Politecnico di Milano
AA 2018-2019

POLITECNICO
MILANO 1863

Computer Science and Engineering
Software engineering 2

TrackMe RASD

Requirements Analysis and Specification Document
Version 1.0
24/10/18

Diego Piccinotti, Umberto Pietroni, Loris Rossi

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this document is to analyze and illustrate specifications for the Data4Help system. This will be accomplished through a detailed analysis of the solution proposed and of the goals, assumptions and requirements that are necessary to realize it.
This document is intended to be used by the clients requiring the development of the system, its end users and all those who are involved in the development process, mainly project managers, analysts and development teams.

Data4Help is designed as a platform to enable third parties to access its users' health and location data. The system continuously collects these data through wearable devices and stores them to enable later access and/or aggregated statistics analysis.
Third parties can request data of some specific individuals, whose collection has to be approved by the user (which grants access to their data from this moment on), or data of a group of users which must be larger than 1000 individuals to allow for proper data anonymization.
Data4Help is then extended in its functionality by **AutomatedSOS**, an additional service built on top of the existing data collection platform, which monitors elderly users' health parameters and automatically requests an ambulance to their location if they exceed a risk threshold determined by a preventive medical checkup.
Finally, Data4Help platform can be used also in run competitions to collect runners' statistics and enable their registration to the runs through another additional service, **Track4Run**, which also enables the run organizers to manage runs' participants and broadcast the live coverage to spectators through a map showing the live position of the runners.

1.2 Scope

In this section we describe the boundaries between the environment and the system to be.
In particular, we distinguish between world, shared and machine phenomena, which we further discuss in paragraph 2.1.
Moreover, we define the goals of the system in order to satisfy the stakeholders' needs.

1.2.1 World, shared and machine phenomena

World phenomena are the ones which the machine cannot observe.
In our context, world phenomena are, for instance:
- Movements of a user from a position to another one;
- The change of a user's health status
- An organizer plans a new run to be held

Phenomena entirely occurring in the machine are considered machine phenomena.
For instance, data manipulation from the database or checking for the proper anonymization of a group of data shall be done by the system to be.

Regarding shared phenomena, which involve both the environment and the system, we have events like:
- A third part requests for a specific user's data
- A user signs up in the system
- The system asks for an ambulance dispatch to the external provider

Further shared phenomena are discussed in the Product Perspective paragraph (2.1).

1.2.2 Goals

G1: The user can be recognized by providing a form of identification
G2: Allow third parties to monitor data about location and health status of individuals.
G3: Allow third parties to access data relative to specific users
G4: Allow third parties to access anonymized data of groups of users
G5: Allow third parties to offer a personalized and non-intrusive SOS service to elderly people so that an ambulance arrives to the location of the customer in case of emergency.
G6: Allow athletes to enroll in a run
G7: Allow organizers to manage runs
G8: Allow spectators to see on a map the position of all runners during the run

1.3 Definitions, acronyms, abbreviations

1.3.1 Definitions

- **User**: individual who accepts to give his location and health status data to Data4Help

- **Third party**: individual or organization registered to Data4Help which can request users' data
- **Data collection**: gathering of users' data through a wearable device
- **Anonymized data**: data about 1000 or more users whose personal information has been previously removed so that they are not directly relatable to the system's users.
- **Risk threshold**: Set of boundary health parameters defined for each AutomatedSOS user. If monitored values of the user's health parameters exceed these boundaries, an ambulance is dispatched to the user's location.
- **Athlete**: user who participates to a run.
- **Run organizer**: third party who can arrange a new run for athletes.
- **Spectator**: public individual who follows a run through a map with runners' positions
- **Wearable**: a personal device provided with biometric sensors and GPS given to each user for free after the registration process
- **Elderly**: user who is subscribed to AutomatedSOS and is older than 60 years old

1.3.2 Acronyms

- **API**: Application Programming Interface
- **GPS**: Global Positioning System

1.3.3 Abbreviations

- **Gn**: n-goal
- **Dn**: n-domain assumption
- **Rn**: n-functional requirement
- **Rn-NF**: n-non functional requirement

1.3.4 Conventions

We'll use "their" and "them" to indicate "his" or "her". [to be expanded]

1.4 Reference Documents

- Specifications document: "Mandatory Project Assignment AY 2018-2019"
- IEEE Standard on Requirement Engineering (*ISO/IEC/IEEE 29148, Dec 2011*)

1.5 Document Structure

(to be written when the document is almost complete)

2 Overall Description

2.1 Product perspective

here we include further details on the shared phenomena and a domain model (class diagrams and state diagrams)

Describes external interfaces: system, user, hardware, software; also operations and site adaptation, and hardware constraints

2.1.1 Phenomena

World: (*world phenomena the machine cannot observe)*

- Users move (their position changes)
- Users' health parameters vary
- Organizers plan runs
- The external provider dispatches an ambulance

Shared: (*controlled by the world and observed by the machine or controlled by the machine and observed by the world)*

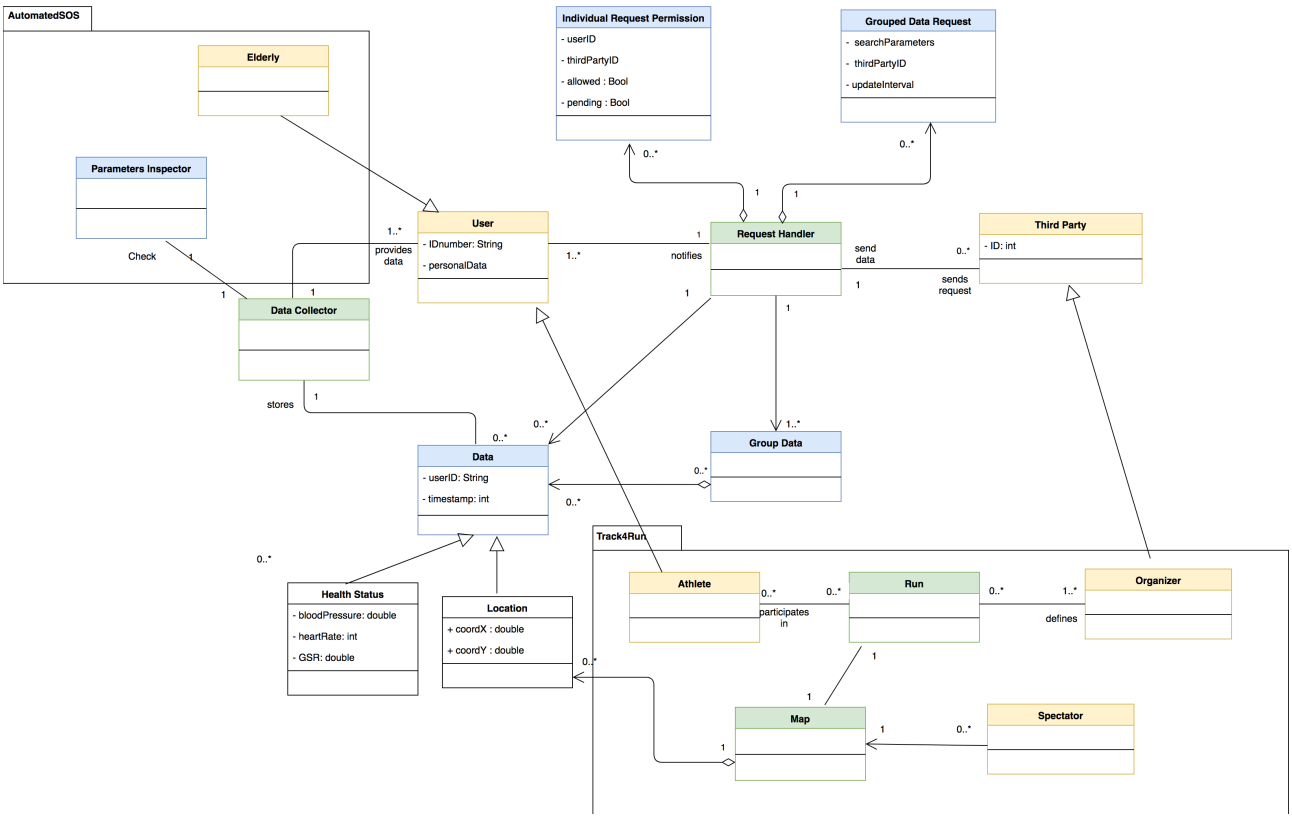
- User and third parties can sign up and login to the system
- User data collection by the system: the data is collected from the sensor on the user's wearable, thus the phenomenon being shared
- Data requests from third parties: the system acts as an interface between user data and third parties
- System sends data to the third parties
- Data collection acceptance/denial by the user: again, the system acts as an interface between two world entities and knows what is happening
- The system asks for an ambulance dispatch to the external provider: the system produces an effect on the world, the dispatching, so the phenomenon is shared
- Position update on map for runners
- Organizers input runs data to the system
- Athletes enroll to runs available in the system: this phenomenon is shared because we assume that enrollment to runs happens only through the system and is not a copy of a physical form.

