

Politecnico di Milano
AA 2018/2019



POLITECNICO
MILANO 1863

Software Engineering 2

TrackMe

Requirement Analysis and Specification Document

Stefano Pecchia

Edoardo Peretti

Deliverable:	RASD
Title:	Requirement Analysis and Verification Document
Authors:	Stefano Pecchia and Edoardo Peretti
Version:	1.0
Date:	November 11, 2018
Download:	https://github.com/Speck1996/PecchiaPeretti
Copyright:	Copyright © 2018, S. Pecchia and E. Peretti – All rights reserved

Contents

1	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.2.1	Description of the given problem	4
1.2.2	Goals	6
1.3	Definitions, Acronyms, Abbreviations	6
1.3.1	Definitions	6
1.3.2	Acronyms	6
1.3.3	Abbreviations	6
1.4	Document Structure	7
2	Overall description	8
2.1	Product perspective	8
2.1.1	Access to specific data	9
2.1.2	Access to group data	9
2.1.3	Data Subscription	9
2.1.4	SOS service	10
2.2	User characteristics	10
2.3	Assumptions, dependencies and constraints	10
2.3.1	Domain assumptions	10
2.3.2	Privacy constraints	10
3	Specific requirements	12
3.1	External Interface Requirements	12
3.1.1	User Interfaces	12
3.1.2	Hardware Interfaces	16
3.1.3	Software Interfaces	17
3.1.4	Communication Interfaces	17
3.2	Scenarios	17
3.2.1	Scenario 1	17
3.2.2	Scenario 2	17
3.2.3	Scenario 3	18
3.2.4	Scenario 4	18
3.3	Functional Requirements	18
3.3.1	Use case diagram	21
3.3.2	Sequence diagrams	28
3.4	Performance Requirements	31
3.5	Design constraints	32
3.5.1	Standards compliance	32

3.5.2	Hardware limitations	32
3.5.3	Other constraints	32
3.6	Software system constraints	32
3.6.1	Reliability	32
3.6.2	Availability	32
3.6.3	Security	32
3.6.4	Maintainability	33
3.6.5	Compatibility	33
3.6.6	Portability	33
4	Formal analysis using Alloy	34
4.1	Alloy model	34
4.2	Worlds generated	39
4.2.1	Ambulance	39
4.2.2	Specific requests	40
4.2.3	Group requests	41
4.3	Alloy results	43
5	Effort spent	44
6	References	45

Chapter 1

Introduction

1.1 Purpose

Data4Help is a service built to collect, organize and distribute the increasing amount of eHealth data generated by wearables.

This service is built for two typologies of users:

- Third Parties: users that want to access Data4Help collected data. These users will have access to the system by joining through a Web Application.
- Individuals: users that want to provide their eHealth data to Data4Help. Data4Help will also collect additional individuals' informations, in order to better organize and distribute the collected data. These users will have access to the system by instead joining through a smartphone application.

Data4Help can be seen as the intermediary between eHealth data providers and eHealth data analyzers: it will gather and store eHealth data from individuals and it will make the collected data accessible for any third party, respecting however individuals' privacy.

The ability to gather eHealth data will be also used to build another service on top of Data4Help, called AutomatedSOS. This service monitors the health status of a subscribed individual and when some vital parameters cross certain thresholds, sends to the user an ambulance.

Improving individuals lifestyle and helping third parties to analyze their health status will be the key drivers for Data4Help features.

Reading this document will be fundamental to better understand what is the role of Data4Help and what this service should provide to both its kind of users.

1.2 Scope

1.2.1 Description of the given problem

In a fragmented market, like the wearable one, Data4Help will act as a central hub for eHealth data. Individuals will be able to use the supported wearables, in order to upload their data to the system through the smartphone application. In particular they will be able

to upload the following data from their wearables: steps taken, heart rate, sleep time and blood pressure.

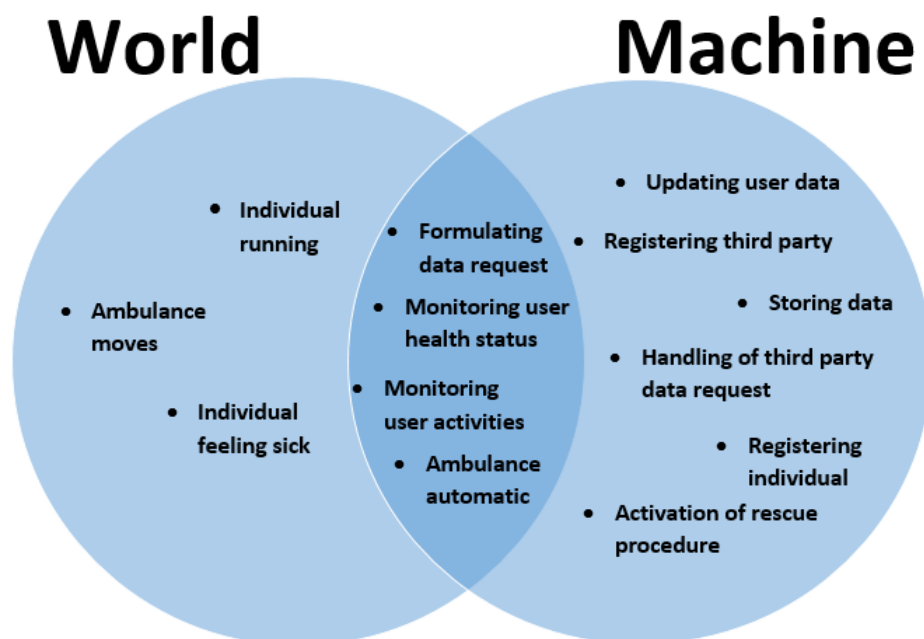
The collected data can be then accessed by third parties in two different ways:

- **Anonymized data:** data of group of individuals, where the group is identified by giving constraints to the requested data. The constraints can be set on some individual attributes by restricting their age, sex, location or on the eHealth data by specifying intervals for each specific data. To maintain a certain level of privacy only requests that detect groups made of a number of individuals that is higher than 1000 are allowed by the system.
- **Specific individual data:** specific individual data can be accessed only after the third party had sent a data access permission request to the individual, and the request was then accepted. The system will act as the intermediary, it will take the third party request and send it to the individual and will then notify the third party with the individual's response, giving access to his data if the request was accepted or by denying the access if the request was not accepted.

The system also gives the opportunity to third parties to be notified when new data is available, for both the specific individual data requests and the group requests.

Individuals can also subscribe to another service, included in the application: Automated-SOS. This service monitors the health status of an individual continuously, and, if certain parameters go below a critical threshold, the application starts automatically a rescue procedure. The rescue procedure should start in under 5 seconds from the crossing of the threshold and consists in sending an ambulance to the position of the user.

To better understand what are the phenomena that are part of the system and what are not, consider the following diagram



1.2.2 Goals

- [G1] Individuals eHealth data is correctly gathered from their wearables.
- [G2] Third parties can have access to accepted group data requests.
- [G3] Third parties can have access to accepted specific individual data requests.
- [G4] Third parties can receive updates whenever new data of their observed groups or individuals is gathered.
- [G5] An ambulance is called whenever an individual vital parameter crosses its critical threshold.
- [G6] Users privacy cannot be violated.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- **Rescue Procedure:** all the operations needed to save a person life. The operations include both the ones handled by the system and humans.
- **eHealth Data:** all the data that can be related to the general health of the user, for example steps taken daily, heart rate, blood pressure, sleep time.
- **Health status:** level of health of a person, obtained analyzing various parameters like heart rate, hours of sleep.
- **Critical Threshold:** value that, referred to a health parameter, must not be passed to guarantee user vital activities.
- **Wearable:** electronics that can be worn on the body, either as an accessory or as part of material used clothing. For Data4Help system, only smartwatches will be initially considered.
- **Data hub:** collection of data from multiple source organized for distribution.

1.3.2 Acronyms

- **TC:** Tax Code
- **DBMS:** Database Management System
- **DDoS:** Distributed Denial of Service
- **GPS:** Global Positioning System

1.3.3 Abbreviations

G_n : n-th goal

D_n : n-th domain assumption

R_n : n-th functional requirement

1.4 Document Structure

This RASD document is organized in the following way: Chapter 1 contains an introductory overview on Data4Help, specifically it describes for which users this service is designed for and what are its main goals.

Chapter 2 contains an overall description of Data4Help, starting from a class diagram that helps to visualize what is the model used to build this service and what is the role of Data4Help in it. A state chart diagram about the AutomatedSOS service is also included in order to better specify the main steps in order to guarantee this service. The chapter also provides a more specific description of Data4Help functionalities, what are the constraints on these functionalities, for which customers the system is designed for and the domain assumptions that are made for its development.

Chapter 3 contains the following external interface requirements: user, hardware, software and communication interfaces. In particular the user interface section contains some mock ups that will serve as guideline to develop a suitable user interface for both the third parties and the individuals. This chapter also includes the functional requirements for the system. These requirements will be explicated through use case and sequence diagrams and with the help of some scenarios that describe concrete examples of Data4Help uses. Non functional requirements are then defined, by describing the performance and the system properties needed to successfully guarantee the functionalities listed before.

Chapter 4 contains the Alloy model used to validate some critical parts, parts that will be specified in an introductory description.

Chapter 5 shows how much time each group member has spent building this document.

Chapter 6 includes the reference documents.

Chapter 2

Overall description

2.1 Product perspective

Data4Help will be a service accessible through a web application for third parties and through a smartphone application for individuals. These applications will be completely developed ground up, while for some back-end and critical activities, external software will be used in order to speed up the development.

