1. ***Introduction***
   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) 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. 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.

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’ subscribed customers acquiring their vital signs through some device (as for Data4Help). This service is thought for elderly people 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 (assuming that the run is a professional one).This service is called Track4Run and, in this case, all the users have to specify their role: the application allows the organizers to set up a run 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.

***1.1.2 Goals***

* G1: Must allow third parties to monitor location and health status of individuals and groups.
* G2: The data related to the users must be anonymized by the system in case of aggregate queries.
* G3: In case of emergency, must guarantee a reaction time (in reporting the emergency) of less than 5 seconds from the time the parameters are below the threshold.
* G4: Must allow users who want to organize a run to define its path
* G5: Must allow participants to enroll to an organized run
* G6: Shall allow spectators 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 customers, 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. So, Data4Help, besides helping users to monitor their health and position statistics, supports companies in the analysis of the mentioned types of users’ data and allow them, for example, to fragment their clients according to their habits, 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). The authorized personnel of the third party can access the data logging in on 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 social security number) and so the third party can 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 he can ask for aggregate data on the base of some parameter (ex: data of all customers with a certain age, with certain body measures, of all customers 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. The request is 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 only if the query is satisfied by at least 1000 users’ data. So, the request for data arrives to the system from the environment, but is observed by the system that provides the appropriate answer after some internal computation (ex: control on the number of individuals that satisfy an aggregate query to verify that anonymity is guaranteed).

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’ data and automatically signals the emergency to the third party that has access to the applicative 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 almost in real time to guarantee a right functioning of the service -> da levare?]. 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 a run and its path, by the participants to a run to enroll for the competition and by the simple users just to follow the evolution of the run. The system offers the possibility to organize professional runs. Each user must authenticate himself when using the application and, if he is an organizer or a runner, he has to prove it through a certificate.

Obviously 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’ user 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 user of the application that exploits it to monitor data of ‘normal’ users, to provide the SOS service or to organize a run*

***1.3.2 Acronyms***

*API = Application Programming Interface*

*GPS = Global Positioning System*

*UI = User Interface*

*TP = Third party*

***1.3.3 Abbreviation***

*Gn = nth goal*

*Dn = nth domain assumption*

*Rn = nth requirement*

***1.4 Revision History [TODO]***

***1.5 Reference documents [TODO]***

***1.6 Document Structure [TODO]***

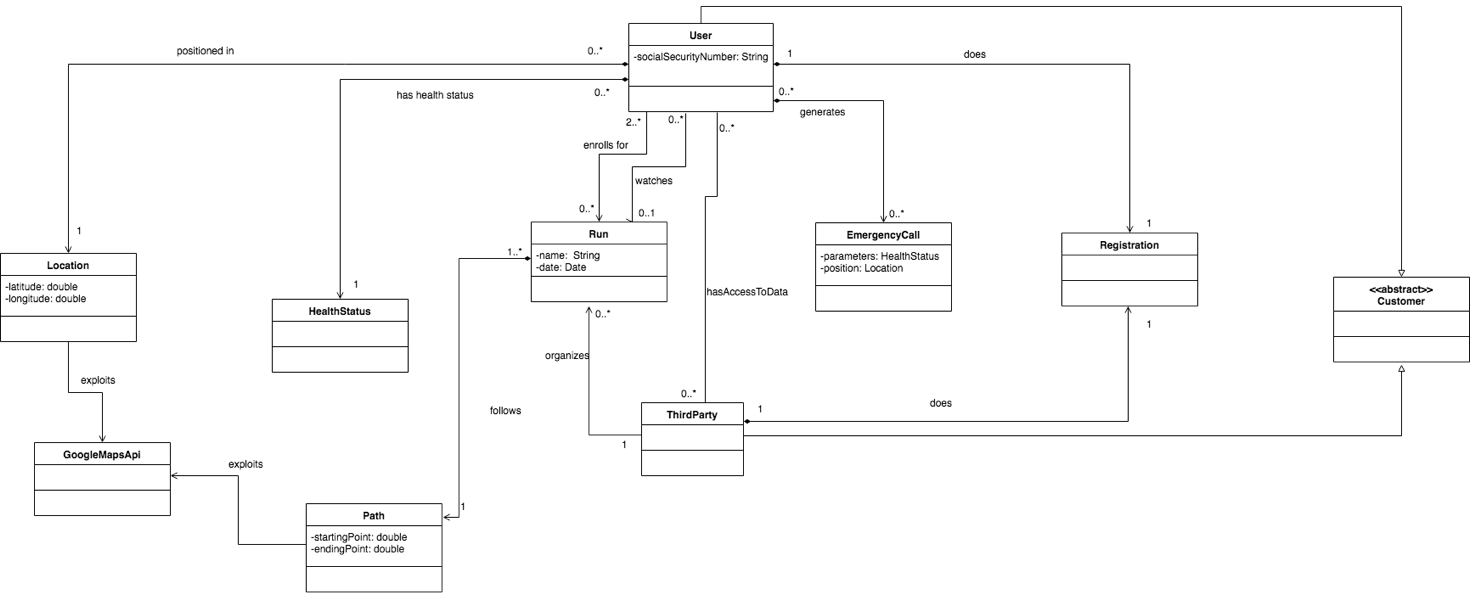
Chapter 1 is an introduction: it describes the purpose of the system also through the goals of the applications and it defines the scope of the system defining in more detail the aim of the project and showing the application domain and the most important shared phenomena*.*

***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 of its user, the application exploits his device’s GPS and to manage the organization of a run the system exploits Google Maps’ APIs and the device’s Calendar app to register the event (ex: for an athlete who wants to participates to a competition or for a user who programs to follow a run programmed in the future).

