

Trackme RASD

REQUIREMENTS ANALYSIS AND SPECIFICATION DOMAIN (RASD)

Release date: 11/11/2018

Version: 0.1 (usiamo 0.xx per indicare le bozze, la 1 sarà per la definitiva)

Come table of content propongo il sommario, che poi sarà pure quello che intende il prof. Come font propongo Georgia o Calibrì light (come quello del progetto di reti logiche), come tema il terzo da sinistra di office. Grazie ciao

Table of content

[[controllare la registrazione] [pon] controllare anche il server [la communcazione??] 3](#_Toc529525280)

[Introduction 3](#_Toc529525281)

[Purpose 3](#_Toc529525282)

[Purpose of the document 3](#_Toc529525283)

[Purpose of the project 3](#_Toc529525284)

[Scope [Analysis of the world and the shared phenomena] 4](#_Toc529525285)

[World phenomena 5](#_Toc529525286)

[Shared phenomena 5](#_Toc529525287)

[Machine phenomena 6](#_Toc529525288)

[Goals for Data4Help 6](#_Toc529525289)

[Goals for AutomatedSos 6](#_Toc529525290)

[Definitions, acronyms, Abbreviations 6](#_Toc529525291)

[Definitions 6](#_Toc529525292)

[Acronyms 7](#_Toc529525293)

[Abbreviations 7](#_Toc529525294)

[Revision history 7](#_Toc529525295)

[Reference Documents 7](#_Toc529525296)

[Document structure 7](#_Toc529525297)

[Overall description 7](#_Toc529525298)

[Product perspective 7](#_Toc529525299)

[Product functions 7](#_Toc529525300)

[User characteristics 7](#_Toc529525301)

[Actors 7](#_Toc529525302)

[Assumptions, dependencies and constraints 8](#_Toc529525303)

[Domain Assumptions 8](#_Toc529525304)

[[sono giuste le text assumptions che abbiamo fatto?] 8](#_Toc529525305)

[Text Assumptions 8](#_Toc529525306)

[Specific requirements 9](#_Toc529525307)

[External Interface Requirements 9](#_Toc529525308)

[User interfaces 9](#_Toc529525309)

[Hardware Interfaces 9](#_Toc529525310)

[Software Interfaces 9](#_Toc529525311)

[Communication Interfaces 9](#_Toc529525312)

[Functional Requirements 9](#_Toc529525313)

[[G1] Provide a form of unique identification (registration/login) of all the clients of the application 9](#_Toc529525314)

[[G2] Prevent third parties from associating a single user to his data without his permission 10](#_Toc529525315)

[[G3] Whenever a user is in danger of life, the application is working properly and there is internet connection, an ambulance is alerted 10](#_Toc529525316)

[[G4] If the user’s health status is not clear due to malfunctions, the user’s family is alerted within an hour 10](#_Toc529525317)

[[G5] Allow the user to see, under request, reports on his vital parameters and data about his health 10](#_Toc529525318)

[[G6] Allow third parties to access data if and only if they could be anonymized 10](#_Toc529525319)

[Non-functional Requirements 11](#_Toc529525320)

[Accuracy 11](#_Toc529525321)

[Performance 12](#_Toc529525322)

[Availability and reliability 12](#_Toc529525323)

[Security 12](#_Toc529525324)

[Scalability/ Design for reusability 12](#_Toc529525325)

[Design Constraints 12](#_Toc529525326)

[Standards compliance 12](#_Toc529525327)

[Hardware limitations 12](#_Toc529525328)

[Any other constraint 13](#_Toc529525329)

[Scenarios 13](#_Toc529525330)

[Scenario 1- Case of emergency 13](#_Toc529525331)

[Scenario 2-Application breakdown 13](#_Toc529525332)

[Scenario 3-. Sensor breakdown 14](#_Toc529525333)

[Scenario 4-Data anonymization and data presentation 14](#_Toc529525334)

[Scenario 5-subscribe to new data 14](#_Toc529525335)

[Scenario 6-Personal Use 15](#_Toc529525336)

[UML modelling 15](#_Toc529525337)

[Formal analysis using alloy 15](#_Toc529525338)

[Effort spent 16](#_Toc529525339)

