POLITECNICO
MILANO 1863

Software Engineering II project

Academic Year: 2018/2019

TRACK ME

Requirement Analysis and Specification Document

Version 1.0 - 10/11/2018

Davide Rutigliano - 903616
Claudio Ferrante - 903417
Davide Matta - 920349

Contents

Use Cases Summary

UML Diagrams List

1 Introduction

1.1 Purpose

This Requirement Analysis and Specification Document [1] is aimed at modeling and describing the system itself, its requirements, its constraints, its components and how it interacts with the world and the users. Moreover, this specifications also addresses some relevant QoS characteristics that the system should guarantee.

This document is directed and highly recommended to software designers and developers interested in the design and deployment of the proposed system.

1.2 Scope and Audience

1.2.1 Description of the given problem

TrackMe is a company that wants to develop an ease of use health monitoring application which offers different services for both young and old people who needs to keep track of their personal data in order to to keep their health safe.

The system should provide an efficient data acquisition facility which gives the possibility to verified *third-party* signed onto the service to request health status information about subscribed customers. The application may also allow *individual* users to connect *external devices* such as smart-watches or similar, to perform a more detailed data acquisition and monitoring.

In addition, the *third-party* may also provide a personalized and non intrusive SOS service for elderly people who requested it.

Moreover, through the *Track4Run* service, *TrackMe* wants to provide to users signed in the application to create a new run: a competition in which other individuals can either participate as *athletes* or watch as *spectators*. Additionally, this feature may exploits other service functionality for keeping track of athletes progresses and for helping them to prevent major accident happen during the run.

1.2.2 World, Machine and Phenomena

This section is intended to provide a full description of the proposed system by meaning of *Domains* and *Phenomena* [2].

The *Machine Domain* is the set of phenomena that the machine can control: data structures, algorithms, devices and inputs it can get from the world.

In contrast, the *World Domain* is the real-life context in which the *Machine* will be introduced. This is the part of the real-world in which the *Machine* actions will be observed.

The World and the Machine are connected too, because the latter should interact with the first one: this interaction is done through *Shared Domain*, whose phenomena are observable both by the Machine and by the World. Shared Phenomena include events in the real world that the Machine can directly sense and actions in the real world that the Machine can directly cause.

Phenomenon	World	Machine	Shared
Registration/Login/Log-out			X
Manage Profile/Change Credentials			X
Application logic		X	
Accept/Reject/Send Request			X
Data Subscription/Sync/View			X
Database Query		X	
Connect External Device			X
GPS Tracking			X
Enable Sos			X
Locate nearest Ambulance			X
Accident	X		
Create/Delete/Enroll/Watch/Start Run			X

1.2.3 Goals

☆ **GOAL - 01** Allow a Guest to register as an Individual.

☆ **GOAL - 02** Allow a Guest to register as a Third-Party.

☆ **GOAL - 03** The Guest should be able to sign in into the application.

☆ **GOAL - 04** The User should be able to change his credentials.

☆ **GOAL - 05** The User should be able to log-out from the system.

☆ **GOAL - 06** The Individual should be able to change his personal data.

☆ **GOAL - 07** The Third-Party should be able to change his organization data.

☆ **GOAL - 08** The Third-Party should be able to send requests to the Individuals.

☆ **GOAL - 09** The Third-Party should be able to make group requests.

☆ **GOAL - 10** The Third-Party should be able to subscribe to new data, once an Individual request is made.

☆ **GOAL - 11** The Third Party should be able to view Individual's data to whom has sent a request.

☆ **GOAL - 12** The Individual should be able to accept or reject a request coming from a Third-Party.

☆ **GOAL - 13** The Individual shall be able connect an external device to the system.

☆ **GOAL - 14** The Third-Party shall be able to activate Automated-SOS service.

☆ **GOAL - 15** The Individual shall be able to activate Automated-SOS service.

☆ **GOAL - 16** The Dispatcher shall be able to assign an ambulance to an incident.

☆ **GOAL - 17** The Organizer shall be able to create a run.

☆ **GOAL - 18** The Organizer should be able to start a run.

☆ **GOAL - 19** The Organizer should be able to delete a run.

☆ **GOAL - 20** The Athlete should be able to enroll a run.

☆ **GOAL - 21** The Athlete should be able to unroll a run.

☆ **GOAL - 22** The Spectator should be able to watch a run.

1.3 Document Overview

This initial part of the document is intended to provide both an overview of the problem and an idea of the proposed solution.

Following section is aimed at giving a more detailed description of the proposed system, by meaning of the application point of view. Additionally, it addresses functionality, user characteristics, dependencies and constraints.

Third chapter shows a detailed analysis of *TrackMe* in terms both functional and non-functional requirements of the system. Furthermore, takes into account requirements mapping on use cases and their detailed description through well known standards for specification documents such as *UML* for Use Case Diagrams [3].

Endmost section illustrates *Alloy* for formal analysis [4] describing models including purpose, proof and explanation. In addition, describes worlds obtained by running them.

Further, there is a list of tables and a list of figures at the beginning of the document representing respectively use cases and diagrams in order to help the reader to understand them and navigate the specification.

1.4 Definition, Acronyms and Abbreviations

1.4.1 Keywords

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in RFC 2119 [5].

1.4.2 Definition of Terms

This document uses several terms that might be more loosely used elsewhere. These terms are defined here as they will be used later on in this document.

Subscribed User an Individual for which one Third Party whom asked to subscribe to the data has accepted the Third Party's request

External Device a device such as a smart-watch or similar, capable of data acquisition

Run a competition organized and managed by an *Organizer* to which *Athletes* and *Spectators* can enroll as participant or watchers

Run Started a competition with at least two *Athletes* enrolled that the *Organizer* has started

Run Created a competition with any number of *Athletes* enrolled

Track A route where athletes will compete in a specific *Run*

Map A *Track* that display all *Athletes* position in a specific *Run*

Accident An event triggered when the monitored user's parameter overcome certain thresholds

TrackMe the *"system to be"*

Data4Help a data monitoring service provided by *TrackMe*

Automated-SOS an SOS service built on top of *Data4Help*

Track4Run run management service offered by *TrackMe* application

Credential as used in this document, is a combination of both username and password used by an *User* to authenticate him/herself during the Log-in phase

Personal Data users' data of different kind either for Individuals (name, surname, age, etc.) or for Third-Parties such as Organization name, number of employees or VAT number

1.4.3 Abbreviations

A-n : Assumption number n

C-n : Constraint number n

D-n : Dependency number n

GOAL-n : Goal number n

REQ-n : Functional Requirement number n

1.4.4 Acronyms

RASD Requirement Analysis and Specification Document

UML Unified Modelling Language

QoS Quality of Service

SSN Social Security Number

FC Fiscal Code

VAT Value Added Tax

BT Bluetooth

NFC Near Field Communication

GPS Global Positioning System

HTTP Hyper Text Transfer Protocol

HTTPS Hyper Text Transfer Protocol over SSL

SSL Secure Socket Layer

SHA Secure Hash Algorithm

SDK Software Development Kit

JSON JavaScript Object Notation

CBOR Concise Binary Object Representation

REST REpresentational State Transfer

API Application Programming Interface

UPS Uninterruptible Power Supply

DBMS Data Base Management System

1.5 References

[1] IEEE Software Engineering Standards Committee et al. "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications". In: *October* 20 (1998), pp. 1–5.

[2] Pamela Zave and Michael Jackson. "Four dark corners of requirements engineering". In: *ACM transactions on Software Engineering and Methodology (TOSEM)* 6.1 (1997), pp. 1–30.

[3] James Rumbaugh, Ivar Jacobson, and Grady Booch. *Unified Modeling Language Reference Manual, the.* Pearson Higher Education, 2004.

[4] Daniel Jackson. *Software abstractions.* Vol. 2. MIT press Cambridge, 2006.

[5] Scott Bradner. "Key words for use in RFCs to Indicate Requirement Levels". In: (1997).

2 Overall Description

2.1 Product Perspective

The system is not part of any already existing software and can be completely developed from scratch. It should interface with a database server (even local), shall be able to interact with several GPS systems from several manufacturers widely used in mobile devices and may be able to do the same with external devices such as smart-watches of different kind. Furthermore, it should be capable of cooperating with an *"ambulance dispatcher"* system in order to make reservations of ambulances in case of accident.

The proposed solution is *TrackMe*, a new health monitoring application which consists of a back-end server that manages users' registration, login and data and allows to keep everything synchronized between the two front-end applications:

- a web-based application to supply to the end user an ease of use interface web interface for *TrackMe*;
- a mobile application that allows the user to easily access the system from the smartphone.

The system should be able to query the database for storing and retrieving both users' credentials and personal data, besides of keeping everything synchronized between all the application instances.