The below class diagram provides a model of the application domain. ‘ThirdParty’ and ‘User’ communicates through the system provided by TrackMe: the third party can make a query for individual or aggregate data to the system 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 to subscribe to users’ data if it wants to receive them as soon as they are produced. The ‘ThirdParty’ has a reference to the ‘User’ so that the system can retrieve the queried data. ‘HealthStatus’, ‘Location’, ‘EmergencyCall’ and ‘Registration’ have been associated through the composition to the ‘User’ because they have no sense (on a logical level) to exist if they are not referenced by their ‘User’ (the same argument holds for ‘Registration’ and ‘ThirdParty’).



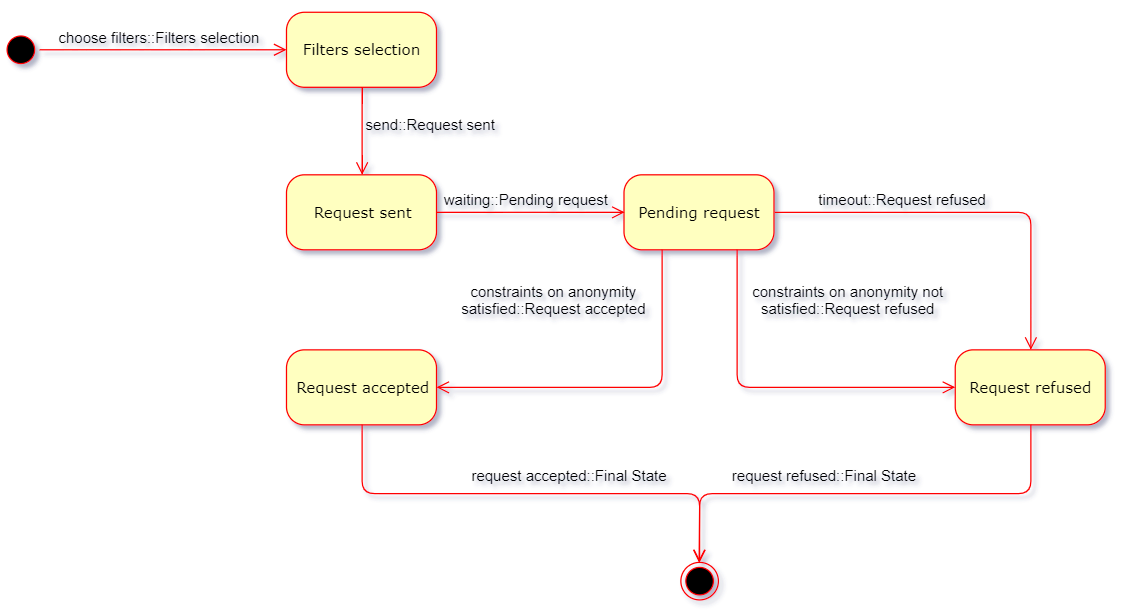
Class diagram

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, the request sent to an individual by a third party to ask for his/her 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, 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 TrackMe can guarantee the privacy of their users.



State diagram 3 - SOS request

This state diagram 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.

The emergency is reported using Internet connection, that’s why an alternative service with simple messages is present in case of malfunctioning.



State diagram 4 - Run organization

Finally, this state diagram 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.

***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.

***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 (for example the birth’s date), while some other data will not (for example the diet). Other data of the user will be collected by the system 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 security number (or the fiscal code, in Italy) of the user, that will accept or refuse to provide the data. In the last 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). 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.

***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 device (for example a smart watch) some vital parameters of the user, like the heartbeat. As soon as those parameters are checked to be behold some thresholds, the system will send location and health status of the user to the nearest registered third party’s structure, so that it will be able to send as soon as possible an ambulance with the appropriate equipment and medical staff.

***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 bodies that will to organize some runs. They will have to enter in the system a certificate that guarantees their authority, in order to be allowed to organize 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, like the path, the starting point, the ending point, the date and the starting time. 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 GPS provided.

***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 (for example his/her own diet) or indirectly by a device that collects the data and send it to the system (for example the location or the heartbeat). 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/her health status in real time, and also the location in case of health emergency (both collected and sent by his/her device).
* Track4Run: the user is a Data4Help’s user who enables this service. He can enroll into run competitions, providing the required data. Also, during the competitions, he provides his/her location in real time to the system, (collected and sent by his/her 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 body, has not only to register to the system, but also to provide a certificate. It will be able to organize and manage runs.
  1. **Assumptions, dependencies and constraints**
* D1: The devices that acquire users’ position provide correct enough location.
* D2: The sensors of devices that acquire users’ health parameters provide correct enough information.
* D3: Each social security 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 device on which the service AutomatedSOS is exploited can provide real time information.
* D7: Third parties that want to organize for a professional run have a certificate to demonstrate that they are acknowledged insitutions.
* D8: Each user that participates to a run wears the device acquiring data during the whole competition

The system does not use any hardware interface. 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 does activity that don’t require data acquisition through device. For example, signing up, logging in, organizing a run, requiring data from a user, accepting or refusing data requests from third party, and so on and so forth. However, some services offered by the system require a specific hardware: in order to locate the user it’s necessary for him to have a GPS device. To get the heartbeat of the user, an appropriate device must be user, such as a smartwatch. Finally, it’s important to underline that AutomatedSOS needs the user to be constantly connected to internet, because in case of emergency the data must be sent immediately. So, a device with a 2G/3G/4G and possibly WiFi connection must be present on the user’s device.

