Politecnico di Milano

AA 2018/2019



RASD – Requirement Analysis and Specification Document

Version 1.0 - 11/11/18

*Authors: Professor:*

Emilio Imperiali Elisabetta Di Nitto

Giorgio Labate

Mattia Mancassola

TABLE OF CONTENTS

1. Introduction
   1. Purpose
   2. Scope
      1. Description of the given problem
      2. Goals
   3. Definition, Acronyms, Abbreviations
      1. Definitions
      2. Acronyms
      3. Abbreviations
   4. Document Structure
2. Overall Description
   1. Product perspective
   2. Product functions
      1. Monitoring mnagement
      2. SOS management
      3. Run Management
   3. User characteristics
   4. Assumptions, dependencies and constraints
3. Specific requirements
   1. External interface requiremens
      1. User interfaces
      2. Hardware interfaces
      3. Software interfaces
   2. Functional requirements
      1. User
      2. Third Party
      3. Requirements
   3. Performance requirements
   4. Design Constraints
   5. Software system attributes
      1. Reliability
      2. Availability
      3. Security
      4. Maintainability

Compatibility

1. **Introduction**

1.1 Purpose

1.1.1 General Purpose

TrackMe is a company that wants to offer some software-based services: Data4Help, AutomatedSOS and Track4Run. The core service is Data4Help and the others are thought as possible integrations of it. This document’s purpose is to deeply describe all the proposed applications to provide a support for the stakeholders.

TrackMe wants to offer the possibility to third parties to monitor the health status and position of users through the Data4Help service. The application has to acquire the users’ data in some way (ex: through a wearable device) with some device and offers the possibility to third parties to access them. Data can be queried in a specific way or in an aggregate way: in the first case the request must be accepted by the user. Third parties can make specific requests or ask to access to data as soon as they are recorded by the application asking for subscription. TrackMe wants to ensure also that the access to aggregate data let them anonymous: Data4Help will make data available only if anonymity can be granted. The application offers also the possibility to the user to insert some of its data and to consult them and its registered data and stats.

TrackMe also wills to exploit the possibility of recording users’ data to offer another service: AutomatedSOS. Its aim is to support third parties in monitoring health status of the applications’ users acquiring their vital signs through some device (as for Data4Help). This service is thought especially for elderly people (although it can be activated by any user) and is thought to automatically activate a request for the emergency services (ex: departure of ambulance) of third parties.

Finally, TrackMe wants to offer a service to track athletes participating to a run. This service is called Track4Run and, in this case, the application allows third parties that wants to organize a run to do it defining its path, the athletes to enroll for a run and offers the possibility to follow the run to every user tracking runners’ position during the manifestation. For this functionality it is assumed that the competitions are organized by third parties that are recognized organisms.

1.1.2 Goals

For the description of the goals the analyzed perspective is the one of the S2B:

* G1: The application must allow third parties to monitor location and health status of individuals (with their permission) and groups..
* G2: The data related to the users must be anonymized by the system in case of aggregate queries.
* G3: Must allow users to consult its correctly registered stats and data
* G4: In case of emergency, must guarantee a reaction time (in reporting the emergency) of less than 5 seconds from the time the health parameters are out of bounds.
* G5: The system must allow third parties to organize and manage a run
* G6: The system must allow users to enroll to an organized run
* G7: The system must allow users to see on a map the position of all runners during a run

1.2 Scope

The Data4Help service is offered to common users and to third parties that want to acquire data (health status and location) about them or, maybe, about their own clients, so it is thought for companies that maybe do not have the appropriate competences internally and have to be supported in the IT management: the service stands in the middle. The S2B will give to the user the possibility to insert its own body measurements, possible pathologies and eating habits (it’s not mandatory to insert these data) and will monitor and register his heartbeat, his position, his body temperature, his walking/running covered distance and his energy consumed. The system will also allow the user to choose which data to register and which not. Data4Help, besides helping users to monitor their health and position statistics and to consult their inserted data, supports companies in the analysis of the mentioned types of users’ data and allow them, for example, to fragment their clients according to their age, their mobility, the places they visit etc. The user can, obviously, accept or refuse the data acquisition’s request by the third party. It must be assumed that users’ devices are capable of acquiring the mentioned data (sensors + GPS): if a sensor for some kind of data is not present that data won’t be available. The authorized personnel of the third party can access the data logging in and querying the TrackMe’s platform on the computer systems of the company (both users and third parties have first to register to the system). The system relies on the fact that all the users can be identified with a unique key (their fiscal code) and so the third party can ask to access their data through it. Data can be queried in two ways: the third part can make a request to the system to retrieve health status’ or location’s data of a single customer or it can ask for aggregate data on the base of some parameter (ex: data of all users with a certain age, with certain body measures, of all users that work in a certain area etc.). The third party can also request to the system to receive users’ data in a live way, as soon as they are produced without the necessity to make a query. Users will be notified of individual requests and will have the possibility to accept or refuse them. Individual requests may also represent requests of subscription to user’s data and, also in this case, users will have the possibility to accept or refuse them. In case of acceptance, they will be then allowed to remove their subscription at any time. The aggregate requests are handled directly by the Data4Help applicative that will provide data only if they can be showed in an anonymous way otherwise it will notify the third party that is impossible to satisfy the request: TrackMe makes data available if, and only if, the query is satisfied by at least 1000 users’ data.

To offer the AutomatedSOS service the user directly agrees to his data processing when adding the service (he won’t be queried every time, but will give his consent only once at the beginning). In this case the service monitors the users’ health parameters and automatically signals the emergency to the third party that is notified when certain health’s parameters go below or over certain thresholds so that an ambulance can be sent to the customer’s location to help him (this responsibility is left to the third party exploiting AutomatedSOS service, AutomatedSOS has just to report theemergency*).* The service should guarantee a reaction time in reporting the emergency of less than 5 seconds from the moment in which the parameters go out of certain bounds : in this case it must be assumed that the users’ device send data in real time to guarantee a right functioning of the service.The system provides the encoding of the call to the ambulance, the location and health status of the person as a reaction to the person’s health problem that belongs completely to the environment. This service is thought to be exploited on one side by the users and on the other especially by third parties as public authorities’ that, having access to such system, want to monitor the mentioned citizens’ parameters and want to protect their health status (it is not very useful for companies that can’t provide emergency services).

For what concerns the Track4Run application, in this case TrackMe offers a service that can be exploited by an organizer of a run to arrange the run and its path, by the users willing to participate to a run to enroll for the competition and also just to follow the evolution of the run. The system offers the possibility to organize runs to third parties recognized by the State and to enroll for them to all users.

Both AutomatedSOS and Track4Run rely on the assumptions made for Data4Help and exploit its features.

1.3 Definitions, acronyms, and abbreviation

1.3.1 Definitions

* **User:** the ‘normal’ customer of the application that exploits the application only to collect his own data or to be monitored for SOS and to enroll for a run or follow its development;
* **Third party:** the customer of the application that exploits it to monitor data of “normal” customers, to provide the SOS service or to organize a run;
* **Customer:** general TrackMe customer, can be a user or a third party.
* **Warning:** a message shown to the customer when it must be notified about something (ex: inserting a wrong input);
* **Health status:** a set of parameters which indicate the health status of a user;
* **Run:** event created by a recognized third party to which the users can enroll;
* **Emergency Call:** a message containing id, position and health status of the sender.
* **Individual Request:** request on some single user’s data made by a third party.
* **Aggregate Request:** request on some users’s grouped data made by a third party.

1.3.2 Acronyms

* API = Application Programming Interface
* GPS = Global Positioning System
* UI = User Interface
* TP = Third party
* S2B = Software To Be
* PC = Personal Computer

1.3.3 Abbreviation

* Gn = nth goal
* Dn = nth domain assumption
* Rn = nth requirement

1.4 Revision History

* Version 1.0:
  + First Release

1.5 Reference Documents

* Specification document: “Mandatory Project Assignment AY 2018-2019”
* IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.
* UML diagrams: <https://www.uml-diagrams.org/>
* Alloy doc: <http://alloy.lcs.mit.edu/alloy/documentation/quickguide/seq.html>

1.6 Document Structure

The RASD document is composed by five chapters, as outlined below:

**Chapter 1** is an introduction: it describes the purpose of the system ~~in words~~ informally and also by making use of the list of goals which the application has to reach. Moreover, it defines the scope, where the aim of the project is defined in greater detail and the application domain and the most important shared phenomena are shown*.*

**Chapter 2** offers an overall description of the project. Here the actors involved in the application’s usage lifecycle are identified and the boundaries of the project are defined, listing all the necessary assumptions. Furthermore, a class diagram is provided, aid to better understanding the general structure of the project, with all the related entities. Then some state diagrams are listed to make the evolution of the crucial objects clear. Finally, the functions offered by the system are here more clearly specified, with respect to the previously listed goals.