- Spectators connect to the system to watch runs: spectators are able to follow the live coverage through the system displaying the map, so the phenomenon is shared

Machine: (*phenomena located entirely in the machine)*

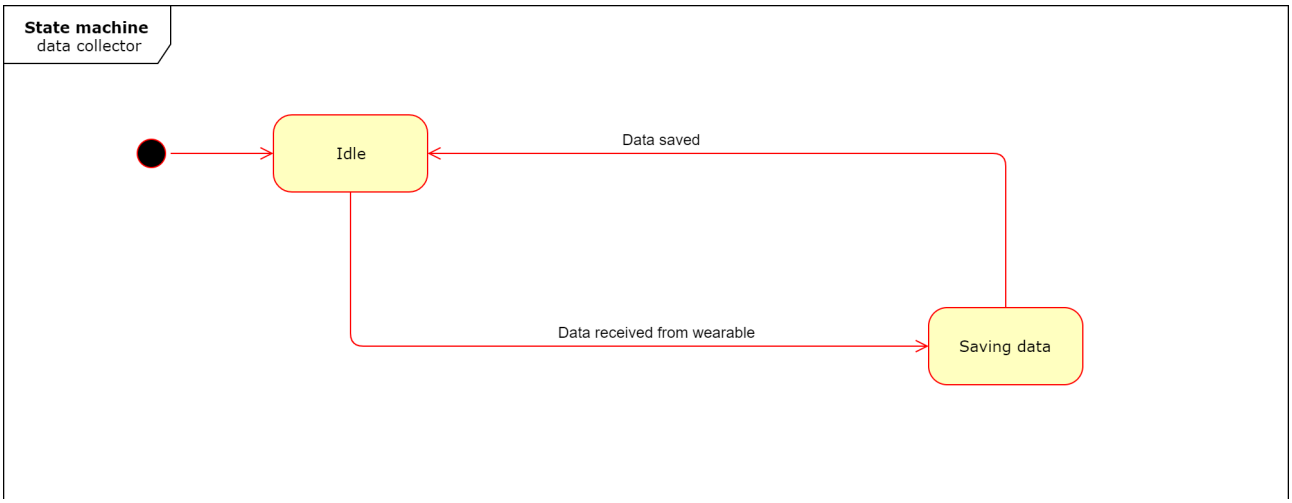
- Database queries
- Anonymization check
- Check of health parameters against risk threshold
- System enforces unicity of usernames

2.1.2 Diagrams

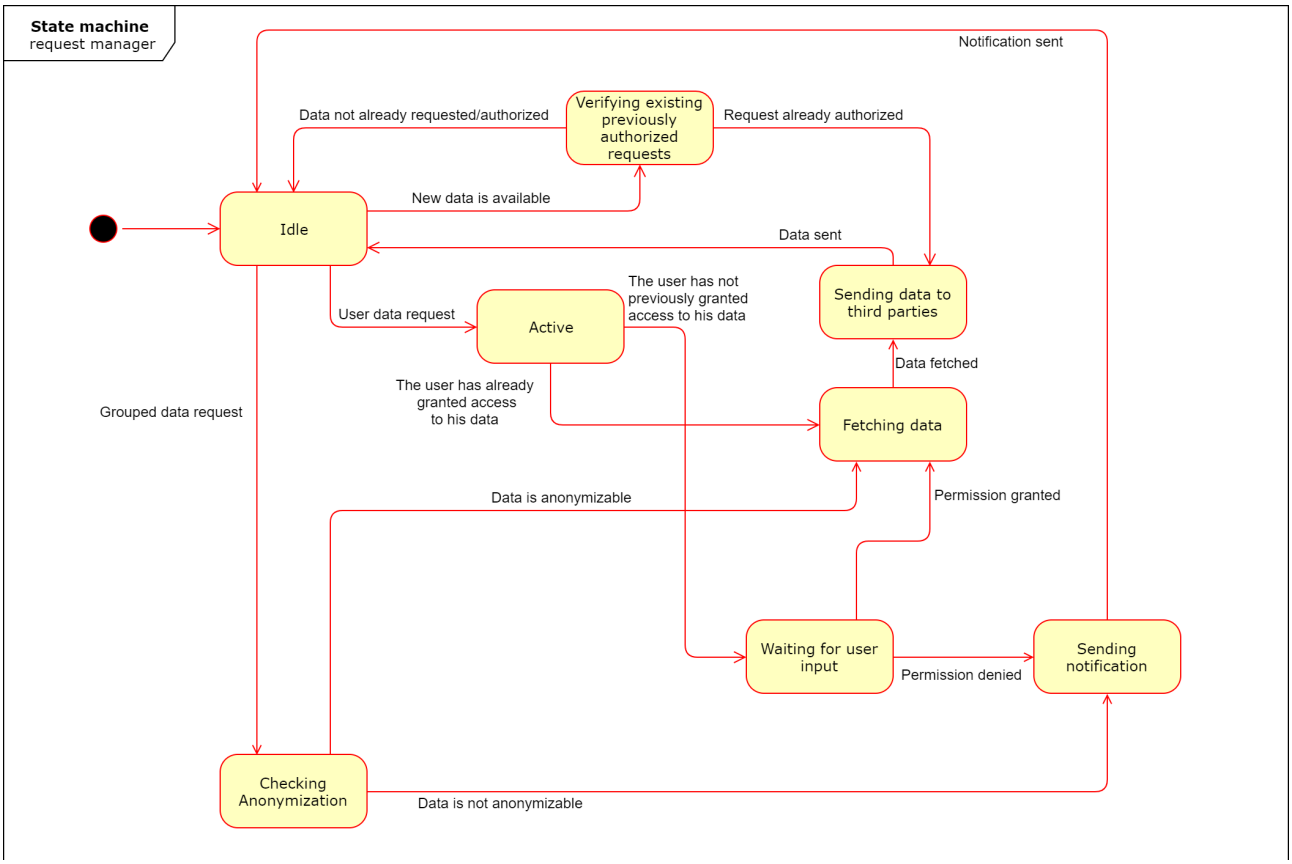


This *class diagram* models how the system is structured and the relations among its parts. One of the main objectives of this diagram is to highlight the system boundaries and in order to facilitate this, actors and Phenomena which are part of the World are colored in yellow, the Shared phenomena in green and, finally, the part of the system which is observed only by the Machine is blue.
User and Third Party are the most important actors because the first one is the provider of Data gathered and stored by the Data Collector class, while the other one triggers the

data retrieval process regulated by the Request Handler class, whose task is to check if the Third-Party requests are accepted and then fetching the desired data.
On top of this system two other services are built: *AutomatedSOS* and *Track4Run*. The former consists, in this diagram, of the Elderly class (which is a subclass of User) and of the Parameters Inspector, that monitors each elderly user's Health Status Data to check if the parameters are below risk thresholds. The latter is composed by Athlete class, a subclass of User that represents all the Run participants, and by the Organizer, a subclass of Third Party that defines the Run.
All runners' Locations can be seen on Map by the Spectator, another type of actor.



