Data4Help

Requirements Analysis and Specification Document
V1.1

Davide Damato Luciano Franchin

January 14, 2019

POLITECNICO
MILANO 1863

Contents

1 Introduction

1.1 Purpose

This document represents the Requirements Analysis ad Specification Document (RASD). The purpose of this document is to specify the functional and non-functional requirements for the development of both Data4Help and AutomatedSOS. The document is aimed to explain every shared phenomena between the world and the system, such that stakeholders and developers requirements will meet unanimously.

1.2 Scope

1.2.1 Current system

Data4Help is a new service that has to be implemented from the foundation in a new environment that has yet to be completely defined. From the problem definition we can state that there are three main participants: public users, third parties companies and first aid services. It is not specified whether or not the service Data4Help will interact with other software applications to obtain new data; in order to keep every possibility open, this project will consider the chance to develop a common interface to share data between Data4Help and external application, but it will not be considered a mandatory requirement.

1.2.2 Description of the given problem

Data4Help is a service aimed to collect user data, the system has to be developed with a distributed registration interface along with hardware devices able to monitor health status. All the data acquired from the users has to be available for third parties given that the user privacy will not be armed. In order to establish an agreement between the user and TrackMe, the user has to accept the terms of use during registration.
Third parties have to register for the usage of Data4Help, once registered they will be able to request data through a browser based interface or an application to install on computer. Every data request has to be accepted by the system before being delivered, by first agreement with the stakeholders a general query has to select at least 1000 user, otherwise, the third party has to have the social security number of the user to access his/her personal data.
AutomatedSOS is a service targeted to elderly people but exceptions may apply. The system will be designed to integrate as much as possible with his main target user, but it may be adaptable to guarantee flexibility and re-usability. The main goal of this service is to guarantee a quick response to emergency situation regarding health issues. In order to make this possible the system will be developed to guarantee a response time of less than 5 seconds to activate the emergency call. AutomatedSOS guarantees that the first aid service will be called in the shortest time possible within agreed time slots. The service will be developed in order to guarantee the accomplishment of the goal from start to finish but will not guarantee the accomplishment of the rescue service's tasks.

1.2.3 Goals

[**G1**]**:** Collect user data.

- [**G1.1**]**:** Allow the user to insert identification data and health data on an application interface.
- [**G1.2**]**:** User informations must be securely stored.

[**G2**]**:** Allow Third Parties to receive data collections.

- [**G2.1**]**:** Allow third parties to select query's parameters.

- [**G2.2**]**:** Allow third parties to select a query even before the data is collected.
- [**G2.3**]**:** Ensure single users' privacy on third party queries.
- [**G2.4**]**:** Ensure group query anonimization on third party queries.

[**G3**]**:** Allow a user to monitor his own parameters.

- [**G3.1**]**:** Allow an user to examine his/her data from previous periods.
- [**G3.2**]**:** Allow an user to see immediately his/her current data.

[**G4**]**:** Allow a user to receive first aid in emergency situations.

- [**G4.1**]**:** Rescue service must be called within 5 seconds from emergency detection.
- [**G4.2**]**:** Allow an user to be exactly located on a map.
- [**G4.3**]**:** Allow first aid service to know user's health data.
- [**G4.4**]**:** Allow the user to get the quickest and nearest rescue service.
- [**G4.5**]**:** Allow the user to be found by activating sounds or lights of his devices.

[**G5**]**:** Allow a system manager to perform updates for maintainability.

- [**G5.1**]**:** System managers should be able to add new First Aid services.
- [**G5.2**]**:** System managers should be able to add new system managers.
- [**G5.3**]**:** System managers should be able to perform any kind of query.
- [**G5.4**]**:** System managers should be able to completely delete user's data under request.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- *Health status:* the set of every measurement on the user's body which can mean something useful to monitor his/her condition. Examples: blood pressure, heart rate, body temperature etc.
- *Third party:* any agency or company, external from TrackMe, that wants to use Data4Help's services.
- *Rescue service, first aid service, SOS service:* all synonyms for an actor able to reach a patient and perform first aid operations.
- *GpsDevice:* any detection device that is not connected to GSM network.
- *GsmDevice:* a device that is connected to GSM network. These devices are divided into GsmSmartphones and GsmSmartwatches.
- *GsmSmartphone:* a device connected to GSM network but usually unable to perform body measurements.
- *GsmSmartwatch:* a device connected to GSM network and able to perform body measurements.

1.3.2 Abbreviations

PU: Public User.

TPU: Third Party User.

SSN: Social Security Number.

GPS: Global Positioning System.

GSM: Global System for Mobile communications.

Gn: n-Goal.

Dn: n-Domain assumption.

Rn: n-Functional requirement.

1.4 Revision history

[**V1.0**]**:** First release, lacks of ”Standard compliance” section.

[**V1.1**]**:** Added Politecnico logo, added ”Standard Compliance” section, added new mockups, bug fixed.

1.5 Reference Documents

- Lecture slides.
- Specification Document:
  “Mandatory Project Assignment AY 2018-2019.pdf”.
- `http://www.cs.toronto.edu/~sme/CSC340F/readings/FoRE-chapter02-v7.pdf`
- `https://en.wikipedia.org/wiki/Requirements_engineering`
- 830-1993 - IEEE Recommended Practice for Software Requirements Specifications `https://ieeexplore.ieee.org/servlet/opac?punumber=3114`

1.6 Document Structure

This RASD is composed by 6 parts:

1. **Introduction:** The first section focuses on the document high level functions, on a generic description of the problem and the environment in which the system will operate.
2. **Overall description:** The second section consists of an overall description of the system. In this part there will be explained more details on the project functions and how they are considered in the system. To explain these aspects, this section underlines the actors and the system boundaries involved in the system's usage life cycle.
3. **Specific requirements:** The third part is composed by the specific requirements, both functional and non functional. This section contains some mock-ups of the user interfaces and the most common use cases.
4. **Formal analysis using Alloy:** Fourth part is embodied with the Alloy model of the system and includes all the relevant details; a proof of consistency and an example of the generated world are also provided.

5. **Effort spent:** Fifth part contains a detailed report of the hours spent to do so.
6. **References:** Sixth part lists all the tools used to redact this document.

2 Overall Description

2.1 Product perspective

The main actor in the software-to-be will be represented by the PUs. Users in general will occupy the major part in the database structure, with a special regard to PUs that will be the main subject for third parties queries. Any user that wants to benefit of AutomatedSOS services must be connected to at least a GsmDevice.
Third party queries are examined by a controller and then delivered to the sender if accepted.