The following class diagram helps to visualize how Data4Help will try to provide services for both the individuals and the third parties by functioning as a central hub of eHealth data

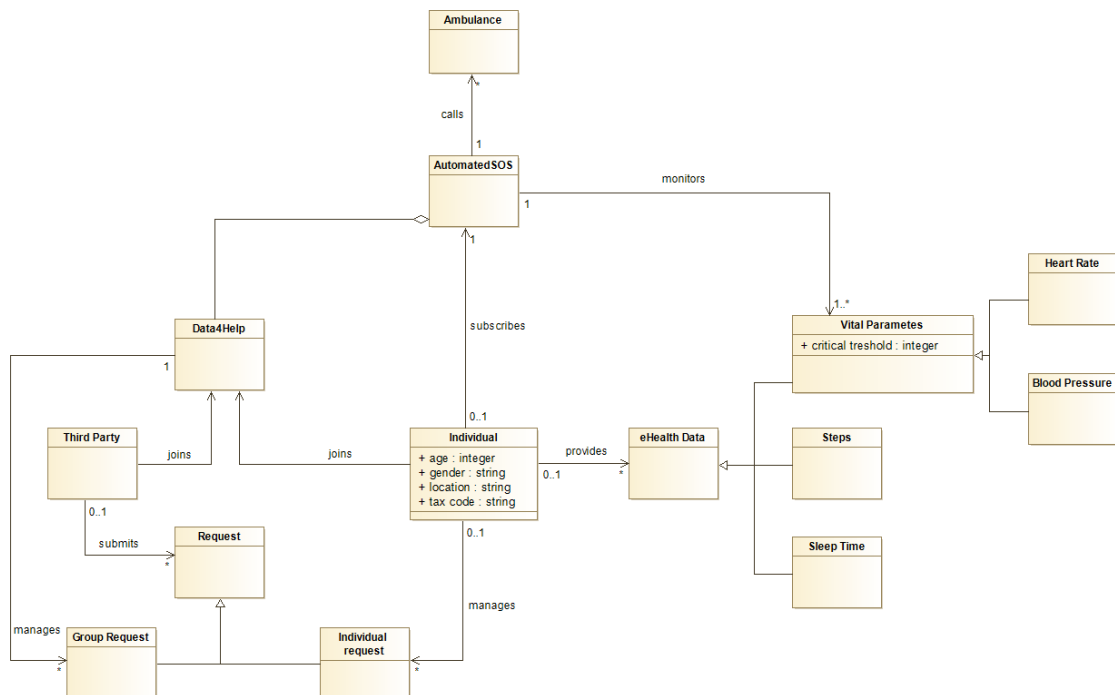


Figure 2.1: Data4Help Class Diagram

Regarding the AutomatedSOS service, the following state chart diagram describes what are the main phases of this service

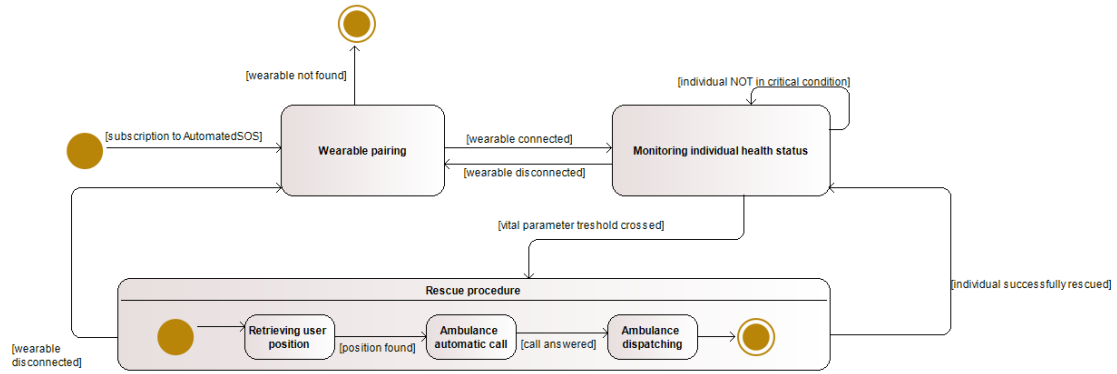


Figure 2.2: AutomatedSOS state chart diagram

2.1.1 Access to specific data

A third party interested in monitoring a specific individual can send a request to the system specifying individual's TC or username. The third party can specify which eHealth data, or individuals' attributes wants to access. The system passes the request to the above mentioned individual who can accept or refuse it. Once the individual has given an answer, the system sends a notification to the third party with the individuals' response: if the individual has given the consent to access his data the third party can then visualize the data.

The data access permission can be managed by the individuals through their application, and can be removed at anytime from when it was accepted. In this case the third party is notified about the individual decision and should formulate another request in order to access again individual's data.

2.1.2 Access to group data

A third party can also request access to anonymized data of a group of individuals. To do so, it must specify some parameters concerning individuals or eHealth data. The specifiable parameters on individuals are: sex, age, country of birth or geographical area. The geographical area can be specified in term of country, region, province, town and district (only for big cities). On the other hand the eHealth data third parties can be filtered giving the range of value desired for steps, sleep time, blood pressure and heart rate. These parameters are provided as averages on a time interval that the third party can choose between a daily, weekly, monthly time horizon.

This type of requests are managed directly by the system. Because TrackMe holds in high regard the privacy of its users, it will satisfy the request only if the number of individuals whose data satisfies the request is higher than 1000.

If the request is positively evaluated, the anonymized data is made available to the third party.

2.1.3 Data Subscription

Third parties can optionally subscribe to data of accepted requests: if new data, matching the third party request, is produced then it will be notified and made available to the requester at most once a week.

The data is provided as long as the data request is valid: if the number of individuals that are part of a group request goes to 1000 or less, or the individual removes the data access, the system stops to update the third party.

Third parties can at any moment, from the subscription to a data request, unsubscribe.

2.1.4 SOS service

The service AutomatedSOS must exploit the real-time stream of eHealth data provided by the underlying Data4Help to offer a personalized and non-intrusive SOS service, especially designed for elderly people. Any individual, correctly registered to Data4Help, can optionally request to activate this service.

If AutomatedSOS has been activated, the system should continuously monitor the parameters of the individuals and compare them using certain specific thresholds. If the parameters are below these thresholds, the system assumes that the individual is having a sudden illness and forwards a request to an emergency service for sending an ambulance to the location of the user. AutomatedSOS must ensure a reaction time of less than 5 seconds from the time the parameters cross their respective thresholds.

2.2 User characteristics

The users of Data4Help and AutomatedSOS services are:

- *Individual* : user that allows the acquisition of the data and can optionally activate AutomatedSOS. He can't use the data request feature of Data4Help.
- *Third party*: user that can request data from the application.

2.3 Assumptions, dependencies and constraints

2.3.1 Domain assumptions

[D1] Location and eHealth data are provided by individuals' devices and assumed to be correct.

[D2] A third party interested in monitoring a specific individual knows the TC or the username of the individual

[D3] Users have access to internet.

[D4] Users have a wearable which is able to measure at least one vital parameter.

[D5] Thresholds for health parameters are provided by medical experts.

[D6] An emergency line is always available.

[D7] Registered users must keep their login credentials secret.

2.3.2 Privacy constraints

The system will collect and elaborate personal data of the individuals and, possibly, it will share them or part of them with third party. For this reason, during the registration activity to the system, all the users must be informed of this practice and they must explicitly confirm their consensus.

In particular, anonymized data can be shared with third parties who request it without the users being further advised. In order to protect its users' privacy and to prevent misuse of data, TrackMe won't share data if the number of individuals whose data satisfy the request is lower than 1000.

Moreover, a third party can request to fully access the data of some specific individual. In this case it is up to the individual to accept or not to share his data with that specific third party.

Chapter 3

Specific requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The following mockups serve as guideline for the application, used by individuals, and the web application, used by third parties. The final product design may differ from these mockups but the functions provided has to be the same.

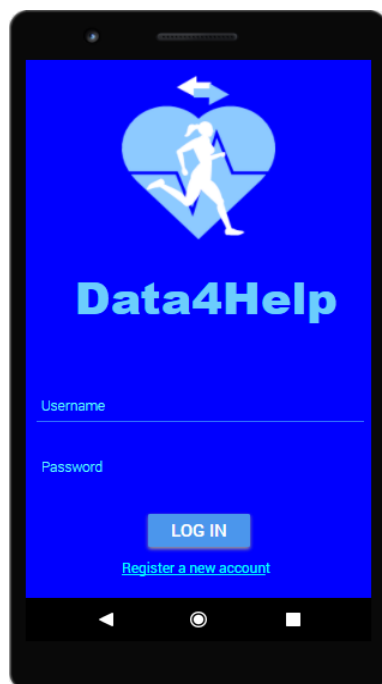


Figure 3.1: App Login screen

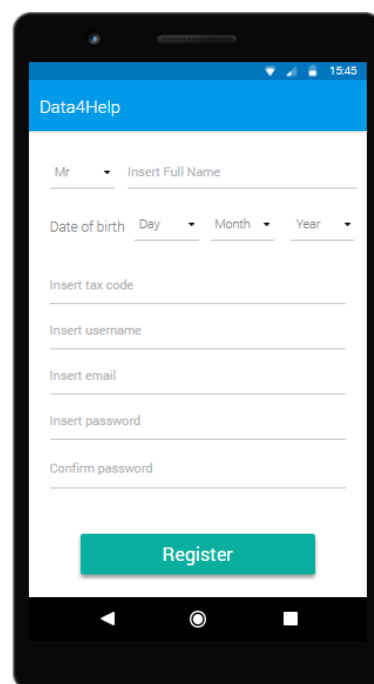


Figure 3.2: Registration screen

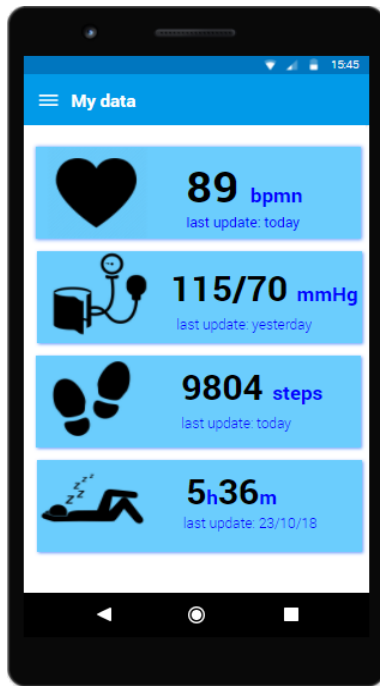


Figure 3.3: App My data screen

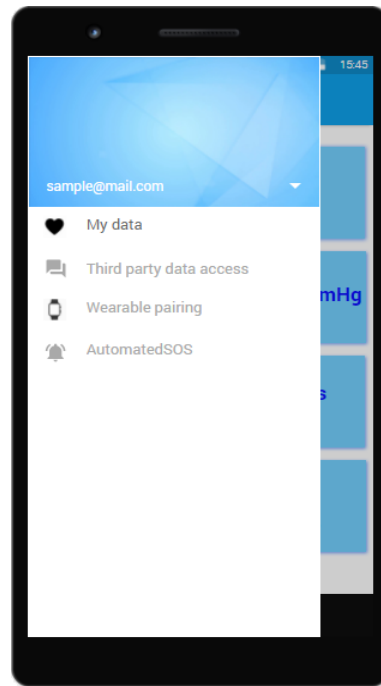


Figure 3.4: App side menu screen

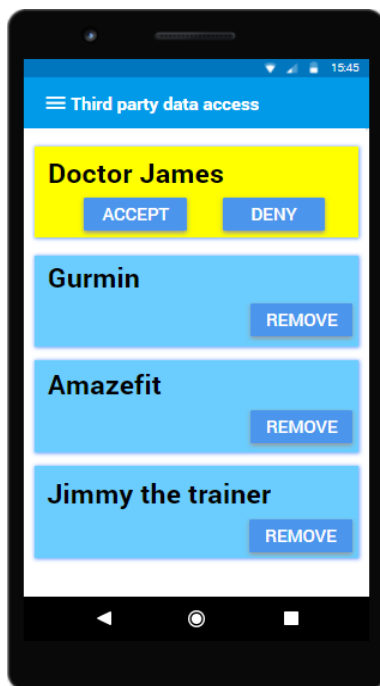
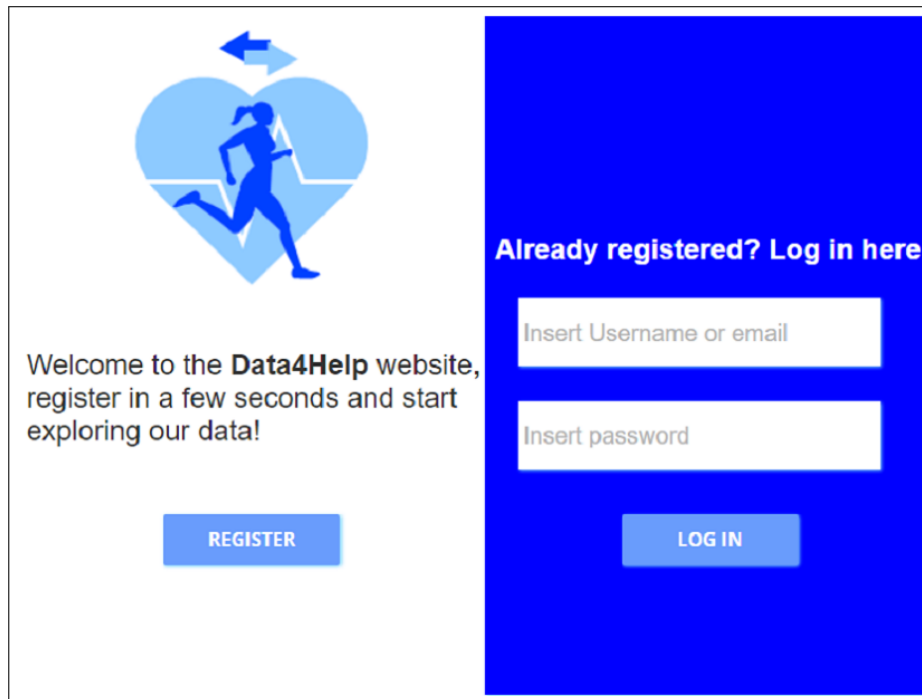