[References 16](#_Toc529525340)

# [controllare la registrazione] [pon] controllare anche il server [la communcazione??]

Nota da inserire [dove va messa secondo te?]:

* *Data4Help* and *AutomatedSos* are supposed to communicate with a huge number of third parties, but we can’t realistically force our system to deal with so many external softwares (the ones of all the third parties), considered that the they could have been developed for different purpose ( the third parties can belong to different markets) : it may even happen that the third party does not have any informative system at all! Due to this, we feel necessary to develop a user interface (e.g. a web app) which all third parties’ managers can use to visualize data or download them in a limited format (e.g. Excel document). To deal with that, our system must provide some forms of understandable presentation of data (e.g. not only tuples), which would have been done by external sofwares if we had made a different modelling of the word

# Introduction

## Purpose

### Purpose of the document

This is the Requirements Analysis and Specification Document of the TrackMe project. Its purpose is to provide a complete description of the system to develop in order to build up the services *Data4Help* and *AutomatedSoS*. In concrete, this means to identify the goals and (most important of all) the requirements (both functional and non-functional), to model the system and the portion of reality it is going to affect in a formal, logic and unambiguous manner, to present all the most common scenarios and use cases, to show relevant constraints and issues.

This document is intended to be a binding yet useful guide for stakeholders, project managers, developers, analysts and testers.

### Purpose of the project

TrackMe wants to develop two software-based services, *Data4Help* and *AutomatedSoS.* The main goal of TrackMe is to provide data collected by the subscribed users to subscribed third parties, helping companies in their business, together with respecting users’ privacy. The two services are not independent each other, but *AutomatedSos* is built on top of the first one: this means that it’s designed as an additional feature which is implemented after Data4Help. Nonetheless, the two offered services differ each other: through Data4Help TrackMe collects data about the position and the health status just to extract useful information for companies of various types, while AutomatedSos is supposed to allow, through data about health status, third parties to provide a medical assistance in case of emergencies. Because of that, A*utomatedSos* needs faster detection, analysis and communication of data. The assistance is defined as non-intrusive, which means that the interactions with users are reduced to the minimum possible, and personalized, which means that the system is able to set personalized tresholds for each (kind of) user: when parameters go under or below these thresholds, assistance is provided

By and large, main beneficiaries of the two services are the third parties, which can make use of some useful tools in examining data: a big number of constrains and parameters helps to define more specific queries, and there are many user-friendly ways to show data.

## Scope

### Analysis of the world and the shared phenomena

TrackMe is a company interested in developing two different services: Data4Help and AutomatedSoS.

First of all, with Data4Help it wants to support third parties to get data about people’s health status and location. To do that, it aims at collecting and storing all these data, in order to make them available to third parties.

The system is able to handle, manipulate and anonymize data. In fact, third parties could request data of a single user or of a group of people, specifying different constraints (e.g. geographical). They also may want to know elaborated data or statistics (e.g. average, maximum, media, sum…).

The system is concerned in guaranteeing in every moment the privacy of the users and knowing always all the people online to struggle misuses: the services are made available only if users submit a registration form, which includes medical history (in order to improve the quality of the analysis) and personal information (in Italy, we accept the fiscal code). In this way all the users can be unique identified, and when data of single users are requested, they are asked to give permission to share their individual data.

All third parties must be registered to the system too, in order to make them recognizable by the system and by the users when they request for individual data.

When data about a group of people are requested, the system handles the request keeping always in mind the privacy issue: because of that, it anonymizes all data shared in order to avoid the third parties to identify the single users who are members of the required group.

[devo parlare dei 1000?]

It may happen that a third party is particularly interested in some data that are not currently available (because they regard a period in the future, because they have not elaborated yet…), or in periodic updates of the same information, so it has the possibility to express some preferences in data and to be informed by the system when there are produced.

The second service offered by TrackMe is *AutomatedSos,* which is in some way inspired from Data4Help and is implemented afterwards. Indeed, having access to health status values of almost all the population, TrackMe realises that it could be useful to provide to elderly people (or with health problems in general) an automated assistance service. *AutomatedSos* exploits all data detected by *Data4Help* making a different use of them: it compares them with personalized thresholds to check whether an immediate emergency is occurring.