2.1.1 Class diagram

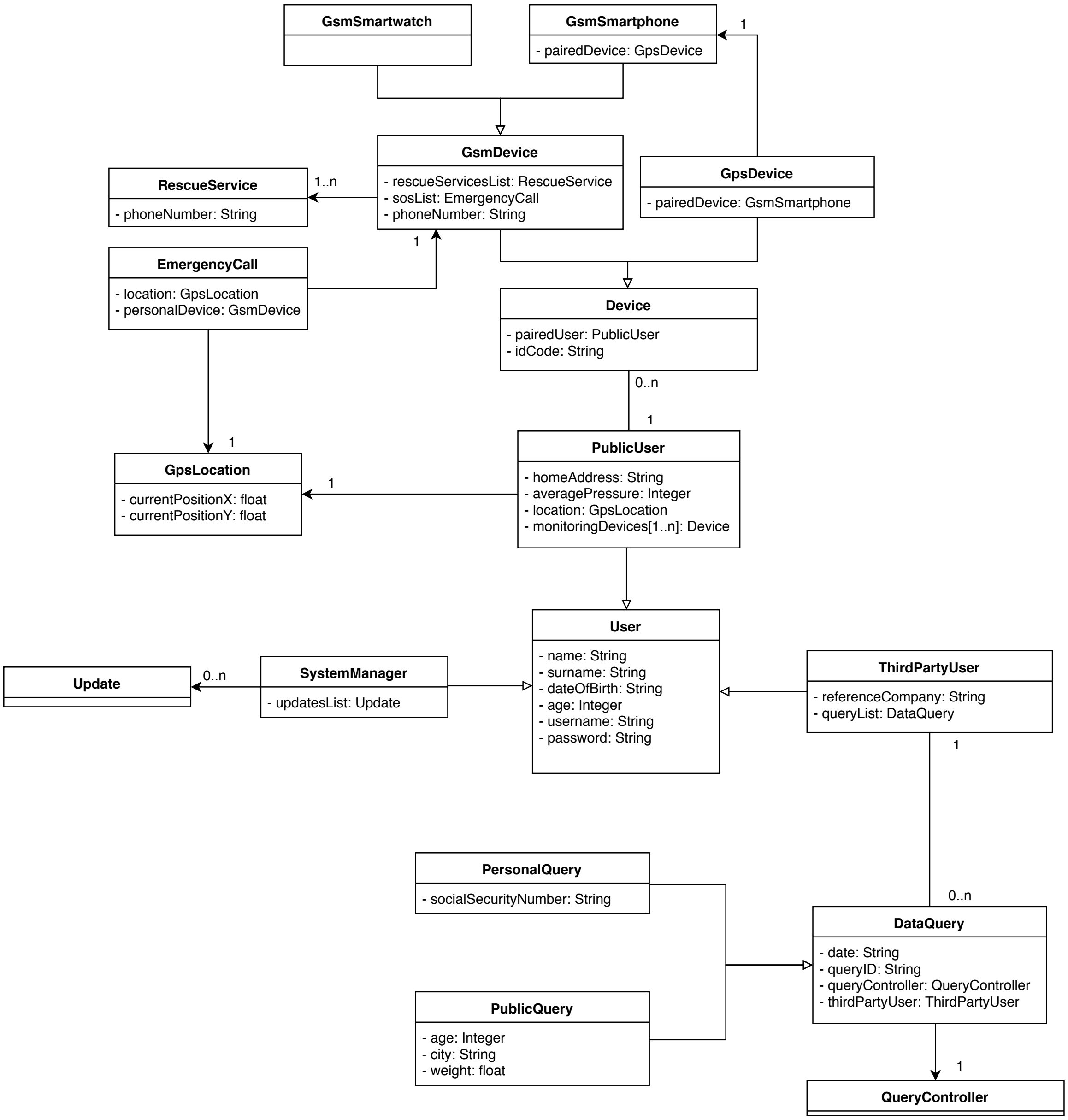


Figure 1: Class diagram

2.1.2 Statechart

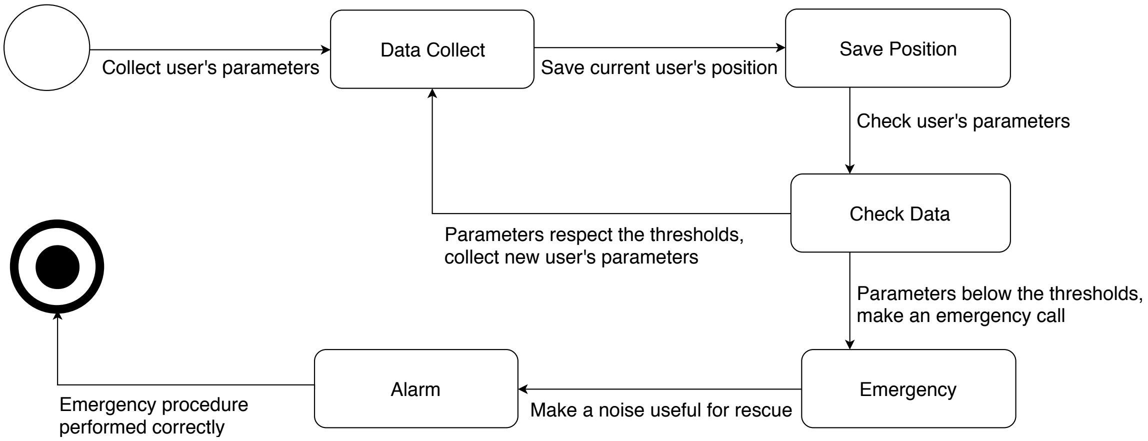

Data Collect
Collect user's parameters
Save current user's position
Save Position
Check user's parameters
Check Data
Parameters respect the thresholds, collect new user's parameters
Parameters below the thresholds, make an emergency call
Emergency
Make a noise useful for rescue
Alarm
Emergency procedure performed correctly

Figure 2: Emergency state diagram

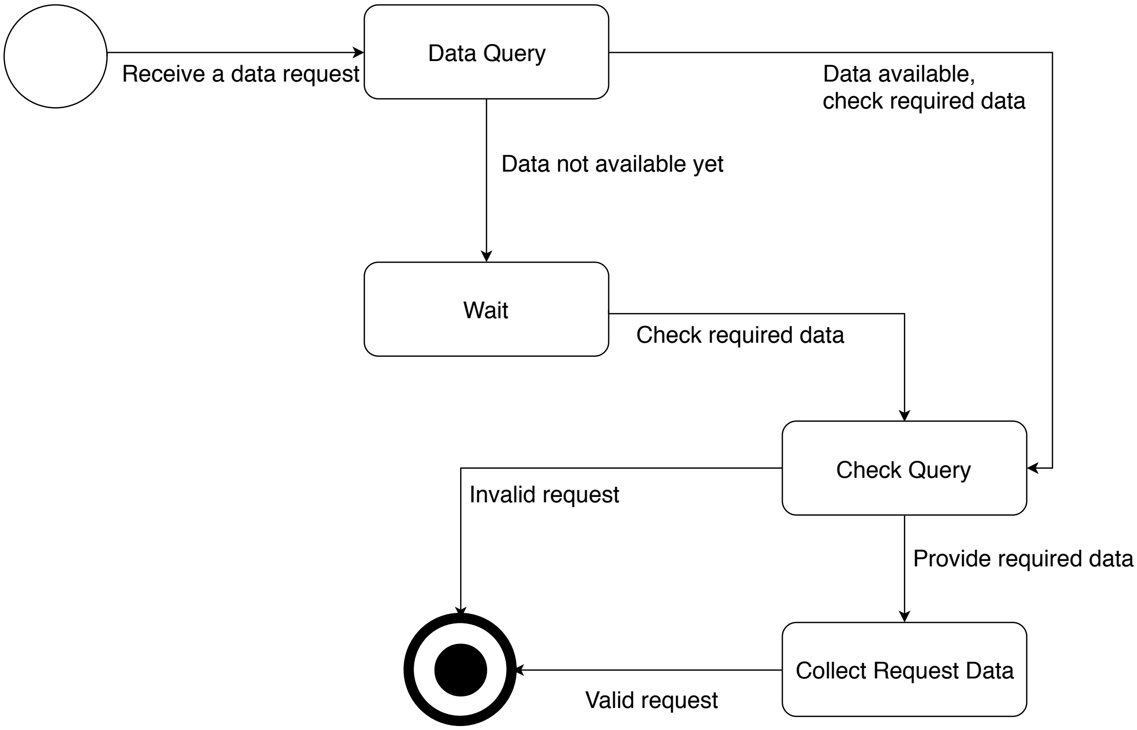


Figure 3: Third party state diagram

2.2 Product functions

Considering the "Data4Help" goals described above, the main two functions can be listed with more accuracy.

2.2.1 Data management

Data management is an important function for various aspects within Data4Help. All the user's data are collected in a database, allowing the system to update or extract easily desired data. Data management is essential for using Data4Help both for users and for third party. Users will be able to look and analyze their previous health data and update personal data which no longer correspond to those previously entered. Instead third party representative user will be able to require a data collection or a single data, searched by fiscal code for example. Data management includes data checks to analyze requests' validity, as in the case of less than a thousand people in a certain area, and after data checks the third party representative user will receive an answer with required data or with an error message.

2.2.2 Health analysis management

Health analysis management is the principal requirement for AutomatedSOS. The user must wear the smartwatch as much as possible for the correct behavior of AutomatedSOS. The system will be able to collect user's health data steadily and compare them with default thresholds, based on user characteristics set by the system at the time of registration, or with a calculated thresholds based on user's past data. If user's health data is found lower than the thresholds the system must be able to make an emergency call within 5 seconds.

2.3 User characteristics

- Public User: every Data4Help user that is registered to use AutomatedSOS or other applications based on Data4Help. A database instance of public user must not figure at the same time as third party user or system manager.
- Third party user: every Data4Help user registered to use the query service on public users. A database instance of third party user must not figure at the same time as public user or system manager.
- System manager: an employee of TrackMe in charge of maintaining and updating the Data4Help system. These users are not able to register themselves to the system but must be inserted by another system manager. This role is necessary to guarantee maintainability of the system even if the aim is to automatize every procedure. A database instance of system manager must not figure at the same time as public user or third party user.
- Hardware Device: every electronic device that is capable of giving health information about the user it is attached to. Since the main goal of this project is to analyze the software specifics, this actor will be as general as possible and its technicalities won't be analyzed.

2.4 Assumptions, dependencies and constraints

[D1]: Every user has a Social Security number or equivalent to be uniquely identified.

[D2]: Once a detection device is paired with an user it will not be used by anyone else, if so, a disconnection procedure is necessary.

[D3]: Every user must protect his/her device from any ill-intentioned and guarantee consistent personal information.

[D4]: Every user must protect his/her personal information to access his/her account.

[D5]: Every user is mentally capable to administrate his/her Data4Help profile, if not, a trusted person will guarantee for him/her and administrate the user profile.

[D6]: Every GsmDevice or GpsDevice can be uniquely identified with a code. (IMEI for example).

[D7]: Detection device is attached to the user body when active so that measurements are valid.

[D8]: Detection device is used properly by the user: regularly charged and worn as much as possible.

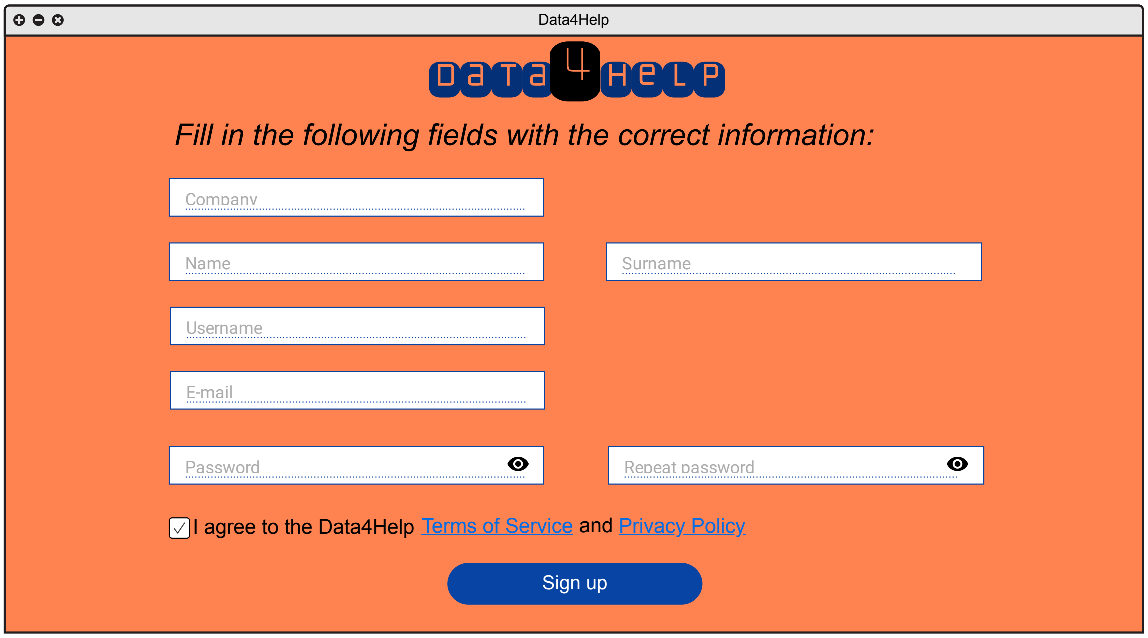
[D9]: Any first aid service that proposes for AutomatedSOS service is sufficiently qualified and will give all the certifications or documentation needed.

[D10]: Every qualified first aid service will have a communication channel to gather emergency information from AutomatedSOS.