Figure 3.5: App Third Party management screen

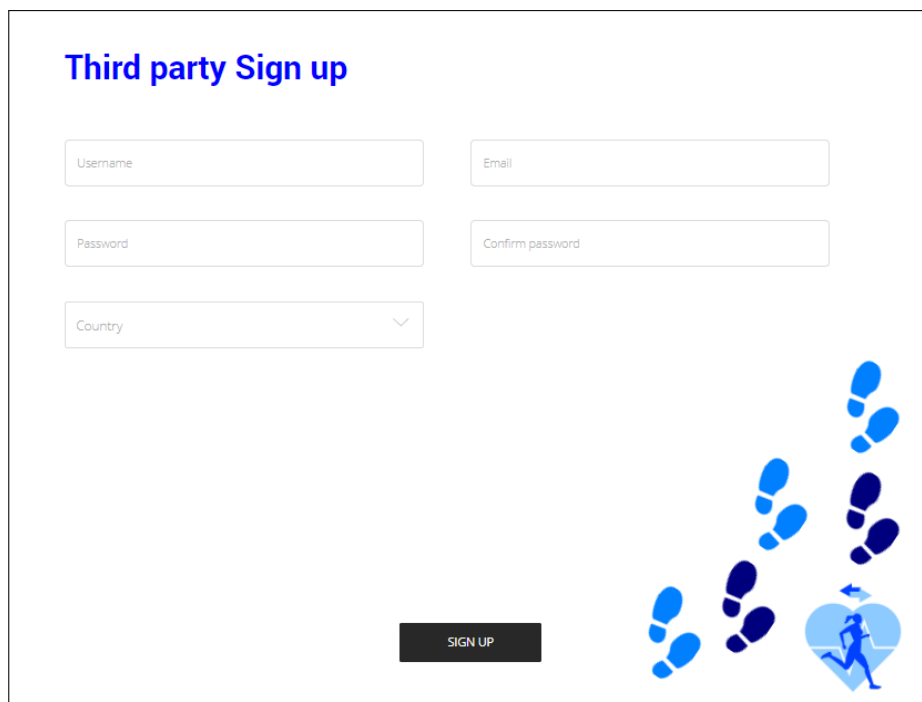


Figure 3.6: App AutomatedSOS screen



The WebApp Home screen is divided into two main sections. The left section has a white background and features a logo at the top: a blue heart with a white line graph and a blue silhouette of a person running. Below the logo, the text reads: "Welcome to the **Data4Help** website, register in a few seconds and start exploring our data!". At the bottom of this section is a blue button labeled "REGISTER". The right section has a solid blue background. At the top, it says "Already registered? Log in here" in white. Below this are two white input fields: "Insert Username or email" and "Insert password". At the bottom of this section is a blue button labeled "LOG IN".

Figure 3.7: WebApp Home screen



The WebApp Sign Up screen has a white background. At the top left, the title "Third party Sign up" is written in blue. Below the title are five input fields arranged in two columns: "Username", "Email", "Password", and "Confirm password" in the first column, and "Country" (a dropdown menu) in the second column. At the bottom center is a black button labeled "SIGN UP". In the bottom right corner, there is a decorative graphic consisting of several blue footprints of varying sizes, with the largest footprint containing the same running person logo seen in the Home screen.

Figure 3.8: WebApp Sign Up screen

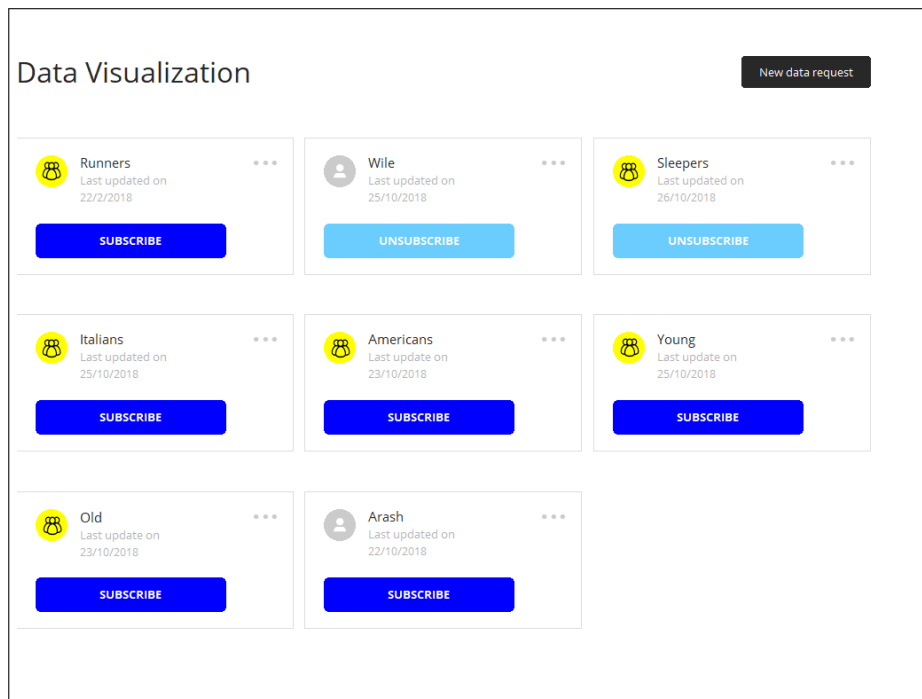


Figure 3.9: WebApp Data visualization screen

The screenshot shows the 'Group data request' form. It is divided into two main sections: 'eHealth data filters' and 'Individuals' attributes filters'. The 'eHealth data filters' section includes dropdown menus for 'Steps' (Daily), 'Sleep time' (Monthly), 'Heart rate' (Weekly), and 'Blood pressure' (Time interval), each with associated 'Minimum value' and 'Maximum value' input fields. The 'Individuals' attributes filters' section includes radio buttons for 'Sex' (Male and Female), input fields for 'Age' (Minimum age and Maximum age), a dropdown for 'Country of birth' (Country), and an 'Insert location' field with a plus icon. A 'SEND' button is located at the bottom center. A decorative graphic of blue footprints is on the right side.

Figure 3.10: WebApp Group data request

Individual data request

Insert individual username or tax code

Write a personalized notification...

Check which eHealth data you want to access:

☐ Sleep time

☐ Steps

☐ Blood pressure

☐ Heart Rate

Check which attributes you want to access:

☐ Sex

☐ Age

☐ Country of birth

☐ Last position

Figure 3.11: WebApp Individual data request

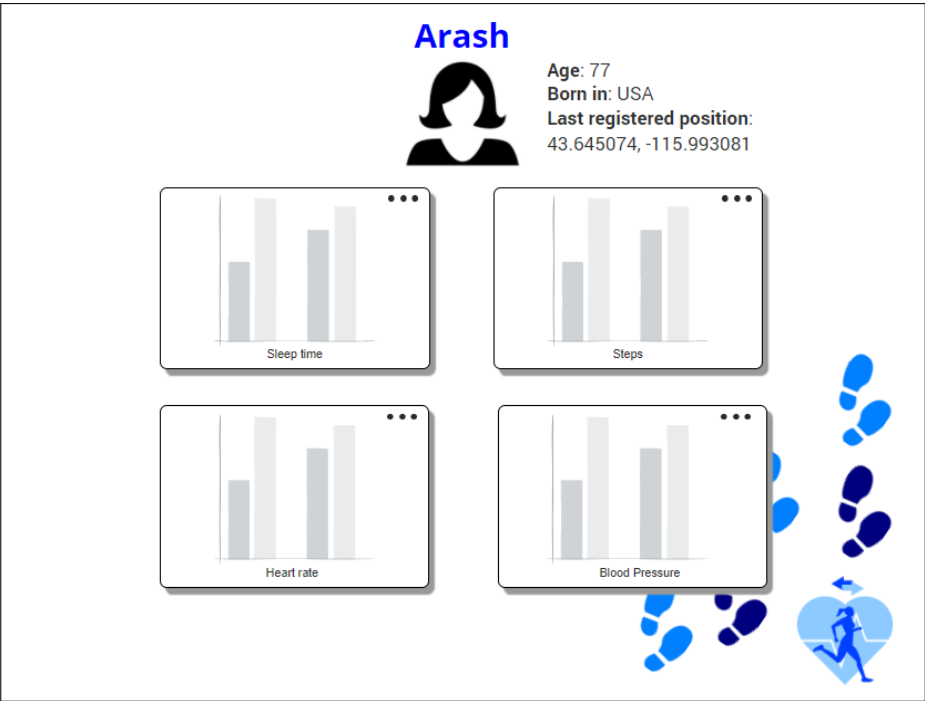


Figure 3.12: Specific Data screen

3.1.2 Hardware Interfaces

Data4Help is an application that can be accessed in two ways, based on the typology of the users:

- Individuals must access to the application through a supported smartphone with working GPS, bluetooth.
- Third parties must access to the application through a compatible web browser with any kind of device. It is suggested, but not necessary to use a device with at least a 10 inches screen to have the best experience using the application.

Data4Help doesn't need any specific hardware interface, all its features will be reachable through the provided software applications.

3.1.3 Software Interfaces

In order to guarantee a quick development Data4Help system will use the following external software services:

- Ambulance call service API: for each country, where the AutomatedSOS service will be enabled, these APIs will be needed in order to implement the automatic call.
- DBMS : this service will be used to manage all users data and satisfy third party queries.
- Location Services: these services will be used to pick the individual position. His position will be used to enrich with more details the eHealth data and for the AutomatedSOS service.
- DDOS security services: these services will be used to protect the servers from DDOS attacks.