Although it, in practise, needs the same permissions of *Data4Help*, ad additional registration is required, both for third parties and users: only the interested members should be addressed by this service, to avoid misuses.

When values go below of over these thresholds, the system sends an alarm to all the subscribed third parties, which can ack the message and immediately send the necessary help to the user. Of course, together with the alert the system communicates the user position ( relying on GPS service) and the health values detected, so that the third parties are able to send the most appropriate and fastest assistance.

Since it is necessary to consider the enormous importance of the efficiency of the service, we may want to detect and handle the cases where the sensor does not provide appropriate data o does not send data at all (i.d. the system is not able to monitor the user’s health status). For this purpose, some regular intervals of time are established, after which the application expects input data. If they are not available (or not readable, or absurd), it can conclude that something is not working properly, and takes the appropriate countermeasures: it asks for confirmation of good health status to the user, and, in case of no answer, it contacts an emergency number.

For the same reason, although we are aware that we are forcing a strict requirement, we state that the all the applications should send data for back up purpose to the system: if an app misses these periodic deadline, the system realizes that it is not working correctly (for various reasons: physical damage, misuse, etc.) and sends immediately a message to the emergency number.

Consequently, *AutomatedSos* needs a real user interface to dialogue with the user (though limited), while for Data4Help the user is merely a subject to observe, with no concrete interaction but the registration.

We imagine that, although health is of course a key issue for everyone, there could be anyway some moments where the user needs to switch off the physical device (e.g. for charging, or maybe in the swimming pool, or in the mountains, or on an airplane, or in some other special cases): to handle this, we give the user the opportunity to set manually the device as “non active”, to stop the *AutomatedSos* service and avoid improper detection of malfunctions of the system.

To summarize all what we have exposed, in the following chapters there can be found some lists.

### World phenomena

* Diseases /absence of disease
* Emergency situations
* Ambulance moves and arrives
* Ambulance breakdown
* GPS does not work
* Server breakdown
* Downloading the app on the smartwatch or smartphone
* The user must switch off the device/special cases

### Shared phenomena

[Controlled by the world]

* Detection of health values
* Request for data from the third part
* Third party’s taking charge of an emergency
* Confirmation of good health status by the user
* Registration to the service by a user or a third party
* Setting the device non-active/active for *AutomatedSos*
* User registration (for *Data4Help* or for *AutomatedSos*)

[The following ones have been put here and not in machine phenomena because, in our modelling, they can be in some ways detected by the server or other components of our machine]

* Sensor breakdown/sensor working properly
* Application breakdown/application working properly
* Network breakdown

[controlled by the machine]

* Sending data to the third parties
* Sending an alert to third parties
* asking the user to confirm his health status

### Machine phenomena

* Database queries
* Database inserts
* Database creation
* Data analysis and comparation with thresholds
* Communication between the application and the server: i.e., sending and receiving messages between the app and the server
* Data elaboration for showing
* Data storing by the application
* Threshold calculation for each user

Having classified the phenomena of interest, we want in the following chapters to state formally all the goals.

### Goals for both services

* [G1] Provide a form of unique identification (registration/login) of all clients of the services;
  + [G1.1] Provide a form of unique identification (registration/login) of all users using the services
  + [G1.2] Provide a form of unique identification (registration/login) of all users using the services

### Goals for Data4Help

* [G2] allow the user to avoid being associated to his data without his permission
* [G3] Allow third parties to request access to data of some specific individuals
* [G4] Allow third parties to request access to anonymized data of groups of individuals
* [G5] Allow third parties to subscribe to new data
* [G6] Allow third parties to obtain the most adapt data for their needs

### Goals for AutomatedSos

* [G7] Third parties are alerted, whenever a user is in danger of life, the application is working properly and there is internet connection
* [G8] If the user’s health status is not clear due to malfunctions, an Emergency number is alerted within an hour