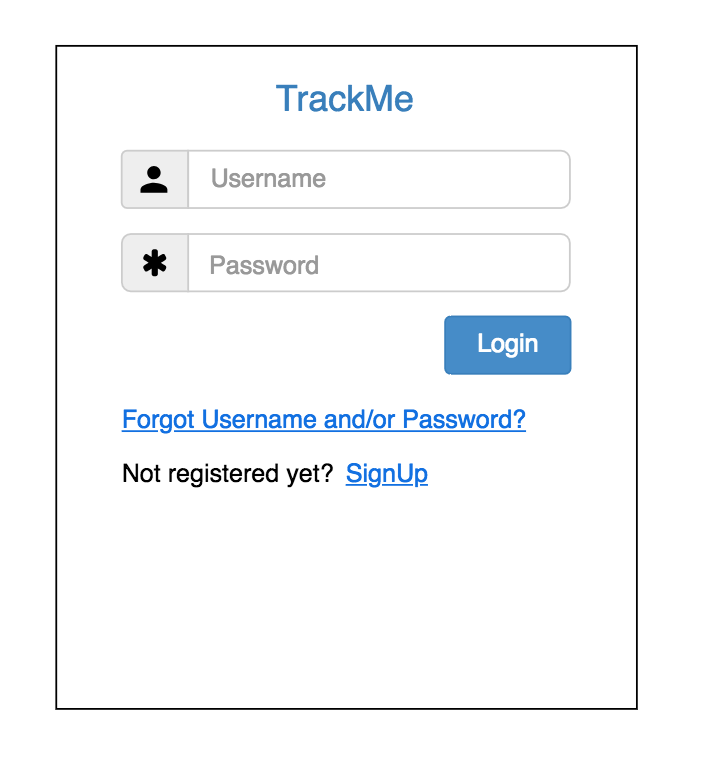
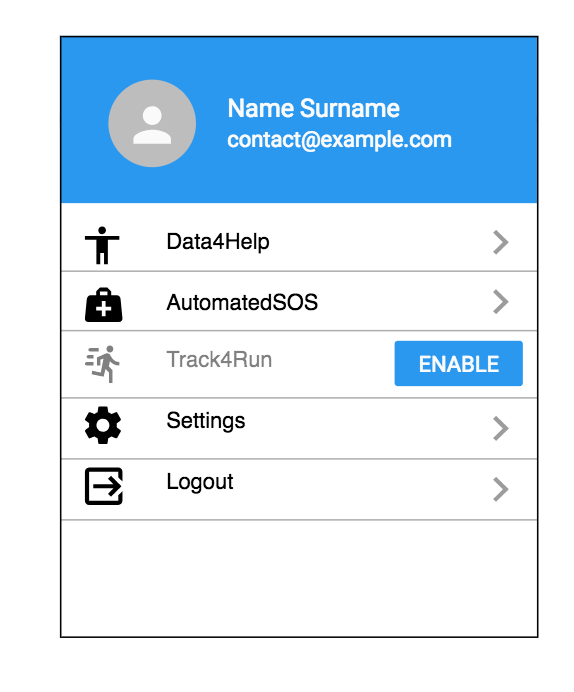
The system uses an external suitable service so that its architecture is simpler. A Google Maps’ API is used when providing the Track4Run service. The organizer will choose the path of his run by managing a map provided by the API. The user that wants to spectate a run, as well, will be provided of a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

***3. Specific requirements***

***3.1 External interface requirements***

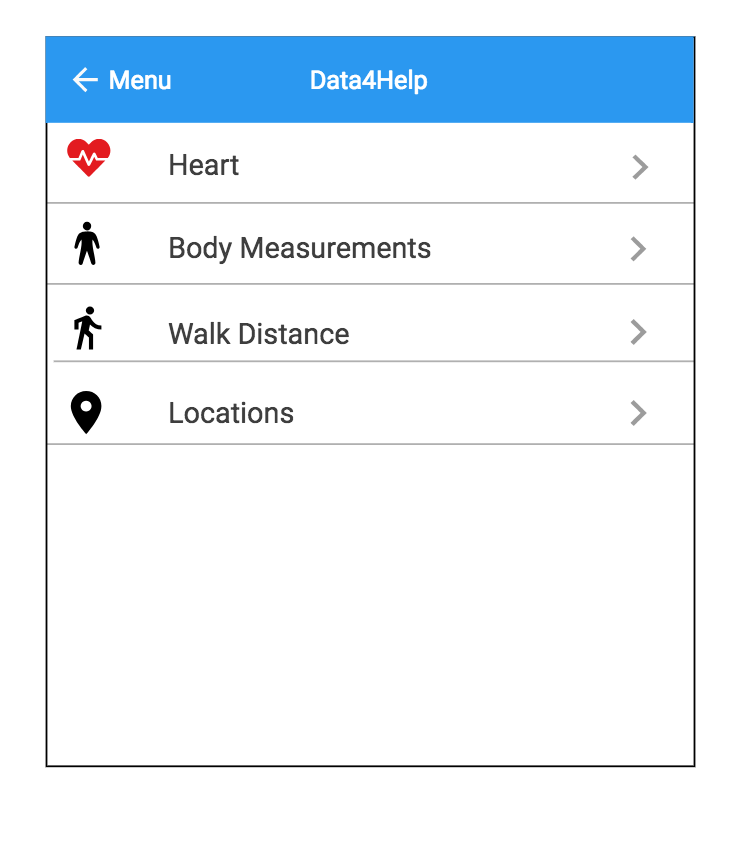
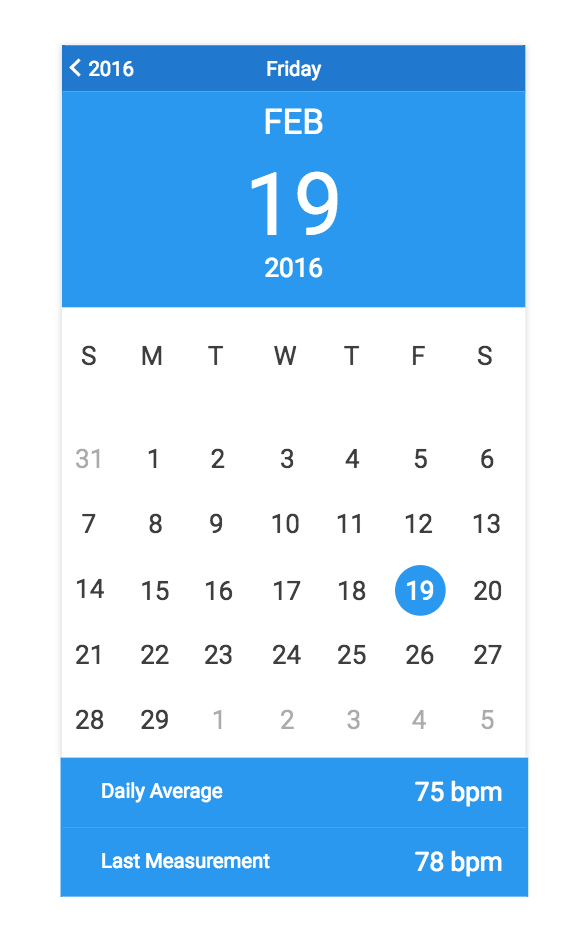
***3.1.1 User Interfaces***