3 Specific requirements

3.1 External interface requirements

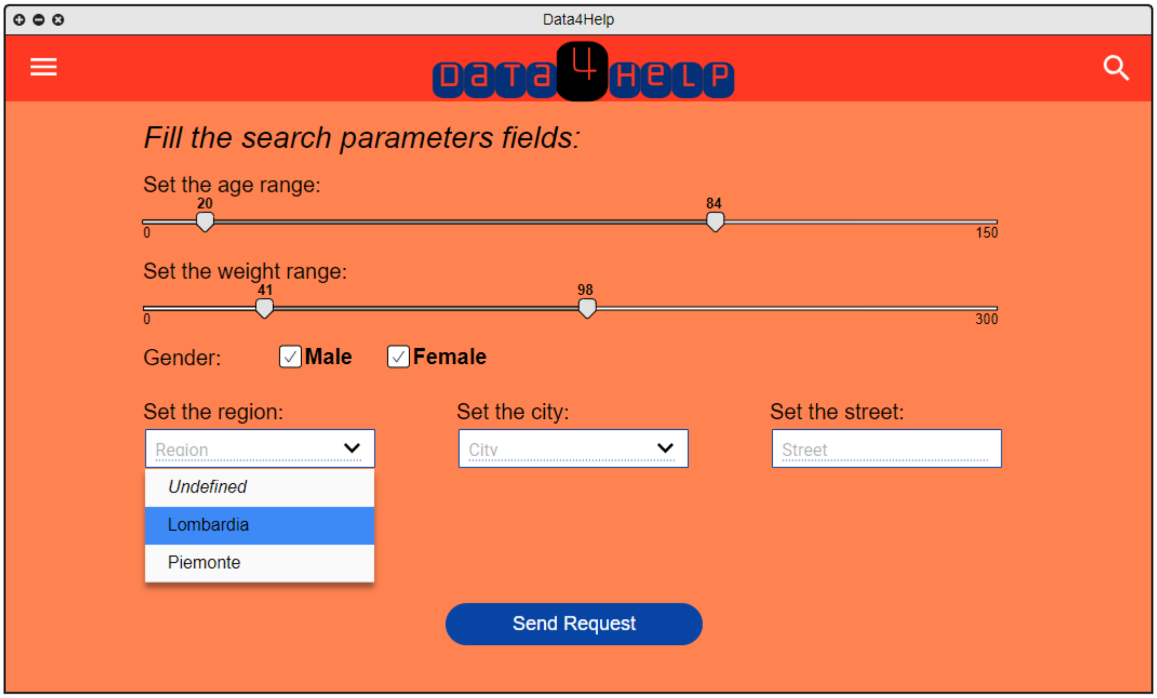
3.1.1 User Interfaces



Fill in the following fields with the correct information:

☑ I agree to the Data4Help Terms of Service and Privacy Policy

Figure 4: Mock up - Data4Help registration page.



Fill the search parameters fields:

Set the age range: 0 — 20 — 84 — 150

Set the weight range: 0 — 41 — 98 — 300

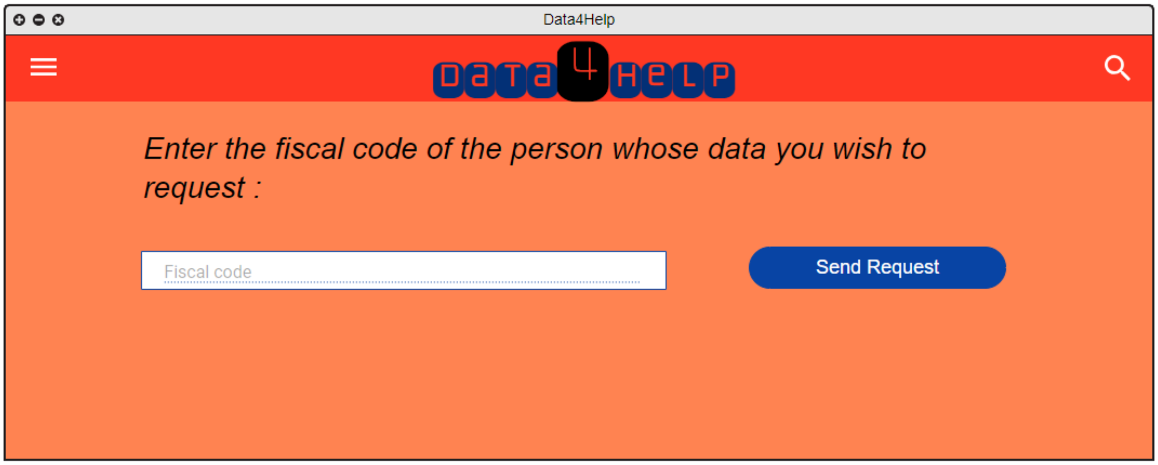
Gender: ☑ **Male** ☑ **Female**

Set the region: Region
- *Undefined*
- Lombardia
- Piemonte

Set the city: City

Set the street: Street

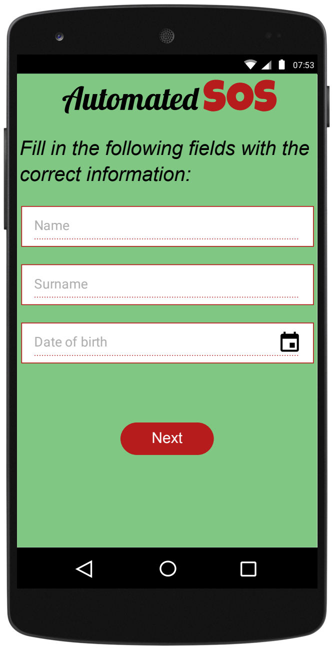
Figure 5: Mock up - Group query page.



Enter the fiscal code of the person whose data you wish to request :

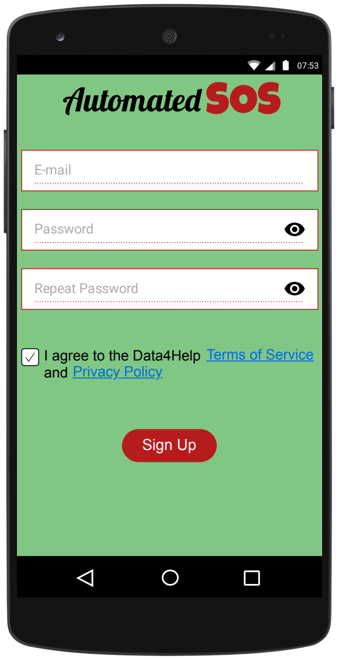
Fiscal code

Figure 6: Mock up - Personal query page.



Fill in the following fields with the correct information:

Figure 7: Mock up - AutomatedSOS first registration page.



☑ I agree to the Data4Help Terms of Service and Privacy Policy

Figure 8: Mock up - AutomatedSOS second registration page.

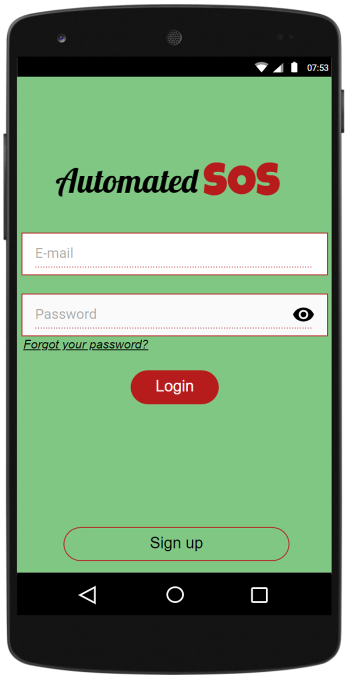
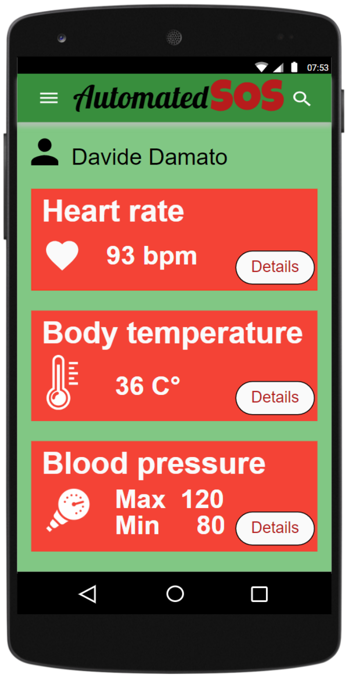


Figure 9: Mock up - AutomatedSOS login page.



Davide Damato

Heart rate: 93 bpm

Body temperature: 36 C°

Blood pressure: Max 120, Min 80

Figure 10: Mock up - AutomatedSOS health status on smartphone.

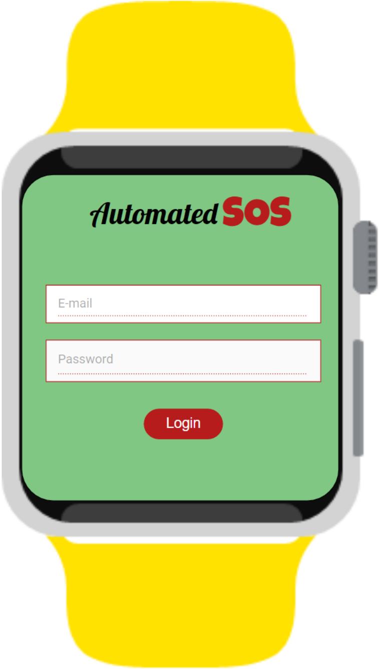


Figure 11: Mock up - Login for Smartwatch.

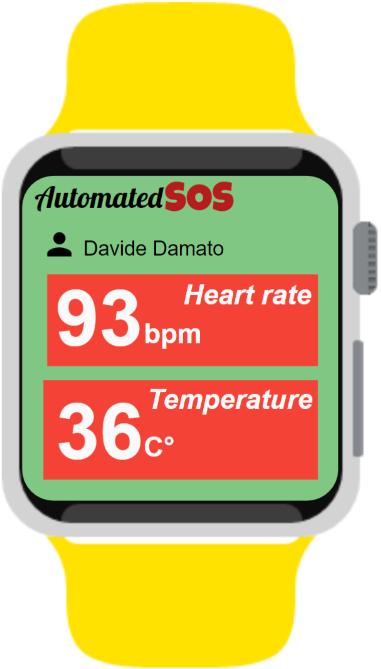


Figure 12: Mock up - Health status.

3.1.2 Hardware Interfaces

AutomatedSOS is an application that requires use of smartphones and smartwatch. The smartwatch must be able to acquire user's health parameters and if it can't support installation of AutomatedSOS then it must be connected to smartphone with bluetooth to communicate only calculated parameters, eventually emergency call will be made by AutomatedSOS installed in the smartphone. If instead the smartwatch can download AutomatedSOS then there is no need for a bluetooth connection with the phone, the smartwatch will be able to analyze user's parameters and make an emergency call autonomously. Obviously, in both cases, the device with AutomatedSOS installed must have Internet and GPS enabled.

3.1.3 Software Interfaces

Data4Help application includes external softwares to improve performance and to lighten the application's structure.

- **GPS service**
  Both for smartwatch and for smartphone, Data4Help uses the GPS service inside the device. For this reason it is necessary to keep the GPS on, whatever the device is. Without GPS, Data4Help can't perform well the principle function.