## Definitions, acronyms, Abbreviations

### Definitions

* “***Health status***”: when in the following parts we state “health status” we are meaning the following values:
  + Blood saturation: it’s an indicator of the status of lungs and of respiratory system in general (e.g. to detect suffocation)
  + Hearth rate: it’s an indicator of hearth diseases (to detect hearth attacks)
  + Blood pression: it hardly ever helps to detect an emergency, but it’s useful for third parties and statistics (blood pressure out of range can indicate/cause a huge number of chronical diseases)
  + Body temperature: it’s an indicator of fever
  + User’s falling: if the user has suddenly fallen there could be various causes and effects that, though other values are not able to detect them, put in serious risk the user’s life.
* ***“Data Anonymization”***: deleting the fiscal code associated to every data tuple obtained by the query;
* ***“Tresholds”***:
* ***“Active/ non active”:***
* ***“Client”:*** everyone using the service;

### Acronyms

* RASD: Requirements analysis and specification domain

### Abbreviations

* [G-n]: n-goal
* [D-n]: n-domain assumption
* [R-n]: n-functional requirement

## Revision history

## Reference Documents

* Specification Document “ A.Y. 2018-2019 Software Engineering 2 Mandatory project”

## Document structure

# Overall description

## Product perspective

## Product functions

## User characteristics

### Actors

* Users (only for *AutomatedSos*): the person who download *AutomatedSos* from the app store, wearing the device and allowing the application to monitor his health status and to manage his data;
* Third party: a company which is interested in monitoring population’s health status and obtaining a useful resource of data (e.g. a health insurance, a pharmaceutical company, the government, an hospital);
* External service: NESSUNO!

## Assumptions, dependencies and constraints

The description of the problem appears to be incomplete and ambiguos: due to this, we feel necessary to make the following feasible assumptions

### Domain Assumptions

* [D1] The user has correctly downloaded the application from the online store on his device (smartwatch, smartphone, etc) [i.e. he is not using a crack version]
* [D2] GPS always works properly indicating the user’s position
* [D3] For every location, there is at least a third party which can send medical assistance
* [D4] The emergency number is correct
* [D5] There is internet connection when the request is submitted

### [sono giuste le text assumptions che abbiamo fatto?]

### Text Assumptions

* !!!!!
* Data4Help is a software-based service included in an application which is presented to the public to have a different purpose (e.g. a pedometer, a diet monitoring application, a simple application to register health data) for marketing reasons. Otherwise, there is no clear advantage for the user to register to the service (which is possible, but unlikely). This happens many times in business models: the goal of the application has nothing in common with the use the people do of it.
* !!!!!
* On the other hand, AutomatedSos is required (by the problem specification) to rely on *Data4Help* data but is deployed as an independent application (cfr. the next text assumption)
* The registration to Data4Help is a necessary condition to register to AutomatedSos, but not all the third parties and the users registered to Data4Help are necessarily registered to AutomatedSos (e.g. all the companies which are interested only in data about location can avoid register to AutomatedSos, because they may have no way to handle emergencies in any case) 🡪 o magari uno è giovane e non gli frega di scaricare AutomatedSos + aggiugi il fatto che quando uno scarica Atuomated comunque per accedere al servizio deve registrarsi a data4help
* [questa è una text assumption o una cosa che diciamo noi???]
* When an emergency is reported, the system sends a notification to all third parties who are subscribed to AutomatedSOS; the first third party that responds to the alert is the one who takes charge of the emergency. When this happens, the emergency is marked as “handling” by the system, and the notification disappears from all the other third parties. 🡪 ci sta metterlo qui ma va aggiunto anche nella sezione 2 tipo in overall description
* All users are identified through their fiscal code (which is also their username), while all third parties are identified through their official email
* When talking about health status we mean some specific parameters, which could be found in the section “*Definitions”*
* The personalized thresholds are calculated by the system and are based on age, gender and clinical history inserted by the user. The algorithm to calculate thresholds has been elaborated with the agreement of a medical equipe.
* Aggiungi il fatto che assumiamo che trackme abbia notato che molte terze parti hanno presentato la volontà di offrire assistenza ai vecchi e quindi trackme sviluppa questo servizio per loro, quindi immaginiamo che siano loro a fornire l’assistenza e non il 118. Se vuoi ci sta aggiungere che si sarebbe potuta immaginare una api del 118 ma che non era secondo noi molto leggera come assunzione e poco coerente con la nostra interpretazione del testo in cui sono le aziende il principale cliente di trackme, più che gli user

# Specific requirements

## External Interface Requirements

### User interfaces

To give a generic idea of what the application will look like we deployed the following mock ups:

[Data4Help]

…….