The following mockups give an approximative 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 users (User and Third Party) are represented.



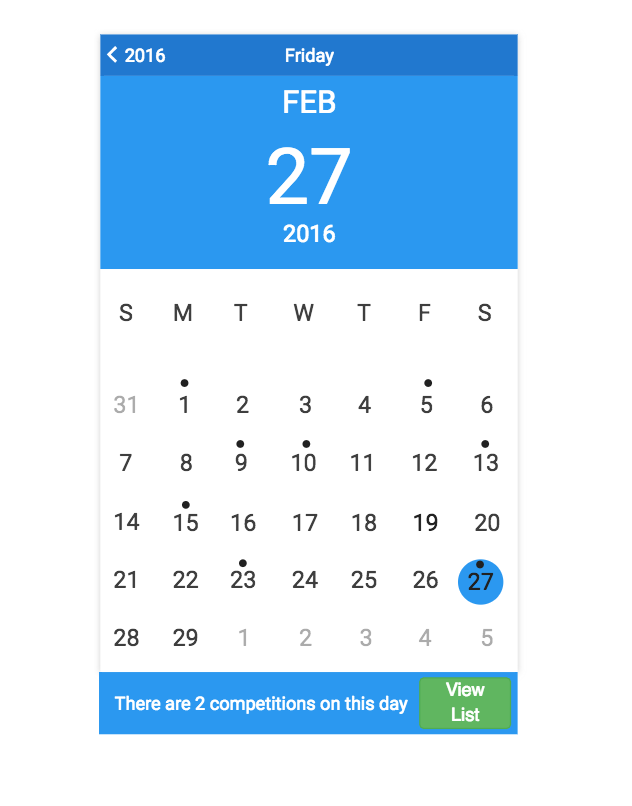
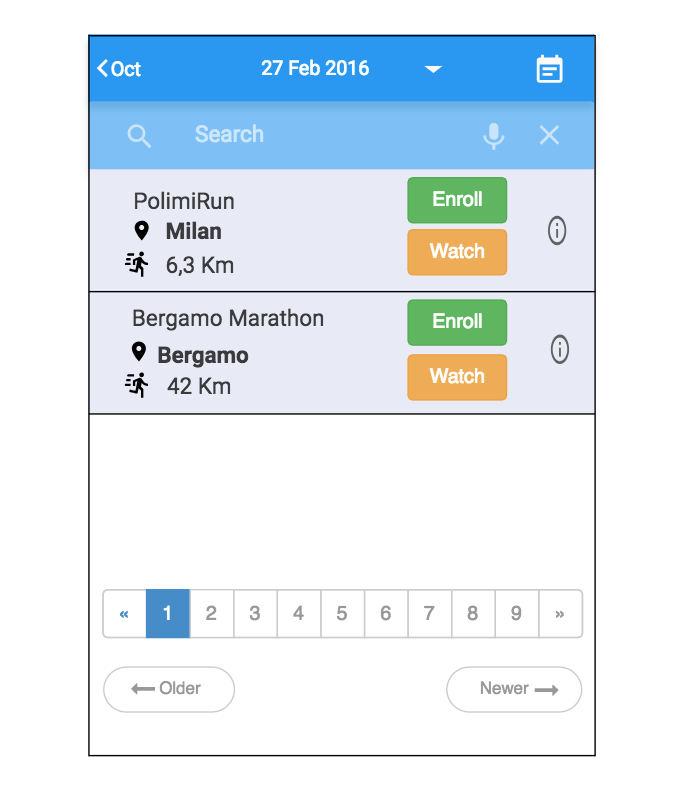
**Figure 2 Mockup - Main menu**

