact CoherentAcceptedRequest {

//Antecedente

all r: Request | all n: NU |all t: TPU | r.sender=t && r.receiver=n && r.status=1 iff

//Postcedente

n in t.followedMan && t in n.follower

}

fact CoherentDevice {

//antecedente

all d: Device | all n: NU | d in n.device iff

//Postcedente

n=d.user

}

//Codes are univocal

fact GoodCode {

all u, u': User | all d, d': Device |

u.code=u'.code iff u=u' and

d.code=d'.code iff d=d'

}

//If a Third Part User follows a normal user, that user is followed by him

fact coherentFollowing {

all n: NU | all t:TPU |

t in n.follower iff n in t. followedMan

}

fact sosIsActivatedIsBoolean{

all n: NU| n.sosIsActivated=0 or n.sosIsActivated=1

}

//SOS

fact sosIsCorrect {

//antecedente

all n: NU | all s:SOS|

//Postcedente

s.normalUser=n implies s.position=n.position

} // Position of the SOS and of the Data and of the User us be the same, sae fior name nd fiscal code

fact SOSIsHandled {

all s: SOS | one x : SosHandler |

s in x.handledSOS}

//PREDICATI

pred sendRequest [r, r': Request] {

r=r+r'

r'.status = 0

}

pred acceptRequest [r, r': Request, t, t': TPU, n, n': NU] {

r'.status = 1

t'.followedMan= t.followedMan+r'.receiver

n'.follower=n.follower+r'.sender

}

pred refuseRequest [r, r': Request, n, n': NU] {

r'.status = 0

}

pred addDevice [d, d': Device, n, n': NU, z, z': DeviceHandler]{

n'.code=n.code

d'.code=d.code

n'.device=n.device+d

d'.user=n'

z'.device=z.device+d

}

pred removeDevice [d : Device, n, n': NU, z, z': DeviceHandler]{

z'.device=z.device-d

n'.device=n'.device-d

}

pred show(z: DeviceHandler, n: NU, t: TPU, r: Request){

#TPU = 2

#NU = 2

#Request = 2

#Device=3

#SOS=2

#DeviceHandler=1}

//ASSERTIONS

assert removeUndoesAdd{

all z, z', z'': DeviceHandler| all n, n', n'': NU| all d, d': Device|

addDevice [d, d', n, n', z, z'] and removeDevice [d, n', n'', z', z'']

implies

(z.device=z''.device) and (n.device=n''.device)}

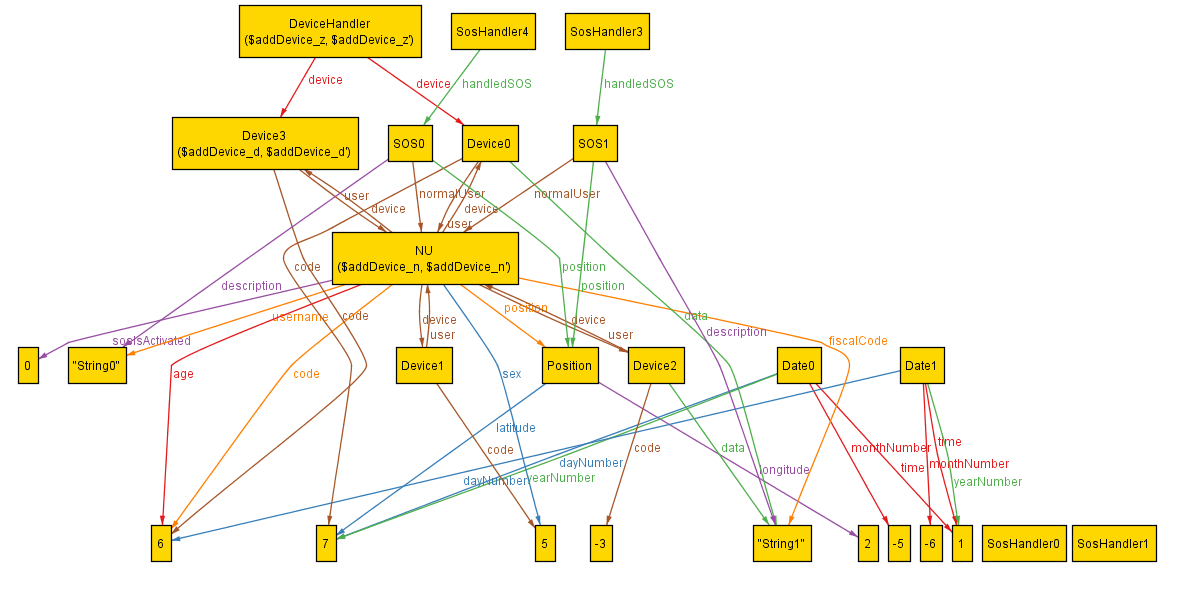
//Run and check

run addDevice for 5 but exactly 2 String

run removeDevice for 2

check removeUndoesAdd for 2

run sendRequest for 2 but exactly 4 String

run show for 1 but exactly 2 String

# 5. EFFORT SPENT

## Luca Grella

|  |  |  |
| --- | --- | --- |
| DATE | TASK | HOURS |
| 17 Oct | Goals | 3h |
| 19 Oct | Requirements | 5h |
| 20 Oct | Domain Assumptions | 3h |
| 22 Oct | Introduction | 6h |
| 24 Oct | State Diagrams | 3h |
| 26 Oct | Scenarios | 2h |
| 29 Oct | Sequence Diagrams | 4h |
| 5 Nov | Mockups | 3h |
| 6 Nov | Other | 2h |
| 7 Nov | Diagrams | 1,5h |
| 9 Nov | Use Case Review | 2h |
| 10 Nov | Mockups Review | 2h |
| 11 Nov | Review | 2,5h |

Tot. 39h

## Daniele Lunghi

|  |  |  |
| --- | --- | --- |
| DATE | TASK | HOURS |
| 18 Oct | Goals | 3h |
| 19 Oct | Requirements | 4h |
| 20 Oct | Domain Assumptions | 4h |
| 21 Oct | Use Case | 6h |
| 24 Oct | Use Case Diagrams | 1h |
| 29 Oct | Mockups | 1h |
| 5 Nov | Alloy | 10h |
| 6 Nov | Other | 2h |
| 7 Nov | Alloy | 2h |
| 9 Nov | Alloy Review | 2h |
| 10 Nov | Scenarios Review | 2,5h |
| 11 Nov | Review | 3,5h |

Tot. 41h

# 6. REFERENCES

1) StarUml 2.8.0  
2) Draw.io

3) Balsamiq  
4) GitHubDesktop 1.0.6  
5) AdobePhotoshop CC 2017

6) alloy4.2