[AutomatedSos]

### Hardware Interfaces

* The sensor which detects data (both for *Data4Help* and *AutomatedSos*): in our modelling it is supposed to be able to detect a huge number of paramenters, but we can accept cases when low level sensors detect only a part of them, provided that the service is not put at risk.

### Software Interfaces

* Data4Help communicates with the user through the application we have suppose it has been created on purpose. Thanks to the facts that it is deployed by the same developers and little communication occurs, there shouldn’t be so many problems in adapting the code and no need for big software interfaces. The only messages they must exchange are the ones to ask for allow third parties to get personal data and the registration form.
* [Alternativa] no sofware interface is required

### Communication Interfaces

* Devo parlare qui del server?

## Functional Requirements

### [G1] Provide a form of unique identification (registration/login) of all the clients of the application

* [R1] If the client does not insert a identifier and password, the application does not let the user access any functionality[login]
* [R2] If the client declares that it has not an account (i.e. it’s the first access), he can fill a registration form to create a new one, providing an identifier and a password[registration];
* [R3] If the identifier provided in the registration form is already associated to a password, the application refuses the registration [unique identification]
* [D1] The client has correctly downloaded the application [i.e. he is not using a crack version]

#### [G1.1] Provide a form of unique identification (registration/login) of all users using the application

* [R4] If the user does not fill the registration form with his fiscal code and all other personal data, the application refuses the registration

#### [G1.2] Provide a form of unique identification (registration/login) of all third parties using the application

* [R5] If the third party does not fill the registration form with his official e-mail and all other public data, the system refuses the registration

### [G2] Allow the user to avoid being associated to his data without his permission

* [R6] If a third part asks for data of a single user, the system asks for the user’s permission
* [R7] The system refuses the request when the user denies access
* [R8] if a third part asks for data that involves less than 1000 people, the application refuses
* [R9] if a third part asks for data that involves more than 1000 people, the application anonymizes data before sending

### [G3] Allow third parties to request access to data of some specific individuals

* [D2] There is internet connection when the request is submitted
* [D3] The third party knows the fiscal code of the specific individual
* [R10] The system retrieves and shows data regarding an individual
* [R6] If a third part asks for data of a single user, the system asks for the user’s permission

### [G4] Allow third parties to request access to anonymized data of groups of individuals

* [D4] There is internet connection when the request is submitted
* [R9] if a third part asks for data that involves more than 1000 people, the system anonymizes data before sending
* [D5] Third parties express realistic constrains
* [R11] The system allows third parties to specify constraints to filter data
* [R12] The system provides data grouped and selected according to the requests

### [G5] Allow third parties to subscribe to new data

* [R13] the system notifies third parties when un update is available
* [D5] Third parties express realistic constrains
* [R14] Third parties can specify whether they are interested in updates
  + [R14.1] Third parties can specify constraints on data which are not yet available (i.e. they haven’t been elaborated yet or they regard a future period of time)
  + [R14.2] Third parties can ask for periodic data

### [G6] Allow third parties to obtain the most adapt data for their needs

* [R15] The system allows third parties to obtain aggregated data and statistics (average, maximum, minimum…)
* [R16] The system offers different options to visualize data
* [R17] Third parties can download data
* [D5] Third parties express realistic constrains

### [G7] Third parties are alerted, whenever a user is in danger of life, the application is working properly and there is internet connection

* [R18] If input data show a severe disease or the user communicates that an emergency is occurring, the machine contacts all third parties (with a notification containing the position and the health values)
* [D6] GPS always works properly indicating the user’s position
* [D7] For every location, there is at least a third party which is able to handle the emergency
* [R19] The application knows the correct thresholds for each type of user
* [D8] The user sets on “active” the application every time he feels he could need medical assistance