- **Health parameters calculation service**
  AutomatedSOS works only with smartwatches that can acquire user's health data and can send this data via bluetooth. This feature is essential for the data analysis and comparison with the threshold values.

3.1.4 Communication Interfaces

The application uses the usual communication protocols for the data analysis. The SOS status occurs when the health parameters analysis detects differences with the thresholds. The device able to make an emergency call send a SOS message to rescue service. This message contains all the user's health parameters and the exactly GPS position. The first aid service must be equipped with the necessary device to receive this type of emergency call, this device has a display in which the user's information will be shown and has a speaker that will emit a prolonged sound when the call is received.

3.2 Functional requirements

[**G1**]**:** Collect user data.

- [**R1**]**:** The system must allow user to register and insert all the needed information.
- [**R2**]**:** The system must be able to acquire user's health data.
- [**R3**]**:** The system must be able to save and store safely all user's data.

- [**D1**]**:** Every user has a Social Security number or equivalent to be uniquely identified.
- [**D4**]**:** Every user must protect his/her personal information to access his/her account.

[**G2**]**:** Allow Third Parties to receive data collection.

- [**R4**]**:** TPU must be able to login to the system using their credentials.
- [**R5**]**:** TPU must be able to select the desidered parameters to start a query.

- [**D2**]**:** Once a detection device is paired with an user it will not be used by anyone else, if so, a disconnection procedure is necessary.
- [**D3**]**:** Every user must protect his/her device from any ill-intentioned and guarantee consistent personal information.

[**G3**]**:** Allow a user to monitor his own parameters.

- [**R6**]**:** User must be able to login to the AuomatedSOS system using their credentials.
- [**R7**]**:** User must be able to analyze and monitor their health parameters.

- [**D4**]**:** Every user must protect his/her personal information to access his/her account.
- [**D6**]**:** Every GsmDevice or GpsDevice can be uniquely identified with a code. (IMEI for example).

[**G4**]**:** Allow a user to receive first aid in emergency situations.

- [**R8**]**:** User must be able to login to the AuomatedSOS system using their credentials.
- [**R9**]**:** User must receive first aid in emergency situations when health parameters are lower than threshold values.

- [**D5**]**:** Every user is mentally capable to administrate his/her Data4Help profile, if not, a trusted person will guarantee for him/her and administrate the user profile.
- [**D7**]**:** Detection device is attached to the user body when active such that measurements are valid.
- [**D8**]**:** Every qualified first aid service will have a communication channel to gather emergency information from AutomatedSOS.

[D9]: Any first aid service that proposes for AutomatedSOS service is sufficiently qualified and will give all the certifications or documentation needed.

[D10]: Every qualified first aid service will have a communication channel to gather emergency information from AutomatedSOS. (Telephone exchange personal for example)

3.2.1 Scenarios

[S1]: A PU gets to know about Data4Help and decides to use its services. Once the user has downloaded the app or has reach the web site he/she registers providing all the necessary information and accepting the terms of use. If any of this steps is not properly completed the PU will not be registered on Data4Help databases and he/she will not be able to use Data4Help services.

[S2]: A registered PU wears his GpsDevice on which AutomatedSOS is installed. Since GpsDevice is not able to make emergency calls, it needs to be paired with a GsmSmartphone able to make emergency calls. The two devices are paired through bluetooth connection and data is transferred from GpsDevice through GsmSmartphones to the Data4Help databases.

[S3]: A TPU gets to know about Data4Help and decides to use its services. Once the TPU reach the web site he/she registers providing all the necessary information and accepting the terms of use. If any of this steps is not properly completed the TPU will not be registered on Data4Help databases and he/she will not be able to use Data4Help services.

[S4]: A TPU connects to the group query web page, he/she selects all his desired options for his/her query and sends it to the system. The system analyzes the query, if it satisfies the minimum requirements then sends the results to the TPU.

[S5]: A TPU connects to the personal query web page, he/she enters a SSN. The system sends a notification to the selected PU to ask for TPU query permissions. If PU grants permission then all his/her data are sent to TPU, otherwise TPU will be notified of denied query.

3.2.2 Use case diagram

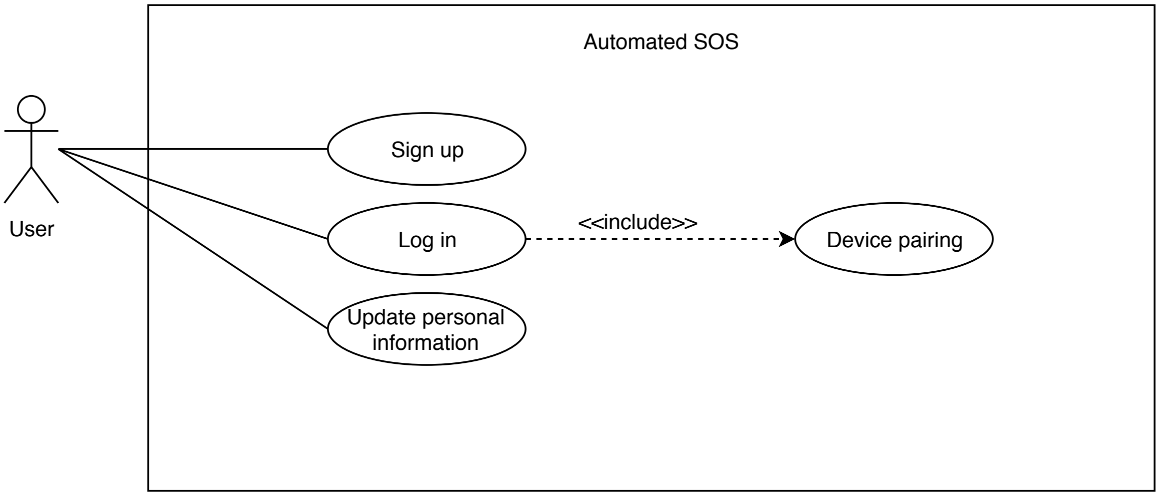


Figure 13: User's use case diagram

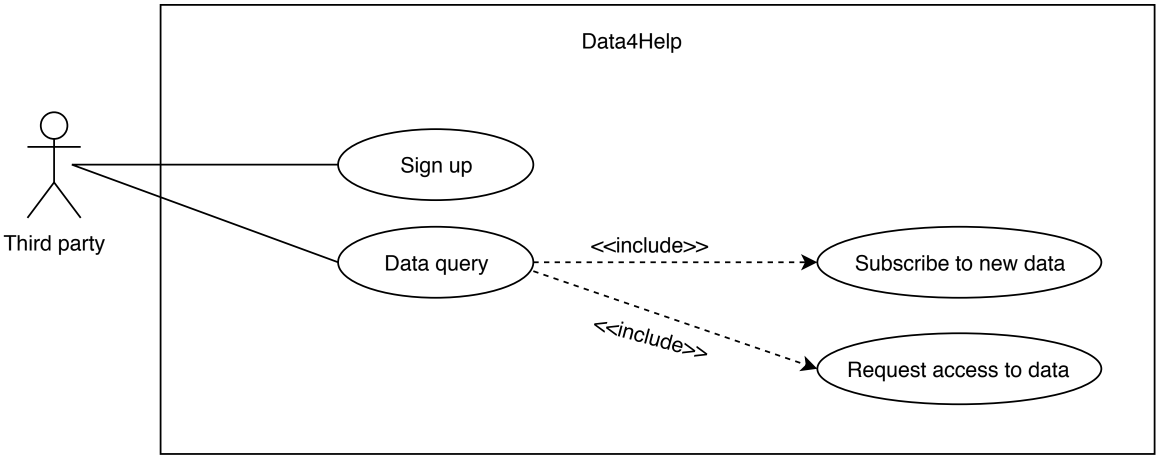


Figure 14: Third party use case diagram

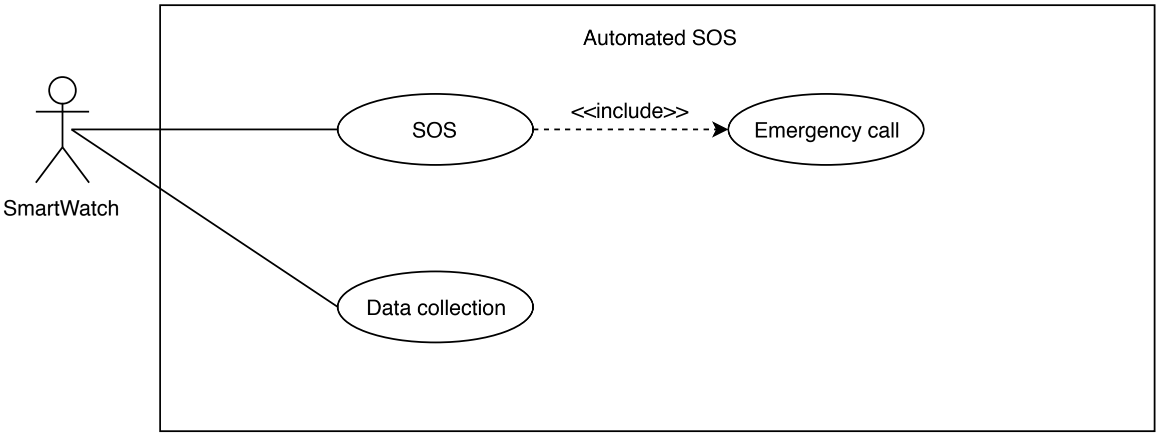


Figure 15: Smartwatch use case diagram

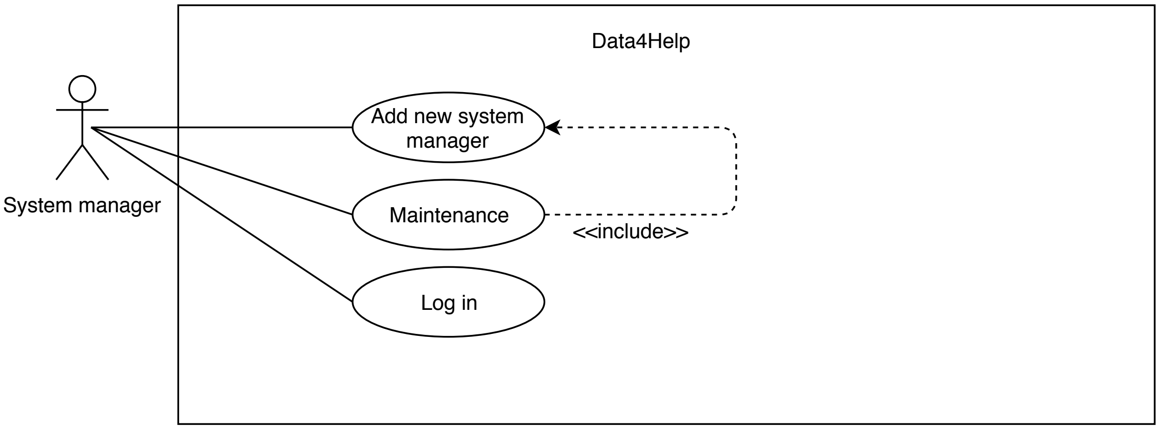


Figure 16: System manager's use case diagram

3.2.3 Use cases

Name	Sign Up
Actor	User
Entry condition	The user has installed the application on his/her device.
Event flow	1. Click on “Sign up” button. 2. Fill in all of the mandatory fields and provide the necessary information. 3. Click on “Confirm” button. 4. The system saves the data.
Exit condition	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 in all of the mandatory fields with valid data. 3. The e-mail is already registered. All the exceptions are handled by notifying the user and taking him back to the sign up activity.