2.1.1 Domain Model

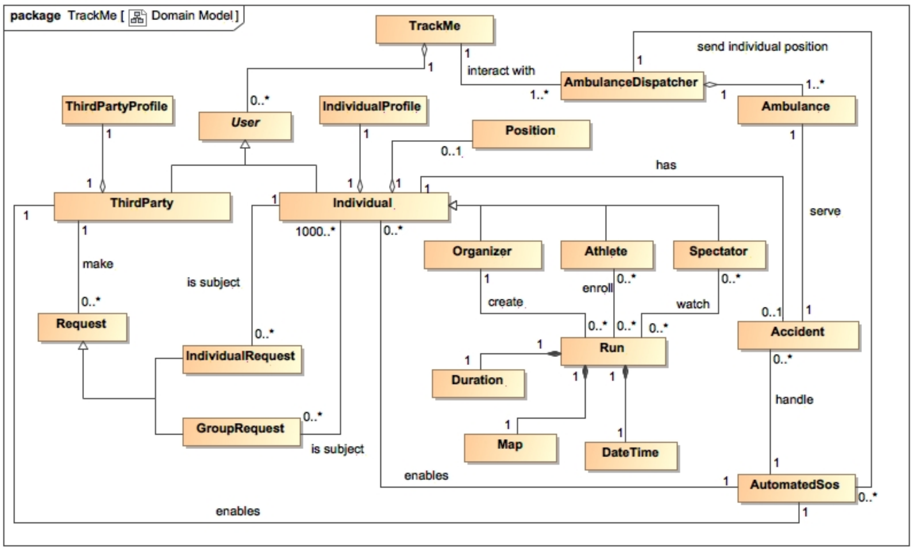


Figure 1: Domain Model

2.1.2 State Charts

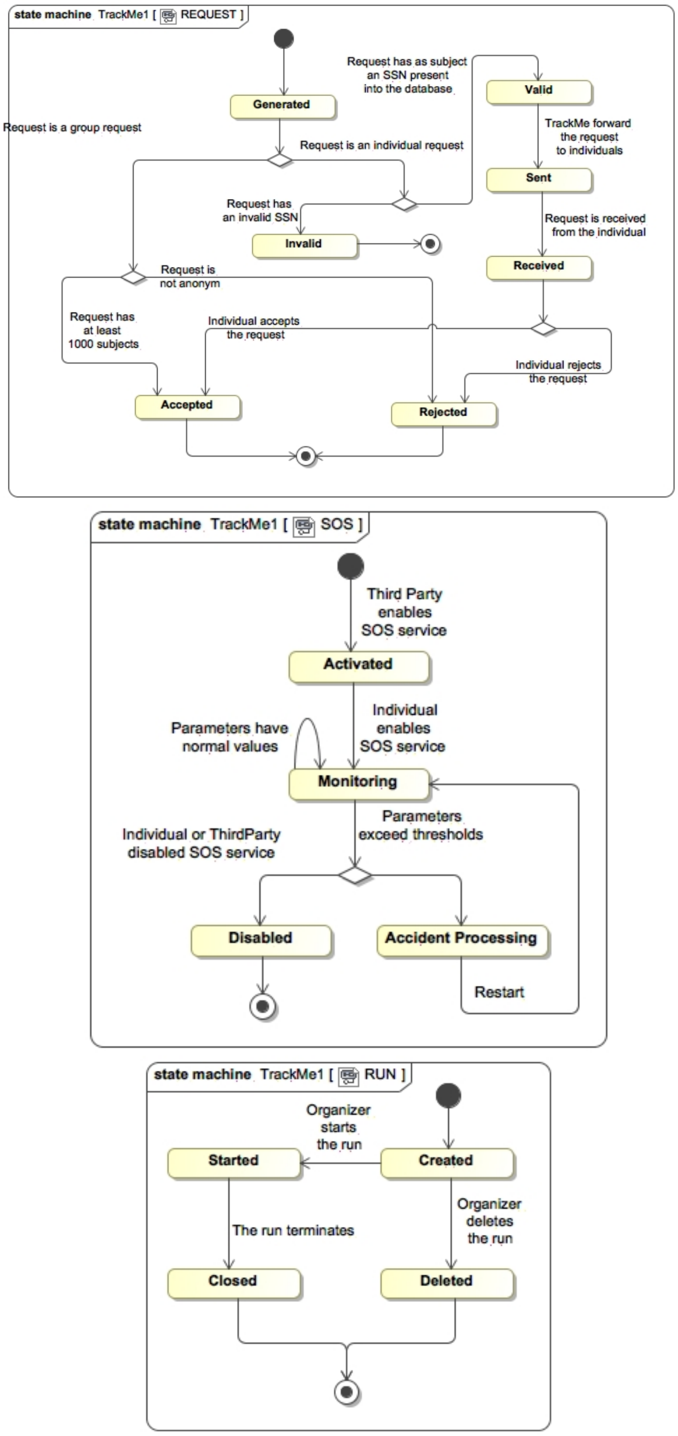

state machine TrackMe1 [REQUEST]
Generated
Request is a group request
Request is an individual request
Request has as subject an SSN present into the database
Valid
TrackMe forward the request to individuals
Sent
Request has an invalid SSN
Invalid
Request is received from the individual
Received
Request is not anonym
Request has at least 1000 subjects
Individual accepts the request
Individual rejects the request
Accepted
Rejected
state machine TrackMe1 [SOS]
Third Party enables SOS service
Activated
Parameters have normal values
Individual enables SOS service
Monitoring
Parameters exceed thresholds
Individual or ThirdParty disabled SOS service
Disabled
Accident Processing
Restart
state machine TrackMe1 [RUN]
Organizer starts the run
Started
Created
The run terminates
Organizer deletes the run
Closed
Deleted

Figure 2: State Charts

2.2 Product Functions

TrackMe will provide three main services: *Data4Help*, *Automated-SOS* and *Track4Run.*

Data4Help : the application will provide both *"Sign-on"* and *"Sign-in"* pages and will be able to register new users and to check credentials used to login.

To register himself an user must provide an username and a password as well as his personal information. To be logged in, the user should send through the login form his credentials which the system will use to authenticate him.

TrackMe may yield an interface that allows individual users to connect external devices and if so, of course a method for acquiring data from it.

The application should also provide to third-parties two possible data request options:

- Individual Data request: sent directly to the individual if they know an individual by his/her social security number or fiscal code in Italy;

- Group Data request: access to anonymous data of groups of individuals, if *TrackMe* that approves them if it is able to properly anonymize the requested data.

Automated-SOS: the system implements *Data4Help* services to manage individual's personal data and provide them to third-parties who requested for. In addition, there should be an interface both for third-party and individual users registered to *Data4Help* that wants to enable an *Automated-SOS* service. This feature should be able to use users' data to monitor their health status and, when such parameters are below certain thresholds, sends the location of the customer to an ambulance. Thus, the application should be able to interact with the GPS system of the customers' devices.

Automated-SOS is built on top of *Data4Help* and provides individuals the capability of subscribing to third-parties who has enabled the service.

Track4Run: the software should provide to users that wants to use *Track4Run* all the facilities they need such as a page for the creation of a new run with all the information (track, date and time) and two different pages to allow both spectators and athletes either watch or enroll/unroll to an existing run.

Track4Run is also built on top of *Data4Help* and allows third parties to create new runs where athletes can participate. Athletes' position can be tracked and it will be sent to all spectators that are watching the run on a map.

2.3 User Characteristics

2.3.1 Data4Help

- **Guest:** a customer that is visiting *TrackMe.* He can only login or register to the system.

- **User:** a *Guest* successfully registered to *TrackMe* and logged in to the system. He can be either an *Individual* or a *Third Party.*

- **Individual:** an *User* that agreed that *TrackMe* can collect his personal data. He is identified by a SSN/FC. Data acquisition can happen through smart watches or similar devices.

- **Third Party:** an *User* that can ask an *Individual* to gain access to their personal data. He is identified by a VAT number. He can also request access to data that belongs to a group of *Individuals*, but the request shall be accepted only if *TrackMe* can guarantee anonymity to the requested data. During a request he can also subscribe to new data that will be received as soon as is it is produced.

2.3.2 Automated-SOS Extension

- **Individual:** he can access to a list of Third Parties that enabled the Automated-SOS service. He can also select a Third Party from the list and subscribe to it.

- **Third Party:** it has the ability to enable the Automated-SOS service so that any Individual can subscribe to it.

2.3.3 Track4Run Extension

- **Organizer:** both an *Individual* and a *Third Party* that created at least one run. He can create, start or delete a run.

- **Athlete:** an *Individual* that enrolled a created run. His position will be sent to all Spectators that joined that run.

- **Spectator:** an *Individual* that joined a started run. He can see the position of all Athletes that are competing in the run until the run is finished.