### [G8] If the user’s health status is not clear due to malfunctions, an Emergency number is alerted within an hour

* [R20] If the application does not read properly input data every 500 Ms (including absurd data), asks for confirmation of good health status.
* [R21] If the user does not respond to confirmation within 5 minutes, the system sends a message to the emergency number, provided through the registration form
* [R22] If the application does not send data for back up purpose every hour, the system sends a message to the emergency number
* [D9] The emergency number is correct
* [R23] If the user does not fill the registration form with his fiscal code and all other personal data, the application refuses the registration

## Non-functional Requirements

Accuracy

This is the non-functional requirement we consider the most relevant. While assuming (here and in all other parts of RASD, cfr, *domain assumptions*) that GPS works properly, our biggest concern is about the sensor which detects values: there is a concrete risk of frequent false positives, caused by some little offsets. Such cases are particularly annoying because we can’t ignore the alarms but at the same time, they make us waste time, money and resources, threatening the correct working of our machine. We pointed out that there is a trade-off between this issue (which would request to wait for a lot of consecutive signals of emergency to be sure) and the issue of performance (which would request to react immediately). We decided to wait for 1,5 sec before alerting, which means, values below thresholds for three times in a row. However, provided that false positives are defined in statistics as *type one errors,* we state that the sensor must have prob (type I err) < 5%.

Performance

Of course, we need a fast reaction to emergency. Concretely, we state that the machine must guarantee a reaction time of less than 5 seconds from the time the parameters are below the thresholds. This means that the application should process, compares data and send a message to the server in less than 3,5 sec (provided that only three consecutive data below the threshold are a clear signal of severe disease and data are sent every 500 Ms).

In addition to this, 4G connection is required to ensure immediate communication with the server.

### Availability and reliability

* Concerning the server: We need a server available 24/7 to handle emergency messages as fast as possible because, among other reasons, there is no way for the app the handle an emergency without the server;
* Concerning the app: we also need availability for the app, but not (with so much relevance) reliability , due to the fact that it takes time to detect a disease ( the sensor does not send data in real time ecc..), in which there could be an app break down with no significative consequences for the service ( provided that recovery time is under 500 ms).

Security

Thinking of a market such as the American one, where health care is subject to negotiation between user s and companies, security of all sensitive information which could advantage malicious companies, is a very important concern for our application. Due to this reason, data encryption should be implemented in communications between the app and the server and the web apps.

### Scalability/ Design for reusability

## Design Constraints

### Standards compliance

### Hardware limitations

Our software needs 2 physical devices to work and communicate properly at the same time. In order to deal with this strong hardware limitation, we came up with some mechanisms of detection of breakdowns

* iOS or Android smartphone with 4G connection and Bluetooth connection
* GPS connection
* Wearable device with Bluetooth connection

In alternative

* iOS or Android smartwatch
* 4G connection
* GPS connection

For visualizing data [Third parties],

* Modern browser able to render graphs and statistical models

### Any other constraint

# Scenarios

## Scenario 1- Case of emergency

Gianni is a 76 years-old man and lives alone quite far from his daughter, Livia.

He suffers from hearts problems, so Livia decides to enrol him to “AutomatedSos”. Then she downloads the app on her father’s phone and buys him a smart bracelet to connect to the system.

When registering Livia indicates its number as emergency number and she indicates, beyond the basics, some details about heart problems that his father has had in the past.

There is no problem connecting to the internet because in the home of Gianni there has been a Wi-Fi network for some years. Gianni also recharges the device every afternoon during the visits of his daughter so that battery is fully charged when he is alone at home.

A day Gianni has a heart attack while alone in the house, and then his heart values fall sharply below the thresholds laid down for him by the application.

The system immediately sends an alarms to the companies offering the assistance service, together with the values of the heart rate and the position of Gianni’s home.

The Policlinic Hospital responds first to the warning transmitted from system and takes charge of the emergency, which is then marked as *"handling*”.

## Scenario 2-Application breakdown

Derek is a man in his fifties and for some years has installed on his smartwatch the application Data4Help, which helps him to keep an eye on his health.

One day a friend told him about the additional service offered by TrackMe, *AutomatedSoS,* which guarantees automated assistance, then Derek, feeling intrigued, decides to add AutomatedSos on his smartwatch.

He has already an account, so he has just to add a phone number to contact in case of emergency: he chooses his wife’s number.