**Chapter 3** represents the body of the document. It contains the ~~external~~ interface requirements, which are: user interfaces, hardware interfaces and software interfaces. It then lists some scenarios to show how the system acts in real world situations, followed by the description of the functional requirements, using use case and sequence diagrams. All the requirements necessary in order to reach the goals are given, linked with the related domain assumptions. Lastly, the non-functional requirements are defined through performance requirements, design constraints and software system attributes.

**Chapter 4** contains the Alloy model of some critical aspects with all the related comments and documentation in order to show how the project has been modeled and represented through the language.

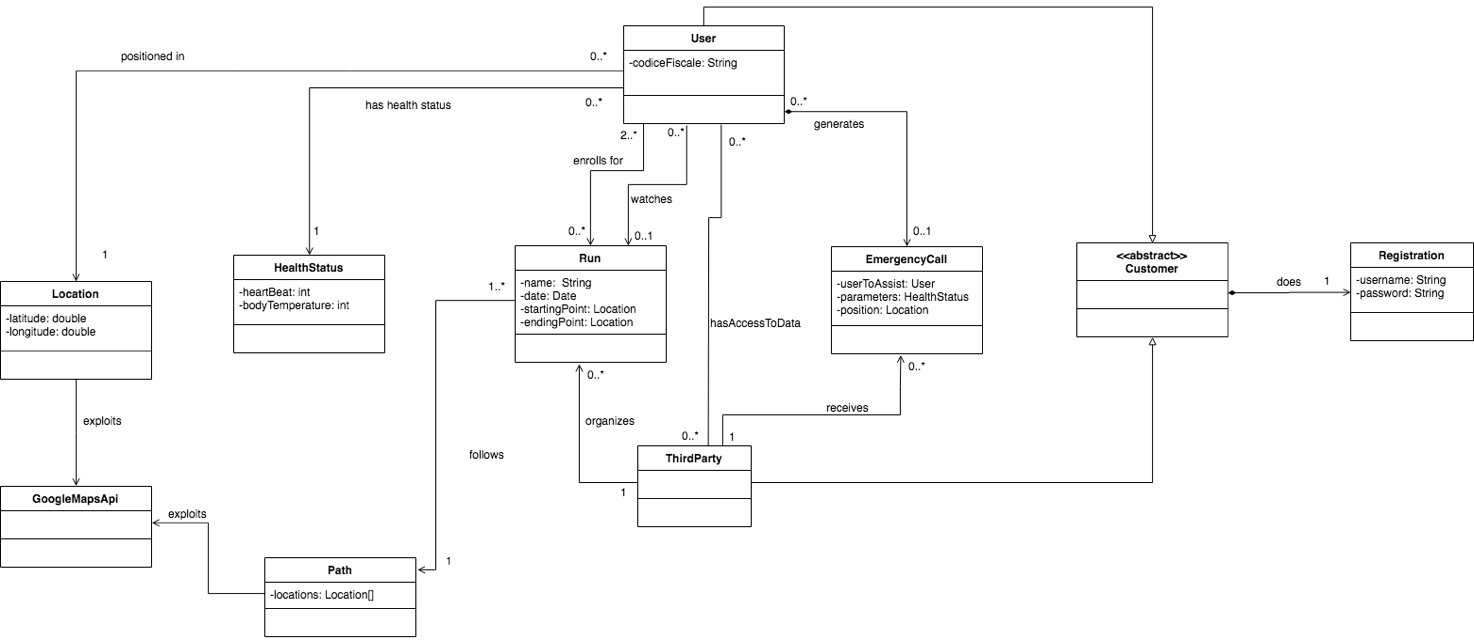
**Chapter 5** shows the effort which each member of the group spent working on the project.

**2. Overall Description**

2.1 Product Perspective

The idea is to build AutomatedSOS and Track4Run upon Data4Help: they are additional services that can integrate Data4Help and they can be activated also in a second moment providing some additional information. To monitor the position and the other data of its user, the application exploits his device’s available sensors and GPS and for the managing of a run the system exploits Google Maps’ APIs.

The below high-level class diagram provides a model of the application domain. Third parties and users communicate through the system provided by TrackMe: the third party can make a query for individual or aggregate data to the system exploiting the Data4Help service and the user, in the first case, can agree or refuse giving an answer to the system, while, in the second case, it can be the system that refuses to provide data if it can’t guarantee their anonymity; the third party can also ask to the system through a request that must be accepted to subscribe to users’ data (both individual or aggregate) if it wants to receive them as soon as they are produced and, also in this case, the user will have the possibility to accept or refuse the subscription in case of individual request. The third parties can also activate the AutomatedSOS service to receive emergency calls and activate the Track4Run service to organize and manage run competions. The same goes for the users that can have the possibility to automatically generate emergency calls and to enroll or follow run competitions activating the appropriate services.



Class diagram

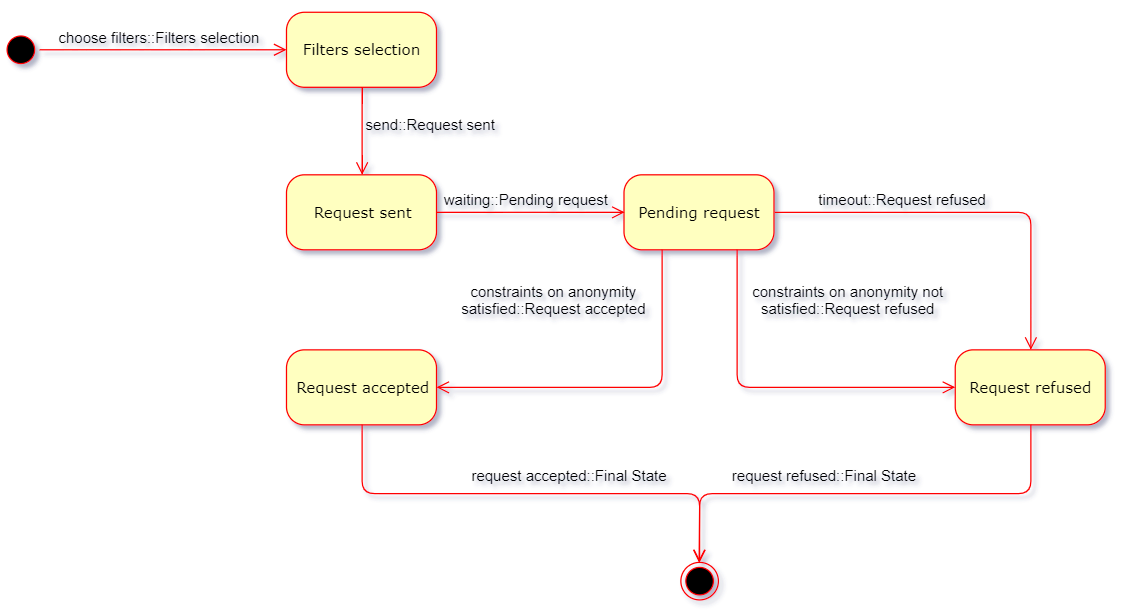
The ‘ThirdParty’ has a reference to the ‘User’ so that the system can retrieve the queried data. ‘EmergencyCall’ has been associated through the composition to the ‘User’ because it has no sense (on a logical level) to exist if it is not referenced by its ‘User’ (the same argument holds for ‘Registration’ and ‘Customer’). ‘EmergencyCall’ contains a reference to the ‘HealthStatus’ and ‘Location’ of ‘User’ to guarantee a direct access to them even if they are retrievable from ‘User’ itself.

Now we are going to analyze some critical aspects of the application, modeling their behaviors and showing the evolution over time of their states through appropriate state diagrams, which are reported below.



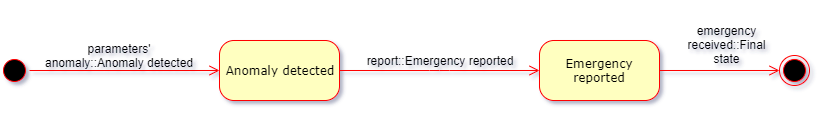
State diagram 1 - Request for individual data

In this first state diagram (State diagram 1), the request sent to an individual by a third party to ask for his data is modeled, from the dispatch to the final state, which depends on the individual’s choice.