2.4 Assumptions and Dependencies

2.4.1 Assumptions

✦ **A - 01** The field ”Password” and ”Repeat Password” match.

✦ **A - 02** Data provided by the individual, such as email and SSN, are valid.

✦ **A - 03** Data provided by the Third-Party (i.e: VAT number) are valid.

✦ **A - 04** The user is already registered.

✦ **A - 05** Username and password provided match the ones previously stored in the registration process.

✦ **A - 06** The user is logged in the system.

✦ **A - 07** The Third-Party provides a SSN/FC of an individual already registered into the system.

✦ **A - 08** The Third-Party provides not contradictory criteria for the selection of the group of individuals.

✦ **A - 09** The Third-Party has sent at least a request to the user whose data she wants to subscribe.

✦ **A - 10** The external device provided by the user is compatible with the system.

✦ **A - 11** The user has activated NFC/Bluetooth on the mobile device.

✦ **A - 12** The user has activated NFC/Bluetooth on the external device.

✦ **A - 13** An Automated-SOS service is enabled from at least one Third Party.

✦ **A - 14** At least one ambulance dispatcher is available for SOS request serving.

✦ **A - 15** The dispatcher is able to communicate with both the individual and the ambulance associated with the accident.

✦ **A - 16** The organizer defines a not empty path.

✦ **A - 17** The Organizer has already created a run.

✦ **A - 18** At least two athletes are enrolled in the run.

✦ **A - 19** There is at least one created run.

✦ **A - 20** The Athlete is enrolled to at least one run.

✦ **A - 21** There is at least one started run.

✦ **A - 22** The individual has approved TrackMe' terms of use.

✦ **A - 23** The individual has received at least an individual request.

✦ **A - 24** The individual has accepted the individual request.

✦ **A - 25** Username is not already taken.

✦ **A - 26** Password is valid (i.e.: at least 8 alphanumerical characters).

✦ **A - 27** The run has not been started.

✦ **A - 28** The individual has received an individual request a specific Third Party.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The mobile application shall provide login and two different registration interfaces for both individual and third parties. The users would also be able to manage their profile page.

TrackMe shall provide an ease of use management interface for requests handling. Moreover, a data management page for Third Parties shall be present in order to allow customers to manage requested data. In addition, for Track4Run users, the system shall provide all the facilities to create, enroll and watch a run.

3.1.2 Hardware Interfaces

The Individuals need a smartphone and an external device capable of data acquisition. The external device also needs a way to communicate with the smartphone that should be NFC and/or BT. The GPS system of the smartphone or the one of the external device (if connected) is needed in order to track customer's position. Furthermore, a separate system to notify the ambulances in case of accident shall be integrated within already existing ambulances' on-board technologies. This system and further interactions with *TrackMe* are out of scope of this specification document.

3.1.3 Software Interfaces

Both the front end and the back-end should be compatible with most of existing PC and mobile operating systems. The mobile and application should indeed be supported at least on Android Lollipop 5.0 and iOS5 or major versions; the web application shall instead be compatible with different browsers such as Google Chrome (with V8 JavaScript engine), Mozilla Firefox v3.0 (or major versions) and Safari v4.0 (or major versions).

Moreover the developer could choose to implement REST services to improve portability, reliability, modifiability and simplicity of the software.

3.1.4 Communication Interfaces

The application shall communicate with the back-end server over HTTPS and may enable O-Auth2 authentication protocol for clients. In order to connect and interact with external devices the mobile application shall also support BT and/or NFC.

Moreover, the system shall be able to interact with a DBMS both for for storing and retrieving user data, also in real-time in an Automated-SOS related scenario. In addition, the client application should be able to interact with the back-end service which will indeed be able get client requests, send responses and make data available to clients using lightweight object oriented communication protocols such as JSON or CBOR.

3.2 Functional Requirements

☆ **GOAL - 01** Allow a Guest to register as an Individual.

➥ **REQ - 01** A customer not signed-on must be able to begin the Individual's registration process to TrackMe providing a username, a password and his personal data.
✦ **A - 25** Username is not already taken.
✦ **A - 26** Password is valid (i.e.: at least 8 alphanumerical characters).
✦ **A - 01** The field "Password" and "Repeat Password" match.
✦ **A - 02** Data provided by the individual, such as email and SSN, are valid.
✦ **A - 22** The individual has approved TrackMe' terms of use.

☆ **GOAL - 02** Allow a Guest to register as a Third-Party.

➥ **REQ - 02** A customer not signed-on must be able to begin the Third Party's registration process to TrackMe providing a username, a password and its organization data.
✦ **A - 25** Username is not already taken.
✦ **A - 26** Password is valid (i.e.: at least 8 alphanumerical characters).
✦ **A - 01** The field "Password" and "Repeat Password" match.
✦ **A - 03** Data provided by the Third-Party (i.e: VAT number) are valid.
✦ **A - 22** The individual has approved TrackMe' terms of use.

☆ **GOAL - 03** The Guest should be able to sign in into the application.

➥ **REQ - 03** The system must provide a log-in interface for already registered users, not previously signed into the application.
✦ **A - 04** The user is already registered.
✦ **A - 05** Username and password provided match the ones previously stored in the registration process.

☆ **GOAL - 04** The User should be able to change his credentials.

➥ **REQ - 04** TrackMe must provide to users the possibility to change their username.
➥ **REQ - 05** TrackMe must provide to registered users the possibility to change their password.
✦ **A - 06** The user is logged in the system.
✦ **A - 05** Username and password provided match the ones previously stored in the registration process.
✦ **A - 25** Username is not already taken.

☆ **GOAL - 05** The User should be able to log-out from the system.

➥ **REQ - 06** The system must provide the possibility to the user of logging out.

✦ **A - 06** The user is logged in the system.

☆ **GOAL - 06** The Individual should be able to change his personal data.

➥ **REQ - 07** The system must provide the possibility to change Individual's personal data.

✦ **A - 02** Data provided by the individual, such as email and SSN, are valid.

✦ **A - 06** The user is logged in the system.

☆ **GOAL - 07** The Third-Party should be able to change his organization data.

➥ **REQ - 08** The system must provide the possibility to change Third Party's organization data.

✦ **A - 03** Data provided by the Third-Party (i.e: VAT number) are valid.

✦ **A - 06** The user is logged in the system.

☆ **GOAL - 08** The Third-Party should be able to send requests to the Individuals.

➥ **REQ - 09** Data4help must allow the Third-Parties to send a request to a particular individual, provided his SSN/FC.

✦ **A - 06** The user is logged in the system.

✦ **A - 07** The Third-Party provides a SSN/FC of an individual already registered into the system.

☆ **GOAL - 09** The Third-Party should be able to make group requests.

➥ **REQ - 10** Data4help must allow the Third-Parties to generated a request for a group of individuals.

➥ **REQ - 11** The system must be able to anonymize users' requested data.

✦ **A - 06** The user is logged in the system.

✦ **A - 08** The Third-Party provides not contradictory criteria for the selection of the group of individuals.

☆ **GOAL - 10** The Third-Party should be able to subscribe to new data, once an Individual request is made.

➥ **REQ - 12** Data4help must allow the Third-Parties to choose if subscribe to new data associated with a particular Individual.

➥ **REQ - 13** The system must be able to periodically query the database in order to get new data as soon as they are produced.

✦ **A - 06** The user is logged in the system.

✦ **A - 09** The Third-Party has sent at least a request to the user whose data she wants to subscribe.

✦ **A - 28** The individual has received an individual request a

specific Third Party.
✦ **A - 24** The individual has accepted the individual request.

☆ **GOAL - 11** The Third Party should be able to view Individual's data to whom has sent a request.

➥ **REQ - 14** Data4help must allow the Third Party to see a list of sent requests and related Individual's data.
✦ **A - 06** The user is logged in the system.

☆ **GOAL - 12** The Individual should be able to accept or reject a request coming from a Third-Party.

➥ **REQ - 15** Data4help must allow the Individual to see a list of the received requests from Third Parties.
➥ **REQ - 16** Data4help must allow the Individual to accept or reject a request from the list.
✦ **A - 06** The user is logged in the system.
✦ **A - 23** The individual has received at least an individual request.

☆ **GOAL - 13** The Individual shall be able connect an external device to the system.

➥ **REQ - 17** Data4help must allow the Individual to connect an external device through BT or NFC.
✦ **A - 06** The user is logged in the system.
✦ **A - 10** The external device provided by the user is compatible with the system.
✦ **A - 11** The user has activated NFC/Bluetooth on the mobile device.
✦ **A - 12** The user has activated NFC/Bluetooth on the external device.

☆ **GOAL - 14** The Third-Party shall be able to activate Automated-SOS service.