During the week-end Derek descends to the cellar to help his son to repair his bike forgetting to set the app status manually "off": unfortunately, the cellar, being underground, is not covered by their home Wi-Fi network.

The application then fails to send the data every hour and the server, not getting the back-up, sends a message to the emergency number to notify that the application is not working properly.

The wife of Derek receives the message, so she decides to go to the cellar in order to check out if everything is ok and remembers her husband to turn the application’s status “off”.

## Scenario 3-. Sensor breakdown

Anna is an elderly lady who has recently retired. Instead of retiring too, her husband works all day outside home so convinces Anna to register to AutomatedSos to be safer when he is not at home.

So she buys a small smartwatch on which she installs the application. She provides her data and indicates her husband’s number as number of emergencies.

She wears the smartwatch every day when her husband is out and recharge it when he is at home.

One day she forgets to take it off before getting into the bath: a bit of water enters the smartwatch, causing a sensor breakdown. the sensor is no more able to send data correctly.

The application, not receiving data for more than 1 minute, sends a notification to Anna in order to know her health status.

Anna sees the message and confirms that she is okay but decides to manually disable the application, putting its status “off”, so that she can bring the smartwatch for repairing, without alerting the number of emergencies.

## Scenario 4-Data anonymization and data presentation

The Saint Francis Medical Clinic would like to open a new geriatric ward then turns to external consultants to figure out if it is convenient, or to understand how many people they might have.

Therefore, the consultants decide to register to Data4Help to collect some prediction data concerning the population living near the clinic.

After registering with the mail at the company, they begin to gather information.

First, ask the average pression and heart rates of people between 70 and 90 years living within a radius of 20 km from the clinic. the application has the data of more than 1500 people, and then accepts the request by providing the media required by the company.

To make things more understandable, the application shows a histogram indicating the number of people in every pression band, to distinguish correctly the big number of people with normal pression values from the little group of people with values very far from the average, which are obviously the target of the clinic.

Then the company asks the number of falls recorded in the last year regarding persons over 60 years, always within 20 km from the clinic. Again, having been recorded more than 2000 falls more or less serious, the data is anonymized and is provided by the system.

The company requires then the number of people who have heart problems in their medical history but who live within a radius of 8 km from the clinic, but this time being that only a small number of users meets these criteria, the request is rejected by the system.

Finally, they ask the application the number of old people signed up to Data4Help who live near the clinic and, of course, who authorize the processing of their personal data.

## Scenario 5-subscribe to new data

The municipal administration of Novate Milanese had a very positive impression about Data4Help before the last elections, finding some data very useful and interesting to get an idea on health of citizens.

In particular, they used the data provided by the application to propose some prevention programs or help for some diseases.

This year the Education Commissioner wants to allocate funds for a smoking-prevention program in all the schools of the municipality, because he is afraid that more and more kids start smoking during high school. Due to the fact that there is not enough money for all projects, and there is no clear evidence of a real increase of smokers between teenagers, his idea is abandoned.

Nonetheless, he decides to take advantage of Data4Help and subscribes to new data on blood saturation of teenagers between 14 the 18 years in the next year.

After 1 year, when the deadline comes, the system notifies that the data he aimed at are now available. Therefore, he examines these data that indicate a decrease in blood saturation of that group of users specified by the query. Consequently, he concludes that his fears were right.

Thanks to these data, the Commissioner manages to convince the administration to invest some funds to sustain his program.

## Scenario 6-Personal Use

Betty has just begun a new fitness program, after being stopped for a few years. At the first lesson, the fitness coaches recommend using Data4Help to check whether the program is too stressful for her health.

Betty downloads the app on her smartphone and buys a device to connect to the application.

Betty enrolled the service providing her social fiscal code, basic information about herself (weight, height, gender) and her health (pre-existing conditions, chronical diseases, pathologies…).

The gym, that want to check out the real impact of their services (e.g. improvements, health stress…) ask via Data4Help for her personal data. Betty receives the request on her smartphone, recognizes the company and accepts without any doubt. Data4Help finds and sends all kind of information it has from Betty’s profile stored in the database.

# UML modelling

# Formal analysis using alloy

# Effort spent

# References