Name	Log in
Actor	User
Entry condition	The user is previously successfully signed up and has the application installed on his/her device
Event flow	1. The user opens the application on his device (smartwatch or smartphone). 2. He enters his credentials in the “Username” and “Password” fields of the home page of “Automated SOS”. 3. The user clicks on the “Log in” button. 4. The user is successfully logged in his/her “Automated SOS” and the system automatically connects the devices.
Exit condition	The user is able to use all the AutomatedSOS functionalities.
Exceptions	1. The user enters invalid Username. 2. The user enters invalid Password. All the exceptions are handled by notifying the user and taking him/her back to the ”Log in” activity.

Name	Update personal information
Actor	User
Entry condition	The user has already logged in.
Event flow	1. The user clicks on the data he wants to modify. 2. Writes the new data. 3. Clicks on ”Confirm” button.
Exit condition	The user has successfully updated the personal information.
Exceptions	1. The user adds incorrect data. All the exceptions are handled by notifying the user and taking him back to the “Update personal information” activity.

Name	Sign up
Actor	Third party
Entry condition	There are no entry conditions.
Event flow	1. Third party representative user connects to Data4Help web page. 2. TPU clicks on “Sign up” button. 3. TPU fills all the mandatory fields and provide the necessary information. 4. TPU clicks on “Confirm” button.
Exit condition	The third party representative 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 e-mail is already registered. All the exceptions are handled by notifying the user and taking him back to the “Sign up” activity.

Name	Data query
Actor	Third party
Entry condition	The third party representative user has already signed up.
Event flow	1. Third party representative user opens the Data4Help web page. 2. TPU goes to the ”data query” page. 3. TPU selects the desired parameters for his/her query. 4. TPU confirms the data request and waits for the results. If the data are not immediately available, TPU will be notified as soon the system can satisfy the query.
Exit condition	The third party representative user has successfully requested a desired data collection.
Exceptions	1. The third party representative user requests an invalid data collection. 2. The data collection request refers to a population lower than 1000 people. All the exceptions are handled by notifying the user and taking him back to the “Data query” activity.

Name	Log in
Actor	System Manager
Entry condition	No entry conditions
Event flow	1. System manager opens the application on his/device. 2. System manager enters his/her credentials in the “Username” and “Password” fields of the home page. 3. System manager clicks on the “Log in” button. 4. System manager is successfully logged in and the system automatically redirects him/her to the options menu.
Exit condition	The system manager is successfully redirected to the options menu.
Exceptions	1. The system manager enters invalid username. 2. The system manager enters invalid Password. All the exceptions are handled by notifying the system manager and taking him/her back to the login activity.

Name	Add new system manager
Actor	System manager
Entry condition	The system manager has already logged in and the system manager to be is an already registered user.
Event flow	1. The system manager opens the options menu. 2. He chooses the option ”Add a new system manager”. 3. He enters the username of the new system manager in the ”Username” field. 4. He clicks on ”Confirm” button 5. The system automatically grant permissions to the new system manager.
Exit condition	The system manager has successfully added a new system manager
Exceptions	1. The system manager enters invalid username. All the exceptions are handled by notifying the user and taking him back to the “Add new system manager” activity.
Special requirements	

Name	Maintenance
Actor	System manager
Entry condition	The system manager has already logged in
Event flow	1. The system manager opens the options menu. 2. The system manager clicks on ”Maintenance” option. 3. The system manager adds the desired system updates. 4. He clicks on ”Save” button.
Exit condition	The system manager has successfully applied the desired updates.
Exceptions	All the exceptions are handled by notifying the user and taking him back to the “Maintenance” activity.

Name	Emergency detection and rescue call
Actor	Smartwatch, Smartphone, Paramedics
Entry condition	The smartwatch must be connected to a smartphone connected to the internet (smartwatch that not support internet connection) or it must have internet connection.
Event flow	1. The smartwatch detects parameters below certain thresholds. 2. If the smartwatch does not supports internet connection, it sends only user's health parameters to the connected smartphone and if parameters represent an emergency situations, then smartphone will make the emergency call in which all informations (GPS position and health parameters) will be communicated. Otherwise the smartwatch makes an emergency call autonomously. 3. The smartwatch will emit a sound useful for rescue.
Exit condition	The rescuers receive all the necessary information and find the person in danger.
Exceptions	

Name	Data collection
Actor	Smartwatch
Entry condition	No entry conditions.
Event flow	1. The smartwatch collect steadily user's health parameters. 2. If the smartwatch not supports AutomatedSOS, it sends the parameters to the connected smartphone. Otherwise it checks and compare it with a determinate thresholds.
Exit condition	The smartwatch has successfully checked user's parameters.
Exceptions	

3.2.4 Sequence diagram

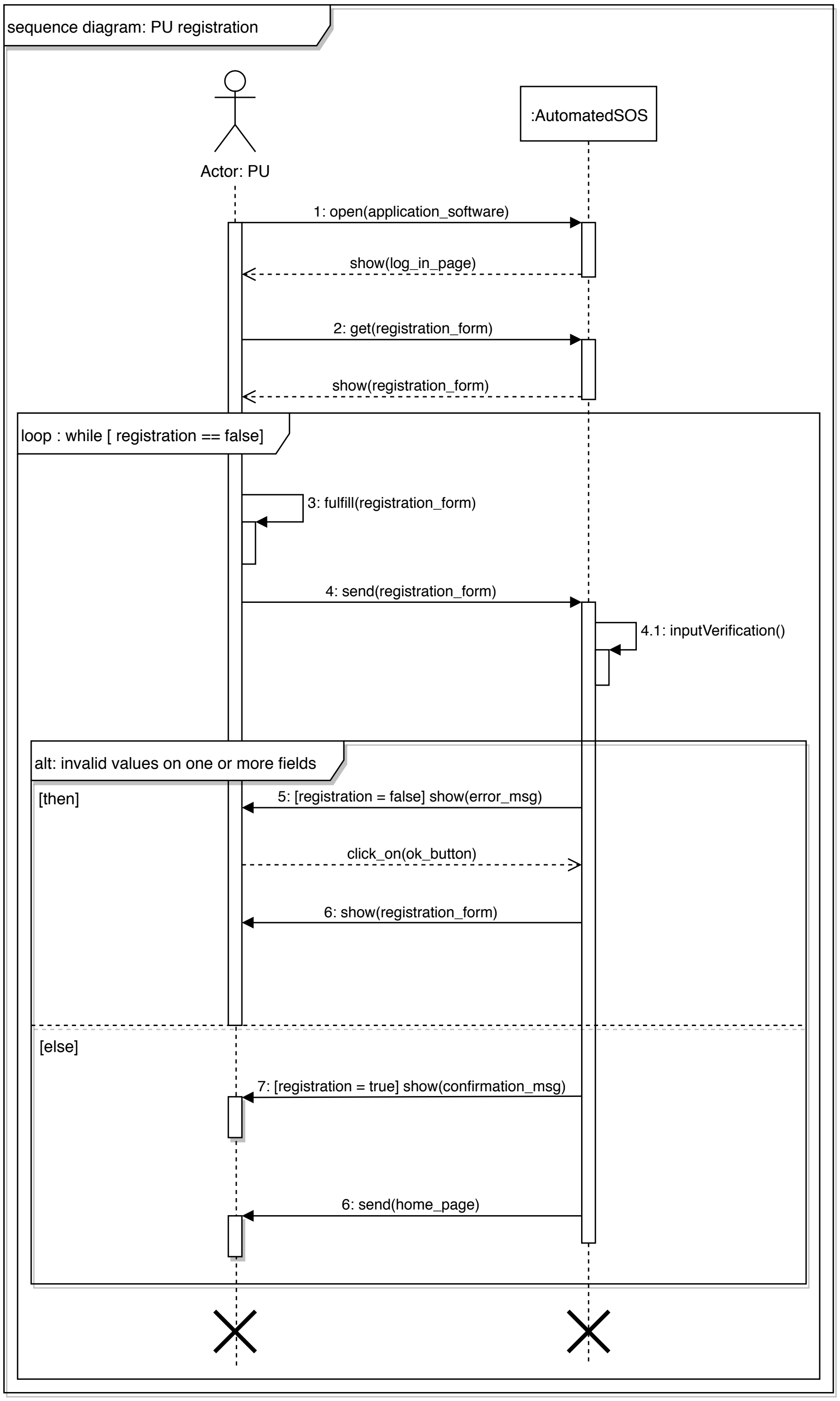

sequence diagram: PU registration
Actor: PU
:AutomatedSOS
1: open(application_software)
show(log_in_page)
2: get(registration_form)
show(registration_form)
loop : while [registration == false]
3: fulfill(registration_form)
4: send(registration_form)
4.1: inputVerification()
alt: invalid values on one or more fields
[then]
5: [registration = false] show(error_msg)
click_on(ok_button)
6: show(registration_form)
[else]
7: [registration = true] show(confirmation_msg)
6: send(home_page)