State diagram 2 - Request for aggregate data

Also in this case (State diagram 2), to be modeled is the request by the third party, but this time it is a request for aggregate data, so the flow is different. Indeed, at first the third party is asked to select the filters it wants to apply (age, geographic area etc.), then the evolution of the states is similar to that in the previous diagram, until the check for anonymity, which results in a positive or negative response, depending on the fact that the system can guarantee the privacy of their users.



State diagram 3 - SOS request

This state diagram (State diagram 3) is inherent to the AutomatedSOS service. Indeed, it shows the evolution of the states necessary to handle an emergency, from the detection of an anomaly in the user’s parameters to the submission of the emergency with all the relevant data to the third party, so that it can send an ambulance to assist the user.



State diagram 4 - Run organization

Finally, this state diagram (State diagram 4) represents the transformation of the event (intended as run) entity. It’s important to note that ‘Data insertion’ and ‘Path selection’ are two different states. This is because the process of path definition is intended to be subsequent to the insertion of data such as the name of the event, the date on which the run will take place etc. Indeed, it could be that some of these data can be necessary for the system to decide if the path defined by the organizer is feasible or not. If an update in the organization of the run is necessary once the event has been created the system will redirect the third party to the ‘Data Insertion’ state regardless of the type of update.

2.2 Product functions

In the following section the most important product functions of the system are reported. It’s important to underline that the AutomatedSOS and Track4Run services are built on top of Data4Help, so if the user enables one or both of them, Data4Help and its product functions are still enabled.

**2.2.1 Monitoring management**

This product function is one of the most important for the system. The system will allow the user to sign up entering a username, a password and his/her data. Some data will be mandatory (the anagraphical ones), while some other data will not (the body measurements, the pathologies and diet informations). Other data of the user will be collected by the S2B periodically, like the heartbeat and the location. The system will have to allow the third party that wants to retrieve some data, to choose between accessing the data of a specific individual or accessing anonymized data of groups of individuals. In the former case, the third party will provide the fiscal code of the user, that will accept or refuse to provide the data. In the latter case, the third party will write a constraint about what kind of individuals must be in the group (for example the individuals whose age is above 50 years) coupled with the required data. In order to protect the privacy of the individuals and avoid a misuse of the data, the request will be automatically refused if the group has less than 1000 people. The system will also allow the third party to subscribe to new data and receive it as soon as they are produced.

**2.2.2 SOS management**

This product function must be guaranteed by the system if and only if the user enables the AutomatedSOS service. It is necessary that some third parties, that in this case are thought to be companies that can provide emergency services, register to the system so that it can send them the parameters of the user in case of emergency, as it will be explained in the following. The system will monitor in real time, through an appropriate wearable device the vital parameters of the user. As soon as those parameters are checked to be out of some thresholds, the system will send location and health status of the user to the nearest registered third party’s application, so that it will be able to send as soon as possible an ambulance with the appropriate equipment and medical staff. The S2B will care only of reporting the emergency: the actions taken to handle it are out of the service’s purpose.

**2.2.3 Run management**

This product function must be guaranteed by the system if and only if the user enables the Track4Run service. In this case the third parties are considered recognized entities that will to organize some runs. The authorized third parties will have the possibility to create the event of a run in the system, and will put some mandatory data about it: the name, the path, the starting point, the ending point, the date and the starting time and the maximum number of participants. The system will allow the users that enabled this service to enroll for an organized run, showing them the list of organized run on a calendar. The system will also provide to users the possibility of following the developing of a run by showing a map and the position of the runners in real time. So, the location of the runners will be captured by the system in real time, using a mobile device provided with GPS.

2.3 User characteristics

The actors of the application are the following users:

1. User: a person that registers to Data4Help and, possibly, to AutomatedSOS and/or Track4Run. The user’s role will be explained better in the following distinctions (depending on the service):

* Data4Help: the user is a person who provides his/her data by explicitly writing it or indirectly by a device that collects the data and send it to the system. The user can see his/her data on the application.
* AutomatedSOS: the user is a Data4Help’s user who enables this service. He provides to the system his health status in real time and his location in case of health emergency (both collected and sent by his device).
* Track4Run: the user is a Data4Help’s user who enables this service. He can enroll into run competitions. Also, during ongoing competitions, he provides his location in real time to the system, (collected and sent by his device).

1. Third party: it is necessary to distinguish what the third party is depending on the service:

* Data4Help: the third party is any organization that wants to access data of the users. It has to register to the system in order to receive data, which must be explicitly asked.
* AutomatedSOS: the third party, supposed to be a company that can provide emergency services, is an organization capable of providing medical aid to the users after receiving the report from the system. It has to register to the system in order to receive the data in case of emergency.
* Track4Run: the third party, supposed to be a recognized institution, will be able to organize and manage runs.

2.4 Assumptions, dependencies and constraints

* D1: The devices that acquire users’ position provide location with an error of 5 meters at most.
* D2: The sensors of devices that acquire users’parameters provide informations with an error of at most 1% (for each type of data) compared to the real values .
* D3: Each fiscal code number is uni­­que.
* D4: A request for data is considered to be anonymous if it is satisfied by at least 1000 individuals.
* D5: Each user that wills to activate AutomatedSOS service always wear the device that acquires data.
* D6: The devices on which the services are exploited can provide real time information.
* D7: Third parties that want to organize a run have demonstrated that they are acknowledged institutions.
* D8: Each user that participates to a run wears the device acquiring data during the whole competition.
* D9: The internet connection works properly without failure.

Any device that can connect to internet through a browser (for example a PC, a mobile phone, a tablet, etc.) is enough for both the user and the third party to do those activities that don’t require data acquisition through devices (ex: signing up, logging in, organizing a run, requiring data from a user, accepting or refusing data requests from third party). To exploit these various functionalities needing an internet connection devices with a 2G/3G/4G and, possibly, WiFi connection, must be present on the customer’s device.

However, some services offered by the system require some other specific hardware: in order to locate the user it’s necessary for him to have a GPS device and to correctly report emergencies all the appropriate sensors must be present on the weareable device (ex: smartwatch).

For what concern the exploited software interfaces, the system uses an external suitable service so that its architecture is simpler. The Google Maps API is used when providing the Track4Run service. The organizer will choose the path of his run by managing a map provided by the API. The user that wants to spectate a run, as well, will be provided of a map by the API.

The S2B is thought to be distributed in Italy since it exploits the fiscal code number to distinguish the users. Finally the S2B must be compliant with the GDPR normative for the privacy.

***3*. Specific requirements**

3.1 External interface requirements

3.1.1 User Interfaces

The following mockups give an approximate idea of how the application’s interfaces should appear: some of the most important screenshots of the interactions between the system and each of its customers (User and Third Party) are represented.