➥ **REQ - 18** The system must allow the Third-Party to activate Automated-SOS service.
➥ **REQ - 19** The system must be able to assign an ambulance dispatcher to the SOS service.
✦ **A - 06** The user is logged in the system.
✦ **A - 14** At least one ambulance dispatcher is available for SOS request serving.

☆ **GOAL - 15** The Individual shall be able to activate Automated-SOS service.

➥ **REQ - 20** The system must allow the Individual to activate Automated-SOS service.

➥ **REQ - 21** The system should allow the Individual to choose between third parties who has enabled the service.
✦ **A - 06** The user is logged in the system.

☆ **GOAL - 16** The Dispatcher shall be able to assign an ambulance to an incident.

➥ **REQ - 22** The system must be able to get the Individual position.
➥ **REQ - 23** The system must be able to send the position to the nearest ambulance.
✦ **A - 13** An Automated-SOS service is enabled from at least one Third Party.
✦ **A - 14** At least one ambulance dispatcher is available for SOS request serving.
✦ **A - 15** The dispatcher is able to communicate with both the individual and the ambulance associated with the accident.

☆ **GOAL - 17** The Organizer shall be able to create a run.

➥ **REQ - 24** Track4Run must allow the organizer to create a run with a date, time, duration and a path.
➥ **REQ - 25** The system must provide an interface that allows the user to define a path on an interactive map.
✦ **A - 06** The user is logged in the system.
✦ **A - 16** The organizer defines a not empty path.

☆ **GOAL - 18** The Organizer should be able to start a run.

➥ **REQ - 26** Track4Run must allow the organizer to start a run previously created.
✦ **A - 06** The user is logged in the system.
✦ **A - 17** The Organizer has already created a run.
✦ **A - 18** At least two athletes are enrolled in the run.

☆ **GOAL - 19** The Organizer should be able to delete a run.

➥ **REQ - 27** Track4Run must allow the organizer to delete a run previously created.
✦ **A - 06** The user is logged in the system.
✦ **A - 17** The Organizer has already created a run.
✦ **A - 27** The run has not been started.

☆ **GOAL - 20** The Athlete should be able to enroll a run.

➥ **REQ - 28** Track4Run must allow the Athlete to enroll to an already existing run.
✦ **A - 06** The user is logged in the system.
✦ **A - 19** There is at least one created run.

☆ **GOAL - 21** The Athlete should be able to unroll a run.

➥ **REQ - 29** Track4Run must allow the Athlete to unroll a run.
✦ **A - 06** The user is logged in the system.
✦ **A - 20** The Athlete is enrolled to at least one run.

☆ **GOAL - 22** The Spectator should be able to watch a run.

➥ **REQ - 30** Track4Run must allow the Spectator to see the position of the Athletes on a map during a run.
✦ **A - 06** The user is logged in the system.
✦ **A - 21** There is at least one started run.

3.2.1 Use Cases Diagrams

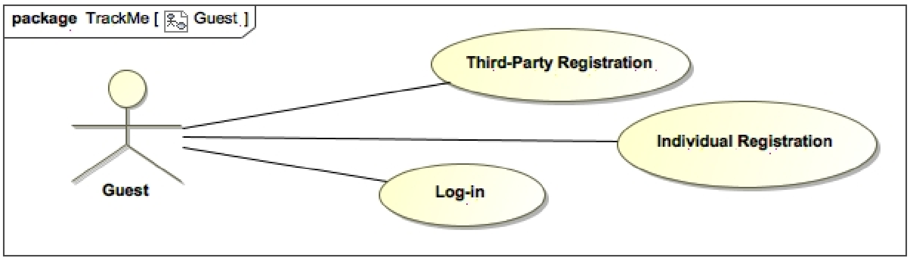


Figure 3: Use Cases: Guest

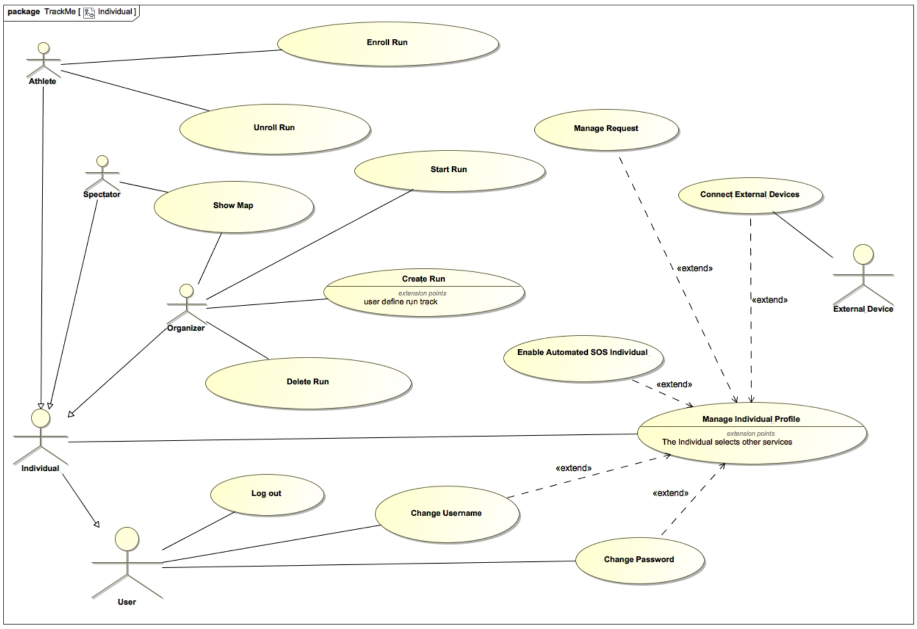


Figure 4: Use Cases: Individual

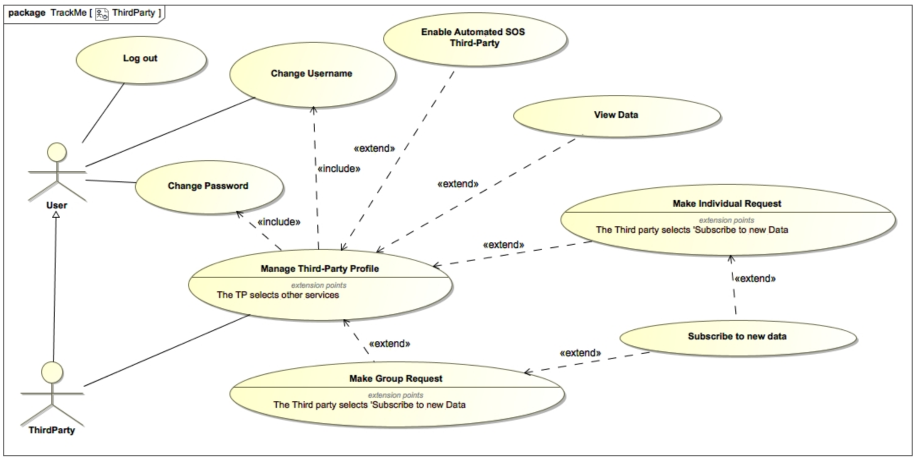


Figure 5: Use Cases: ThirdParty

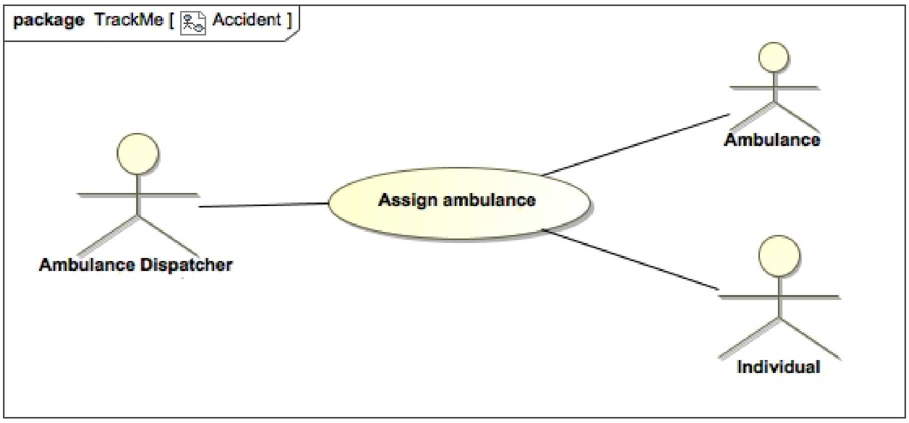


Figure 6: Use Cases: Dispatcher

3.2.2 Use Cases

ID	01
Name	Individual Registration
Brief Description	A guest sign up to the application as an Individual.
Actor	Guest, TrackMe.
Pre-Conditions	The individual has a valid SSN/FC and he has selected the *Individual* option in the Registration page.
Basic Flow	a) TrackMe asks to provide username, password, SSN/FC (based on region) and user personal information such as name, surname, age, gender, e-mail, nationality and optional notes about his health condition. b) The individual fills out the requested information and presses the confirm button. c) TrackMe registers that user as an Individual and adds it to the Database.
Post-Conditions	The individual is successfully registered as an Individual.
Exception	• The Guest provide a username already present on the database. In this case an error message is printed out the guest must chose a correct username. • If the password chosen is too short an error message is printed out and the Guest must chosen an other one.