**Figure 1 Mockup - Login**



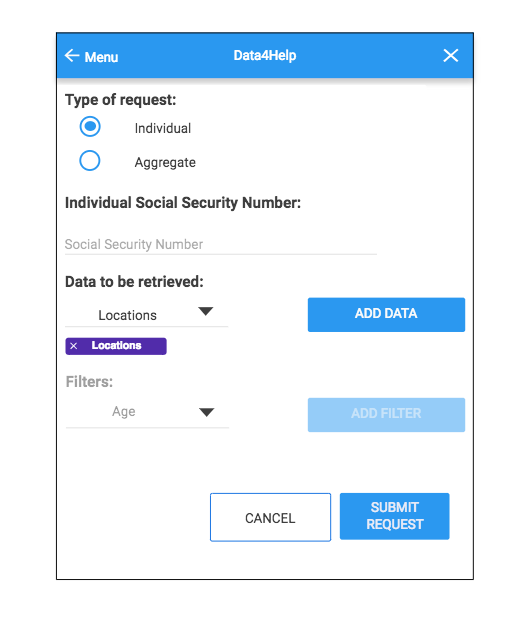
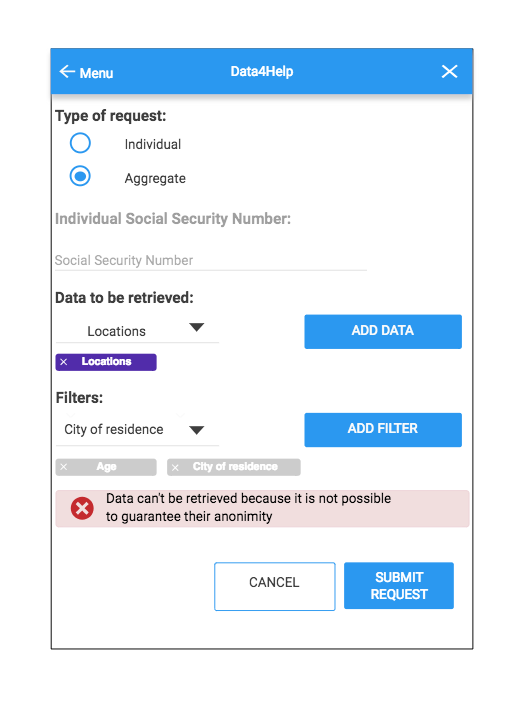
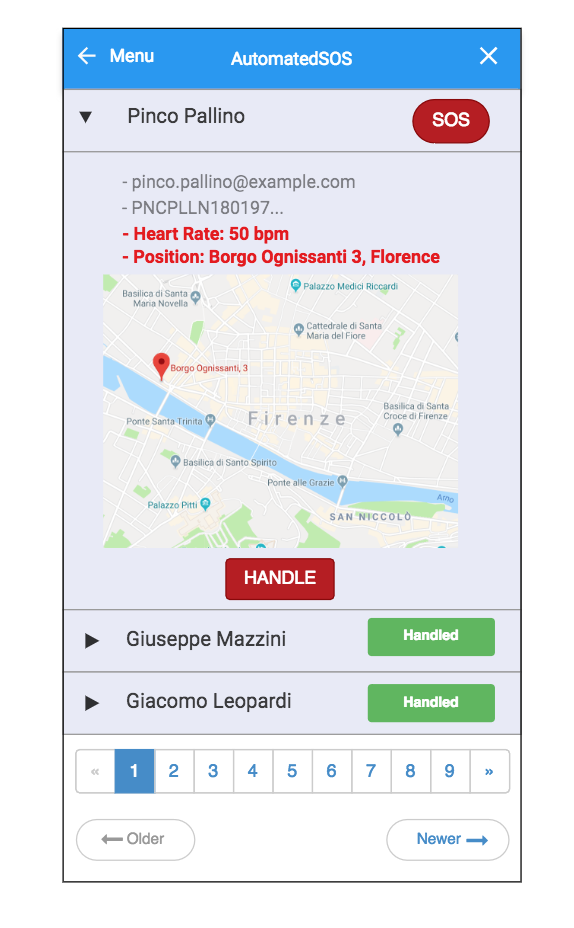
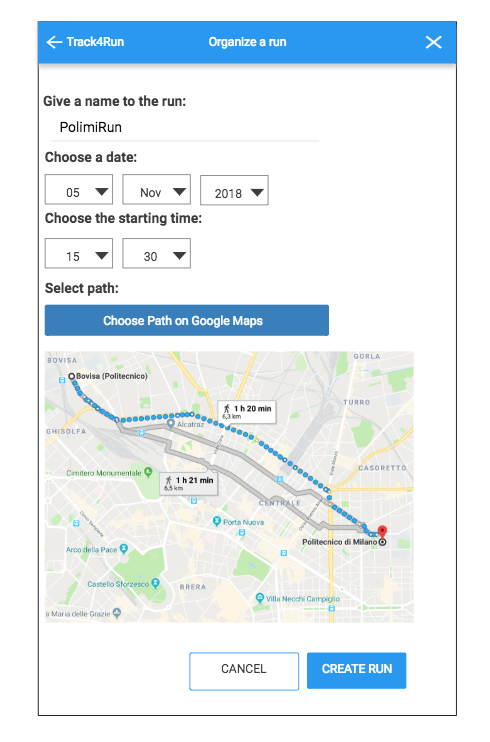
**Figure 3 Mockup – User UI: Data4Help menu**

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

******

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

**Figure 6 Mockup – User UI Track4Run: list of scheduled competition**

******

**Figure 9 Mockup – TP UI AutomatedSOS: list of reportings**

**Figure 10 Mockup – TP UI Track4Run: organize a run**

**Figure 8 Mockup – TP UI Data4Help: group request**

**Figure 7 Mockup – TP UI Data4Help: individual request**

Figure 1 and Figure 2 represent a possible login and menu interface for any kind of user (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***

As mentioned before, the only external API used by the system is the Google Maps’ API, in order to provide the Track4Run service.