3.1.4 Communication Interfaces

The smartphone application and the web application will both use HTTP to ensure a connection with Data4Help system. The communication between the individuals' smartphone application and the wearable will use instead the bluetooth protocol.

3.2 Scenarios

3.2.1 Scenario 1

Rebecca has worked very hard in the last months and she has been a long stressful period. Recently she had to skip several working days because she didn't feel very well. She consulted her doctor John who didn't notice anything alarming. John then discovered Data4Help and he thought that it could have been useful to monitor the health status of Rebecca on daily basis for the next weeks. Then, John asked Rebecca to register to Data4Help and to use her smartwatch to collect some data. Rebecca immediately registered to the service and accepted the request that John sent to her. Now John can have a better look at his patient's health status and can make better diagnosis.

3.2.2 Scenario 2

Walter is a 75 years old man who lives alone in a small town. His doctor has been monitoring him thanks to Data4Help because he has a cardiac disease. Last week, Walter was at the supermarket and suddenly he fainted. Luckily, the store was crowded and therefore someone helped him and called an ambulance. He is afraid it could happen again and because he lives alone, he decides to activate the AutomatedSOS service.

3.2.3 Scenario 3

FatBit is a startup that wants to launch its first wearable. To optimize their limited resources and to improve their ads, Fatbit's managers want to know which are the countries with healthier people, which one have the more active, and the average age of their potential consumers. Collecting this data it's not easy, especially when joining the market for the first time. Fortunately a new service, Data4Help, has just launched and its main features it's the distribution of eHealth data, organized according to different parameters, included the ones FatBit managers are interested in (age,location,steps taken daily). Therefore FatBit's managers decide to register with a unique account as third party, and start their data analysis thanks to Data4Help.

3.2.4 Scenario 4

Wile is an engineering student who wants to get back in shape. He decides to join the gym and to buy a wearable to see its progress over time. Initially Wile is satisfied by his wearable and the built in application but later he starts noticing that more and more wearables with improved design and many new features are launching in the market. Wisely, he notices that the built in application collects the data only for the producer's devices and so if he wanted to change his wearable without losing data he were practically forced to buy a new watch from the same company. Therefore he decides to use Data4Help, allowing him to change his wearable with the one he likes the most without losing data.

3.3 Functional Requirements

[G1] Individuals eHealth data is correctly gathered from their wearables.

[D1] Location and eHealth data are provided by individuals' devices and assumed to be correct.

[D3] Users have access to internet.

[R1] Users are able to login with the username and the password associated to their account.

[R2] Data4Help is able to pair with the user wearable.

[R3] Data4Help is able to download eHealth data from the individual's wearable.

[R4] Individuals can upload their data through Data4Help app.

[R4.1] Data4Help is able to store the data provided by individuals.

[R4.2] Data4Help is able to organize data provided by individuals.

[G2] Third parties can have access to accepted group data requests.

[D3] Users have access to internet.

[R4.1] Data4Help is able to store the data provided by individuals.

[R1] Users are able to login with the username and the password associated to their account.

[R4.2] Data4Help is able to organize data provided by individuals.

[R5] Third parties can formulate requests to access anonymized data of groups.

[R6] Third parties can apply filters on data while formulating their request for data of groups.

[R7] The system checks if the number of individuals in the group detected by the third party request is higher than 1000.

[R7.1] If the groups has 1000 or less individuals the system denies the request.

[R8] The system is able to distribute the requested data to the third party.

[G3] Third parties can have access to accepted specific individual data requests.

[D3] Users have access to internet.

[D2] A third party interested in monitoring a specific individual knows the TC or the username of the individual

[R4.1] Data4Help is able to store the data provided by individuals.

[R1] Users are able to login with the username and the password associated to their account.

[R4.2] Data4Help is able to organize data provided by individuals.

[R9] Data4Help is able to forward the request to the individual specified in the third party.

[R10] The individual to whom the request will be forwarded is able to reply.

[R10.1] The system notifies the third party about the individual reply.

[R11] The system can check if the individual has accepted the data access request by the third party.

[R8] The system is able to distribute the requested data to the third party.

[G4] Third parties can receive updates whenever new data of their observed groups or individuals is gathered.

[D3] Users have access to internet.

[R1] Users are able to login with the username and the password associated to their account.

[R4.1] Data4Help is able to store the data provided by individuals.

[R4.2] Data4Help is able to organize data provided by individuals.

[R12] Third parties can specify to be updated whenever new data of groups detected by their data requests, or observed individuals is gathered. The time interval between an update and the next one can be set by the third party, in a limit that goes from one hour to 1 year.

[R13] The system is able to update third parties with new data, respecting their preferences on the interval timing.

[G5] An ambulance is called whenever an individual vital parameter crosses its critical threshold.

[D3] Users have access to internet.

[R1] Users are able to login with the username and the password associated to their account.

- [R14] Individual can subscribe to the AutomatedSOS service.
- [R2] Data4Help is able to pair with the user wearable.
- [D4] Users have a wearable which is able to measure at least one vital parameter.
- [R15] The system is able to recognize if the wearable can measure at least a vital parameter.
- [R16] The system is able to continuously monitor individual vital parameters.
- [R17] If the wearable is disconnected the AutomatedSOS service is suspended, untill the individual connects his wearable again.
- [D5] Thresholds for health parameters are provided by medical experts.
- [D6] An emergency line is always available.
- [D1] Location and eHealth data are provided by individuals' devices and assumed to be correct.
- [R18] Whenever a vital parameter cross its respective threshold, the system calls an ambulance in under 5 seconds, giving to the ambulance driver the location of the individual.

[G6] Users privacy cannot be violated.

- [D7] Registered users must keep their login credentials secret.
- [R19] The system doesn't allow third parties to access specific individuals data without first asking for their permission.
- [R20] The system stops to update third parties about new data whenever observed individuals decide to remove data access permissions, or whenever the number of individuals of an observed groups goes below 1001.
- [R21] Only users that know their username and password can access to their respective accounts

3.3.1 Use case diagram

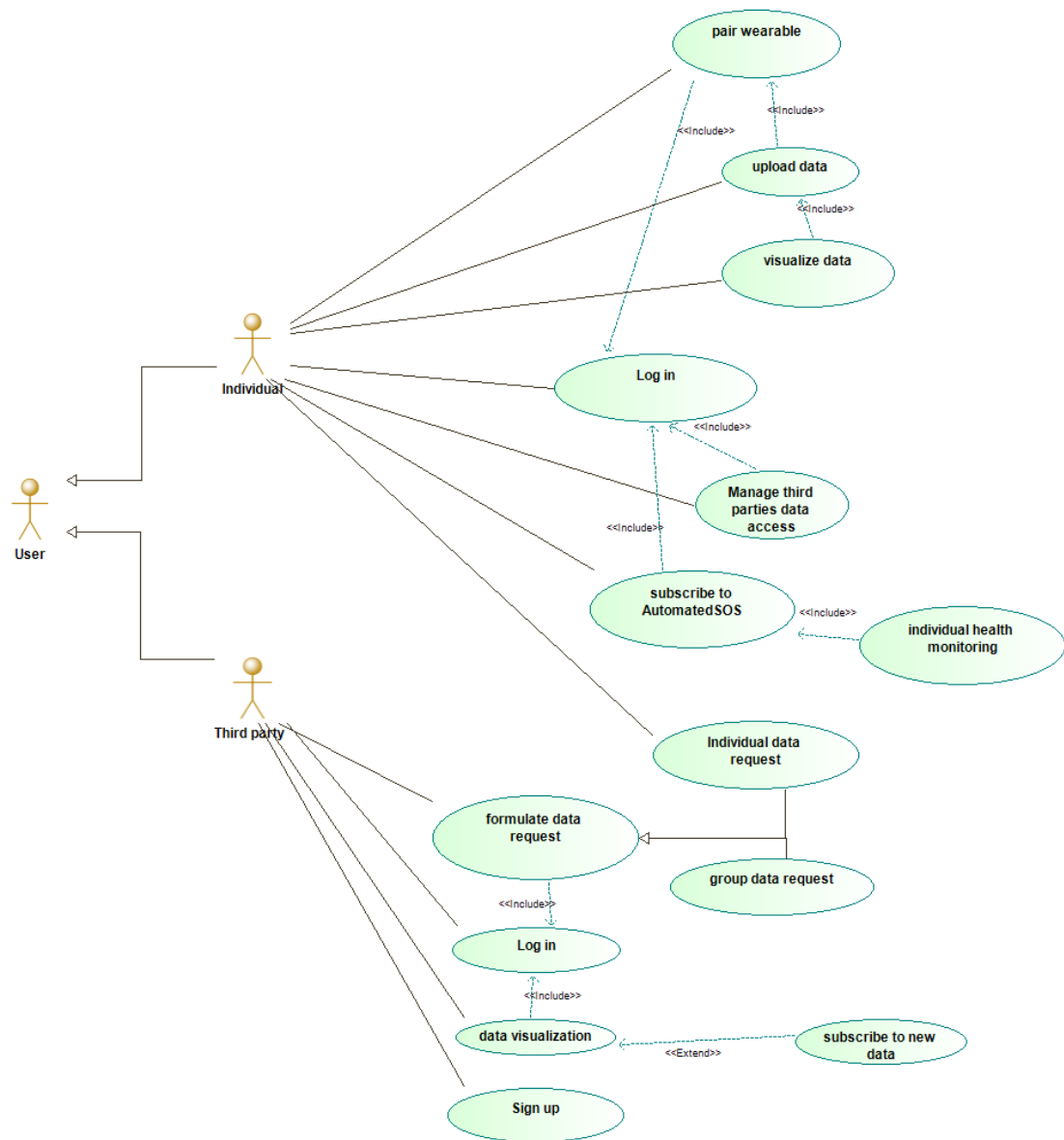


Figure 3.13: Data4Help Use case diagram

Name	Smartphone application sign up
Actor	Individual
Entry condition	The individual installed the application and is connected to the internet
Events flow	<ol style="list-style-type: none"> 1. The individual opens the application 2. The individual taps on "Register a new account" 3. The individual fills all the mandatory fields with correct information 4. The individual clicks on "Register".
Exit Condition	Individual credentials have successfully been added to the system, the individual can now log in
Exceptions	<ol style="list-style-type: none"> 1. The individual username or his email is already present in the system database. 2. Individual didn't fill all the mandatory fields. 3. Individual provided invalid data. <p>All the exceptions above are notified to the individual who is taken back to the registration form.</p>

Name	Web application sign up
Actor	Third party
Entry condition	The third party downloaded the web application and is connected to the internet
Events flow	<ol style="list-style-type: none"> 1. The third party clicks on "Register a new account" 2. The third party fills all the mandatory fields with correct information 3. The third party clicks on "Sign up".
Exit Condition	Third party credentials have successfully been added to the system, the third party can now log in
Exceptions	<ol style="list-style-type: none"> 1. The third party username or his email is already present in the system database. 2. The third party didn't fill all the mandatory fields. 3. The third party provided invalid data. <p>All the exceptions above are notified to the third party, which is taken back to the registration form.</p>

Name	Smartphone Log In
Actor	Individual
Entry condition	The individual has already signed up, has installed the application and is connected to the internet.
Events flow	<ol style="list-style-type: none"> 1. The individual opens the application 2. The individual fills the username and password field in the app homepage. 3. The individual taps on "Log in". 4. The username and the password inserted are sent to the system. 5. The system checks if the credentials are valid.
Exit Condition	The individual is successfully logged in and the app redirect him to the wearable pairing screen.
Exceptions	The inserted credentials are not found in the system database. The application notifies that the inserted credentials are wrong.