Use Case 1: Individual Registration

ID	02
Name	Third Party Registration
Brief Description	A Guest sign up to the application as a Third-Party
Actor	Guest, TrackMe.
Pre-Conditions	The third party has a valid VAT and he has selected the *Third-Party* option in the Registration page.
Basic Flow	a) TrackMe asks to provide a VAT number and organization personal information. b) The Third-Party fills out the requested information and presses the confirm button. c) TrackMe asks to provide username and password. d) The system registers that user as a Third-Party and adds it to the Database.
Post-Conditions	The Guest is successfully registered as a Third-Party.
Exception	• The guest provide a username already present on the database. In this case an error message is printed out the guest must chose a correct username. • If the password chosen is too short an error message is printed out and the guest must chosen an other one.

Use Case 2: Third Party Registration

ID	03
Name	Log-in
Brief Description	A Guest sign in to access the application.
Actor	Guest, TrackMe.
Pre-Conditions	The Guest is already registered and it's on the main page.

Basic Flow	a) TrackMe provides to the *Guest* a form composed by two text boxes where the user can write and a button to send the form. The first box is the *’username’*, while the latter is the *’password’* field. b) The Guest writes his username on the *’username’* box and his associated password on *’password’* box and presses the Log-in button. c) TrackMe checks the username is on the DB and that its associated password is correct and checks whether he is registered as individual or Third Party. d) TrackMe finds that username and its associated password are valid and let the guest log-in into the system.
Post-Conditions	The Guest is successfully authenticated either as Individual or Third-Party.
Exception	• If the username is not present in the database an error message is showed and the access to the system is not granted • If the password inserted is not associated with the given username, an error message is showed and the guest cannot access the system

Use Case 3: Log-in

ID	04
Name	User Changes Username
Brief Description	A user changes his username
Actor	User, TrackMe.
Pre-Conditions	The User is logged in the system and is in the change credentials section.
Basic Flow	a) The User clicks on the *'Change credentials'* button and select "Username". b) TrackMe redirects the User to a page and he can now insert a new username writing it into a box. c) The User clicks on the submit button. d) The user's row in the database is updated with his new username.
Post-Conditions	The User has successfully changed his username.
Exception	• The new username is already present on the database and an error mistake is showed

Use Case 4: User Changes Username

ID	05
Name	User Changes Password
Brief Description	A user changes his password
Actor	User, TrackMe.
Pre-Conditions	The User is logged in the system and is in the change credentials section.
Basic Flow	a) The User clicks on the *'Change credentials'* button and select "Password". b) TrackMe asks to the user to insert the old password, a new password and to repeat the latter once. c) The User clicks on the submit button. d) The User's row in the database is updated with his new password.
Post-Conditions	The User has changed his password.

Exception	• If the new password is too short and an error message is showed. • If the old password inserted is not correct an error message is printed out.

Use Case 5: User Changes Password

ID	06
Name	User Log-out
Brief Description	A user logs out from the system
Actor	User, TrackMe.
Pre-Conditions	The User is logged in the system and is on the main page.
Basic Flow	a) The User clicks on the *'Log-out'* button. b) TrackMe asks to the user if he is sure he wants to log-out the system. c) The User select the button *'Confirm'*.
Post-Conditions	The User is not logged anymore into TrackMe.

Use Case 6: User Log-out

ID	07
Name	Manage Individual Profile
Brief Description	An Individual can change the setting of his profile.
Actor	Individual, TrackMe.
Pre-Conditions	The User is logged in the system as an Individual and he's on the profile page.
Basic Flow	a) TrackMe provides a box with the User's description, and a box with the user's image b) Individual can choose above several actions: • The Individual can change any of its personal data. • The Individual can click on the image to update his picture. c) The Individual clicks on the *Save Changes* button provided by TrackMe.
Post-Conditions	The Individual has modified his profile data.

Extension	The Individual clicks on the *Manage Request*, *Connects External Devices* or *Enable Automated SOS button.*

Use Case 7: Manage Individual Profile

ID	08
Name	Manage Third-Party Profile
Brief Description	A Third-Party can change the settings of his profile.
Actor	Third-Party, TrackMe.
Pre-Conditions	The User is logged in the system as a Third-Party and he's on the profile page.
Basic Flow	a) TrackMe provides a box with the user's description, and a box with the user's image. b) Third Party can choose above several actions: • The Third-Party can change any of its personal data. • The Third-Party can click on the image to update his picture. c) The Third-Party clicks on the *'Save Changes'* button provided by TrackMe.
Post-Conditions	The Third-Party has modified his profile data.
Extension	The Third-Party clicks on one of the following buttons: *Enable Automated SOS* , *Make Individual Request*, *Make Group Request*, *View Data.*

Use Case 8: Manage Third-Party Profile

3.2.3 Data4Help

ID	09
Name	Make Individual Request
Brief Description	A Third Party sends a request to an Individual to view his data.
Actor	Third Party, TrackMe.
Pre-Conditions	The Third-Party is logged-in, has selected the *'Make Individual Request'* option and has a valid SSN/FC of a registered individual.

Basic Flow	a) TrackMe provides a box called *'SSN/FC'* where the third party writes the individual *'SSN/FC'* b) Third Party presses the *'Submit'* button. c) TrackMe sends the data request to the individual and informs the Third-Party user that the operation succeed.
Post-Conditions	The third party has access to the data of the individual who received the request.
Extension	The Third-Party selects the 'Subscribe to new data' option.

Use Case 9: Make Individual Request

ID	10
Name	Make Group Request
Brief Description	A third party make a group request that allows to see data of a group of individuals.
Actor	Third Party, TrackMe.
Pre-Conditions	The third-party is logged-in and has selected the 'Make Group Request' option.
Basic Flow	a) TrackMe provides an interface in which a third-party can insert search criteria for group requests. b) Third party fills out the form and then presses the button *'Send Request'*. c) TrackMe checks if the number of individuals of the group who matches the search criteria defined on point *a)* is greater than 1000. d) TrackMe informs the Third-Party that the operation succeed.
Post-Conditions	The Third Party has access to the anonymized data that belongs to a specific group of Individuals.
Extension	The Third-Party selects the 'Subscribe to new data' option.
Exception	• If the number of individual that satisfy the request is lower than 1000 an error message is printed and the request is rejected.

Use Case 10: Make Group Request

ID	11
Name	Subscribe to New Data
Brief Description	A Third Party subscribes to the data of an individual or a group.
Actor	Third Party, TrackMe.
Pre-Conditions	The Third-Party is logged-in and is on requests page.
Basic Flow	a) TrackMe provides a check-box to allow a Third-Party to choose if she wants to be subscribed to new data. b) Third party checks the *'Subscribe'* box. c) TrackMe queries periodically the database to check if data has changed and if so will send the new data to the subscribed third-party.
Post-Conditions	The third party is now subscribed to the data and will receive new data as soon as they are produced.

Use Case 11: Subscribe to New Data

ID	12
Name	Manage Request
Brief Description	An Individual accepts the request to view his personal data.
Actor	Individual, TrackMe.
Pre-Conditions	The individual is logged-in and has selected the 'Manage Request' option in the main page.
Basic Flow	a) TrackMe sends to the individual a request to view his data. The individual can see that the request was sent by a specific third-party. TrackMe provides two buttons called *'Accept'* and *'Reject'*. b) The individual presses the accept button. c) TrackMe sends to the Third-Party a confirmation about his request for that Individual.
Post-Conditions	The third party has access to the Individual data.

Use Case 12: Manage Request

ID	13
Name	View Data
Brief Description	A Third-Party can view the data of the Individuals that have received its request.

Actor	Third-Party, TrackMe.
Pre-Conditions	The Third-Party is logged-in and is in the section to view the data collected by the application.
Basic Flow	a) The system provides a list of all request sent by the Third-Party, both individual and group requests b) Third Party can choose above several actions: • Third Party select the user to which has sent at least an individual request • Third Party clicks on the group request she has sent
Post-Conditions	The Third-Party can view the Individual data, create graphs, publish analysis.