The system doesn’t provide any API.

***3.2 Scenarios***

3.2.1 Scenario 1

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.

3.2.2 Scenario 2

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.

3.2.3 Scenario 3

Giovanni suffers from heart's problems, and a private hospital is taking care of him. Going outside riding a bike has always been is passion, and this shouldn't cause any problem, but his doctors are just too afraid to let him go. The fact is that the woods are his preferred place to ride his bike, and in case any kind of injury they couldn't know where to find him. Giovanni then discovers Data4Help, and he finds out that the private hospital could request to it his location and his health status while he is riding the bike. That specific data can be requested by providing Giovanni's fiscal code, and he just has to accept that request through the application. The hospital could also do a subscription and get the data as soon as it is produced, without any further future request. Now Giovanni can ride his bike freely in the woods, and the hospital can monitor his location and his health status in real time, avoiding any risk.

3.2.4 Scenario 4

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.

3.2.5 Scenario 5

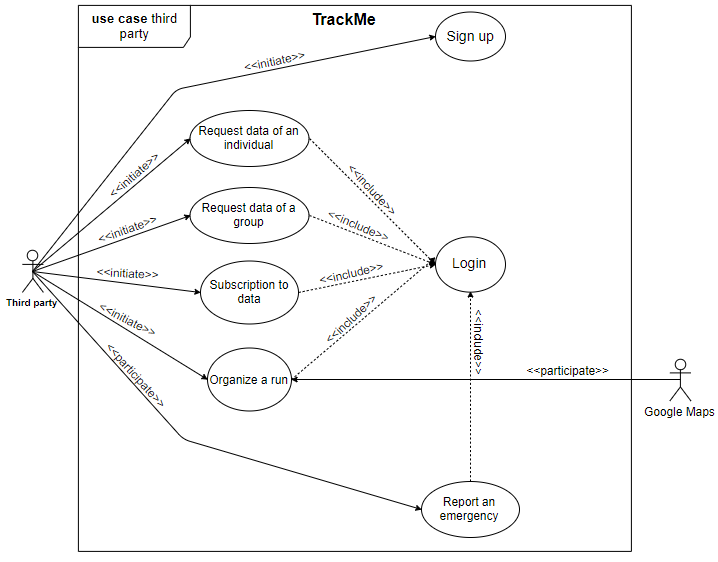
Polytechnic of Milan decides to organize the annual run called PolimiRun. This time, however, it will collaborate with the INRC and for this purpose it wants to make use of a third parties service, called Track4Run, offered by the company TrackMe.

Once registered to the service, 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.

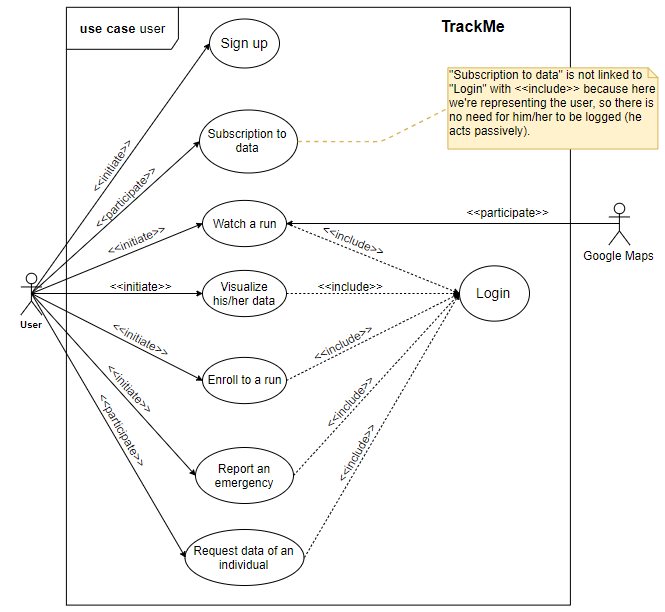
Furthermore, with this service, Polytechnic can ask for the health data of the user involved into the run, so that the INRC can use them for their research projects.

***3.3 Functional requirements***

***3.3.1 Use case diagram***

******

Use case diagram – Third party



Use case diagram - User

|  |  |
| --- | --- |
| 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 sign in; 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.   All the above exceptions are captured by the system after the user chooses the confirmation option. |

|  |  |
| --- | --- |
| Name | Log in |
| 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 “Log in” 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 his/her data |
| Actor | User |
| Entry conditions | 1. The user has successfully logged in. |
| Events flow | 1. The user opens the application; 2. The user can see his/her data and navigate between the related tabs. |
| Exit conditions | The user has access to his/her data. |
| Exceptions | \ |

|  |  |
| --- | --- |
| Name | Request data of a group |
| Actor | Third party |
| Entry conditions | The third party has already done the “Log in” activity. |
| Events flow | 1. The third party chooses the “Request data of a group” option; 2. The third party fills the fields with the constraints of which groups of individuals the third party is looking for; 3. The third party chooses the confirmation option; 4. The system queries its database with the third party’s request; 5. 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 | Request data of an individual |
| Actor | Third party, user |
| Entry conditions | The third party has already done the “Log in” activity. |
| Events flow | 1. The third party chooses the “Request data of an individual” option; 2. The third party writes the identifier of the individual (his security number or his fiscal code); 3. The third party chooses the confirmation option; 4. The system queries its database with the third party’s request and finds who is the user; 5. The system forwards to the user the third’s party request; 6. The user does the “Log in” activity; 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. This exception is captured after the third party chooses the option. |

|  |  |
| --- | --- |
| Name | Subscription to data |
| Actor | Third party, user |
| Entry conditions | Either “Request data of a group” or “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 has been shown by the system; 2. The systems registers that the third party is subscripted to that data; 3. The group of users/the user 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 him as soon as it is produced. |
| Exceptions | \ |