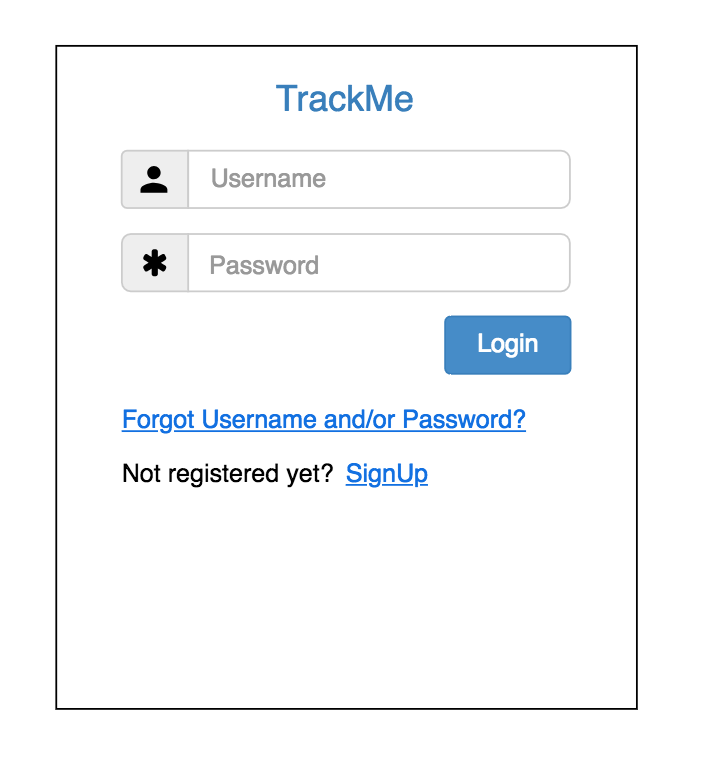
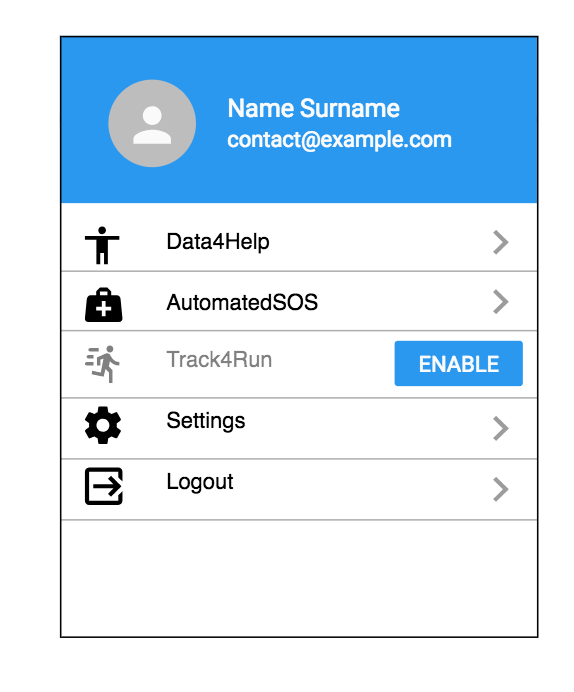
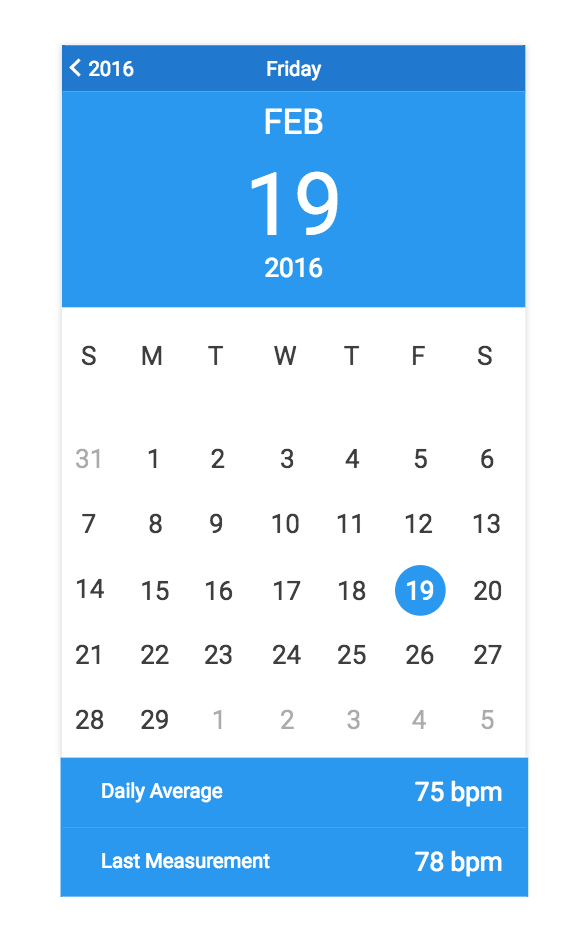


Figure 2 Mockup - Main menu

Figure 1 Mockup - Login



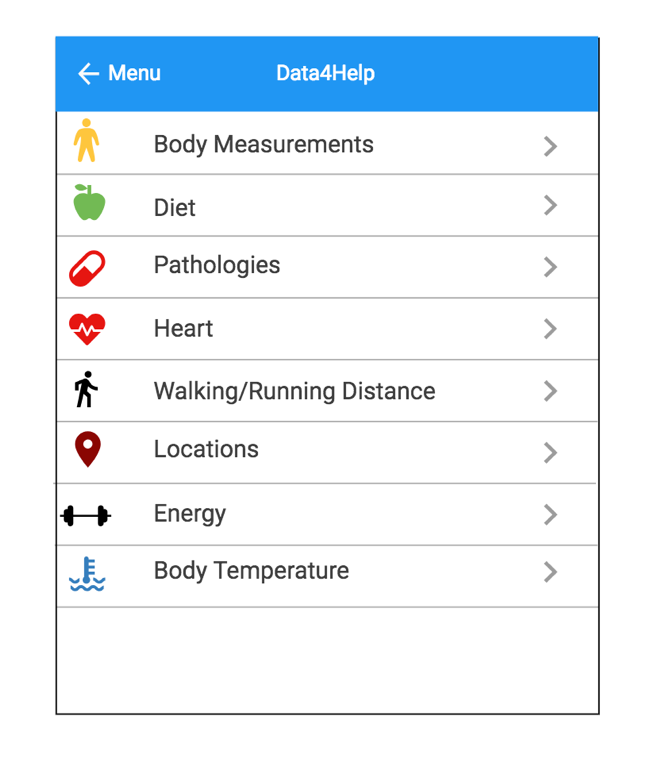


Figure 4 Mockup – User UI: access to ‘Heart’ functionality

Figure 3 Mockup – User UI: Data4Help menu

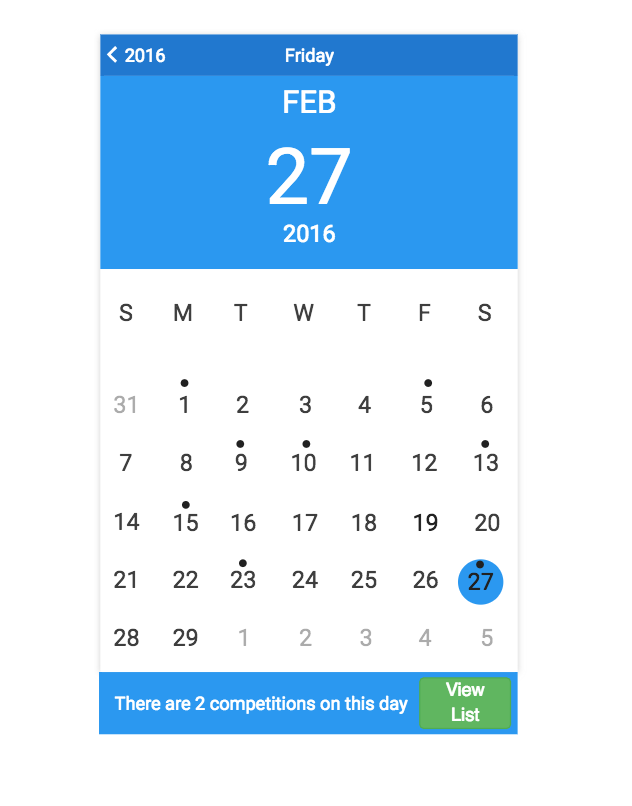
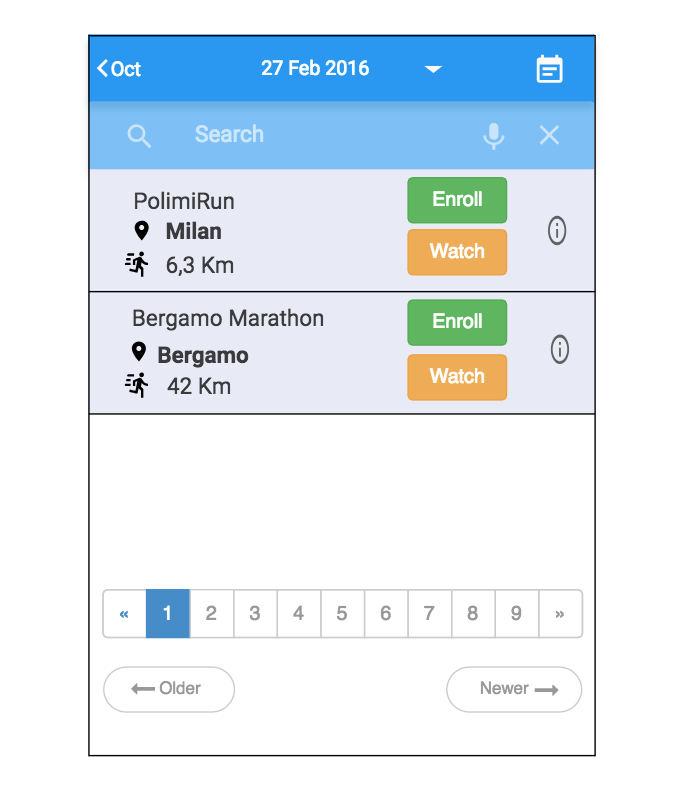
******