Use Case 13: View Data

3.2.4 Connection of External Devices

ID	14
Name	Connect External Device
Brief Description	An Individual can connect External Device that allows the system to collect data.
Actor	Individual, External Device, TrackMe.
Pre-Conditions	The Individual is in the home page and he pressed the *'Connect External Device'* button.
Basic Flow	a) The system provides two drop-down menus that allows the user specify the type of device and which kind of wireless connection uses: Bluetooth or NFC. b) The user selects the desired option and checks the type of connection, than presses the *Confirm* button. c) Track4Run asks now to connect the device in the way previously chosen. d) The Individual connects the device.
Post-Conditions	The system can now collect data from the external device.

Use Case 14: Connect External Device

3.2.5 Automated-SOS

ID	15
Name	Enable Automated-SOS Third Party
Brief Description	A Third-Party wants to enable the Automated-SOS service.
Actor	Third party, Ambulance Dispatcher, Individual, TrackMe.
Pre-Conditions	The Third-Party is logged-in, it has pressed the Enable Automated-SOS on the main page and has not already activate this service.
Basic Flow	a) TrackMe checks if the Third Party has already enabled the service. b) TrackMe asks if the Third Party wants to enable Automated-SOS. It provides two buttons called *'Confirm'* and *'Back'*. c) Third-Party presses the *'Confirm'* button.
Post-Conditions	Third-Party has enabled Automated-SOS and individuals that enabled Automated-SOS can see the third-party in the Automated-SOS third-party list. An ambulance dispatcher has been assigned to that Third-Party by the system.
Exception	• If no dispatcher is available an error message will be printed out by the system.

Use Case 15: Enable Automated-SOS Third Party

ID	16
Name	Enable Automated-SOS Individual
Brief Description	An Individual wants to enable Automated-SOS. The individual will choose the Third-Party that will provide the Automated SOS functionality.
Actor	Individual, Third-Party, Automated-SOS.
Pre-Conditions	The Individual is logged-in, it has pressed the Enable Automated SOS on the main page and has not already activate this service.

Basic Flow	a) TrackMe checks if the user has already enabled the service. b) TrackMe asks if the user wants to enable Automated-SOS. It provides two buttons called *'Confirm'* and *'Back'*. c) Individual presses the *'Confirm'* button. d) TrackMe provides a table with the list of Third-Parties that have enabled Automated SOS. e) The Individual selects a Third-Party desired and presses *'Confirm'*.
Post-Conditions	Individual has enabled Automated-SOS and has selected the Third-Party that will provide that service.
Exception	• If no Third-Parties has activated Automated SOS an error message is printed.

Use Case 16: Enable Automated-SOS Individual

ID	17
Name	Assign Ambulance
Brief Description	The Dispatcher will assign an ambulance to an individual, related to a particular Third-Party, whose actual data has overcame a certain limit.
Actor	Third Party, Individual, Ambulance Dispatcher, Ambulance.
Pre-Conditions	At least one ambulance must be available; the Third Party must have enabled Automated-SOS. The Individual has selected that Third-Party to provide the service of Automated SOS. The Individual health parameters are under a certain threshold that depends on the health parameter.

Basic Flow	a) TrackMe checks the Individual's health parameters and finds that he has overcame the thresholds. b) TrackMe sends to the Ambulance Dispatcher a request with the GPS position of the subject in danger. c) The Ambulance Dispatcher checks in the ambulance database the nearest one available. d) The Ambulance Dispatcher request the ambulance found in point c). e) The ambulance selected should accept the request, otherwise Ambulance Dispatcher will request the second ambulance available and so on until the request is accepted.
Post-Conditions	An ambulance will go to the position of the Individual.

Use Case 17: Assign Ambulance

3.2.6 Track4Run

ID	18
Name	Create Run
Brief Description	An Organizer creates a new run.
Actor	Organizer, TrackMe.
Pre-Conditions	The Organizer has clicked on the *Plus* button in the Track4Run section.

Basic Flow	a) TrackMe provides a set of boxes called *'Name'*, *'Country'*, *'Date' Time* and two buttons called *'Confirm'* and *'Back'*. b) The Organizer fills the boxes and then presses the ('Confirm') button. c) TrackMe asks the user to defines a *path* using an interactive map d) The Organizer defines the *path* using the tools provided by Track4Run to design a path on the map and presses the *Confirm* button e) TrackMe adds the *run* information to the DB. f) TrackMe informs the Organizer that the operation succeed.
Post-Conditions	The run is created and has been added to the database and its state is *created*. The list of available runs is updated.
Exception	• If the data and time provided for the run are previous the current one an error message is printed.

Use Case 18: Create Run

ID	19
Name	Start Run
Brief Description	An Organizer starts a run that he has previously created.
Actor	Organizer, TrackMe.
Pre-Conditions	The Organizer is logged-in and is in the Track4Run section. The run he want to start must have at least two athletes enrolled.
Basic Flow	a) TrackMe provides a list of runs that the Organizer has created. Furthermore, the system provides two buttons near the each run: *Play* and *Cross*. b) Organizer Presses the *Play* button near the run he wants to start. c) TrackMe asks the Organizer to confirm his choice. d) The Organizer clicks on the *start* button

Post-Conditions	The selected run state is now *started* and Athletes and Spectator will be notified.

Use Case 19: Start Run

ID	20
Name	Delete Run
Brief Description	An Organizer deletes run that he has previously created.
Actor	Organizer, TrackMe.
Pre-Conditions	The Organizer is logged-in and is in the Track4Run section. At least one run has been created.
Basic Flow	a) TrackMe provides a list of runs that the Organizer has created. Moreover, the system provides two buttons *Cross* and *Play.* b) Organizer select a *created* run and presses the ’*Cross* ’ button. c) The system asks if the Organizer wants to delete the run, and provides two buttons *yes* and *no* d) The Organizer clicks on the *yes* button.
Post-Conditions	Any Athlete that was subscribed to that run is no longer subscribed. Athletes can no longer subscribe to that run. That run is deleted from the list of available runs.

Use Case 20: Delete Run

ID	21
Name	Enroll Run
Brief Description	The Athlete is logged-in and is in the Track4Run section. At least one run has been created
Actor	Athlete, TrackMe.
Pre-Conditions	The Athlete is logged-in and is in the Track4Run section. At least one run has been created

Basic Flow	a) TrackMe provides a list of runs that the user can select. these runs must have the state *not-started.* TrackMe also provides two buttons near each created run: *'Enroll'* and *'Watch'* . b) The Individual clicks the *'Enroll'* button of the chosen run. c) TrackMe informs the Athlete that the operation succeed.
Post-Conditions	The Athlete successfully enrolls the run and his position will be sent to all *Spectators* that spectate that run.

Use Case 21: Enroll Run

ID	22
Name	Unroll Run
Brief Description	An Athlete Unrolls to a run previously enrolled.
Actor	Athlete, TrackMe.
Pre-Conditions	The Athlete is logged-in and is in the Track4Run section. He is also enrolled to at least one run.
Basic Flow	a) TrackMe provides the list of runs to which the athlete is enrolled. Near each created run the buttons *'Unroll'* and *'Watch'* . b) Individual selects one run from the list and presses the *'Unroll'* button. c) TrackMe informs the Athlete that the operation succeed.
Post-Conditions	The Athlete is no longer enrolled to that run.

Use Case 22: Unroll Run

ID	23
Name	Watch Run
Brief Description	A Spectator wants to watch a *run.*
Actor	Spectator, TrackMe.
Pre-Conditions	The Spectators is logged-in and is in the Track4Run section. At least one run has been *Started.*

Basic Flow	a) TrackMe provides a list of runs with state started the state *started*. TrackMe also provides two buttons: a *Watch* and *Enroll*. b) Individual selects one run from the list and presses the *'Watch'* button.
Post-Conditions	The spectator can view a map with all athletes positions that are enrolled in that run.