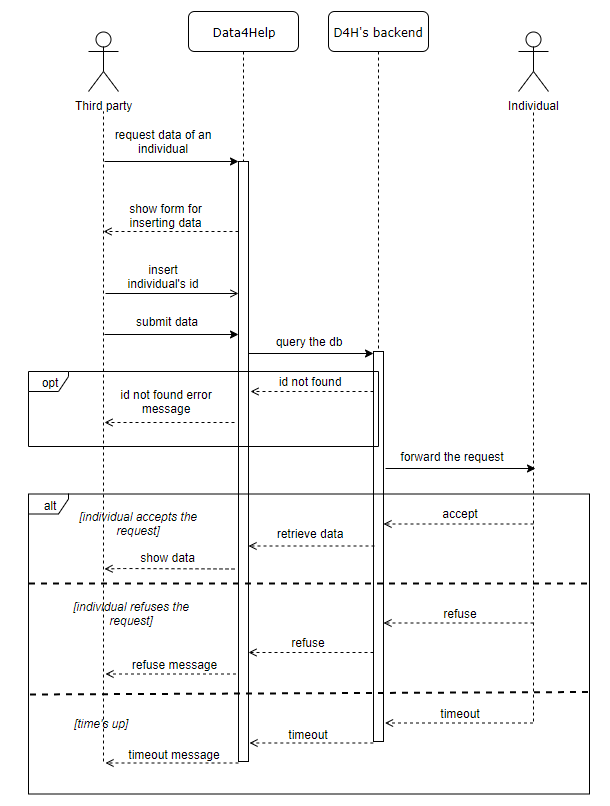
|  |  |
| --- | --- |
| Name | Report an emergency |
| Actor | User, third party |
| Entry conditions | The user is a member of the service AutomatedSOS. |
| Events flow | 1. The system detects that the parameters of the monitored user are below or above the defined threshold; 2. The system sends an alert to the third party, notifying the emergency and sending the location and the status of the user; 3. The third party sends an ambulance to assist the user. |
| 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 | 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 on which the run will take place; 4. The third party defines the path which the participants will travel; 5. 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, such an invalid name or date for the run, or he defines an unfeasible path.   A warning is showed, indicating the wrong parameter(s) and the motivation. |

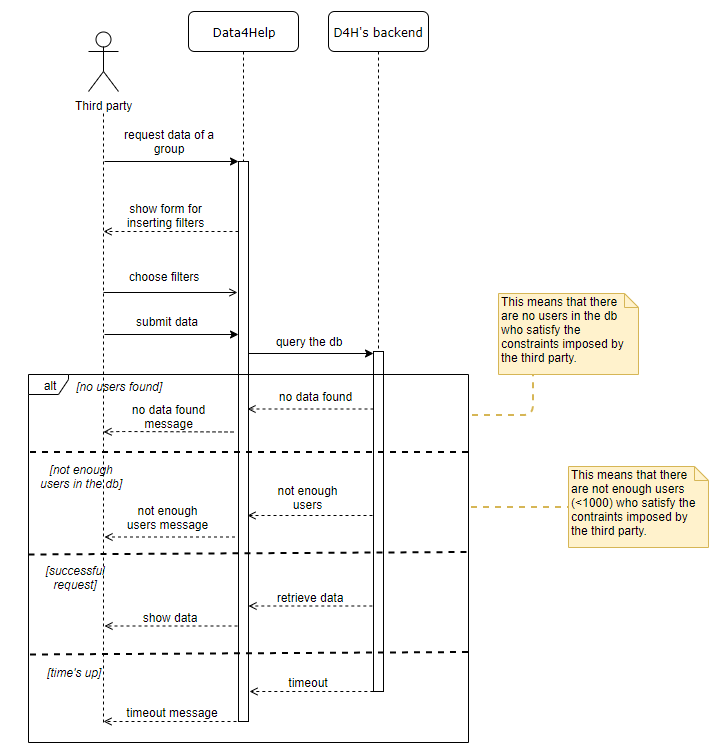
|  |  |
| --- | --- |
| Name | Watch a run |
| Actor | User |
| Entry conditions | 1. The user has activated the service Track4Run; 2. The user has successfully logged in. |
| Events flow | 1. The user selects the run he wants to see. |
| Exit conditions | The user can see the progress of the run and the participants’ position on the map. |
| Exceptions | \ |

|  |  |
| --- | --- |
| 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 to; 3. The user sees a recap of the information about the run and his/her id; 4. The user confirms the participation. |
| Exit conditions | The user is regularly registered to the run. |
| Exceptions | \ |