Figure 17: Public user registration

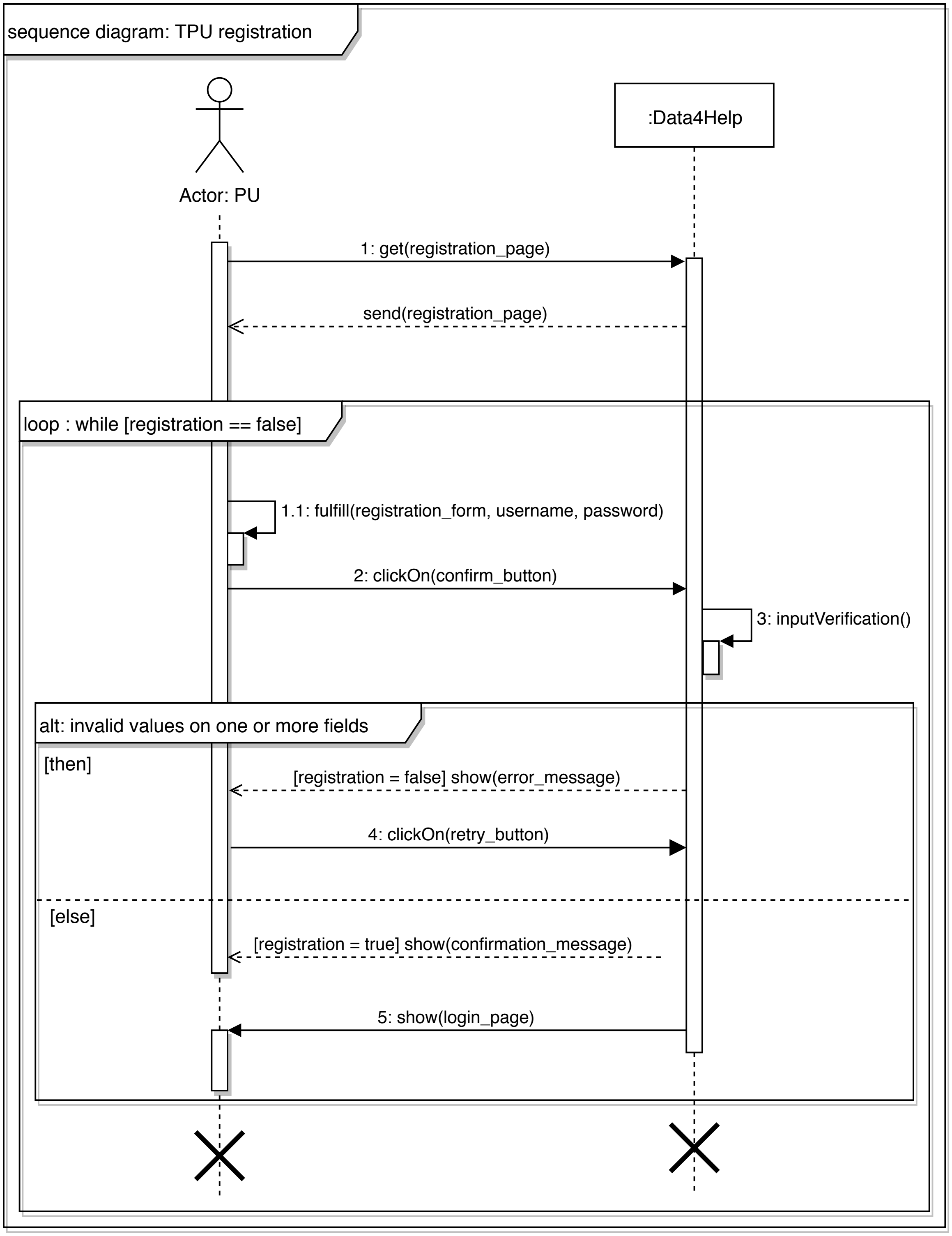

sequence diagram: TPU registration
Actor: PU
:Data4Help
1: get(registration_page)
send(registration_page)
loop : while [registration == false]
1.1: fulfill(registration_form, username, password)
2: clickOn(confirm_button)
3: inputVerification()
alt: invalid values on one or more fields
[then]
[registration = false] show(error_message)
4: clickOn(retry_button)
[else]
[registration = true] show(confirmation_message)
5: show(login_page)

Figure 18: Third party user registration

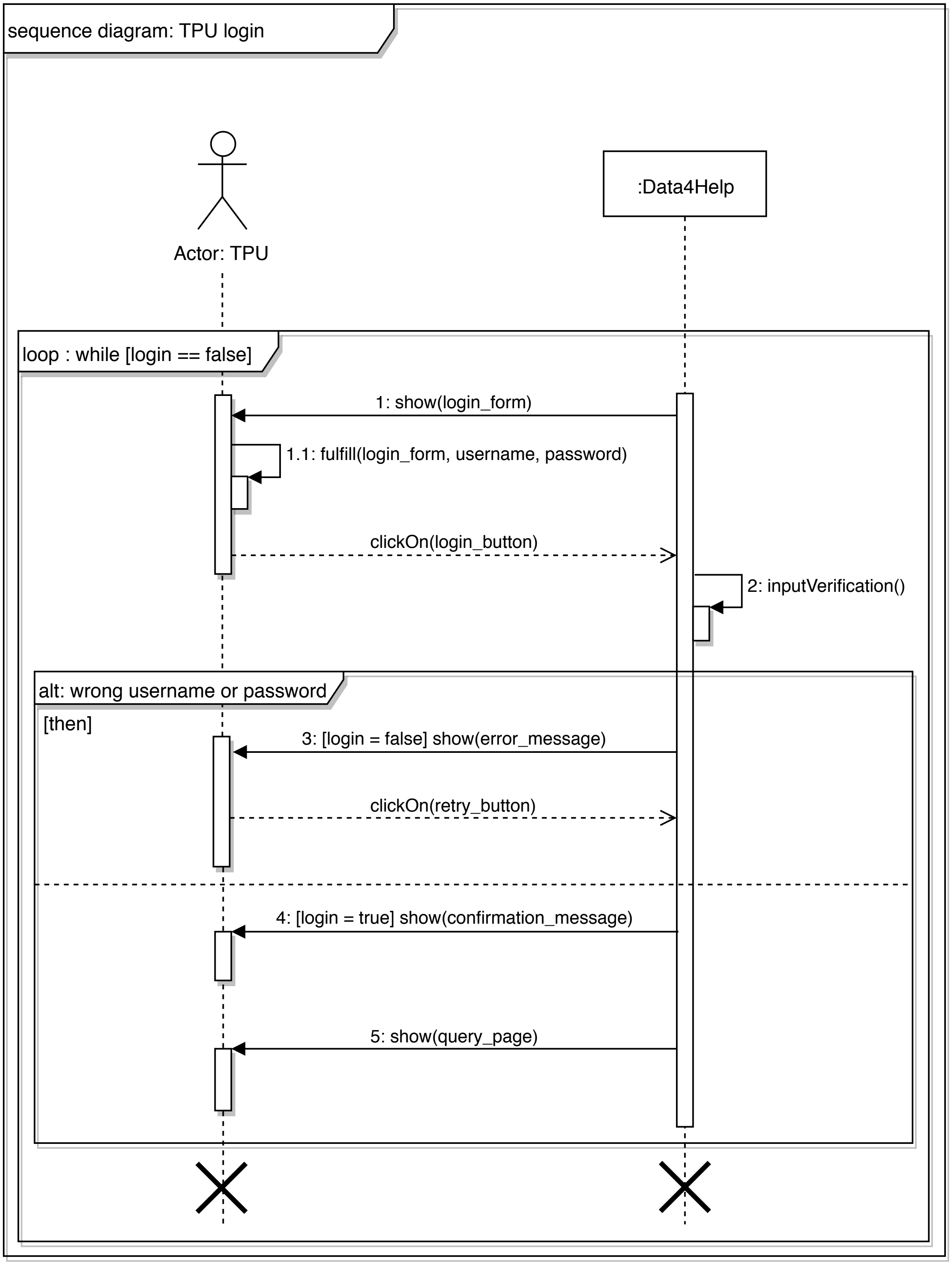

sequence diagram: TPU login
Actor: TPU
:Data4Help
loop : while [login == false]
1: show(login_form)
1.1: fulfill(login_form, username, password)
clickOn(login_button)
2: inputVerification()
alt: wrong username or password
[then]
3: [login = false] show(error_message)
clickOn(retry_button)
4: [login = true] show(confirmation_message)
5: show(query_page)