Figure 6 Mockup – User UI Track4Run: list of scheduled competitions

Figure 5 Mockup – User UI Track4Run: view of days with at least one competition

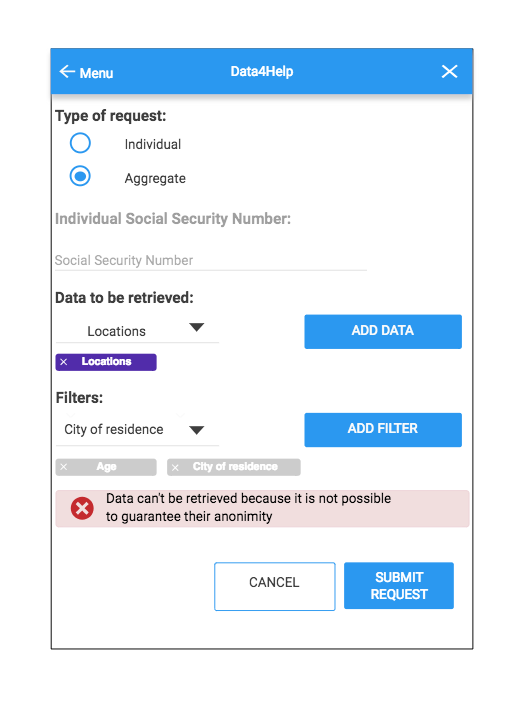
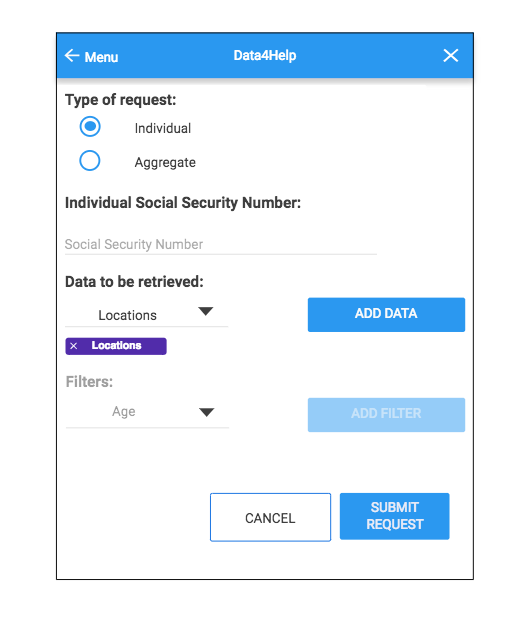
******

Figure 7 Mockup – TP UI Data4Help: individual request

Figure 8 Mockup – TP UI Data4Help: group request

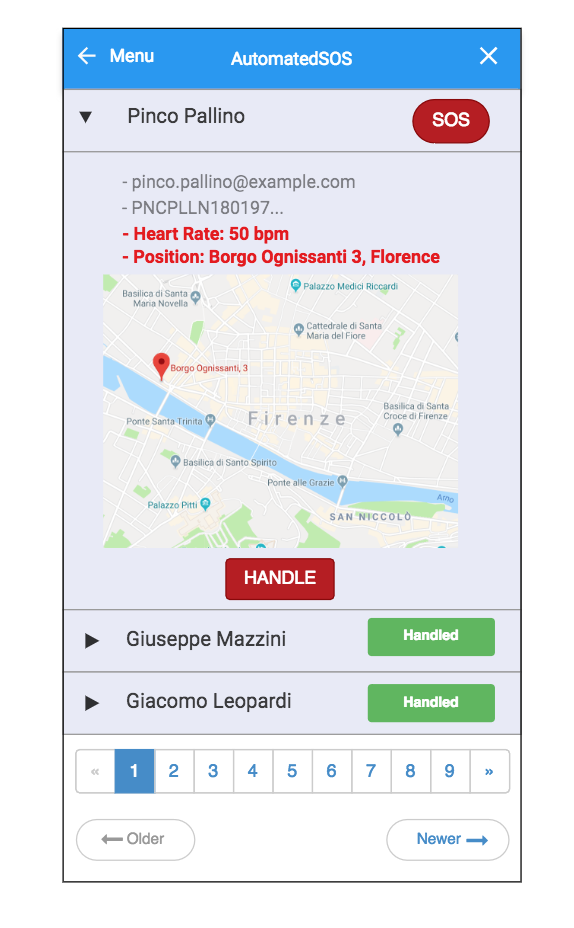
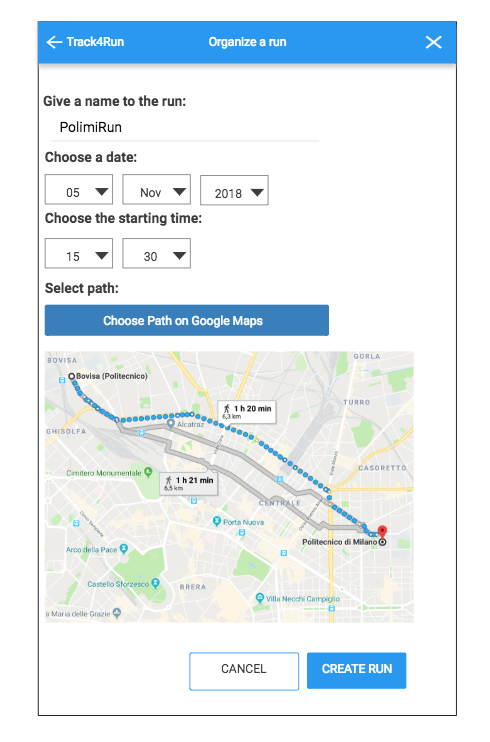
******

Figure 10 Mockup – TP UI Track4Run: organize a run

Figure 9 Mockup – TP UI AutomatedSOS: list of reportings

Figure 1 and Figure 2 represent a possible login and menu interface for any kind of customer (both User and Third Party) while the other mockups represent just one side of the interaction with the system offered by TrackMe (specified in the captions).

3.1.2 Hardware interfaces

The system has no hardware interface.

3.1.3 Software interfaces

The system doesn’t provide any API.

3.2 Functional requirements

3.2.1 User

*Scenarios*

Scenario 1

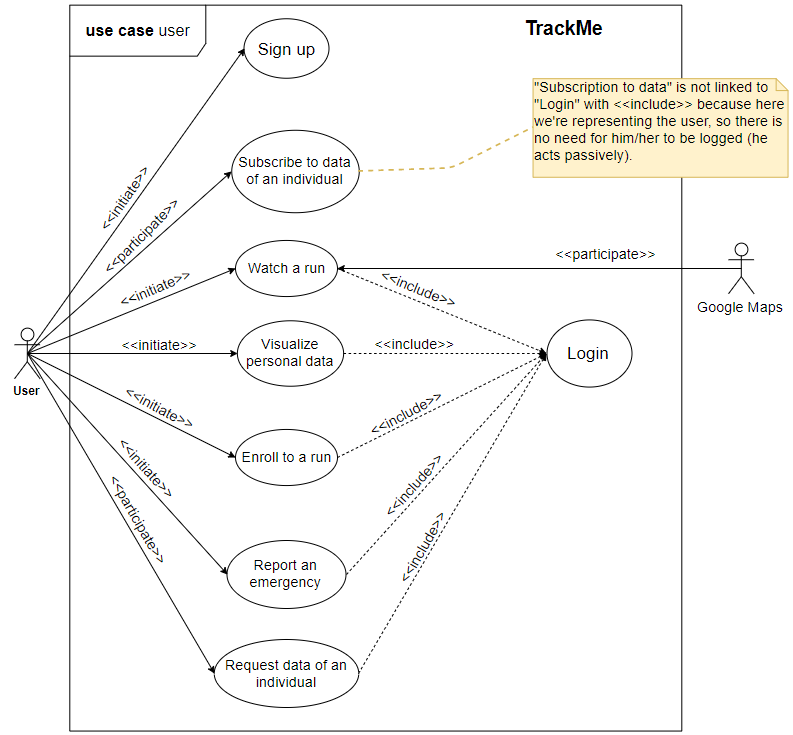
Tyrion, an elderly man who lives alone, taking advantage of the sunny day, decides to fix up the yard, despite the fact that his doctor ordered him to not push himself too hard to avoid unpleasant inconveniences.

Indeed Tyrion should have listened to the doctor, because after a little more than one hour he starts to feel fatigued. Fortunately he is wearing a wearable devices with the AutomatedSOS service offered by the company TrackMe installed, which detects that the man's parameters are below the threshold and immediately calls an ambulance, allowing Tyrion to be still alive.