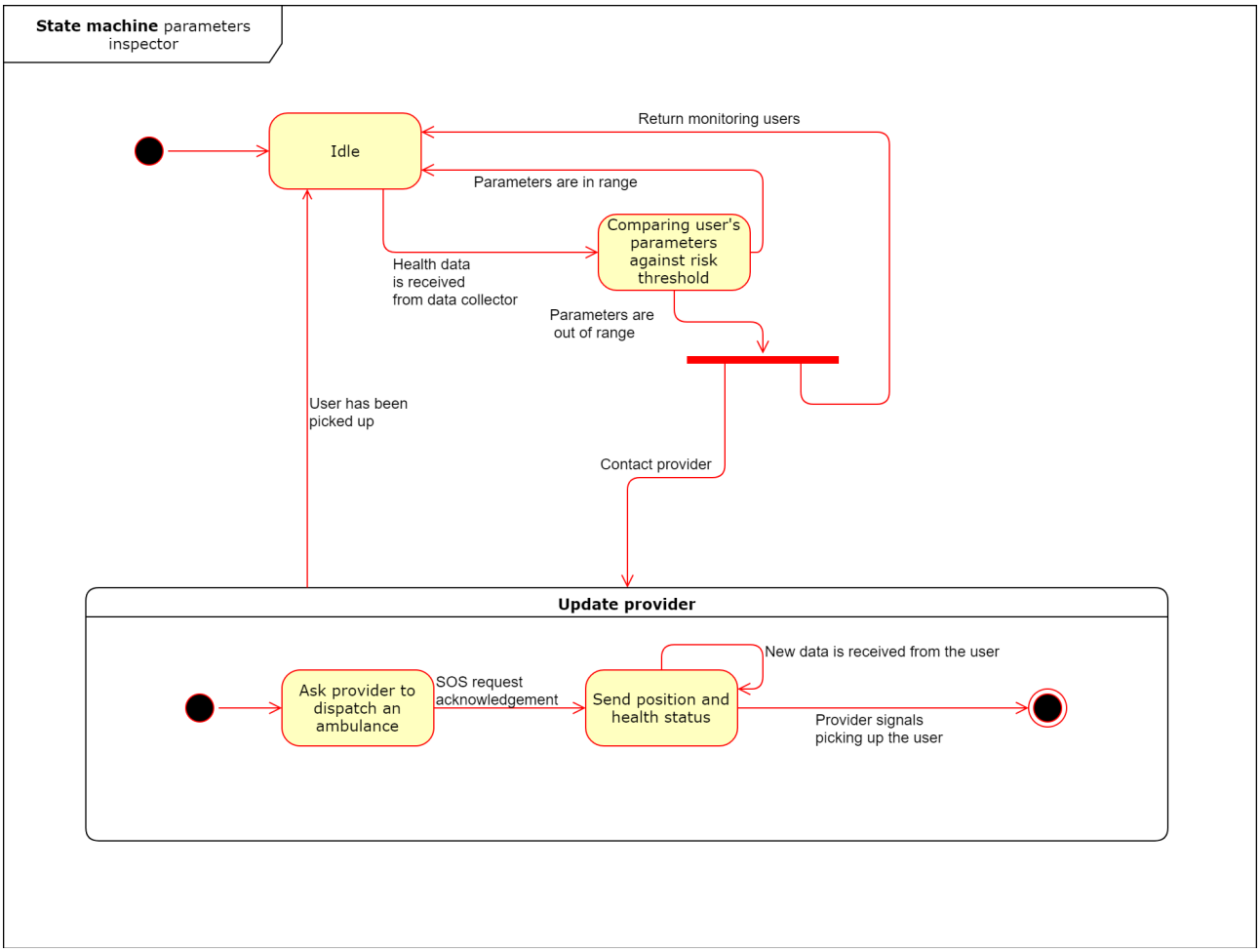
This state diagram shows the behavior of the *Data Collector* class, whose duty is to collect data from the sensors placed on the users' wearable devices. The class exhibits a cyclic behavior: starting from an idle state it awakens after data is received from a wearable device, then it proceeds to store it in the system. Once the save is completed, the class returns idle.



The request manager class, as the name suggests, manages data requests from third parties and fulfils them when data is available. The availability of the data depends on different conditions regarding the nature of the request:

- If the data request concerns a *single user's data*, the class shall verify if the user has previously granted the permission to allow access to their data from that specific third party. If the permission was not already granted, the class shall proceed to ask the user for permission and fulfil the request if they grant it. Otherwise it shall notify the third party of the unavailability of the data.
- If the request is concerning grouped data, the class shall verify that the data can be anonymized (1000 or more users) and, if this is possible, send the data properly anonymized. Otherwise it shall notify the third party of the unavailability of such data.

Finally, the class is able to manage incoming new single user's data and dispatch it to the third parties, if they already placed an authorized request for such data. Regarding grouped data requests, instead, after a regular time interval has expired the state machine proceeds to check if anonymization on that group is still possible and then will proceed to satisfy or deny the request as usual.



The *parameters inspector* class has the responsibility to monitor single users' health parameters and, if they are below a risk threshold, ask the external ambulance provider to dispatch an ambulance and pick the user up to transport him to an health center. The state machine stays in an idle state until it receives health and GPS data from *data collector* class, then it activates and compares the user's parameters against the risk threshold stored in the system and acts as follows:

- If the parameters are in range, the class returns idle since there is no risk situation detected.
- Else, if the parameters are out of range, the state machine enters a substate set called *update provider* to communicate all necessary information to the ambulance provider: first it places a request for an ambulance dispatch and then, when it receives acknowledgement for such request, starts sending health and position data of the user until the ambulance provider signals the system to have completed picking up the user. Finally, the state machine goes back to idle, exiting the substate set.

The fork after parameters out of range are discovered means that the state machine can monitor multiple users simultaneously and does not get fully occupied by a single emergency. To prevent ignoring other users getting sick while that specific emergency is being serviced, the *update provider* substate set shall be handled as a separated task that does not block the machine waiting for its completion.

2.2 Product Functions

(here we include the most important requirements - Summary of major functions)

Data4Help

Data4Help's purpose is to allow third parties to search and collect data about specific individuals or groups for their business or research projects. To enable this, individuals, who are users of the system willingly providing their data for collection from Data4Help, must be uniquely identified to allow to retrieve their data. Since users' privacy is nowadays an important concern, *Data4Help* won't allow third parties to monitor a user unless he explicitly accepts such behavior.

Data queries

To perform a single user's data query, a third party must be registered and access the system with their credentials. Then the system shall require the desired user's identifier, specifically their ID card number, and check if they have already granted permission to the third party to collect their data. If this is not the case, the user shall be notified through an email and asked if they want to allow data collection from that specific third party. If the user grants access to their data, then the system shall immediately provide it to the third party and send new data as soon as it is available, if this is required by the third party. In case the user denies the data access request, the system shall notify the third party of the unavailability of such data.

Grouped data queries are also allowed by the system but *Data4Help*, aiming to protect the privacy of its users, will guarantee their satisfaction only if they are concerning a group larger than 1000 individuals. In this case, the third party will specify a criterion to select users being sampled (e.g. people living in Città Studi block, with age between 18 and 25) and the system will immediately provide anonymized data of users matching the criterion, without previously asking permission to them.
When the anonymization threshold is not reached, the query shall not be satisfied and the concerned third party notified of the unavailability of such data.

For both types of query, the system enables third parties to request updates on future data, allowing to continuously track a single user or a group of individuals and receive data as soon it is available in the system.

AutomatedSOS

AutomatedSOS aims to be a "smart" SOS service for elderly people. Taking advantage of modern biometric sensors and wireless connections, the service monitors its users' real-time health status and position. An elderly person, that has reached a minimum age, can get access to AutomatedSOS service upgrading their *Data4Help* account by inputting their risk threshold, that has been evaluated through a preventive checkup.