Figure 19: Third party user login

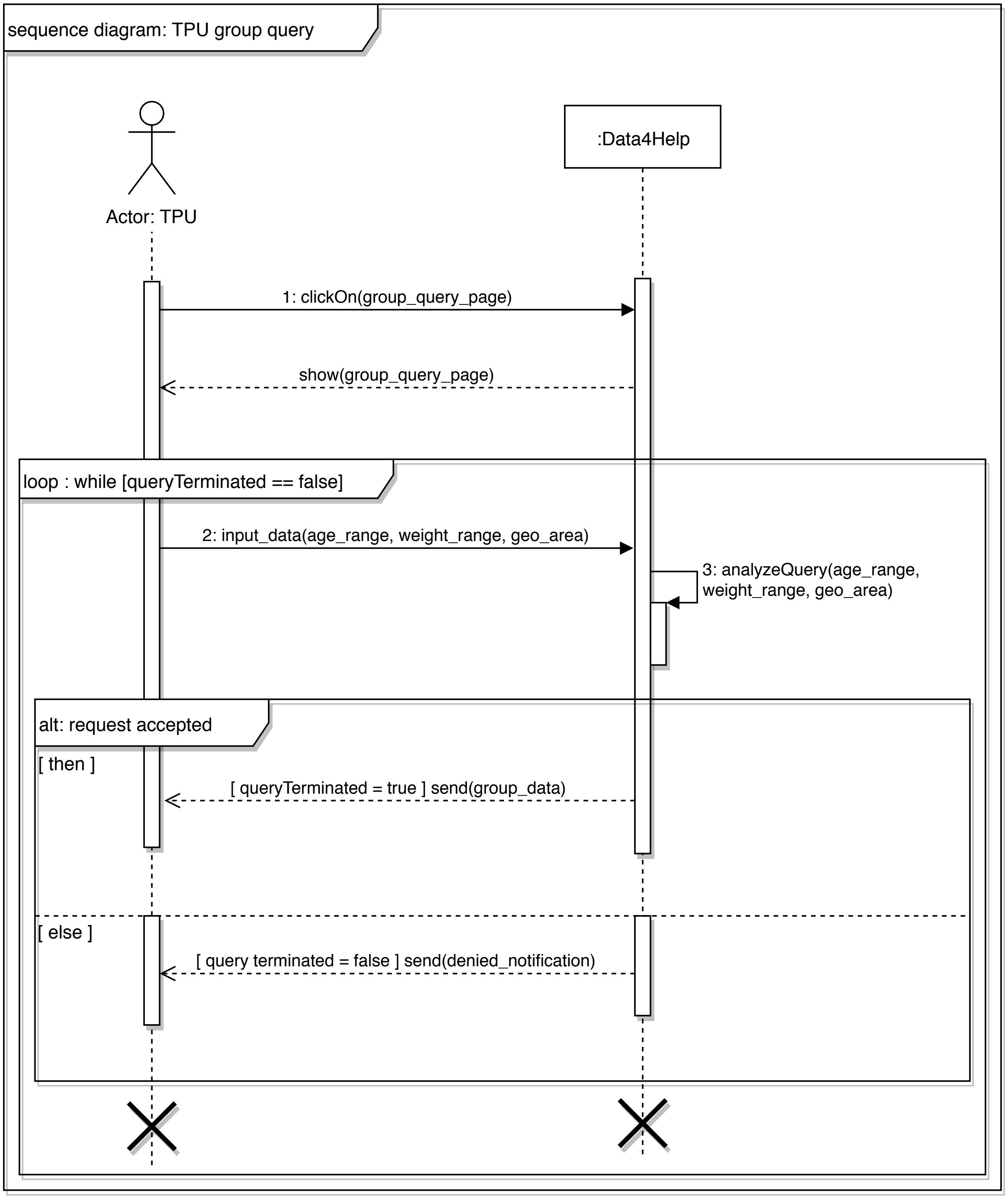

sequence diagram: TPU group query
Actor: TPU
:Data4Help
1: clickOn(group_query_page)
show(group_query_page)
loop : while [queryTerminated == false]
2: input_data(age_range, weight_range, geo_area)
3: analyzeQuery(age_range, weight_range, geo_area)
alt: request accepted
[then]
[queryTerminated = true] send(group_data)
[else]
[query terminated = false] send(denied_notification)

Figure 20: Third party user's group query

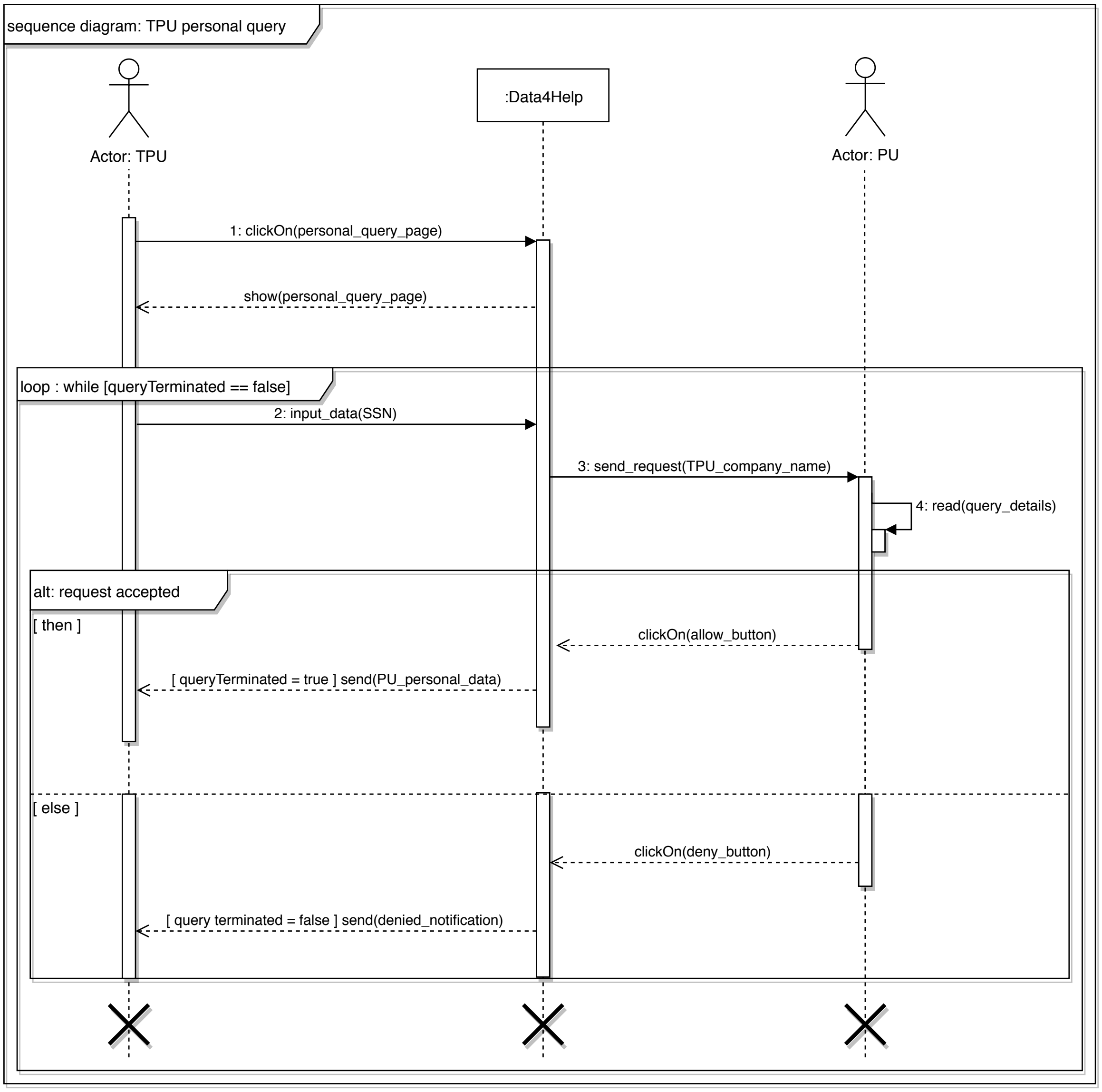

sequence diagram: TPU personal query
Actor: TPU
:Data4Help
Actor: PU
1: clickOn(personal_query_page)
show(personal_query_page)
loop : while [queryTerminated == false]
2: input_data(SSN)
3: send_request(TPU_company_name)
4: read(query_details)
alt: request accepted
[then]
clickOn(allow_button)
[queryTerminated = true] send(PU_personal_data)
[else]
clickOn(deny_button)
[query terminated = false] send(denied_notification)

Figure 21: Third party user's personal query

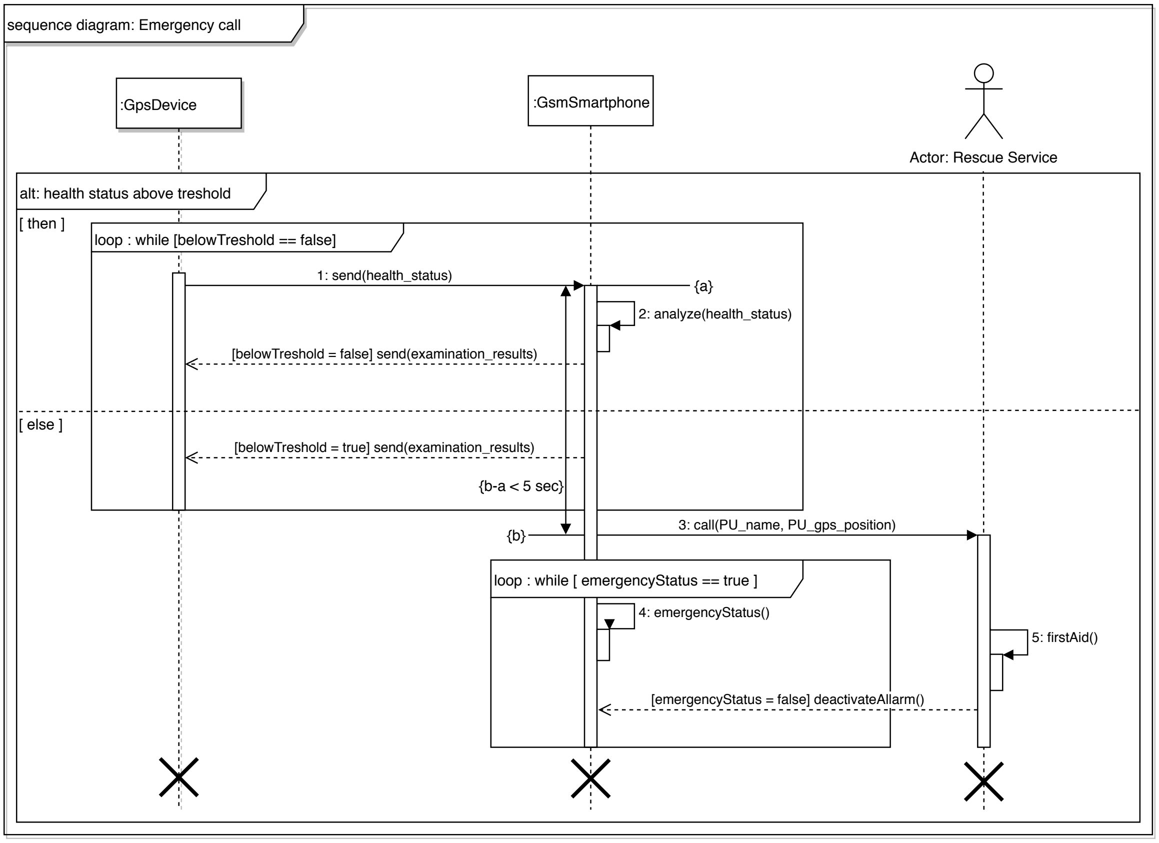

sequence diagram: Emergency call
:GpsDevice
:GsmSmartphone
Actor: Rescue Service
alt: health status above treshold
[then]
loop : while [belowTreshold == false]
1: send(health_status)
{a}
2: analyze(health_status)
[belowTreshold = false] send(examination_results)
[else]
[belowTreshold = true] send(examination_results)
{b-a < 5 sec}
{b}
3: call(PU_name, PU_gps_position)
loop : while [emergencyStatus == true]
4: emergencyStatus()
5: firstAid()
[emergencyStatus = false] deactivateAllarm()

Figure 22: Emergency call

3.3 Performance requirements

The system is projected to be delivered to a predefined amount of users at the beginning. Since no specific requirements have been specified for this section all the numbers provided below are pure assumptions and may change according to the will of the stakeholders.

- Data4Help
  1. Terminals to be supported: 3.000 - 10.000
  2. Simultaneous users to be supported: 2.000 - 6.000
  3. Amount of information to be handled: 300.000 - 1.000.000 PUs' data
- Automated SOS
  1. Terminals to be supported: from 9.000 to 30.000
  2. Simultaneous users to be supported: from 3.000 to 10.000
  3. Amount of information to be handled: any user device should be able to constantly or periodically monitor his/her health status.

3.4 Design constraints

3.4.1 Standards compliance

This document follows the IEEE-Standard for the format of Software Requirements specifications.