Scenario 2

Marco is a competitive and sportsman person, so he wants to enroll for a run. He decides to use the Track4Run service, in order to find a run. He finds it and enrolls for it. The run will start in a month, so he decides to train for it with his friend Federico, that will also partecipate in the competition. While they are running together, Marco stumbles into a stone and breaks his knee. Unfortunately he won’t be able to take part to the competition, but he really would like to support his friend Federico at least. He finds out that it is possibile to monitor the position of a runner in real time during the competition, thanks to the Track4Run service. So, while he is lying down on the bed of his hospital, he is able to keep track of how well is Federico’s run going.

*Use Case Diagram*



Use case diagram - User

*Use Cases*

|  |  |
| --- | --- |
| Name | Sign up |
| Actor | User |
| Entry conditions | The user has opened the application on his/her device. |
| Events flow | 1. The user chooses the “Sign up” option. 2. The user fills the mandatory fields. 3. The user fills the optional fields with not mandatory data. 4. The user chooses the confirmation option. 5. The systems saves the data. |
| Exit conditions | The user is registered and the system has his/her data stored. |
| Exceptions | 1. The user was already registered. In this case the system warns the user and suggests him/her to do the login. 2. The username is already taken. In this case the system warns the user and suggests him to change the username. 3. The username doesn’t fill all the mandatory fields. In this case the system warns the user and notifies him which fields were left unfilled. |

|  |  |
| --- | --- |
| Name | Login |
| Actor | User |
| Entry conditions | 1. The user has opened the application on his/her device. 2. The user has already done the “Sign up” activity. |
| Events flow | 1. The user chooses the “Login” option. 2. The user enters username and password in the respective fields. 3. The user chooses the confirmation option. |
| Exit conditions | The user is logged in and the system allows the user to visualize and manage his account and his data. |
| Exceptions | 1. The user enters the wrong username. 2. The username enters the wrong password.   In both cases, the system warns the user and notifies him which field is wrong, suggesting to correct it. |

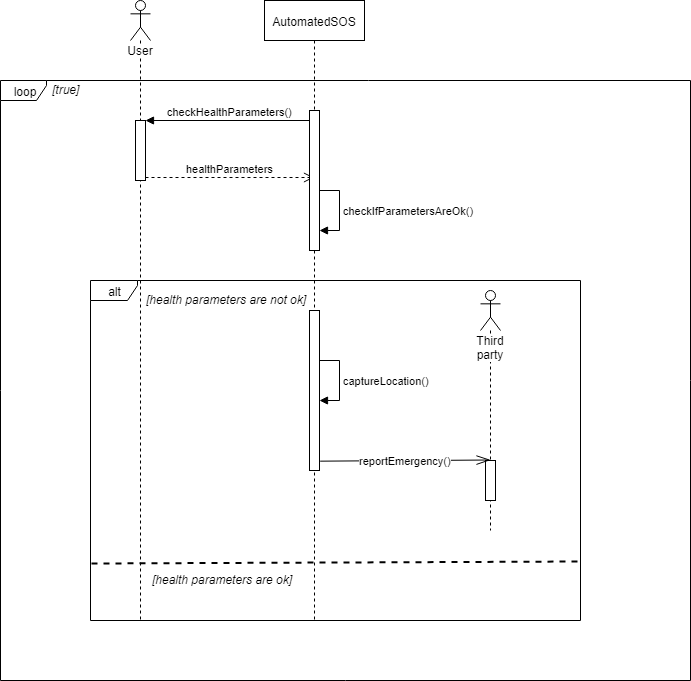
|  |  |
| --- | --- |
| Name | Visualize personal data |
| Actor | User |
| Entry conditions | The User has already done the “Login” activity. |
| Events flow | 1. The user selects the type of data that he wants to consult. 2. The system provide the single data if they are of ‘static’ type (data inserted manually by the user such as his weight) or with average and last measurement if they are of ‘dynamic’ type (data measured by the system such as the heartbeat). |
| Exit conditions | The user is provided with the requested data. |
| Exceptions | / |

|  |  |
| --- | --- |
| Name | Report an emergency |
| Actor | User, Third party |
| Entry conditions | The user is a member of the service AutomatedSOS. |
| Events flow | 1. The system detects that the parameters of the monitored user are below or above the defined threshold. 2. The system sends an alert to the third party, notifying the emergency and sending the location and the status of the user. 3. The third party receives the emergency call. |
| Exit conditions | The emergency is correctly reported to the third party. |
| Exceptions | \ |
| Special Requirements | The alert is received by the third party within 5 seconds from the detection of parameters below or above the threshold. |

|  |  |
| --- | --- |
| Name | Watch a run |
| Actor | User |
| Entry conditions | 1. The user has activated the service Track4Run. 2. The user has successfully logged in. |
| Events flow | 1. The user selects the run he wants to see. 2. The user is allowed to select a runner to follow him or to navigate on the map and focus on a certain area. |
| Exit conditions | The user can see the progress of the run and the participants’ position on the map. |
| Exceptions | 1. The user selects a run that is not going on now, but is already ended or not still begun. The system notifies the user with a warning. |

|  |  |
| --- | --- |
| Name | Enroll to a run |
| Actor | User |
| Entry conditions | 1. The user has activated the service Track4Run. 2. The user has successfully logged in. |
| Events flow | 1. The user opens the service Track4Run. 2. The user selects the run he wants to enroll for from a list provided by the system containing only the competitions to come. 3. The user sees a recap of the information about the run. 4. The user confirms the participation. 5. The system send to the user a receipt of his enrollment. |
| Exit conditions | The user is regularly registered to the run. |
| Exceptions | 1. The user tries to enroll for a run that reached the maximum number of participants. The system notifies the user with a warning. |

*Sequence Diagrams*



Sequence diagram 1 – Report an emergency

3.2.2 ThirdParty

*Scenarios*

Scenario 3

Giovanni suffers from heart’s problem, and a private hospital is taking care of him. After some analysis, the doctors of the hospital find out that his health has improved, and it is only necessary to monitor his heartbeat. The hospital would like to allow Giovanni to go home so that his bed is free for another patient, but monitoring his heartbeat constantly is needed. The hospital finds out that monitoring his heartbeat is possibile even though he is at home or in any other place, thanks to Data4Help. It just needs to sign up and log in to the service and ask Giovanni’s heartbeat by providing his fiscal code. Giovanni, who also needs to sign up and log in to the service, will just accept the request and the hospital will see his heartbeat. The hospital can also ask for a subscription in order to get new data as soon as they are produced.

Scenario 4

ISTAT is doing some research activity in order to analyze the health status of smoking people. In order to do that, it requests to the system to access the data of those who live in Milan and smoke. Specifically, ISTAT wants the heartbeat of those people. Data4Help provides this data to ISTAT.

Scenario 5

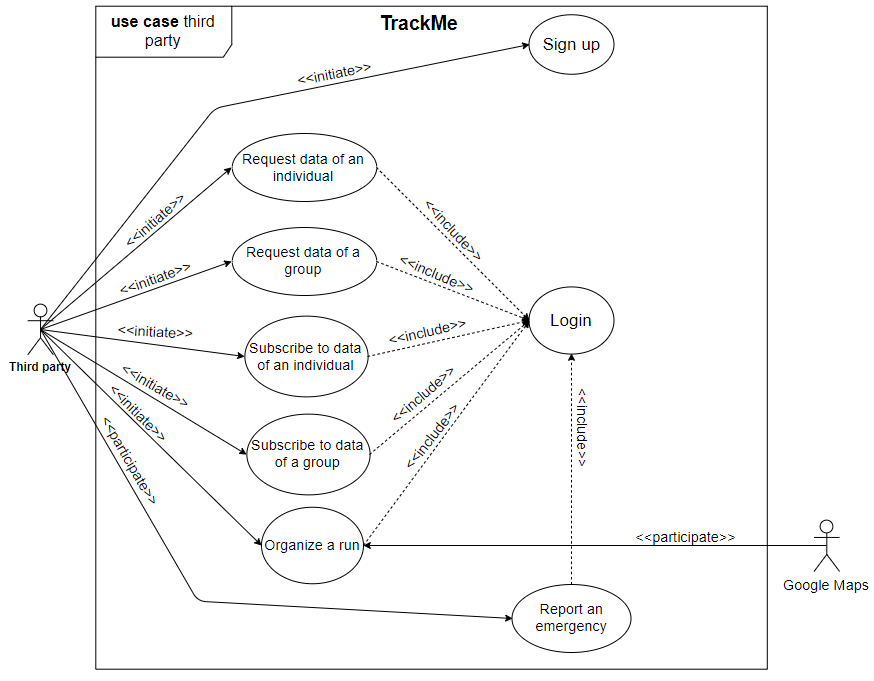
An emergent start-up, located in a small city, provides a food delivery service, and has the peculiarity of selling vegan food only. The company wants to know the location of vegan people in the small city, in order to do some targeted advertising. So they ask that piece of information to Data4Help, that stores the eating habits of their users. Because the city is small, there are only 853 vegan people in there, so Data4Help refuses to provide the requested data in order to protect their anonymity.