Whenever an AutomatedSOS user's health parameters get below a risk threshold, an external ambulance provider is contacted by the service and the user's location and health status are immediately signaled to allow for a steady pick up and an easier on-site treatment. Furthermore, sick user's data get continuously broadcasted to the ambulance provider to keep him updated until he confirms picking up the user.

Track4run

Track4Run is an additional service which exploits *Data4Help* platform to allow an easy management of running competitions and gathering of statistics about runners' performance.
Race organizers can register to the system as third parties and, after logging in, can insert new competitions into the system. Another feature available to them is to manage participants, adding or removing them manually from the participants' list that is shown in the competition detail page.
Standard TrackMe users can upgrade their account to a *runner* profile to participate to runs available in the system.
After logging in, the system shows them a list of available races, which they can search through and select a single competition to enroll to. After selecting the competition, the details are shown to the user, who can confirm the enrollment or go back to the run list to check other competitions.
After completing a run, each user can check their data from their personal page and the competition's result.
Spectators are unregistered people who can spectate a run through a dedicated page, showing a live map with positions of each runner and a live leaderboard with runners' times. The desired run can be selected from a run competitions that are actually live or will begin shortly.

2.3 User characteristics

2.3.1 Actors

- **User**: an individual registered in the system who has accepted to give his location and health data, through a wearable device provided by the company.
- **AutomatedSOS user:** elderly user who is subscribed to AutomatedSOS service.
- **Third party**: an individual or organization registered in the system. It can login to the system, and then request user data, either a specific user's data or data about group of users (properly anonymized).
- **Run organizer**: third party that can schedule run competitions through the Track4Run system. It can schedule a new run, define the path for a new run, check the athletes participating in a run and add or delete them.

- **Athlete**: a user who participate to a run organized through the Track4Run system. He can login to the system, see a list of all the scheduled runs and participate to the desired ones. He can see the data relative to his past run competitions.
- **Spectator:** an unregistered individual who can see a list of active runs in a public website. After selecting the desired one, the spectator can see a live map with the runners' positions on the track.

2.4 Assumptions, dependencies and constraints

2.4.1 Domain Assumptions

D1: Users are uniquely identified by their ID number or fiscal code. [G1, G2]
D2: Information provided by the user during the registration process are assumed to be true. [G1, G2, G3]
D3: User's position is available through GPS. [G1]
D4: User's health related data (heart rate and blood pressure) values are available through a wearable personal device [G1]
D5: A partner of TrackMe provides an ambulance service 24/7. [G4]
D6: The risk threshold for each user is obtained through a preventive hospital check. [G5]
D7: There is an external provider offering a map service [G8]

3 Specific Requirements

(the IEEE standard suggests 8 different templates for this section, we may have a look at them –
All the requirements go in here)

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communication Interfaces

3.2 Scenarios

(Heuristics for finding scenarios:
What are the primary tasks that the system needs to perform?
What data will the actor create, store, change, remove or add in the system?
What external changes does the system need to know about?
What changes or events will the actor of the system need to be informed about?)

Scenario 1
Kate would like to watch his friend Umberto running in the Run4Fun run competition organized by the company HealthFun. She opens the Track4Run website and scrolls through the list of active runs, until she finds Run4Fun's. Then she clicks on the "Watch on live map" button, and the browser shows a live map with the current runners' position, including Umberto's.

Scenario 2
The HealthAnalytica company would like to monitor the health status of people aged from 18 to 25 and living in the Città Studi block in Milan. Jacob, an HealthAnalytica employee, logs into the Data4Help website and enters the "Group Data" section.
Then he selects the settings for the desired data: mininum and maximum age, and the geographical area to consider. Next, he sets a Daily Update for this data, and eventually he clicks on Subscribe. Since there are not enough users satisfying the chosen criteria, the website shows an error message because it can’t properly anonymize the data. So, the HealthAnalytica employee changes the search settings, and clicks again on Subscribe. Now Data4Help shows a confirmation message to let Johnny know the request has been accepted successfully.

Scenario 3

Andy, a teenager registered to Data4Help, found out that Data4Help can help elderly people in case of need. Therefore, he tells his grandpa Mike, which is 66 years old, to subscribe to the AutomatedSOS service, so that Mike can feel safer when he is alone. Mike goes to the doctor for a medical checkup, and after the visit he subscribes to *AutomatedSOS* and inputs his health status values into his *Data4Help* personal area, together with risk thresholds.

Scenario 4
Arnold, a personal trainer working in the gym StayFit based in Milan, wants to write a personal work schedule for a gym member.
Since each member has included in his season pass also a subscription to Data4Help and has already agreed to send data to StayFit, Arnold wants to look up the member's data. In particular, in order to give him a schedule with exercises adequate to his level, Arnold is looking for the member's health status. Therefore, he opens Data4Help's web page, logs in with the gym account and searches for the member's data using his ID number and after this he is able to write the right work schedule for him.

3.3 Functional requirements

(Taken from Prof document:
Definition of use case diagram, use cases and associated sequence/activity diagrams, and mapping on requirements. Scenarios)
It's important to keep track of the relation between use cases and requirements – ***Traceability Matrix*** *)*

3.3.1 Data4Help

Users: source users and third-party services.

G1: The user can be recognized by providing a form of identification
[D1]: Users are uniquely identified by their ID number or fiscal code.
[D2]: Information provided by the user during the registration process are assumed to be true
[R1]: The system shall allow registration of individuals through the creation of a username and a password.
[R2]: The system shall guarantee the unicity of usernames.

[R3]: The system shall ask the user to provide his personal data (birthdate, gender, residency address, ID number / fiscal code).
[R4]: The system shall ask the user to agree to a policy that specifies that, by registering, users agree that TrackMe acquires their data.

G2: Allow third parties to monitor data about location and health status of individuals.
[D3]: User's position is available through GPS.
[D4]: User's health data (heart rate and blood pressure) are available through a wearable personal device
[R5]: The system shall store past position and health data of every single user.
[R6]: The system shall support the registration of third parties and provide them with a unique identifier (ID).
[R7]: Third parties shall be allowed to subscribe to new data and the system sends data as soon as they are produced. *[also in G3, G4]*

G3: Allow third parties to access data relative to specific individuals
[D2]: Users are uniquely identified by their ID number or fiscal code.
[R7]: see above
[R8]: The third parties shall be able to request and receive a specific individual's data through his ID number or fiscal code.
[R9]: Upon every data collection request, the system shall check if the permission to access that data has already been granted by the user, otherwise it shall ask them.

G4: Allow third parties to access anonymized data of groups of individuals
[R7]: see above
[R25]: The system shall allow third parties to search for the desired group of individuals.
[R10]: The system shall accept a grouped data collection request only if the data can be properly anonymized. Anonymization is considered proper if the number of people involved in the request is greater than 1000. *(Same comment as R9)*
[R26]: The system shall allow third parties to subscribe to periodic updates relative to a certain group of individuals provided that the data contained in each update can be properly anonymized.

3.3.2 AutomatedSOS

Users: elderly people and third parties.

G5: Allow third parties to offer a personalized and non-intrusive SOS service to elderly people so that an ambulance arrives to the location of the customer in case of emergency.
[D6]: The risk threshold for each user is obtained through a preventive hospital check.
[D13]: The system shall allow Data4Help users to subscribe to AutomatedSOS service by providing their risk threshold.

[R11]: Frequently enough, health parameters are monitored by the system and compared against the threshold to detect risk situations.
[R12]: The system shall allow the ambulance provider to notify when the user has been picked up, in order to stop data transmission.

