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 (both professional or not). 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, should 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 don’t 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 acquistion’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 platform installed by TrackMe 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 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 personnel user 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 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. The system provides the encoding of the call to the ambulance, the location of the person and, eventually, some information that the person manage to send 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 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 both professional and non-professional runs. Each user must authenticate himself when using the application and, in case of a professional run, if he is an organizer or a runner, he has to prove it through a certificate, while for amateur runs this is not necessary.

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

***1.4 Definitions, acronysms, and abbreviation***

***1.4.1 Definitions***

***1.4.2 Acronyms***

*API = Application Programming Interface*

*GPS = Global Positioning System*

***1.4.3 Abbreviation***

***1.5 Revision History***

***1.5 Reference documents***

***1.6 Document Structure***

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 exploit Google Maps’ APIs and the device’s Calendar app to register the event (ex: for an athlete that wants to participates to a competition or for a user that programs to follow a run programmed in the future).

***3. Specific requirements***

***3.1 External interface requirements***

***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***

|  |  |
| --- | --- |
| 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 data stored |
| Exceptions | 1. The user was already registered. In this case the system warns the user and suggests him 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 below 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 | 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 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 | The connection goes out. This is handled by an alternative SMS option. |
| 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 | 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 gives a name to the event; 3. The user defines the data on which the run will take place; 4. The user defines the path which the participants will travel; 5. The user sets the maximum number of participants; 6. The user publishes the event. |
| Exit conditions | The event for the run is online and joinable. |
| Exceptions | 1. The user enters invalid data, such an invalid name, data or number of participants for the run, or he defines an unfeasible path. A warning is showed, indicating the wrong parameter(s) and the motivation. 2. The system fails in publishing the event. A warning is showed and the user is asked to try to commit again. |

***3.4 Performance requirements***

***3.5 Design constraints***

***3.6 Software system attributes***