Scenario 6

Politecnico di Milano decides to organize the annual run called PolimiRun. Once registered to the Track4Run service offered by the TrackMe company, the organizers can define all the useful information to make the run enjoyable by the participants, including the path which the latter will have to travel.

Furthermore, through the services offered by TrackMe, Politecnico can ask for the data of the user involved into the run to eventually analyze them.

*Use Case Diagram*

**

Use case diagram – Third party

*Use Cases*

|  |  |
| --- | --- |
| Name | Request data of an individual |
| Actor | Third party, user |
| Entry conditions | The third party has already done the “Login” activity. |
| Events flow | 1. The third party chooses the “Request data of an individual” option. 2. The third party insert the type of data it wants to retrieve. 3. The third party writes the identifier of the individual (his fiscal code). 4. The third party chooses the confirmation option. 5. The system queries its database with the third party’s request and finds who is the user. 6. The system forwards to the user the third’s party request. 7. The user sees the request forwarded by the system. 8. The user chooses the “Accept” option. 9. The system provides to the third party the requested data about the user. |
| Exit conditions | The third party obtains the data about the requested individual. |
| Exceptions | 1. The user chooses the “Refuse” option instead of the “Accept” option. In that case, the system notifies the third party that the request was refused by the user. 2. There is no user registered in the system with the identifier specified by the third party. In that case, the system warns the third party and suggest him to try with another identifier. |

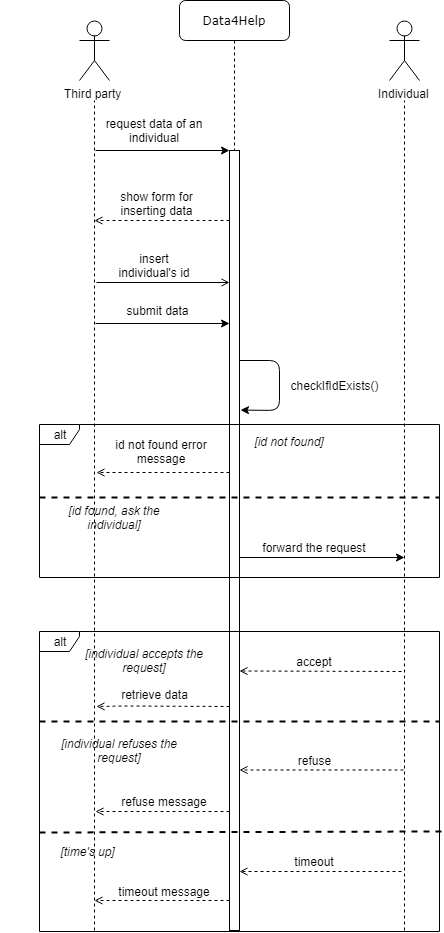
|  |  |
| --- | --- |
| Name | Request data of a group |
| Actor | Third party |
| Entry conditions | The third party has already done the “Login” activity. |
| Events flow | 1. The third party chooses the “Request data of a group” option. 2. The third party inserts the type of data it wants to retrieve. 3. The third party fills the fields with the constraints of which groups of individuals the third party is looking for. 4. The third party chooses the confirmation option. 5. The system queries its database with the third party’s request. 6. The system gives the requested data to the third party. |
| Exit conditions | The third party obtains the data about the groups of individuals that respect the imposed constraints. |
| Exceptions | 1. The individuals that respect the constraints found by the system are less than 1000. In this case, the system notifies the third party that the requested data can’t be given in order to protect the anonymity of the users. |

|  |  |
| --- | --- |
| Name | Subscribe to data of an individual |
| Actor | Third party, User |
| Entry conditions | “Request data of an individual” activity has been successfully done. |
| Events flow | 1. The third party chooses the “Subscribe to data” option that appears as soon as the data requested have been shown by the system. 2. The system forwards the request of subscription to the user. 3. The user sees the request forwarded by the system. 4. The user chooses the “Accept” option. 5. The systems registers that the third party is subscripted to that data. 6. The user produces some new data. 7. The system provides the data to the subscripted third party. |
| Exit conditions | The third party is registered to the data and the system provides them to it as soon as they are produced. |
| Exceptions | 1. The user refuses the request for subscription. The third party is notified by the system with a warning. |

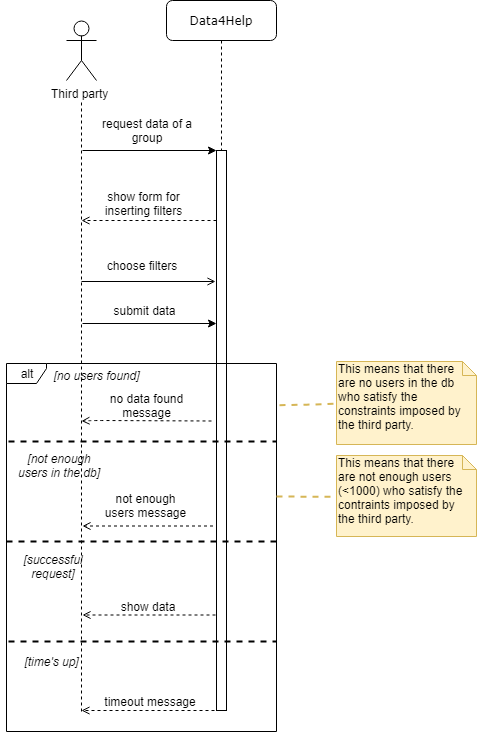
|  |  |
| --- | --- |
| Name | Subscribe to data of a group |
| Actor | Third party |
| Entry conditions | “Request data of a group” activity has been successfully done. |
| Events flow | 1. The third party chooses the “Subscribe to data” option that appears as soon as the data requested have been shown by the system. 2. The systems registers that the third party is subscripted to that data. 3. The group of users produce some new data. 4. The system provides the data to the subscripted third party. |
| Exit conditions | The third party is registered to the data and the system provides them to it as soon as they are produced. |
| Exceptions | 1. The number of users satisfying the request has become less than 1000 so the subscription is canceled until the matching users become again more than 1000 and the third party is notified with a warning. |

|  |  |
| --- | --- |
| Name | Organize a run |
| Actor | Third party |
| Entry conditions | 1. The third party has activated the service Track4Run. 2. The third party has successfully logged in. |
| Events flow | 1. The third party opens the service Track4Run. 2. The third party gives a name to the event. 3. The third party defines the date and time on which the run will take place. 4. The third party defines the path which the participants will travel. 5. The third party defines the maximum number of participants for the competition. 6. The third party publishes the event. |
| Exit conditions | The event for the run is online and joinable. |
| Exceptions | 1. The third party enters invalid data (an invalid name or date and time for the run) or he defines an unfeasible path.   A warning is showed, indicating the wrong parameter(s) and the motivation. |