Use Case 23: Watch Run

3.2.7 Traceability Matrix

Raw ID	Goal ID	Req ID	Use Case ID	Comments
r1	G. 01	R. 01	UC. 01	
r2	G. 02	R. 02	UC. 02	
r3	G. 03	R. 03	UC. 03	
r4	G. 04	R. 04	UC. 04	
r5	G. 04	R. 05	UC. 05	
r6	G. 05	R. 06	UC. 06	
r7	G. 06	R. 07	UC. 07	
r8	G. 07	R. 08	UC. 08	
r9	G. 08	R. 09	UC. 09	
r10	G. 09	R. 10	UC. 10	
r11	G. 09	R. 11	UC. 10	
r12	G. 10	R. 12	UC. 11	
r13	G. 10	R. 13	UC. 11	
r14	G. 11	R. 14	UC. 13	
r15	G. 12	R. 16	UC. 12	
r16	G. 12	R. 15	UC. 12	
r17	G. 13	R. 17	UC. 14	
r18	G. 14	R. 18	UC. 15	
r19	G. 14	R. 19	UC. 15	
r20	G. 15	R. 20	UC. 16	
r21	G. 15	R. 21	UC. 16	
r22	G. 16	R. 22	UC. 17	
r23	G. 16	R. 23	UC. 17	
r24	G. 17	R. 24	UC. 18	
r25	G. 18	R. 25	UC. 19	
r26	G. 18	R. 26	UC. 19	
r27	G. 19	R. 27	UC. 20	
r28	G. 20	R. 28	UC. 21	
r29	G. 21	R. 29	UC. 22	
r30	G. 22	R. 30	UC. 23	
Raw ID	**Goal ID**	**Req ID**	**Use Case ID**	**Comments**

3.3 Design and Implementation Constraints

3.3.1 Regulatory Policies

In order to grant individual privacy and data anonymity, the system should be able to manage both individual and group requests:

- In case of individual requests, the system shall check that the third-party knows individual's security social number and should guarantee that only trusted third-parties can access his data;

- In case of group requests, *TrackMe* should be able to check in the number of individuals which will receive the request will be greater enough to guarantee user privacy. TrackMe will approve this type of requests if it is able to properly anonymize the requested 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. Then, to avoid this risk and the possibility of a misuse of data, TrackMe will not accept the request. For simplicity, we assume that TrackMe will accept any request for which the number of individuals whose data satisfy the request is higher than 1000.

3.4 Software System Attributes

3.4.1 Usability

The software application shall be usable for all people including elder ones and not *computers* or *smart-things* experts. Ease of use of both the mobile and the web application are (strictly) needed.

3.4.2 Reliability

The system should work without failures in 99,99% of the cases. The possibility of the service success depends on the stability of the Internet connection and on the availability of the external services such as the Ambulance Dispatcher. In this case the users wont be able to access the system and, in case of accident, no ambulance will receive a call and the user must be informed that the service is unavailable. In case of general system failure the mean time to repair should be less than 2 days. In case of specific Automated-SOS service failure the mean time to repair should be less than 6 hours.

3.4.3 Availability

To guarantee the maximum profit by the service, in particular the one provide by Automated-SOS, the system must be available 24 hours per day and 7 day per week; availability shall also be granted by at least one "ambulance dispatcher". For the two other features *Data4Help* and *Track4Run*, small deviations from this requirement will be acceptable.

3.4.4 Security

The user allows the system to collect their health data. Furthermore, the position of the user is also present in the system, to provide services such as

Automated SOS and Track4Run. If there is a leak of information the privacy of the user is compromised. For this reason at least these security features shall be granted:

- The users passwords are stored in the database hashed with SHA-256.
- The connection between the client and server is encrypted and sent over SSL to guarantee integrity of data.

3.4.5 Maintainability

The system is backed up 2 times a week, in order to guarantee the database integrity and consistency. To avoid loss of data the database is saved in at least two copies and the power supply is provided by a particular system that keeps online at least one database. Moreover an UPS for each *Data Center* shall be present.

3.4.6 Portability

The system is expected to have a long life and it's expected to run on major os such as Windows, MacOs and Linux. The front-end application should work on at least on Android Lollipop 5.0 and iOS5 or major versions.

The developers shall consider that the *E-Health* applications area is a wide and promising market slice thus, in order to be competitive, they should guarantee an high level portability through interactions with other existing widely used software applications and hardware interfaces.

3.4.7 Performances Requirements

In case of accident, the application should be able to get users' position and send it to the ambulance dispatcher in reasonable time, then the dispatcher will forward the position to the nearest ambulance; all in less than five seconds.

4 Formal Analysis using Alloy

Alloy model samples

4.1 Signatures

Listing 1: Data4Help model definition

```
module trackMe
open util/boolean

sig Username, Password{}

abstract sig User{
   username: Username,
   password: Password
}

sig Name, OrganizationName, Ssn, Vat,Position{}

sig Individual extends User{
   name: Name,
   incomingRequests: set IndividualRequest,
   ssn: Ssn,
   enableSos: Bool,
   position:Position
}

abstract sig Request{
   sender: ThirdParty,
   receiver: some Individual,
   accepted : Bool
}

sig IndividualRequest extends Request{
   ssn: Ssn,
}

sig GroupRequest extends Request{}

sig ThirdParty extends User{
   organization: OrganizationName,
   sentRequests: set Request,
   subscribedUsers: set Individual,
   vat: Vat
}
```

Listing 2: Automated-SOS model definition

```
sig Ambulance{
   available: Bool
}

sig AutomatedSos{
   provider: ThirdParty,
   customers: set Individual,
   dispatcher: AmbulanceDispatcher
}

sig AmbulanceDispatcher{
   ambulances: some Ambulance
}

sig Accident{
   position:Position,
   ambulance:Ambulance
}
```

Listing 3: Track4Run model definition

```
sig Track, Duration, Date, Time{}

sig Run{
   state: RunState,
   organizer: Organizer,
   track: Track,
   duration: Duration,
   date: Date,
   time: Time,
}

abstract sig RunState{}

one sig Created extends RunState{}
one sig Started extends RunState{}
one sig Closed extends RunState{}
one sig Deleted extends RunState{}

sig Athlete extends Individual{
   enrolledRuns: set Run
}

sig Spectator extends Individual{
   watchedRuns : set Run
}

sig Organizer extends Individual{
   organizedRuns : set Run
}
```

4.2 Facts

Listing 4: Common facts

```
-----------------------> FACTS <------------------------

fact dataUniqueness{
   no disj u1,u2: User | u1.username = u2.username //
       username
   no disj i1,i2: Individual | i1.ssn = i2.ssn //SSN
   no disj p1,p2: ThirdParty | p1.vat = p2.vat //VAT
   no disj p1,p2: ThirdParty | p1.organization = p2.
       organization //organization name
}
```

Listing 5: Data4Help facts

```
----------------------- DATA4HELP ------------------------

fact individualRequestsAreUnary{
   all r:IndividualRequest | #r.receiver = 1
}

fact sentRequestAreRecorded{
   all r: Request , t: ThirdParty | (r in t.sentRequests) iff
        (r.sender = t)
}

fact receivedRequestAreRecorded {
   all r: IndividualRequest, i: Individual |( r in i.
       incomingRequests) iff (r.receiver = i)
}

fact requestSsnPointAtCorrectReceiver{
   all r: IndividualRequest, i: Individual | (r.ssn = i.ssn)
        iff (i.ssn = r.receiver.ssn)
}

-- a groupRequest will be accepted only if it is anonymous
fact grantAnonimity{
   all r: GroupRequest | isTrue[r.accepted] iff hasAnonimity
       [r]
}

-- to subscribe to an individual's new data, a third party
   must have sent a request that has been accepted
fact subscriptionMustBeAccepted{
   all t:ThirdParty, i:Individual, r:IndividualRequest | i
      in t.subscribedUsers => (requestBetween[r, t, i] and
      isTrue[r.accepted])
}
```

Listing 6: Automated-SOS facts

```
----------------------- AUTOMATEDSOS -----------------------

-- a Third party can provide only one automated-sos service
fact uniqueAutomatedSosService {
   all t: ThirdParty | no disj a1, a2: AutomatedSos |
      enabledService[a1, t] and enabledService[a2, t]
}

-- an Individual can be an Automated-SOS customer only if he
    has activated the service
fact enabledSosMeansCustomer{
   all a: AutomatedSos, i: Individual | isCustomer[i, a] =>
      isTrue[i.enableSos]
}

-- an Individual can be subscribed to only one Automated-SOS
    provider
fact justOneAutomatedSosSub{
   all i: Individual | no disj a1,a2: AutomatedSos |
      isCustomer[i, a1] and isCustomer[i, a2]
}

-- ambulances are managed by some dispatcher
fact ambulancesAreManaged{
   all a:Ambulance | some d:AmbulanceDispatcher | a in d.
      ambulances
}

-- ambulances can have only one dispatcher
fact ambulancesArentShared{
   all a:Ambulance | all disj d1,d2:AmbulanceDispatcher | a
      in d1.ambulances => a not in d2.ambulances
}

-- an accident must have happened where there is an
   individual
fact accidentHasAnIndividualPosition{
   all a:Accident | some i:Individual | a.position = i.
      position
}

-- an ambulance associated with an accident is not available
```

```
fact ambulancesAvailability{
   all a:Ambulance, acc: Accident | isManaging[a, acc] =>
       isFalse[a.available]
}

-- an accident must has managed by only ony ambulance
fact ambulanceManagesOnlyOneAccident{
   all disj acc1,acc2:Accident | all a:Ambulance |
       isManaging[a, acc1] => not isManaging[a, acc2]
}

fact dispatcherWorksForAutomatedSos{
   all d:AmbulanceDispatcher | some a:AutomatedSos | a.
       dispatcher = d
}
```