Name	Web application Log In
Actor	Third party
Entry condition	The third party has already signed up, has downloaded the web application and is connected to the internet.
Events flow	<ol style="list-style-type: none"> 1. The third party fills the username and password field in the web application homepage 2. The third party clicks on "Log in". 3. The username and the password inserted are sent to the system. 4. The system checks if the credentials are valid.
Exit Condition	The third party is successfully logged in and can access his personal area.
Exceptions	The inserted credentials are not found in the system database. The application notifies that the inserted credentials are wrong.

Name	Wearable pairing
Actor	Individual
Entry condition	The individual has logged in and both his wearable and his phone have bluetooth on.
Events flow	<ol style="list-style-type: none"> 1. The individual reaches the wearable pairing screen. 2. The individual ensures that the wearable is on and close to the smartphone. 3. The individual taps on pair wearable.
Exit Condition	Individual wearable is paired and the app is ready to gather individual eHealth data through it.
Exceptions	<ol style="list-style-type: none"> 1. The application isn't able to recognize a wearable. 2. Bluetooth goes down during the pairing. <p>The exception above is notified by the application. The individual can repeat the pairing procedure.</p>

Name	Upload Data
Actor	Individual
Entry condition	The individual is logged in and the wearable is paired.
Events flow	<ol style="list-style-type: none"> 1. The wearable has gathered some individual eHealth data. 2. The individual goes to the home application screen. 3. The application downloads the data from the wearable. 4. The application uploads the data on the system.
Exit Condition	Individual eHealth data is successfully uploaded and stored in the system
Exceptions	<ol style="list-style-type: none"> 1. Connection falls down during the upload phase. 2. The application fails the download of eHealth data from the individual's wearable. <p>The application notifies that there were problem during the upload phase. The individual can repeat the procedure by swiping down.</p>

Name	Visualize data
Actor	Individual
Entry condition	The individual is logged in and has already uploaded some data to the system
Events flow	<ol style="list-style-type: none"> 1. The individual goes to "My data" section. 2. The individual selects which data he wants to see in details. 3. The application downloads the data from the system.
Exit Condition	Individual eHealth data is successfully loaded on the device screen.
Exceptions	<ol style="list-style-type: none"> 1. Connection falls down during the upload phase. 2. The application fails the download of eHealth data. <p>The application notifies that there were problem during the upload phase. The individual can repeat the procedure by selecting again the data.</p>

Name	Accept third parties data access requests.
Actor	Individual, Third party
Entry condition	The individual is logged in and a data access request notification from a third party is received.
Events flow	<ol style="list-style-type: none"> 1. The individual goes to the third parties access app section. 2. The individual answers the data access request, giving or not giving the permission to the third party to access his data. 3. The application forward the individual response to third party.
Exit Condition	The third party is notified about the individual decision.
Exceptions	<ol style="list-style-type: none"> 1. Connection falls down while the individual is answering the request. <p>The application notifies that the device is disconnected.</p>

Name	Remove third parties data access
Actor	Individual
Entry condition	The individual is logged in and wants to remove a third party data access
Events flow	<ol style="list-style-type: none"> 1. The individual goes to the third parties management app section. 2. The individual clicks on the remove button associated to the third party.
Exit Condition	The system correctly change data access permissions and notifies the third party of the change
Exceptions	<ol style="list-style-type: none"> 1. No third party already has access to individual's data. The app notifies the individual about it. 2. The connection goes down during the process. The app notifies this to the individual, who will have to repeat the procedure when the connection is back.

Name	Subscribe to AutomatedSOS service
Actor	Individual
Entry condition	The individual is logged in and the wearable is paired
Events flow	<ol style="list-style-type: none"> 1. The individual goes to the AutomatedSOS app section. 2. The individual click the subscribe button. 3. The app checks if the wearable paired can gather the necessary data for the AutomatedSOS service. 4. The app starts monitoring the individual.
Exit Condition	The wearable starts monitoring individuals vital parameters
Exceptions	<ol style="list-style-type: none"> 1. Wearable cannot monitors individuals vital parameters. 2. Wearable unpairs during the procedure. 3. The connection falls down during the procedure. <p>The exception above are notified to the individuals. The individual must repeat the whole procedure.</p>

Name	Individual data request
Actor	Third party
Entry condition	The third party is logged in and knows the individual's username or TC
Events flow	<ol style="list-style-type: none"> 1. The third party selects the data request button. 2. The third party insert the individual identifier (either his username or his TC). 3. The system forwards the request to the individual.
Exit Condition	A notification of a new data access request is sent to the individual
Exceptions	<ol style="list-style-type: none"> 1. The individual identifier is not found in the system. <p>The exception above is notified to the third party, which is sent back to the data request screen.</p>

Name	Group data request
Actor	Third party
Entry condition	The third party is logged in
Events flow	<ol style="list-style-type: none"> 1. The third party goes in the data request section. 2. The third party selects the group data request. 3. The third party can apply filters on his request. To be precise it can filter data by individuals age, sex, location (country, city) and can specify which data it is interested in (steps taken ,sleep hours, average heart rate, average blood pressure). The time interval of the each requested data can be specified. 4. The system analyzes the request and distributes the requested data to the third party.
Exit Condition	The requested data is visualized by the third party
Exceptions	<ol style="list-style-type: none"> 1. The group of individuals detected by the third party has a number of members of 1000 or less. <p>The exception above is notified to the third party, which is sent back to the data request screen.</p>

Name	Visualize data
Actor	Third party
Entry condition	The third party is logged in and has access to some data (either group or individual)
Events flow	<ol style="list-style-type: none"> 1. The third party goes in the data visualization section. 2. The third party clicks on the subscribe button of the data request he is interested in. 3. The system starts automatically to update the data to which the third party is subscribed to.
Exit Condition	The application notifies the third party of the new subscription.
Exceptions	<ol style="list-style-type: none"> 1. The number of individuals of the group detected by the data request which the third party wants to subscribe to went to 1000 or below. 2. The individual removed the permission to access his data from the third party. <p>The exceptions above are notified to the third party that will not be updated anymore with new data.</p>

3.3.2 Sequence diagrams

The following diagrams model the main features of Data4Help: data uploading by individuals, data request by third parties and AutomatedSOS subscription. Some assumptions were considered in order to simplify these diagrams, the system should act without them:

- The login credentials provided by the users are correct;
- The wearable used during the pairing is compatible with the applications;
- User devices are connected to the internet for the whole time.