3.3.3 Track4Run

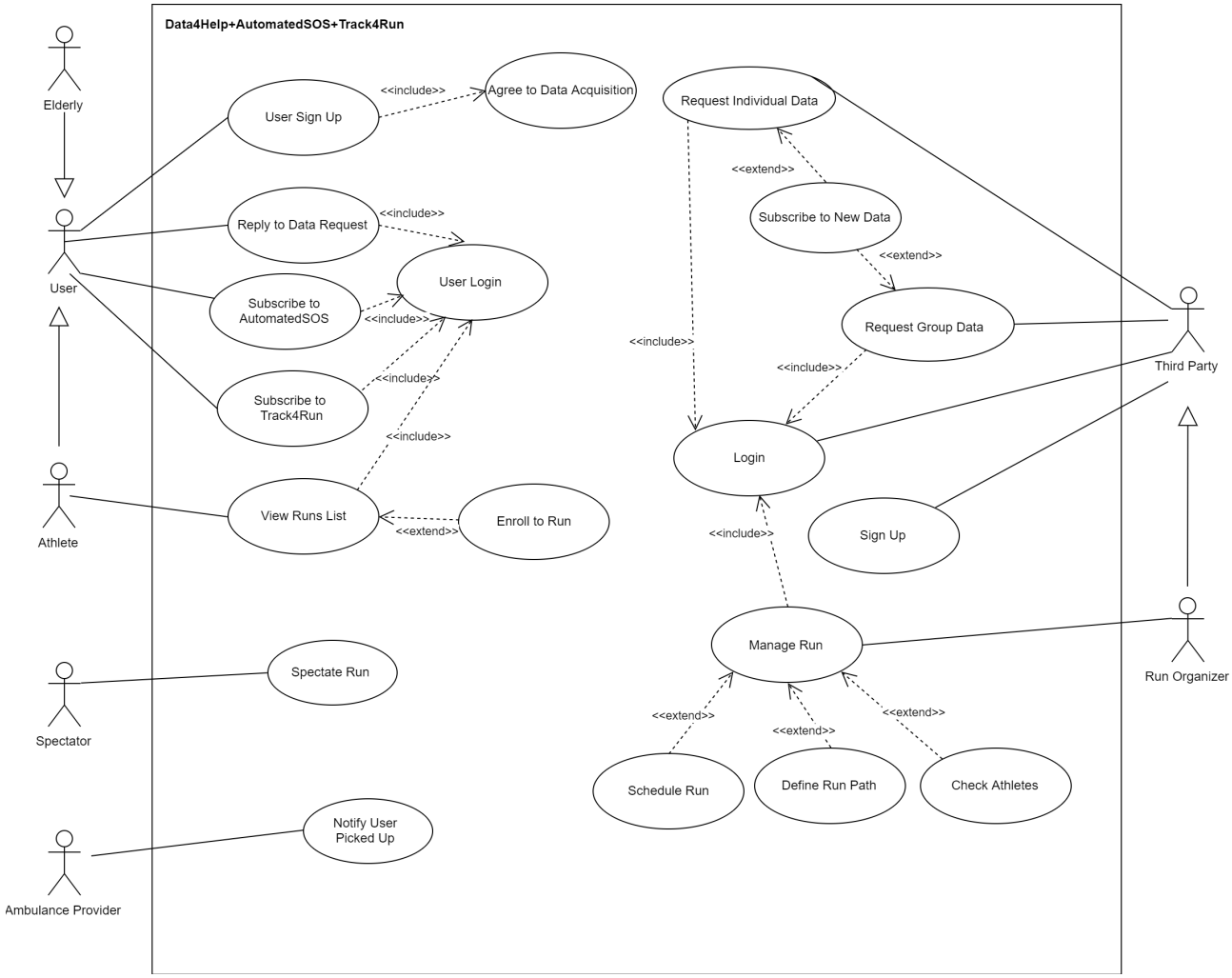
Users: Runners, organizers and spectators

G6: Allow athletes to enroll in a run
[R13]: The system shall allow participants to register to the system.
[R14]: The system shall allow participants to check a list of available runs.
[R15]: The system shall provide the ability to enroll to the desired run only to registered athletes.
[R16]: The system shall allow enrolling only if the user has already agreed to share publicly his location for the duration of the run.

G7: Allow organizers to manage runs
[R17] The system shall allow organizers to register to Track4Run platform.
[R18] The system shall allow organizers to create and delete races.
[R19] The system shall allow organizers to add a path for the run.
[R20] The system shall allow organizers to check a participants list.
[R21] The system shall allow organizers to add or remove participants before the start of the race.

G8: Allow spectators to see on a map the position of all runners during the run
[D7] There is an external provider offering a map service
[R22] The system shall provide a public list of live runs
[R23] The system shall allow the spectator to see a map of the desired run, with live participants' position
[R24] The system shall update the positions of the runners on the map as soon as new data is received

3.3.4 Use Case Diagram



3.3.5 Use Cases

Name	Sign Up
Actor	Third Party
Entry Condition	This use case starts when Third Party clicks the "Sign up as third party" button on Data4Help webpage
Flow of Events	1. The system shows the registration form to Third Party 2. Third Party fills all the mandatory fields and provides the necessary information. 3. They click on the "Confirm" button. 4. The system generates an unique ID which identifies the Third Party

Exit Condition	This use case terminates when the registration is completed and from now on the Third Party is able to login with the ID provided by the system
Exceptions	1. Third Party misses to fill some mandatory information or the data inserted is not valid. This exception is handled by the system who notifies the Third Party and disable the confirm button while the data are not complete or valid. 2. The Third Party is already registered to the system. This exception is handled by notifying the Third Party and taking him back to the homepage.
Special Requirements	

Name	Login
Actor	Third Party
Entry Condition	The third party clicks on "Login" button on Data4Help webpage
Flow of Events	1. The system shows the login form 2. The third party enters the requested credentials 3. The third party clicks on "Confirm" button
Exit Condition	The login is successful, and the system shows the third party's personal area.
Exceptions	1. The third party enters invalid data in the form. The system shows an error message and then returns to the form page.

Name	Request Individual Data
Actor	Third Party
Entry Condition	The Third Party clicks on “Search Individual” button on their Data4Help webpage
Flow of Events	1. Third Party inserts the single individual’s ID number or fiscal code on an input form 2. The system shows the searched User 3. Third Party clicks on “Request Data” 4. The system sends a notification to User
Exit Condition	The request is successful and it’s waiting for user’s response

Exceptions	1. There is no individual related to the input and no user is shown to the Third Party. The system shows again the input form to them. 2. The user has not granted access to their data, the system notifies the third party
Special Requirements	

Name	Request Group Data
Actor	Third Party
Entry Condition	The Third Party clicks on "Search Group" button on their Data4Help webpage
Flow of Events	1. Third Party inserts the search parameters (geographical area, gender, age and so on) 2. The system fetches all the data which satisfy the parameters 3. The system verifies that it is able to anonymize the requested data 4. The system sends group data to Third Party
Exit Condition	Group data is sent to Third Part
Exceptions	1. Group data can't be anonymized. The system reacts by notifying the Third Party with an error message.
Special Requirements	

Name	Subscribe to Individual Data
Actor	Third Party
Entry Condition	A data request has just completed successfully
Flow of Events	1. A dialog window asking if the Third Party wants to subscribe to new data for the same request is shown by the system 2. The Third Party clicks on the "Yes" button
Exit Condition	The dialog window is closed and the subscription is registered in the system
Exceptions	1. The Third Party clicks "No". In this case the system will not send new data to the Third Party about that specific individual

Special Requirements	

Name	Subscribe to Group Data
Actor	Third Party
Entry Condition	A group data request has just completed successfully
Flow of Events	1. A dialog asking if the Third Party wants to subscribe to new data for the same request is shown by the system 2. The Third Party specifies the time interval between each new data update 3. Third Party clicks “Confirm” button
Exit Condition	The dialog window is closed and the subscription is registered in the system
Exceptions	1. Third Party selects “No Update” option. In this case the system will not send new data to the Third Party about that group
Special Requirements	

Name	User Sign Up
Actor	User
Entry Condition	The user clicks on the “Sign up as user” button on Data4Help webpage
Flow of Events	1. The system shows the user registration form 2. The user fills all the mandatory fields and provides necessary information 3. The user clicks on the "Submit data" button
Exit Condition	The registration is successful and from now on the user can grant access to their data
Exceptions	1. The user misses to fill some mandatory information or the data inserted is not valid. This exception is handled by the system who notifies the User and disables the confirm button as long the data are not complete or valid