3.4.2 Hardware limitations

The only requirement for this application is a smartphone or smartwatch that can use GPS service.

- AutomatedSOS
  - iOS or Android smartphone
  - Smartwatch able to analyze a person's health status
  - GSM connection
  - GPS
- Web Interface
  - Modern browser

3.4.3 Any other constraint

- Regulatory policies

The user who wants to use this application will have to accept that their personal data could be used for market analysis or other research activity. The application also will have to constantly know the position of the users to ensure correct operation. Regarding the use of the application by third parties, in the case of request of specific user's data, to get this data the user will have to accept the request that will receive by mail.

3.5 Software system attributes

3.5.1 Reliability

The system must guarantee a 24/7 service. There may be only brief moments when the application will not be accessible, for example during maintenance. Data4Help will perform such maintenance as often as possible in the time slots with less use of the application by users.

3.5.2 Availability

The system has been designed to work only when there is network coverage. Moreover, if the GPS position is lost at the moment of the emergency, to ensure the maximum availability the system can retrieve the last saved position.

3.5.3 Security

User passwords and sensitive information will be protected and stored safely. All other user's information will be used by third party for marketing analysis, the user will be asked for permission at the time of registration.

3.5.4 Maintainability

The use of internal device functions such as gps or health parameters acquisition service allow system managers to intervene mainly on the data management part. This type of structure that includes the use of device services ensures greater stability to the application.

3.5.5 Portability

AutomatedSOS is an application compatible with Android and iOS systems for smartphone. Regarding smartwatch, the application can be installed only on types of smartwatch that can make a call (for example AppleWatch). Instead Data4Help is reachable only by website, and it is compatible with all modern browser.

4 Formal analysis using Alloy

4.1 Signatures

```
sig GpsPosition{
        latitude: one Int,
        longitude: one Int
} {
        latitude > -90 and latitude <90
        longitude > 0 and longitude<360
}

abstract sig User{
        email: one String,
        password: one String
}

sig PublicUser extends User{
        socialSecurityNumber: one String,
        averageMinPressure: one Int,
        averageMaxPressure: one Int,
        location: one GpsPosition,
        monitoringDevice: set Device
}{
        averageMinPressure > 60 and averageMinPressure < 80
        averageMaxPressure > 100 and averageMaxPressure <120
}

sig SystemManager extends User{
        update: Update
}

sig Update{}

abstract sig Device{
        pairedUser: one PublicUser,
        idCode: one String
}

sig GpsDevice extends Device{
        pairedDevice: one GsmSmartphone
}

abstract sig GsmDevice extends Device{
        phoneNumber: one String,
        rescueService: some RescueService
}

sig GsmSmartwatch extends GsmDevice{}

sig GsmSmartphone extends GsmDevice{
        pairedDevice: some GpsDevice
}

sig ThirdPartyUser extends User{
        referenceCompany: one String,
        queries: DataQuery
}

sig DataQuery{
        queryId: one String,
        thirdPartyUser: one ThirdPartyUser
}
```

```
sig PersonalQuery extends DataQuery{
        socialSecurityNumber: one String
}

sig PublicQuery extends DataQuery{}

sig EmergencyCall{
        location: one GpsPosition,
        personalDevice: one GsmDevice
}

sig RescueService{
        phoneNumber: one String
}
```

4.2 Facts

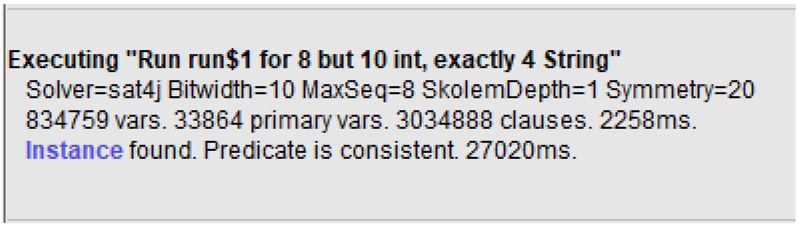
```
-- an email can be used for the same account only if they're for different use
fact noMultiplePUAccountEmails{
        all pu1, pu2: PublicUser |
                pu1.email = pu2.email implies pu1=pu2
        all tpu1, tpu2: ThirdPartyUser |
                tpu1.email = tpu2.email implies tpu1=tpu2
        all sm1, sm2: SystemManager |
                sm1.email = sm2.email implies sm1=sm2
}

-- general identification parameters must be unique
fact uniqueRestraints{
        no disjoint gsmDev1, gsmDev2: GsmDevice |
                gsmDev1.phoneNumber = gsmDev2.phoneNumber

        no disjoint resServ1, resServ2: RescueService |
                resServ1.phoneNumber = resServ2.phoneNumber

        no resServ: RescueService, gsmDev: GsmDevice |
                resServ.phoneNumber = gsmDev.phoneNumber

        no disjoint d1, d2: DataQuery|
                d1.queryId = d2.queryId

        no disjoint dev1, dev2: Device |
                dev1.idCode = dev2.idCode

        no disjoint pu1, pu2: PublicUser |
                pu1.monitoringDevice = pu2.monitoringDevice
}

-- a device can't be paired to multiple users
fact noMultipleUserDevice{
        no disjoint pu1, pu2: PublicUser |
                pu1.monitoringDevice = pu2.monitoringDevice
}

--every phone must be associated to at least one Rescue Service
fact atLeastOneRescueService{
        all gsmDev: GsmDevice |
                #gsmDev.rescueService >0
}

-- all pairing between devices must be symmetric
fact symmetricPairing{
        all dev: Device, pu: PublicUser |
                dev.pairedUser=pu
                implies pu.monitoringDevice = dev

        all gpsDev: GpsDevice, gsmSM: GsmSmartphone |
                gpsDev.pairedDevice=gsmSM
                implies gsmSM.pairedDevice=gpsDev
}
```

```
--two paired devices must be paired to the same user
fact sameUserPairing{
        all gpsDev: GpsDevice, gsmSM: GsmSmartphone |
                gpsDev.pairedDevice = gsmSM
                implies gpsDev.pairedUser = gsmSM.pairedUser
}
```

4.3 Results

```
run {
        #PublicUser = 2
        #ThirdPartyUser = 1
        #SystemManager = 1
} for 8 but  10 int, exactly 4 String
```

4.4 Proof of consistency



```
Executing "Run run$1 for 8 but 10 int, exactly 4 String"
   Solver=sat4j Bitwidth=10 MaxSeq=8 SkolemDepth=1 Symmetry=20
   834759 vars. 33864 primary vars. 3034888 clauses. 2258ms.
   Instance found. Predicate is consistent. 27020ms.
```

Figure 23: Proof of consistency

4.5 Generated world

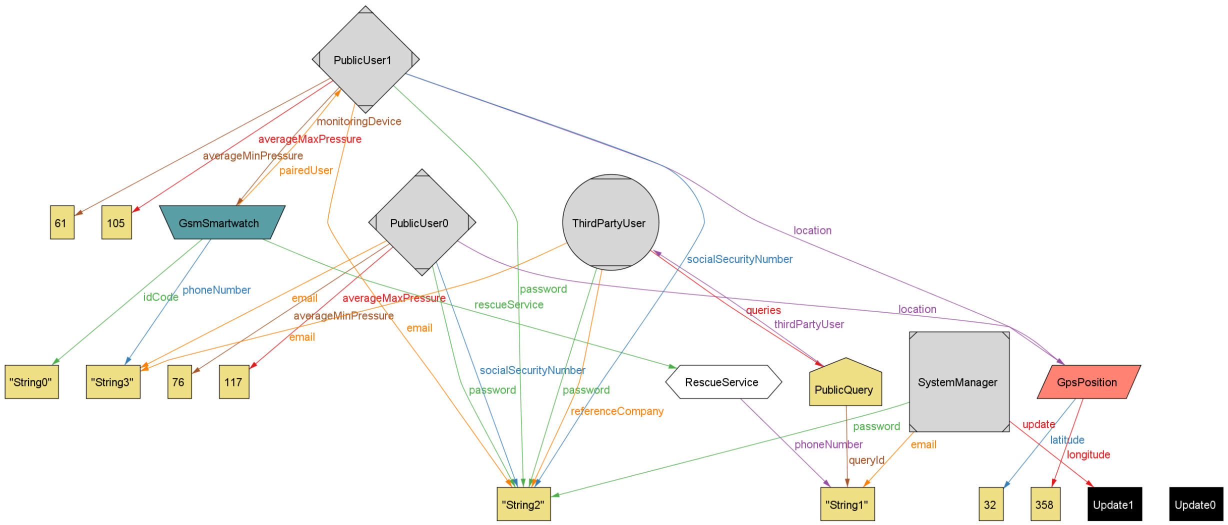


Figure 24: Alloy generated world

5 Effort spent

5.1 Davide Damato

Date	Task	Hours
02/11/2018	Introduction	4
05/11/2018	Use case definition	4,5
06/11/2018	Use case definition	1,5
06/11/2018	Goals definition	1
06/11/2018	State diagram	2,5
07/11/2018	Product functions	2
07/11/2018	Hardware-Software Interfaces	1
08/11/2018	Communication Interfaces	1
08/11/2018	Hardware Limitations - Constraint	1,5
08/11/2018	Mockup	3,5
09/11/2018	Mockup	2
09/11/2018	Software system attributes	2.5
10/11/2018	Mock up fix - Maintainability	1
10/11/2018	Hardware/Software/Communication interfaces fix	1.5
11/11/2018	Requirements - Standard Compliance	1
	Total	30

Table 1: Davide Damato work hours detail

5.2 Luciano Franchin

Date	Task	Hours
02/11/2018	Introduction - Initial latex layout	4
05/11/2018	Use case definition	3
06/11/2018	Use case definition	1
06/11/2018	Scope definition - Uml class diagram	2
06/11/2018	Product Perspective	1.5
07/11/2018	Domain Assumptions	2
07/11/2018	Scenarios - Abbreviations - Definitions	2
08/11/2018	Performance requirements - UML diagrams modification	2.5
08/11/2018	Sequence Diagrams	3.5
09/11/2018	Sequence Diagrams - Images layout	3.5
09/11/2018	Performance Requirements review - Alloy	4
10/11/2018	Alloy - General review	6
10/11/2018	Alloy	3
11/11/2018	Alloy - Functional Requirements	4
	Total	41.5

Table 2: Luciano Franchin work hours detail

6 References

6.1 Used tools

- `overleaf.com`, Overleaf © 2018
- `draw.io`, a trading name of JGraph Ltd.