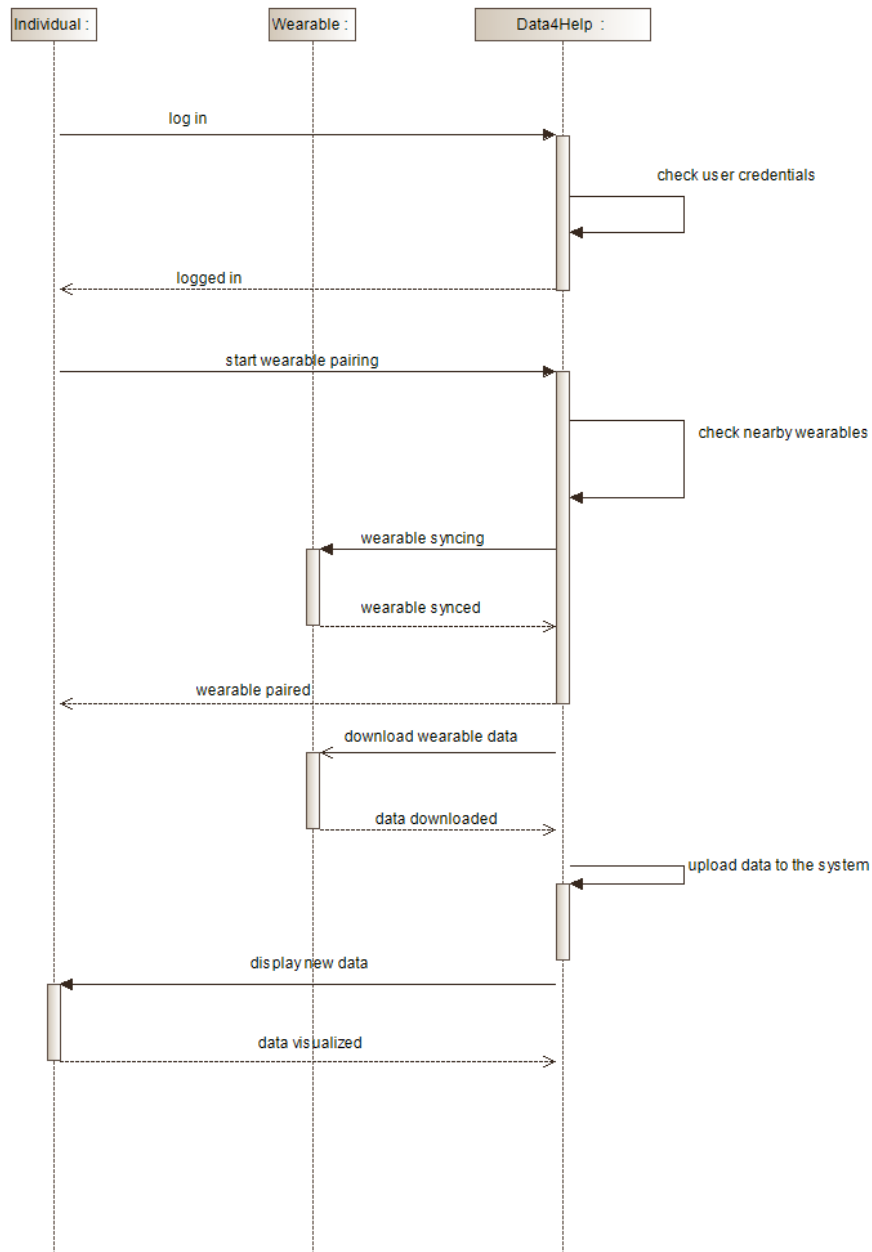


Figure 3.14: Individual data upload screen

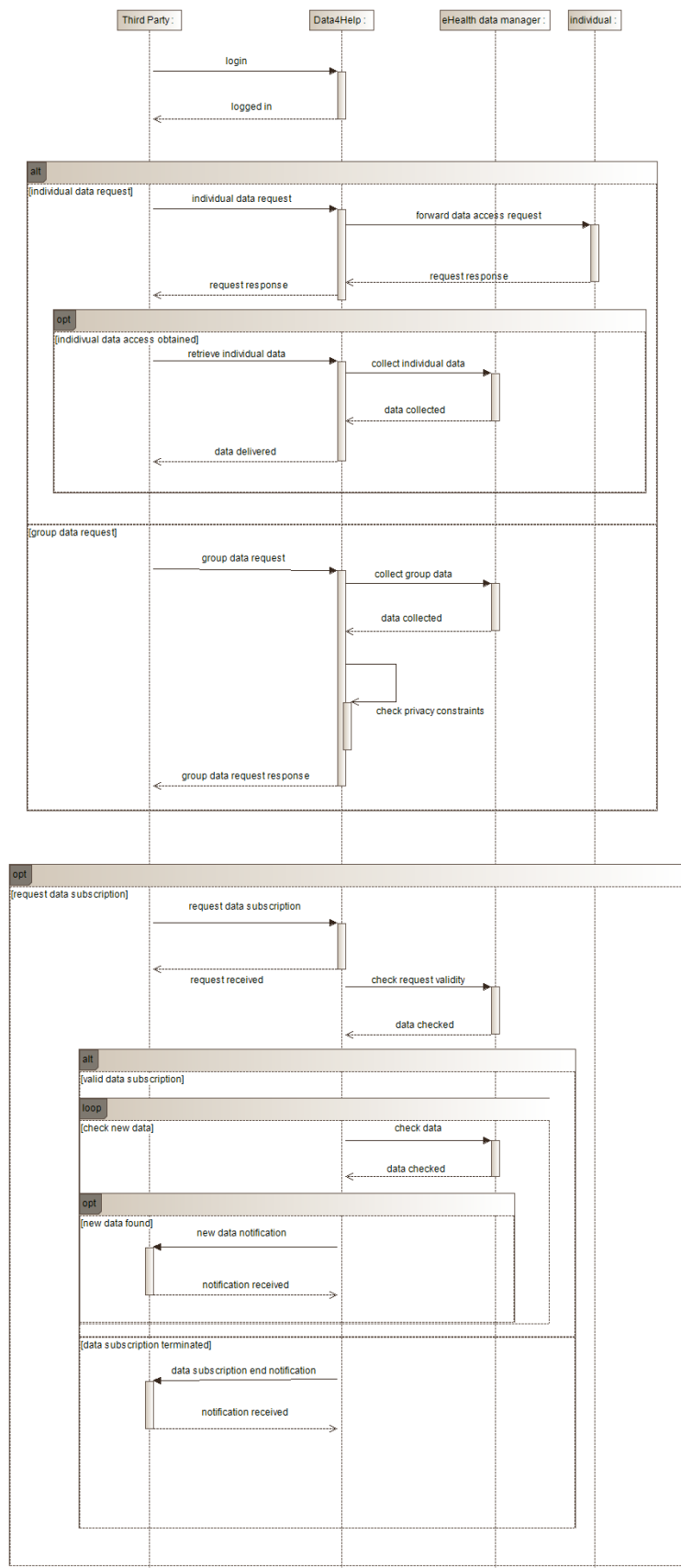


Figure 3.15: Third Party data request sequence diagram

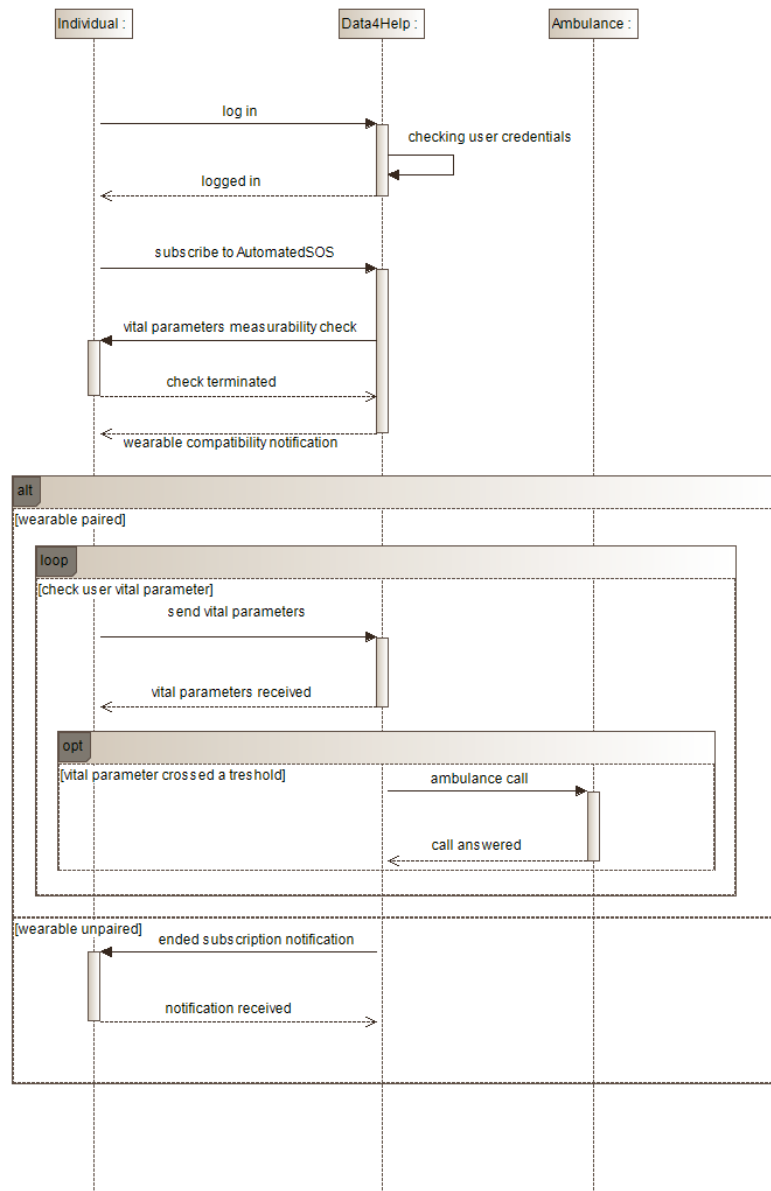


Figure 3.16: AutomatedSOS subscription sequence diagram

3.4 Performance Requirements

Data4Helps is a service that strongly depends on the amount of data gathered, therefore the initial phase will focus on recruiting as more individuals as possible.

Giving a realistic estimation on the numbers of users is not possible but a total amount of 100000 individuals and 500 third parties will be considered a good starting point.

In order to expand the user base and have from the start good feedbacks the development will be focused on providing a reliable and persistent service.

Regarding the AutomatedSOS service, the ambulance call should be done with a delay time from the crossing of the critical thresholds that is under 5 seconds.

3.5 Design constraints

3.5.1 Standards compliance

- The app only requests the minimum permissions in order to guarantee its core functionalities: storage access, location and wearable access.
- The app only supports portrait mode.
- All the data gathered through individuals wearables will be uploaded and stored in the system database.

3.5.2 Hardware limitations

Individuals will use a smartphone app in order to access the service. The following devices are the one that will be compatible with the app:

- iOS or Android smartphone with following capabilities:
 - 2G/3G/4G connection;
 - Bluetooth connection;
 - GPS;

These are instead the smartwatches that will be initially supported in the app:

- watchOS or wearOS smartwatch

Third parties will have access to the service through a WebApp, accessible with any browser compatible with html5 and java.

3.5.3 Other constraints

The app must make clear that **AutomatedSOS service DOES NOT substitute** the manual emergency call since that the availability of the service rely on external factors such as internet connection, wearable hardware and only a couple of vital parameters can be monitored.

The Automated SOS service was born as an additional service that can anticipate the emergency call, not as a service that substitutes it.

3.6 Software system constraints

3.6.1 Reliability

The application aims to provide a 24/7 service, but, especially in the initial phase, some time of unavailability for maintenance is expected.

3.6.2 Availability

The initial target of availability will be a system that is up for the 95.00% of the time. This value should improve over time as the service expands and more resources are available.

3.6.3 Security

Users passwords will be encrypted and stored in a specific database. In order to prevent DDoS attacks an external security service on the servers will be enabled.

3.6.4 Maintainability

The whole system will be designed in a modular way in order to separate each component from the other ones, guaranteeing an high level of flexibility.

AutomateSOS is a first, clear example of the expandability nature of Data4Help.

3.6.5 Compatibility

Gathering the possible largest amount of eHealth data is the fundamental feature of the service. Although, due the fragmentation of the wearable market, only watchOS and wearOS smartwatches will be initially supported.

TizenOS smartwatches will get official support in no longer than 3 months after the app initial release.

The app will support both Android and iOS smartphones while the WebApp will be compatible with any browser that supports html5 and java applets.

3.6.6 Portability

In order to optimize resources the application will be initially only developed for iOS and Android, the two major players in the smartphone market and their corresponding smartwatch operative systems.

Since Data4Help strongly relies on the amount of its registered individuals, any new operative system that gets at least a 10% share of the market will be considered for an application port.

Chapter 4

Formal analysis using Alloy

It can be useful to have a formal model for some of the main functionalities of Data4Help and AutomatedSOS. The following Alloy model explains in details the three core features of the system:

- The sending of an ambulance whenever the vital parameters of an individual, who has activated AutomatedSOS, crosses the given threshold. In particular, it is modeled the maximum reaction time.
- The possibility that a third party can monitor the data of a specific individual. In particular, the model focuses on the fact that the individual must give permission to the third party in order to access his data.
- The access to a collection of data by a third party. In particular, the systems must enforce a certain level of privacy.

For simplicity and without any loss of generality, all the entities that have a numeric value are represented with integer values in small intervals. Moreover, only upper thresholds for vital parameters are considered.

4.1 Alloy model

```
1 open util/integer
2 open util/boolean
3
4 sig Location {
5     coordX: one Int,
6     coordY: one Int
7 }
8 {coordX >= -3 and coordX <= 3 and coordY >= -6 and coordY <=
   6}
9
10 sig Time {
11     time: one Int
12 }
13 {time >= 0 and time <= 6}
14
15 sig Individual {
```

```

16   hasSOS: one Bool,
17   data: lone EHealthData,
18   location: lone Location,
19   illness: lone Illness
20 }
21
22 sig EHealthData {
23   heartRate: lone Int,
24   bloodPressure: lone Int,
25   steps: lone Int,
26   sleepTime: lone Int
27 }
28 {heartRate >= 0 and heartRate <= 6 and bloodPressure >= 0
   and bloodPressure <= 6 and steps >= 0 and steps <= 6 and
   sleepTime >= 0 and sleepTime <= 6}
29
30 sig Illness {
31   startTime: one Time
32 }
33
34 sig Ambulance {
35   startTime: one Time,
36   rescuee: one Individual,
37   location: one Location
38 }
39
40 sig EHealthDataGroup {
41   data: some EHealthData
42 }
43
44 sig ThirdParty {
45   request: set Request,
46   monitor: set Individual,
47   groupData: set EHealthDataGroup
48 }
49
50 abstract sig Request {
51   approved: one Bool
52 }
53
54 sig GroupRequest extends Request {
55   groupData: one EHealthDataGroup
56 }
57
58 sig SpecificRequest extends Request {
59   individual: one Individual
60 }
61
62