	2. The username provided by the user is not available. The system shows an error message to inform the user and asks them to insert a new one

Name	Agree to Data Acquisition
Actor	User
Entry Condition	The user is about to complete the registration process and has clicked the "Submit data" button
Flow of Events	1. The system shows the user a dialog asking to confirm their will to agree to personal and biometric data collection, along with a "Yes" and a "No" button 2. The user clicks "Yes" 3. The user is shown a confirmation page
Alternative flow	1. The system shows the user a dialog asking to confirm their will to agree to personal and biometric data collection, along with a "Yes" and a "No" button 2. The user clicks "No" 3. The user is shown a warning message stating that he has to accept data collection in order to complete the registration 4. The user dismisses the message 5. The user clicks "Yes" 6. The user is shown a confirmation page
Exit Condition	The user exits the confirmation page
Exceptions	1. The user closes the web page before clicking "Yes", the data is deleted and any new sign up for that user will start from scratch
Special Requirements	

Name	User Login
Actor	User
Entry Condition	User has landed on the login page
Flow of Events	1. The user types login credentials in the login form 2. The user clicks on the "Login" button
Exit Condition	The system recognizes the credentials inserted by the user

Exceptions	1. The credentials inserted by the user are invalid, the system notifies the user and asks to try again.
Special Requirements	

Name	Reply Data Request
Actor	User
Entry Condition	The user receives an email containing a request from a third party to access his data
Flow of Events	1. The user clicks on "Yes" or "No" links to respectively grant or deny access to his data
Exit Condition	The system notifies the involved third party about the user's choice
Exceptions	1. The user doesn't make any choice. The system does not provide any data (or denial notification) to the third party until a decision is made.

Name	Subscribe to AutomatedSOS
Actor	User
Entry Condition	User has clicked on the “Subscribe to AutomatedSOS” button, located in their personal area of the website
Flow of Events	1. The system shows the user a page explaining the AutomatedSOS terms of service. 2. The user clicks on the “Accept” button 3. The system updates the enabled services for this specific user
Exit Condition	The system has correctly added AutomatedSOS to the user’s services
Exceptions	1. The user clicks on “Cancel” or closes the web page before confirming the terms of service, the system does not update the enabled services for that user.
Special Requirements	

Name	Subscribe to Track4Run
Actor	User

Entry Condition	User has clicked on the “Subscribe to Track4Run” button, located in their personal area of the website
Flow of Events	1. The system shows the user a page explaining the Track4Run terms of service. 2. The user clicks on the “Accept” button 3. The system updates the enabled services for this specific user
Exit Condition	The system has correctly added Track4Run to the user’s services
Exceptions	1. The user clicks on “Cancel” or closes the web page before confirming the terms of service, the system does not update the enabled services for that user.
Special Requirements	

Name	View Runs
Actor	Athlete
Entry Condition	The Athlete has clicked on the “Available runs” button in his personal area
Flow of Events	1. The system provides an available run list to the user 2. The user clicks on the “Show details” button associated with a single run 3. The system shows a page containing the run details to the user
Exit Condition	The user can view the details page
Exceptions	1. The user does not select a run to detail, no run is shown
Special Requirements	

Name	Enroll to Run
Actor	Athlete
Entry Condition	The athlete clicks on the "Enroll" button in the run details page
Flow of Events	1. The system shows the enrollment form 2. The athlete provides any additional required data, apart from the information already collected from Track4Me 3. The athlete clicks on "Confirm" button
Exit Condition	The enrollment is successful

Exceptions	1. The athlete fails to fill in some mandatory information, or the data inserted is not valid. The system notifies the athlete and disables the confirm button as long the data are not complete or valid. 2. The athlete inserted invalid credentials. The system shows an error message and asks to re-input correct credentials.

Name	Manage Runs
Actor	Run Organizer
Entry Condition	Run Organizer has successfully logged in
Flow of Events	1. A page containing all the runs managed by the organizer is shown by the system 2. The organizer selects a run to manage through the "Select" button relative to it 3. The system shows an administration page for the desired run
Exit Condition	The page is successfully shown to the organizer
Exceptions	1. The organizer does not select any run to manage, no run is shown
Special Requirements	

Name	Schedule Run
Actor	Run Organizer
Entry Condition	The run organizer has clicked the "Add new run" button in the personal area
Flow of Events	1. The run scheduling form is presented 2. The run organizer fills in the required fields (such as start and end time and date of the run) 3. The run organizer clicks the "Define run path" button 4. The run organizer has correctly defined the run path
Exit Condition	The run data is stored successfully
Exceptions	1. The organizer inputs past dates or starting time for the run, an error message is shown and the organizer is asked to input correct data

	2. The organizer clicks the "Cancel" button or closes the page before confirming data, the run data is not saved to the system 3. The organizer does not define the run path, the run is not stored in the system
Special Requirements	

Name	Define Run Path
Actor	Run Organizer
Entry Condition	The run organizer has clicked the "Define run path" button while adding a new run
Flow of Events	1. The run path definition dialog is shown 2. The organizer inputs the run path 3. The organizer clicks the "Save path" button
Exit Condition	The path data is correctly stored in the system
Exceptions	1. The organizer clicks cancel or closes the dialog before having specified the run path, the schedule run page is shown again
Special Requirements	

Name	Check Athletes
Actor	Run Organizer
Entry Condition	The run organizer has clicked the "Show Athletes" button in the administration page of a selected run
Flow of Events	1. The system shows a list of the Athletes participating in the run 2. The Run Organizer select an Athlete by clicking on his name 3. The system shows the Athlete's detail to the Run Organizer
Exit Condition	The Run Organizer close the Athlete's list and return to the previous page
Exceptions	1. No one is participating to the run. The system shows a message and returns to the previous page.
Special Requirements	

Name	Spectate Run
Actor	Spectator
Entry Condition	Spectator lands on "Live runs" page
Flow of Events	1. The system shows a list of live runs or about to start 2. The spectator selects a run through the relative "Select" button 3. The system shows a page relative to the run containing the live map and the scoreboard.
Exit Condition	The spectator leaves the page
Exceptions	1. The spectator does not select a run to spectate, no live map page is shown
Special Requirements	