Listing 7: Track4Run facts

```
----------------------- TRACK4RUN -----------------------

fact organizedRunsAreRecorded{
   all r:Run, o:Organizer | (r.organizer = o) iff (r in o.
       organizedRuns)
}

-- there can't exist two runs that have the same track in
    the same date
fact noDuplicatedRun{
   no disj r1,r2: Run | isSameRun[r1,r2]
}

-- an athlete can't enroll 2 runs that happens in the same
    date/time
fact noMultipleEnrollement{
   all a:Athlete | all disj r1,r2: Run | (isEnrolled[r1,a]
       and isEnrolled[r2,a]) => not isSameDate[r1,r2]
}

-- a spectator can't enroll 2 runs that happens in the same
    date/time
fact noMultipleWatch{
   all s:Spectator | all disj r1,r2:Run | (isEnrolled[r1,s]
       and isEnrolled[r2,s]) => not isSameDate[r1,r2]
}

-- an athlete can't watch the runs where he is also enrolled
fact athletesCantWatch{
   no s: Spectator, a: Athlete | isSameIndividual[s,a] and #
       a.enrolledRuns > 0 and hasCommonRuns[s,a]
}

-- if a run exists it must be organized by some organizer
fact runMustBeOrganized{
   all r:Run | some o:Organizer | hasOrganized[r,o]
}

-- athletes can enroll to a run only if that run state is "
    created"
fact athletesEnrollOnlyCreatedRuns{
   all a:Athlete, r:Run | r in a.enrolledRuns => r.state =
       Created
}

-- spectator can enroll to a run only if that run state is "
    started"
fact spectatorsEnrollOnlyCreatedRuns{
   all s:Spectator, r:Run | r in s.watchedRuns => r.state =
       Started
}
```

4.3 Predicates

Listing 8: Data4Help predicates

```
-----------------------> PREDICATES <-----------------------

----------------------- DATA4HELP -----------------------

pred isSameRequest[r1,r2:Request]{
   r1.receiver = r2.receiver and r1.sender = r2.sender
}

pred requestBetween[r:Request, t:ThirdParty, i:Individual]{
   r.sender = t and r.receiver = i
}

pred isSubscribedToData[t:ThirdParty, i:Individual]{
   i in t.subscribedUsers
}

pred hasAnonimity[r: GroupRequest]{
   #r.receiver > 1000
}
```

Listing 9: Automated-SOS predicates

```
-------------------- AUTOMATED-SOS ---------------------

pred enabledService[a: AutomatedSos, p: ThirdParty]{
   a.provider = p
}

pred isCustomer[i:Individual, a:AutomatedSos]{
   i in a.customers
}

pred isManaging[a:Ambulance, acc:Accident]{
   acc.ambulance = a
}
```

Listing 10: Track4Run predicates

```
----------------------- TRACK4RUN -----------------------

pred isSameIndividual[s:Spectator, a:Athlete]{
   s.ssn = a.ssn
}

pred hasCommonRuns[s:Spectator, a:Athlete]{
   a.enrolledRuns & s.watchedRuns != none
}

pred isEnrolled[r: Run, a:Athlete]{
   r in a.enrolledRuns
}

pred isEnrolled[r: Run, s:Spectator]{
   r in s.watchedRuns
}

pred hasOrganized[r:Run, o:Organizer]{
   r in o.organizedRuns
}

pred isSameDate[r1, r2 : Run]{
   r1.date = r2.date and r1.time = r2.time
}

pred isSameRun [r1, r2 : Run]{
   isSameDate[r1,r2] and r1.track = r2.track
}
```

Listing 11: Common predicates

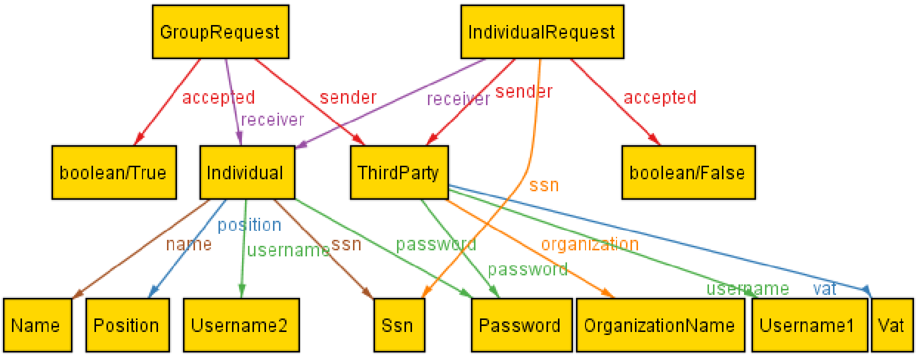
```
---------------------------------------------------------
pred disableData4Help{
   #Request = 0
   #ThirdParty.subscribedUsers = 0
}

pred disableAutomatedSos{
   #AutomatedSos = 0
   #Ambulance = 0
}

pred disableTrack4Run{
   #Athlete = 0
   #Spectator = 0
   #Organizer = 0
   #Run = 0
}
```

```
pred data4Help{
   disableTrack4Run
   disableAutomatedSos
   some IndividualRequest
   some GroupRequest
}

pred automatedSos{
   disableData4Help
   disableTrack4Run
   some Ambulance
   some AutomatedSos
   #AutomatedSos.customers > 0
}

pred track4Run{
   disableData4Help
   disableAutomatedSos
   some Organizer
   some Athlete
   some Spectator
   some Run
}

pred showAll{}
```

4.4 Worlds

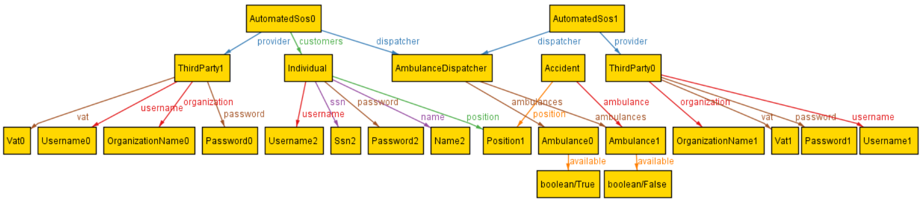
Listing 12: Run data4Help

```
run data4Help for 3 but exactly 1 Individual
```



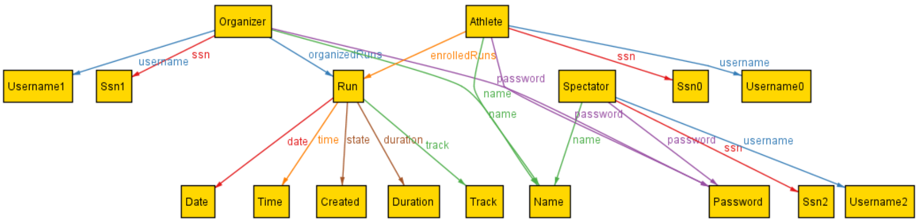
Listing 13: Run automatedSos

```
run automatedSos for 3 but exactly 1 Individual
```



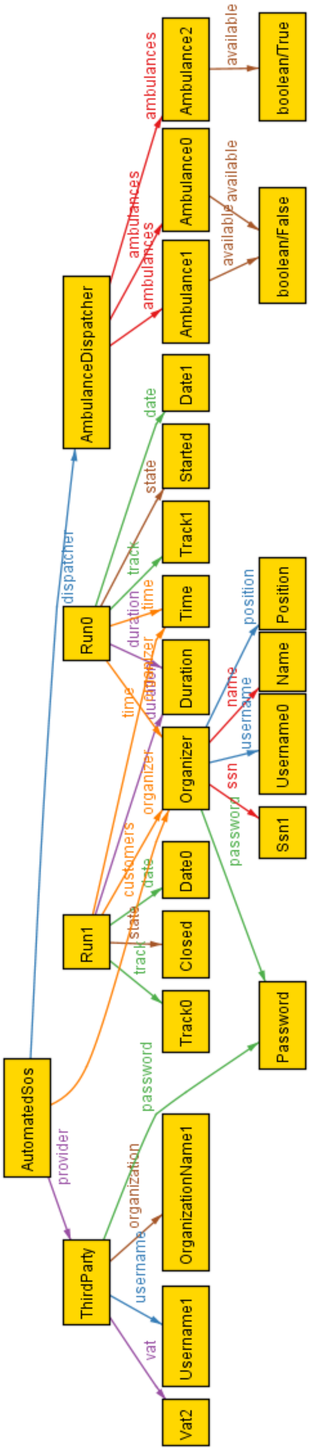
Listing 14: Run track4Run

```
run track4Run for 3 but exactly 1 Run
```



Listing 15: Run showAll

```
run showAll for 3
```



5 Effort Spent

- **Davide Rutigliano: 50h**
- **Davide Matta: 50h**
- **Claudio Ferrante: 50h**