```

```

63
64 /*
65     Ambulance sending
66 */
67
68 -- Every EHealthData belongs to a single Individual
69 fact DataBelongsToSingleIndividual {
70     (all d: EHealthData | one i: Individual | i.data = d)
71 }
72
73 -- There is a Illness iff threshold have been crossed
74 fact IllnessIndividual{
75     (all i: Individual | i.hasSOS = True and
76         (i.data.heartRate >= 5 or i.data.bloodPressure >= 5)
77         implies one il: Illness | il = i.illness) and (all
78         il: Illness | one i: Individual | il = i.illness and
79         i.hasSOS = True and (i.data.heartRate >= 5 or
80         i.data.bloodPressure >= 5))
81 }
82
83 -- Prevent two individual with same Illness
84 fact IllnessSigleIndividual {
85     all il: Illness | no disj i1, i2: Individual | il =
86         i1.illness and il = i2.illness
87 }
88
89 -- An ambulance is sent for a good reason with a reaction
90 -- time of less than 2
91 fact AmbulanceIndividual {
92     all a: Ambulance | a.rescuee.location = a.location and
93         #a.rescuee.illness = 1 and a.startTime.time >=
94         a.rescuee.illness.startTime.time and
95         minus[a.startTime.time,
96             a.rescuee.illness.startTime.time] <= 2
97 }
98
99 -- If an Individual is having an illness and he hade
100 -- activate AutomatedSOS then an Ambulance is directed to
101 -- his location.
102 -- Constraint also that only one ambulance has been sent
103 fact IndividualAmbulance {
104     all i: Individual | #i.illness = 1 implies one a:
105         Ambulance | a.rescuee = i
106 }
107
108 /*
109     Requests

```

```

98 */
99
100 -- Every request has been sent by a single ThirdParty
101 fact RequestThirdParty {
102     (all r: Request | one t: ThirdParty | r in t.request )
103 }
104
105
106 /* Specific Requests */
107
108 -- A third party monitor only individuals who has accepted
    to be monitored
109 fact IndividualMonitored {
110     all t: ThirdParty | all i: Individual | i in t.monitor
        implies          one r: SpecificRequest | i =
            r.individual and r.approved = True and r in t.request
111 }
112
113 -- If a SpecificRequest is approved, a ThirdParty is
    monitoring the specific Individual
114 fact SpecificRequestApproved {
115     all t: ThirdParty | all r: SpecificRequest | r in
        t.request and r.approved = True implies r.individual
            in t.monitor
116 }
117
118 -- No multiple SpecificRequest for the same Individual from
    the same ThirdParty
119 fact NoMultipleSpecificRequest {
120     all t: ThirdParty | no disj r1, r2: SpecificRequest | r1
        in t.request and r2 in t.request and r1.individual =
            r2.individual
121 }
122
123
124 /* Group Requests */
125
126 -- If TrackMe can properly anonymized those a
    EHealthDataGroup, a GroupRequest to this group is
    automatically accepted
127 fact GroupDataPrivacy {
128     all r: GroupRequest | #r.groupData.data > 3 implies
        r.approved = True else r.approved = False
129 }
130
131 -- If a ThirdParty have access to a group of data then there
    is an approved request for them
132 fact GroupDataRequest {

```

```

133   all t: ThirdParty | all g: EHealthDataGroup | g in
      t.groupData implies (one r: GroupRequest | r in
      t.request and r.groupData = g and r.approved = True)
134 }
135
136 -- If a GroupRequest is accepted, then the ThirdParty have
      access to EHealthDataGroup
137 fact {
138   all t: ThirdParty | all r: GroupRequest | r in t.request
      and r.approved = True implies r.groupData in
      t.groupData
139 }
140
141 -- No multiple GroupRequest for the same group from the same
      ThirdParty
142 fact {
143   all t: ThirdParty | no disj r1, r2: GroupRequest | r1 in
      t.request and r2 in t.request and r1.groupData =
      r2.groupData
144 }
145
146
147
148
149 pred ambulanceWorld {
150   some i: Individual | #i.illness > 0 and some a: Ambulance
      | a.rescuee = i and a.startTime.time >
      i.illness.startTime.time
151 }
152
153 run ambulanceWorld for 4 but 1 Ambulance, 1 Individual, 1
      Illness, 5 Int
154
155
156 pred groupRequestWorld {
157   some r1, r2: GroupRequest | r1.approved = True and
      r2.approved = False
158 }
159
160 run groupRequestWorld for 6 but 2 Request, 1 ThirdParty, 5
      Int
161
162
163 pred specificRequestWorld {
164   some r1, r2: SpecificRequest | r1.approved = True and
      r2.approved = False
165 }
166

```

```

167 run specificRequestWorld for 5 but 2 Request, 2 Individual, 1
    ThirdParty, 2 EHealthData, 0 EHealthDataGroup, 5 Int
168
169 assert privacyMaintained {
170     no d: EHealthDataGroup | some t: ThirdParty | d in
        t.groupData and #d.data <= 3
171 }
172
173 check privacyMaintained for 15
174
175 assert monitorAccepted {
176     no i: Individual | some t: ThirdParty | i in t.monitor
        and no r: SpecificRequest | r.approved = True and r in
        t.request
177 }
178
179 check monitorAccepted for 15
180
181 assert ambulanceCalling {
182     no i: Individual | #i.illness > 0 and no a: Ambulance |
        a.rescuee = i and a.location = i.location
183 }
184
185 check ambulanceCalling for 15
186
187 assert ambulanceReactionTime {
188     no i: Individual | #i.illness > 0 and some a: Ambulance |
        a.rescuee = i and a.location = i.location and
        minus[a.startTime.time,
        a.rescuee.illness.startTime.time] > 2
189 }
190
191 check ambulanceReactionTime for 15

```

4.2 Worlds generated

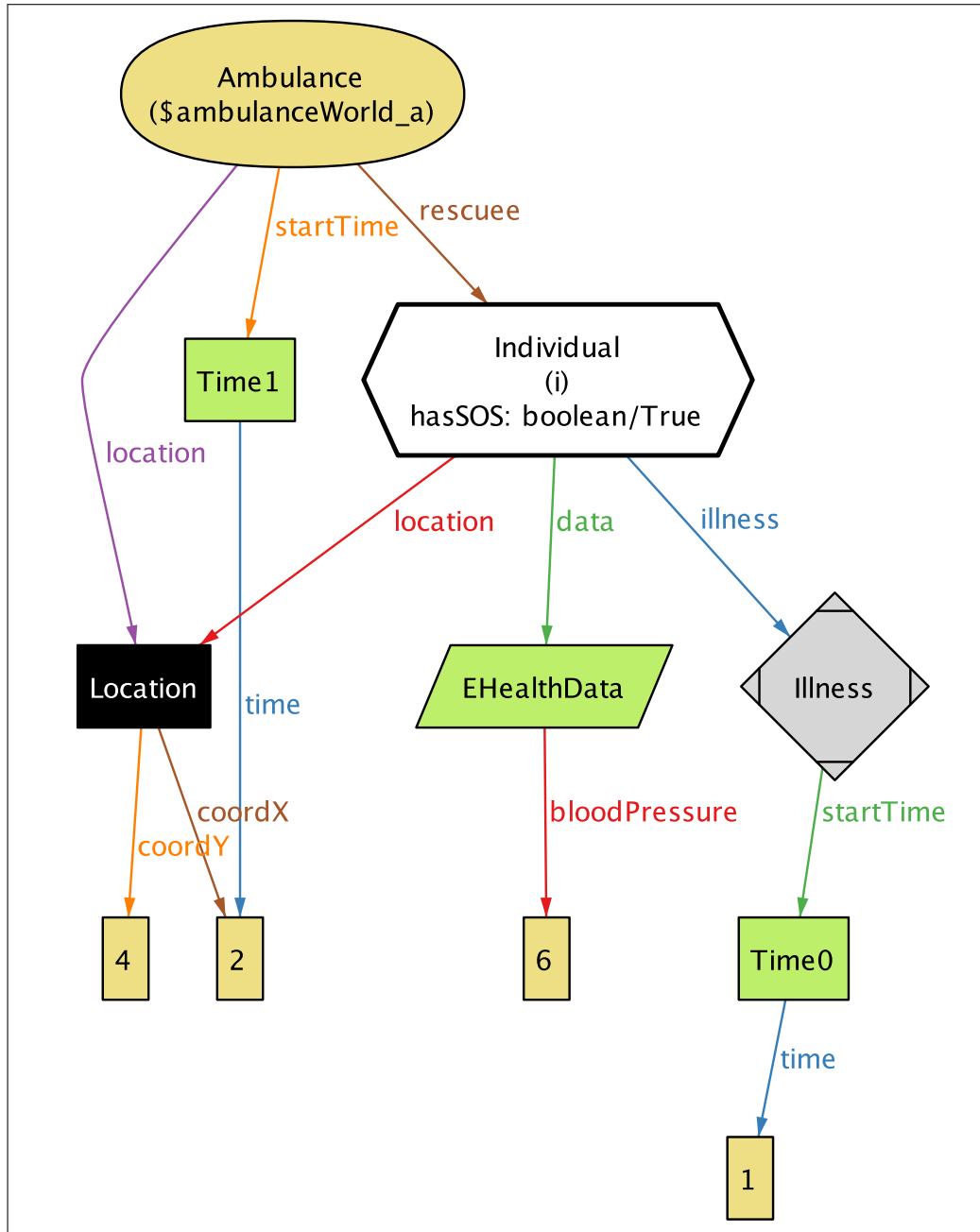
4.2.1 Ambulance

In Figure 4.1, it is shown that an ambulance is sent to the location of an Individual that he doesn't feel good. Note that it can be there a (bounded) delay between the moment in which the vital parameters cross the thresholds (start of the Illness) and the time in which the ambulance is alerted.

```

run ambulanceWorld for 4 but 1 Ambulance, 1 Individual, 1
    Illness, 5 Int

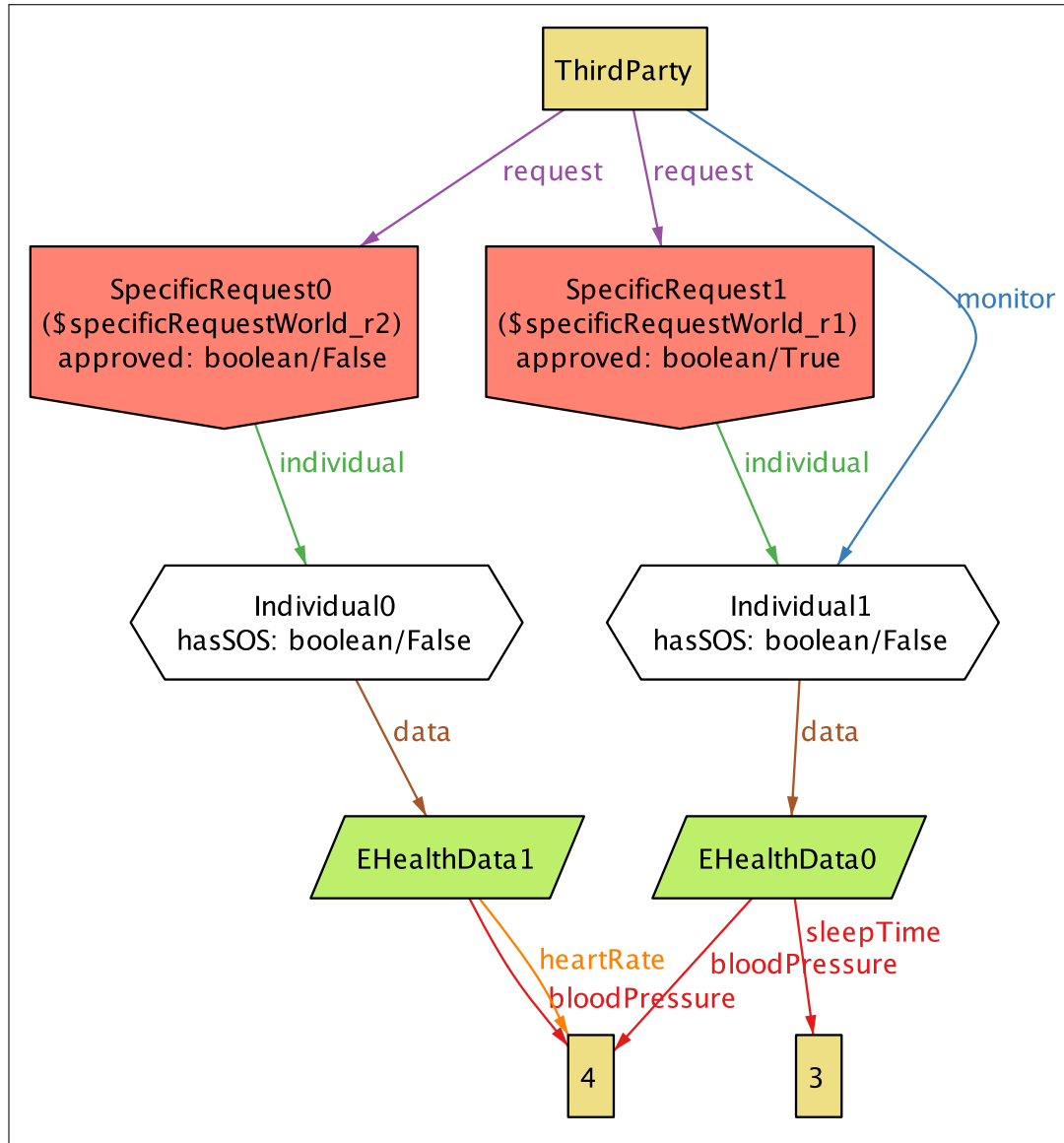
```