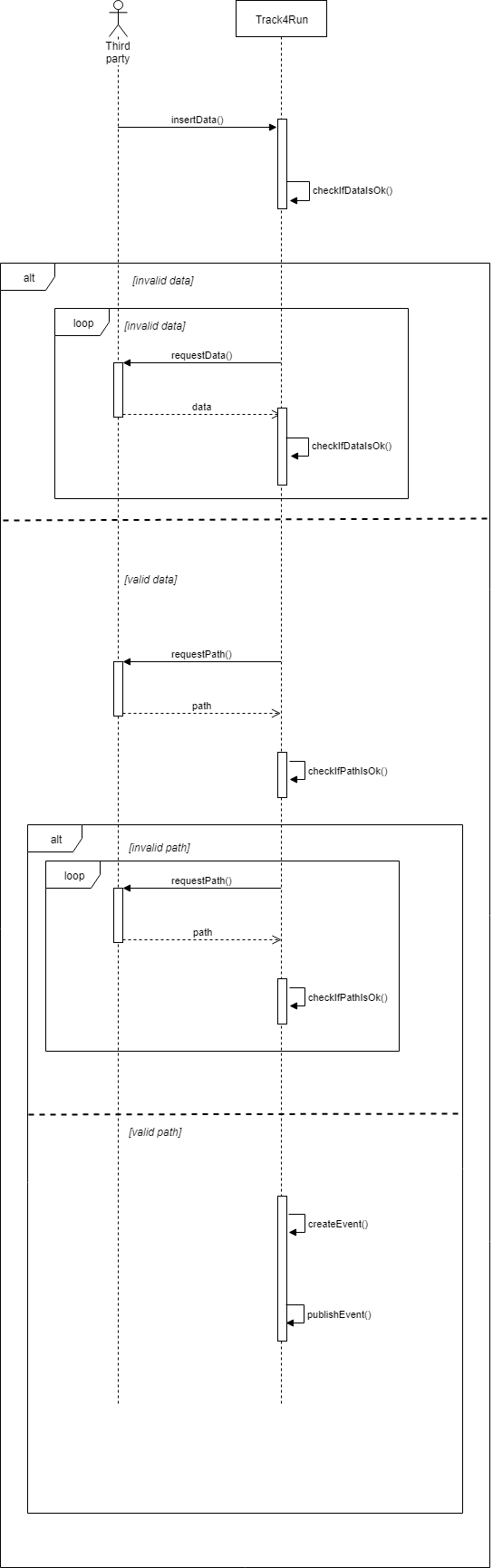
*Sequence Diagrams*

******

Sequence diagram 2 - Request data of an individual

******

Sequence diagram 3 - Request data of a group



Sequence diagram 4 – Organize a run

3.2.3 Requirements

In this section we show that the requirements ensure the satisfaction of the goals in the context of the domain assumptions: the list of requirements and domain assumptions under each goal have this purpose.

**G1: The application must allow third parties to monitor location and health status of individuals (with their permission) and groups.**

* **R1**: The system must allow ~~to~~ all the customers to sign up to the desired services specifying a username, a password and filling at least the anagraphical mandatory fields.
* **R2**: The system must allow to all the customers to log in to the desired services through username and password.
* **R3**: The system must save the collected data of users registered for Data4Help in real time.
* **R4**: The system has to allow the third party that wants to retrieve some data to choose between an individual request or an aggregate one.
* **R5**: In case of a query for data of an individual the system has to ask to the third party the individual’s fiscal code.
* **R6**: In case of a query for aggregate data the system has to ask to the third party which parameters to use to filter data.
* **R7**: When a request for a specific individual’s data is made the system must allow the individual to accept the request or not.
* **R8**: When a request for data is approved the system has to make the previously saved data available to the third party.
* **R9**: The system must provide to the third party the possibility to subscribe to new data and receive them as soon as they are produced both for individuals and for group of individuals.
* **R10**: The system must allow the user to accept or refuse a possible request for subscription by a third party.
* **D1**: The devices that acquire users’ position provide location with an error of 5 meters at most.
* **D2**: The sensors of devices that acquire users’ parameters provide informations with an error of at most 1% (for each type of data) compared to the real values.
* **D3**: Each fiscal code is unique.
* **D6**: The devices on which the services are exploited can provide real time information.
* **D9**: The internet connection works properly without failure.

**G2: The data related to the users must be anonymized by the system in case of aggregate queries.**

* **R11**: The requests by third parties on aggregate data must provide them if and only if the number of individuals whose data satisfy the request is higher than 1000.
* **R12**: When the number of users satisfying an aggregate request (for which a subscription was made) becomes less than 1000 the subscription is automatically canceled until the matching users become again more than or equal to 1000.
* **D4**: A request for data is considered to be anonymous if it is satisfied by at least 1000 individuals.

**G3: Must allow users to consult their correctly ­registered stats and data**

* **R1**: The system must allow to all the customers to sign up to the desired services specifying a username, a password and filling at least the mandatory fields.
* **R2**: The system must allow to all the customers to log in to the desired services through username and password.
* **R3**: The system must save the collected data of users registered for Data4Help in real time.
* **R12**: The system must allow the User to analyze its own data and stats providing him a way to access to all registered data and stats and giving him the possibility to consult both their aggregate (ex: daily average) values and precise measurements.
* **D1**: The devices that acquire users’ position provide location with an error of 5 meters at most.
* **D2**: The sensors of devices that acquire users’parameters provide informations with an error of at most 1% (for each type of data) compared to the real values .

**G4: In case of emergency, must guarantee a reaction time (in reporting the emergency) of less than 5 seconds from the time the health parameters are out of bounds.**

* **R13**: The application on wearable devices, if AutomatedSOS is activated, must send the health parameters and location to the third party as soon as it detect that parameters are out of the defined bounds.
* **D5**: Each user that wills to activate AutomatedSOS service always wear the device that acquires data.
* **D6**: The devices on which the services are exploited can provide real time information.
* **D9**: The internet connection works properly without failure.

**G5: The system must allow third parties to organize and manage a run**

* **R14**: The system has to allow third parties that have activated the Track4Run service to schedule a run providing name, starting and ending point coordinates, the path, the date and time of the competition and the maximum number of participants.
* **D7**: Third parties that want to organize a run have demonstrated that they are acknowledged institutions.
* **D9**: The internet connection works properly without failure.

**G6: The system must allow users to enroll to an organized run**

* **R15**: The system has to allow users who have activated the Track4Run service to enroll for an organized run showing them the organized competitions on a calendar.
* **D9**: The internet connection works properly without failure.

**G7: The system must allow users to see on a map the position of all runners during a run**

* **R16**: The system has to allow users who have activated the Track4Run service to follow the development of a run selecting an ongoing competition from a list that identifies runs by their name.
* **R3**: The system must save the collected data of users registered for Data4Help in real time.
* **D6:** The devices on which the services are exploited can provide real time information.
* **D8**: Each user that participates to a run wears the device acquiring data during the whole competition
* **D9**: The internet connection works properly without failure.

3.3 Performance requirements

The system has to be able to serve a great number of users and third parties simultaneously. It has to guarantee quick, reactive and correct responses. Also, it’s important to underline that AutomatedSOS provides a critical and vital service, so it must be ensured a reaction time of less than 5 seconds from the time the parameters are below the threshold.

3.4 Design constraints

**Standards compliance**

The system adopts precise units of measure (the units used in the USA):

* Heartbeat: [bpm]
* Distance: [Km]
* Temperature: [Degrees]
* Consumed Energy: [Kcal]

**Hardware limitations**

As specified in the “Dependencies and constraints” section, hardware requirements are present only in relation to specific functionalities. For example, every device with an internet connection is fine in order to sign up or log in, while in order to use the AutomatedSOS service, instead, the GPS and other sensors are needed. So in the following, to be concise, are reported all the hardware requirements needed to use every functionality:

* Connection to internet (Wi-Fi/4G/3G/2G)
* GPS
* Sensor for heartbeat monitoring, temperature measuring and to measure walking/running covered distance

**Other constraints**

The system must respect privacy policies, in particular the privacy of the users. In that sense, this is a critical aspect because third parties can request sensible data of the users. In case of a data request of a specific user, no data will be given to the third party unless the user authorizes it. Moreover, the user will see precisely what data he is asked for. In case of a data request of a group of users, it won’t be given if the matching group has less than 1000 persons, in order to avoid a misuse of the data (for instance, if the third party is asking for data about 10-year-old children living in a certain street in Milano and the number of these children is two, then the third party could be able to derive their identity simply having people monitoring the residents of the street between 8.00 and 9.00 when kids go to school).

3.5 Software system attributes

3.5.1 Reliability

The system must be able to run continuously without any interruptions. In order to do that, it must be ensured that the system is fault tolerant. For example, the central server, which contains the data, should be duplicated, just like the running processes, which provide the services. Some techniques, like the FloodSet algorithm, can be adopted to ensure the required reliability.

3.5.2 Availability

As mentioned before, a fault tolerant architecture is needed. It’s important to make different assumptions for each one of the services. Indeed, if Data4Help and Track4Run can be expected to be available 99.9% of the time, AutomatedSOS contains some critical aspects tied to the nature of the service itself. That’s why it is expected to have an availability of 99.999%.

3.5.3 Security

The data provided by the user contains sensitive information, so the security aspect is of primary importance. The central database on which the data reside must be protect by all the necessary measures to avoid any external and internal attack and also to handle malfunctions of the hardware. For the purpose of sending the data, encryption technique must be used in order to guarantee privacy and consistency.

3.5.4 Maintainability

The development of the application must be done so that in the future it will be easy to fix and modify it, according to the circumstances, and also in order to let cost of those operations be cheap. Appropriate design patterns will be used, as it will be better explained in a further document.

3.5.5 Compatibility

The application offers multiple services and can be used by a variety of people. That is why it must be compatible to as many devices and technologies as possible, in order to meet the constraints contained in “Hardware and Software interfaces” sections.