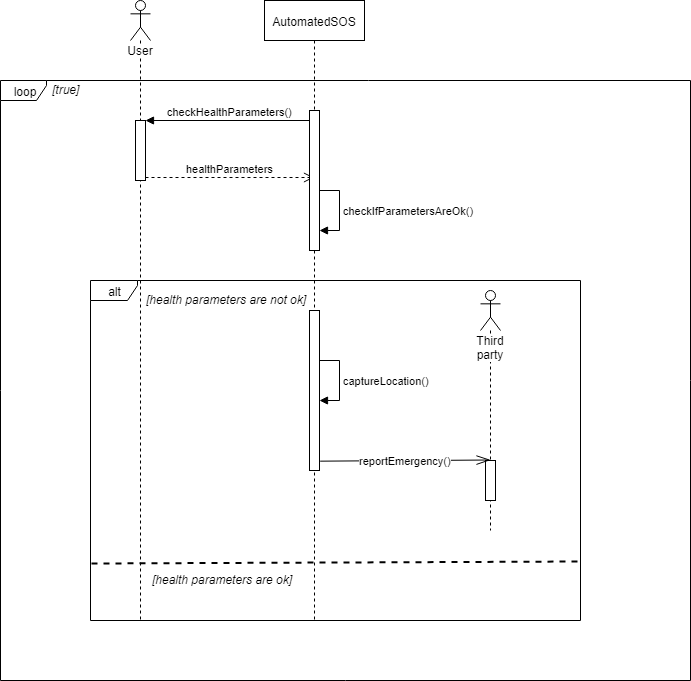
* + 1. ***Sequence diagrams***

******

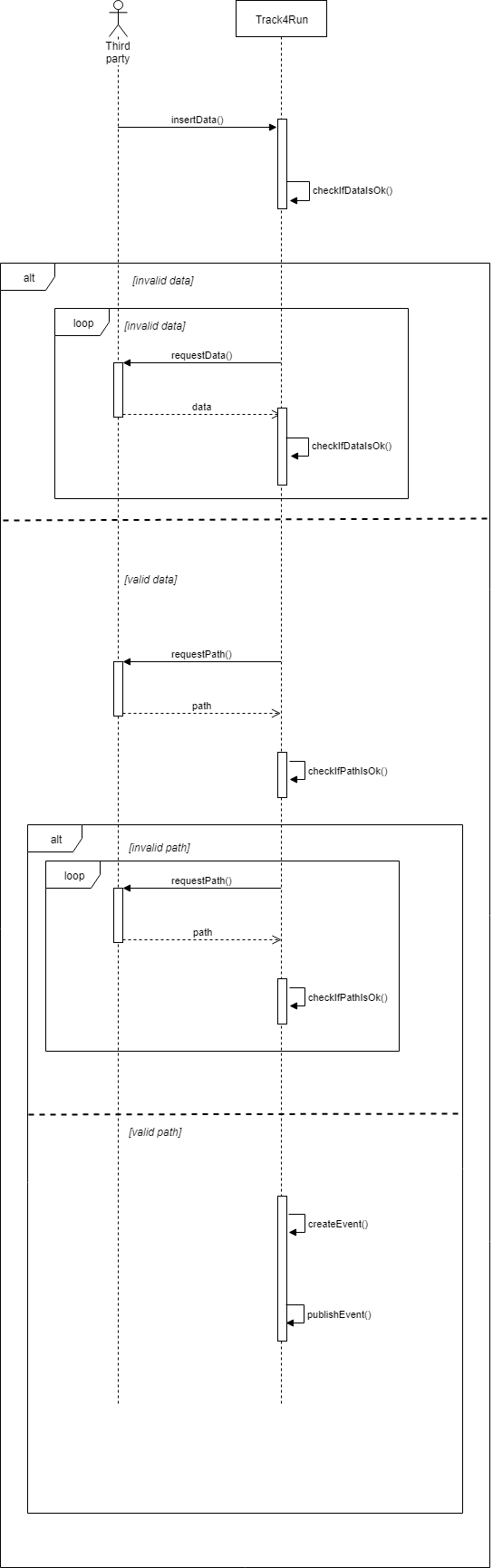
Sequence diagram 1 - Request data of an individual

******

Sequence diagram 2 - Request data of a group



Sequence diagram 3 – Report an emergency



Sequence diagram 4 – Organize a run

***Requirements (TODO: choose where to insert these section)***

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 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 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 collect data of users registered for Data4Help periodically.
* **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 social security number.
* **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.
* **D1**: The devices that acquire users’ position provide correct enough location.
* **D2**: The sensors of devices that acquire users’ health parameters provide correct enough information.
* **D3**: Each social security number is unique.

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

* **R10**: 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.
* **D2**: The sensors of devices that acquire users’ health parameters provide correct enough information.

**G3: In case of emergency, the system must guarantee a reaction time (in reporting the emergency) of less than 5 seconds from the time the parameters are below the threshold.**

* **R11**: The application on wearable devices, if AutomatedSOS is activated, must send the health parameters and location 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 device on which the service AutomatedSOS is exploited can provide real time information.

**G4: The system must allow users who want to organize a run to define its path**

* **R12**: The system has to allow organizers to schedule a run providing name, starting and ending point coordinates, the path and the date of the competition.
* **R13**: If a third party wills to organize a professional run, the application has to provide him the possibility to upload his certificate in order to be authorized. After the first organization the certificate will not be asked anymore until its expiration date.
* **D7**: Third parties that want to organize for a professional run have a certificate to demonstrate that they are acknowledged institutions.

**G5: The system must allow participants to enroll to an organized run**

* **R14**: The system has to allow users to enroll for an organized run showing them the list of organized run on a calendar.

**G6: The system shall allow spectators to see on a map the position of all runners during a run**

* **R15**: The system has to allow users to follow the development of a run selecting an ongoing run within a distance range that has to be chosen by the user from a list that identify competitions by their name.
* **R16**: The application on wearable devices must send the updated position in real time.
* **D8**: Each user that participates to a run wears the device acquiring data during the whole competition

***3.4 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.

***3.5 Design constraints***

***Standards compliance***

The system adopts precise units of measure:

* Heartbeat: [bpm]
* Body temperature: [°C]
* Distance: [Km]

***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 is 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
* Sensor for body temperature monitoring

***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 to authorize the send. 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).

No data given by the user will be used for commercial uses.

***3.6 Software system attributes***

***3.6.1 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.

***3.6.2 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%.

***3.6.3 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.

***3.6.4 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.

***3.6.5 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.