Figure 4.1: World generated by `ambulanceWorld` predicate

4.2.2 Specific requests

In Figure 4.2, it is shown a case in which a third party had sent two distinct specific request but only one is being accepted. Note that the third party can effectively monitor only the individual who has accepted the requests.

```
run specificRequestWorld for 5 but 2 Request, 2 Individual, 1
  ThirdParty, 2 EHealthData, 0 EHealthDataGroup, 5 Int
```

Figure 4.2: World generated by `specificRequestWorld` predicate

4.2.3 Group requests

Similarly to the previous, Figure 4.3 shows a third party that had forwarded two group data requests. Note that only one respect the privacy constraint and therefore the third party can access only to one group data.

```
run groupRequestWorld for 6 but 2 Request, 1 ThirdParty, 5
Int
```

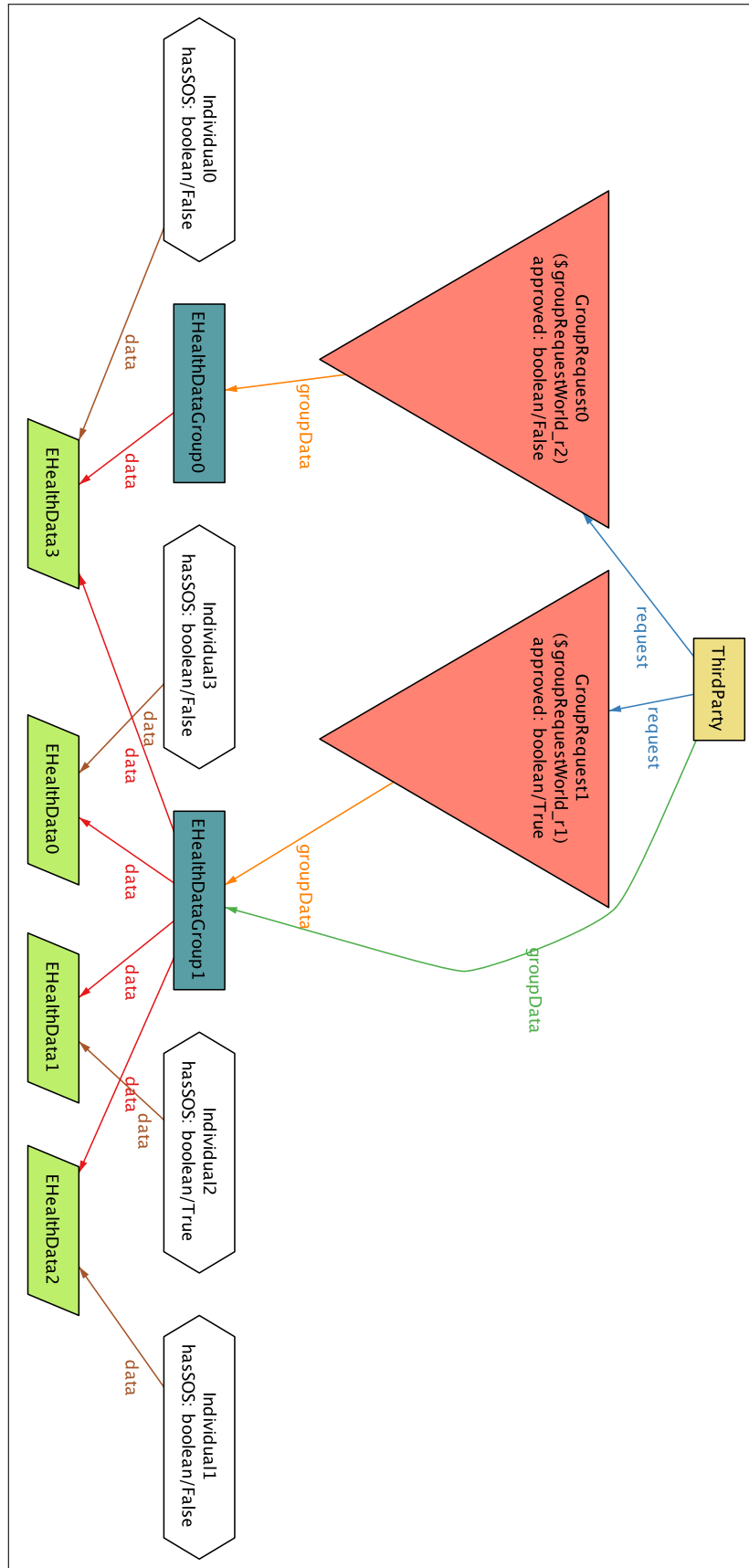
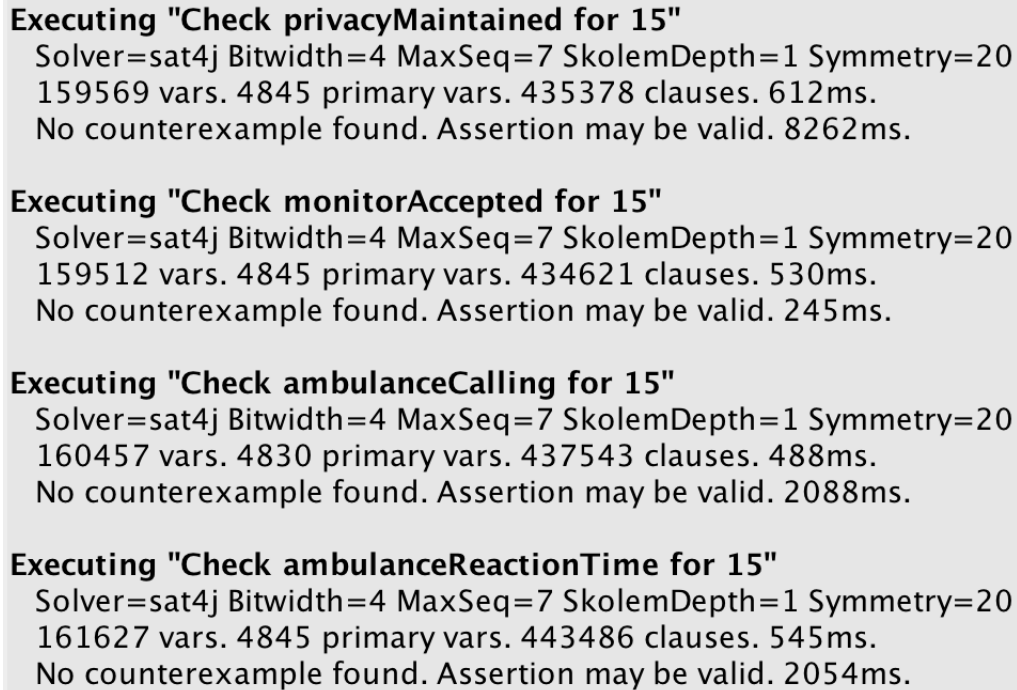


Figure 4.3: World generated by `groupRequestWorld` predicate

4.3 Alloy results

The checking of the four assertions defined in the model produce the results shown in Figure 4.4

The image shows a screenshot of Alloy results for four assertions. Each assertion is listed with its name, the solver used (sat4j), and various parameters (Bitwidth, MaxSeq, SkolemDepth, Symmetry). The results for each assertion are: 1. "Check privacyMaintained for 15": 159569 vars, 4845 primary vars, 435378 clauses, 612ms. 2. "Check monitorAccepted for 15": 159512 vars, 4845 primary vars, 434621 clauses, 530ms. 3. "Check ambulanceCalling for 15": 160457 vars, 4830 primary vars, 437543 clauses, 488ms. 4. "Check ambulanceReactionTime for 15": 161627 vars, 4845 primary vars, 443486 clauses, 545ms. All assertions resulted in "No counterexample found. Assertion may be valid." with the total time for each assertion.

```

Executing "Check privacyMaintained for 15"
  Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
  159569 vars. 4845 primary vars. 435378 clauses. 612ms.
  No counterexample found. Assertion may be valid. 8262ms.

Executing "Check monitorAccepted for 15"
  Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
  159512 vars. 4845 primary vars. 434621 clauses. 530ms.
  No counterexample found. Assertion may be valid. 245ms.

Executing "Check ambulanceCalling for 15"
  Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
  160457 vars. 4830 primary vars. 437543 clauses. 488ms.
  No counterexample found. Assertion may be valid. 2088ms.

Executing "Check ambulanceReactionTime for 15"
  Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
  161627 vars. 4845 primary vars. 443486 clauses. 545ms.
  No counterexample found. Assertion may be valid. 2054ms.

```

Figure 4.4: Checking of the four assertions

Chapter 5

Effort spent

In the following tables the time spent for each section of the project are presented, the specific requirements section is analyzed in details:

Stefano Pecchia

Section	Hours
Introduction	6
Overall description	4
External interfaces	8
Scenarios	1
Functional requirements	8
Non-functional requirements	1
Alloy	1

Edoardo Peretti

Task	Hours
Introduction	1
Overall description	4
External interfaces	0.5
Scenarios	1
Functional requirements	2
Non-functional requirements	0.5
Alloy	17

Chapter 6

References

- Specification document "Mandatory Project Assignment AY 2018-2019.pdf"
- ISO/IEC/IEEE 29148 - Standard on requirement engineering