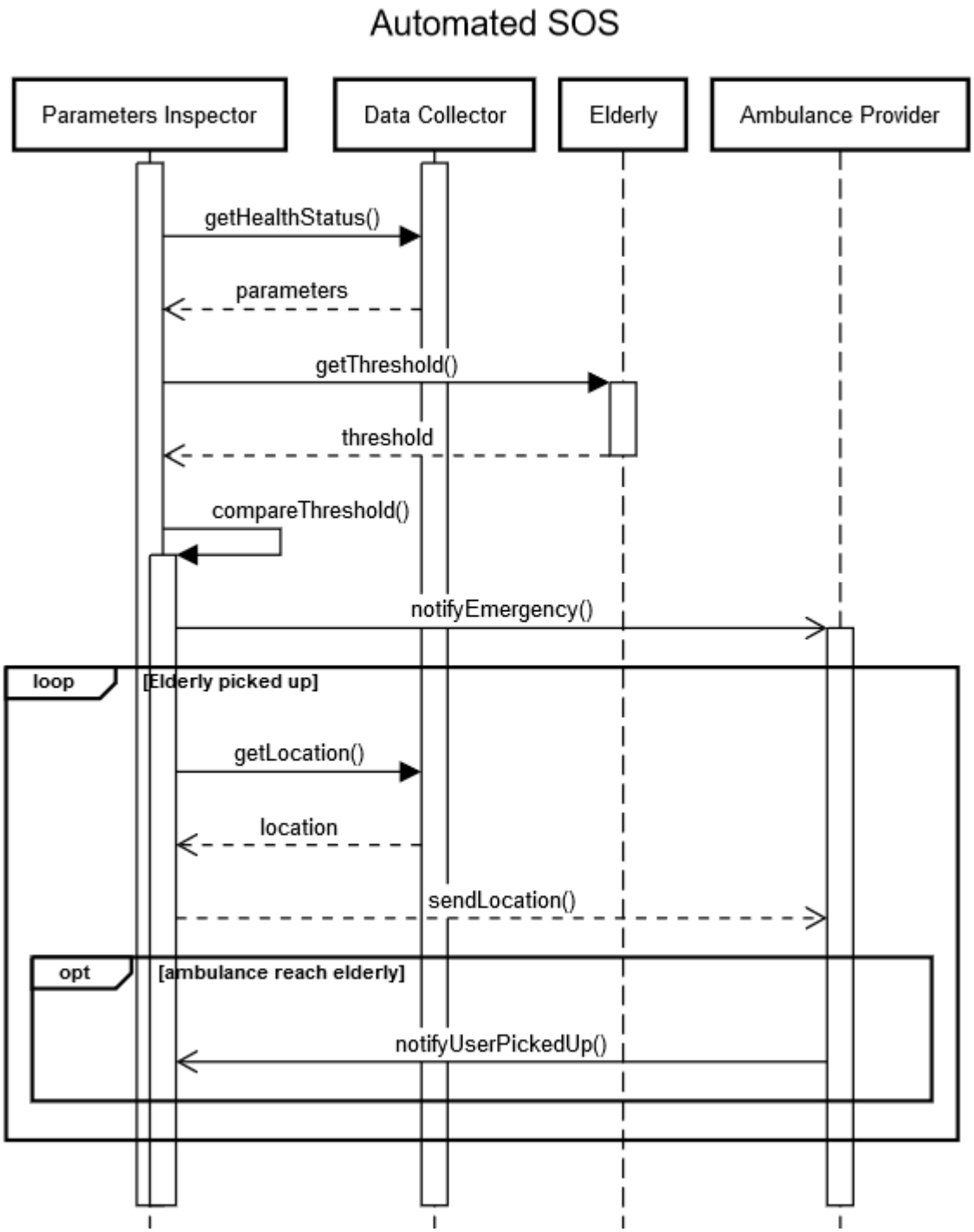
Name	Notify User Picked Up
Actor	Ambulance Provider
Entry Condition	The ambulance provider logs into the system
Flow of Events	1. The system shows the provider an active SOS requests list 2. The provider clicks the "Completed" button relative to the desired request 3. The system asks the provider to confirm its choice through a dialog 4. The provider clicks the "Yes" button 5. The system shows a confirmation message
Exit Condition	The system acknowledges the SOS request ending
Exceptions	1. The provider clicks the "No" button on the confirmation dialog or closes the page before confirming, the SOS request is not terminated and data keeps flowing to the provider
Special Requirements	

3.3.6 Sequence Diagrams

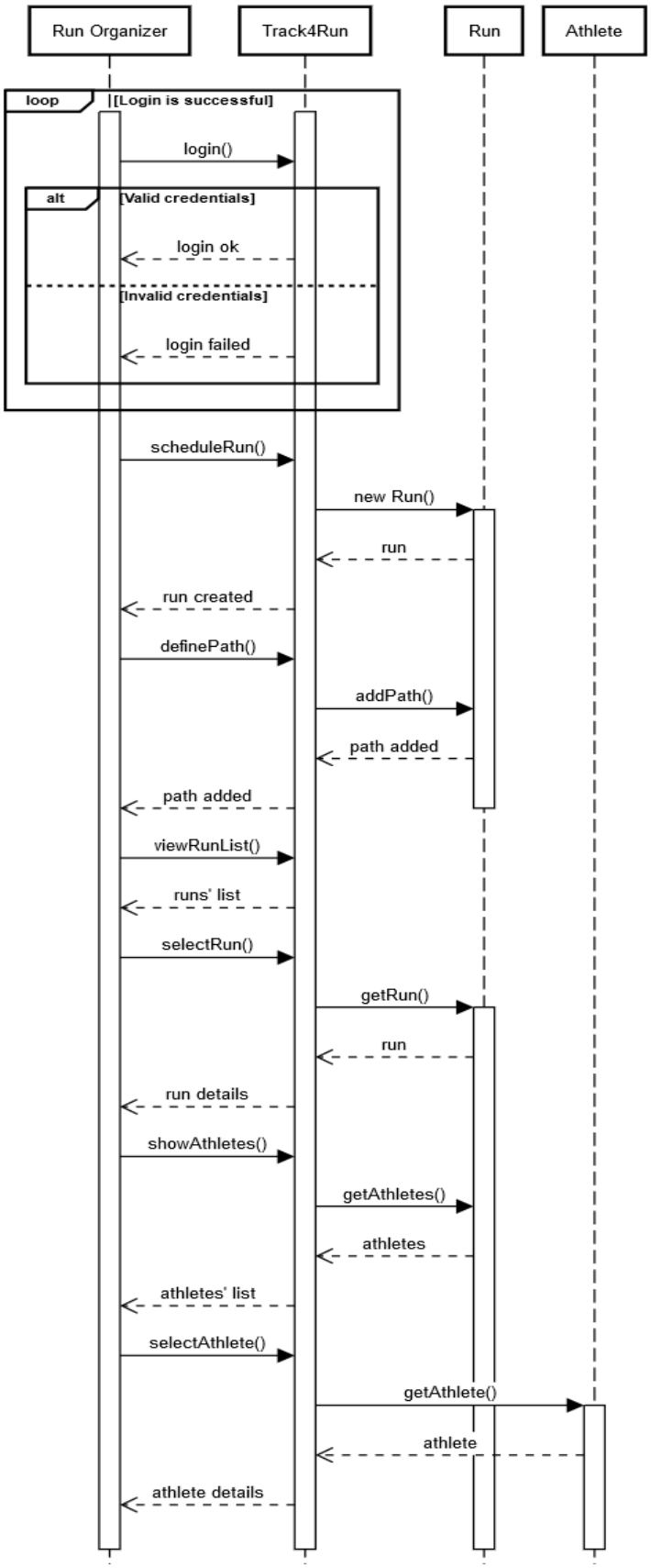
(In the RASD sequence diagram must focus on the interaction between user and the system. In the DD the sequence diagram can focus more on the architectural issues)

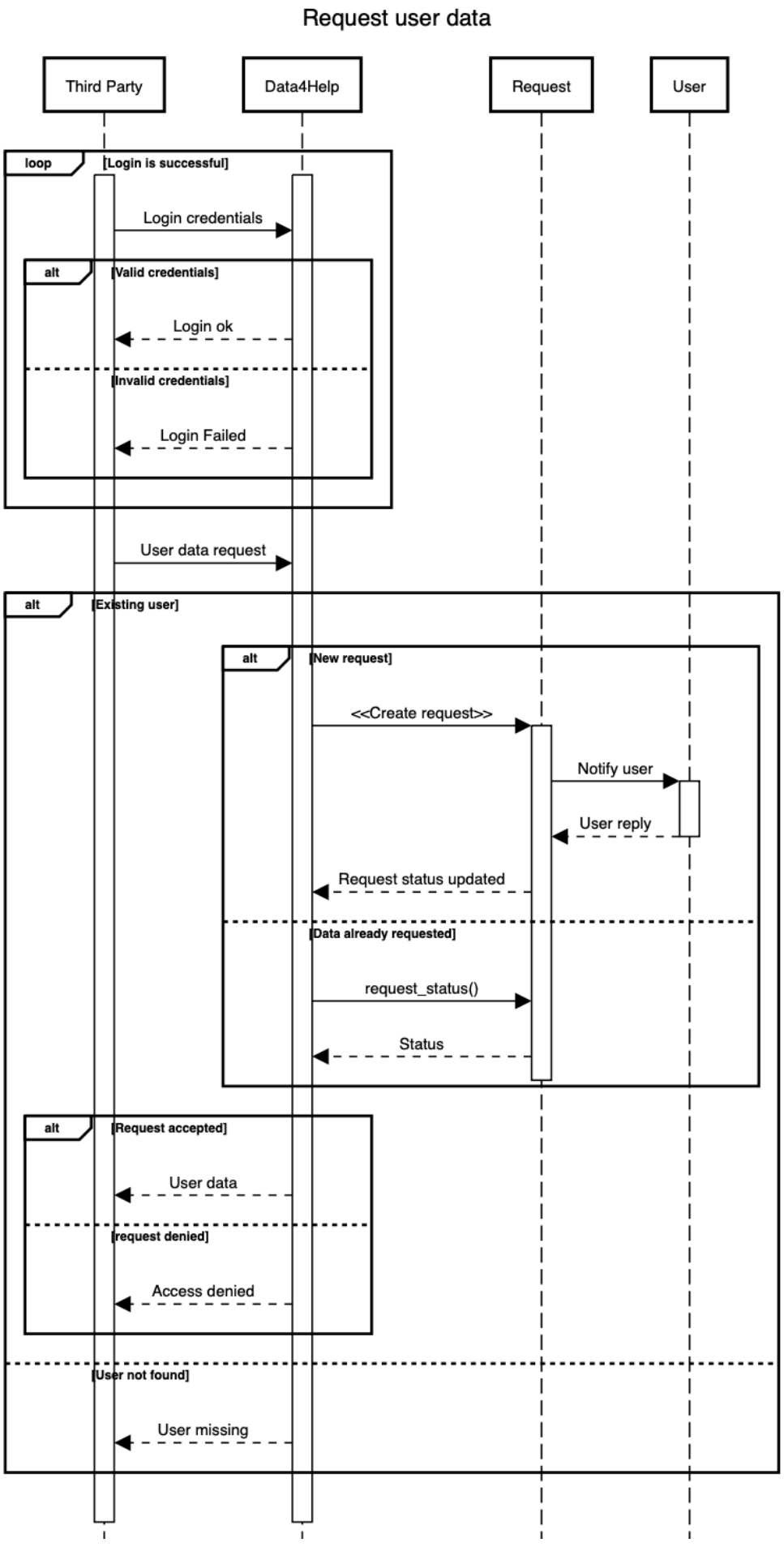
-individual data request
-group data request

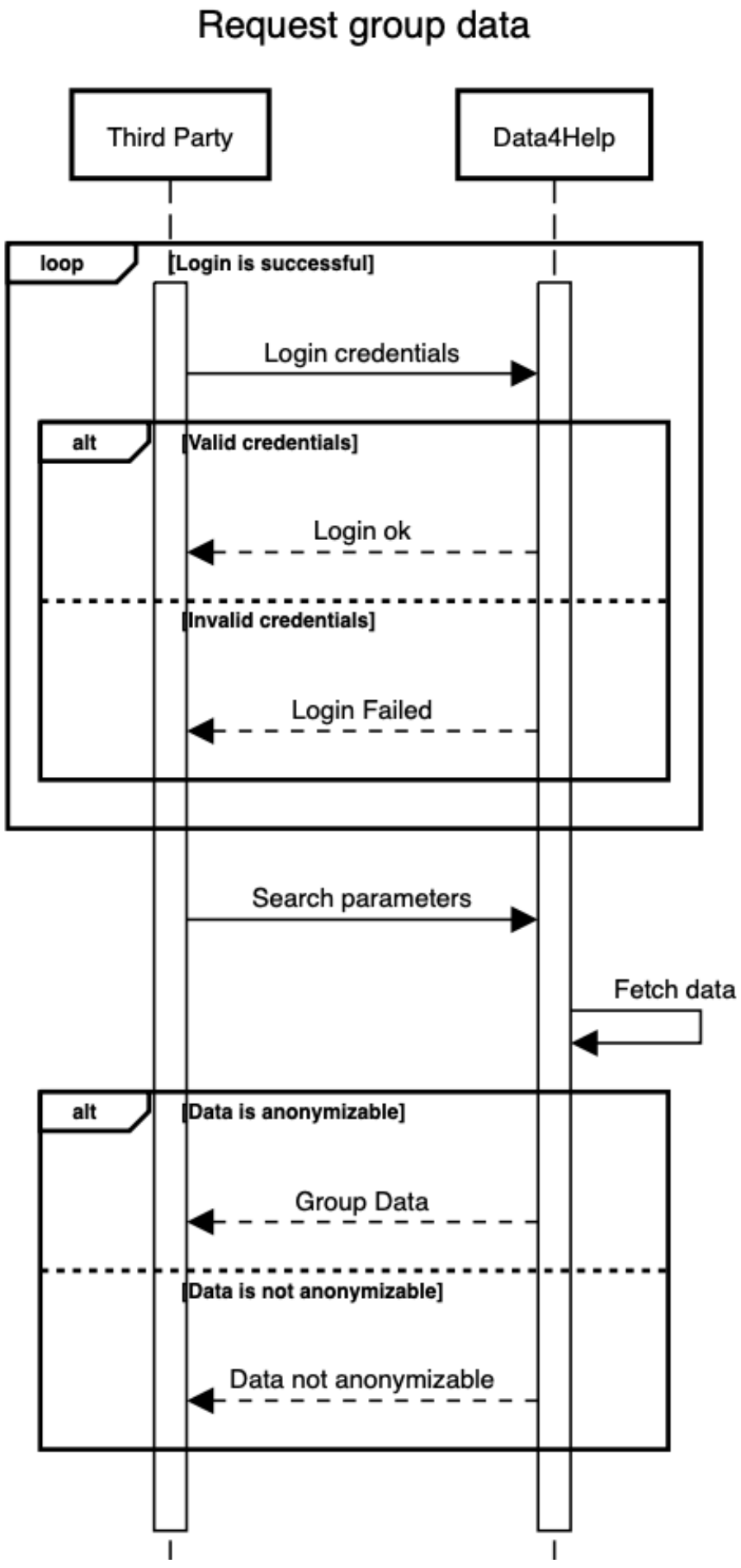
-health status below threshold
-manage runs
-user replies to request


Automated SOS
Parameters Inspector
Data Collector
Elderly
Ambulance Provider
getHealthStatus()
parameters
getThreshold()
threshold
compareThreshold()
notifyEmergency()
loop
[Elderly picked up]
getLocation()
location
sendLocation()
opt
[ambulance reach elderly]
notifyUserPickedUp()

Manage Run


Run Organizer
Track4Run
Run
Athlete
loop
[Login is successful]
login()
alt
[Valid credentials]
login ok
[Invalid credentials]
login failed
scheduleRun()
new Run()
run
run created
definePath()
addPath()
path added
path added
viewRunList()
runs' list
selectRun()
getRun()
run
run details
showAthletes()
getAthletes()
athletes
athletes' list
selectAthlete()
getAthlete()
athlete
athlete details


Request user data
Third Party
Data4Help
Request
User
loop
[Login is successful]
Login credentials
alt
[Valid credentials]
Login ok
[Invalid credentials]
Login Failed
User data request
alt
[Existing user]
alt
[New request]
<<Create request>>
Notify user
User reply
Request status updated
[Data already requested]
request_status()
Status
alt
[Request accepted]
User data
[request denied]
Access denied
[User not found]
User missing


Request group data
Third Party
Data4Help
loop
[Login is successful]
Login credentials
alt
[Valid credentials]
Login ok
[Invalid credentials]
Login Failed
Search parameters
Fetch data
alt
[Data is anonymizable]
Group Data
[Data is not anonymizable]
Data not anonymizable

3.4 Performance requirements

G5: Allow third parties to offer a personalized and non-intrusive SOS service to elderly people so that an ambulance arrives to the location of the customer in case of emergency.
[R1-NF]: A reaction time of less than 5 seconds from the time the health parameters are below the threshold shall be guaranteed by the system.

3.5 Design constraints

3.5.1 Standards Compliance

3.5.2 Hardware limitations

3.5.3 Any other constraint

3.6 Software System Attributes

3.6.1 Reliability

3.6.2 Availability

3.6.3 Security

3.6.4 Maintainability

3.6.5 Portability

4 Formal Analysis using Alloy

(*In this section you will include your Alloy model. We require you to comment on the model by discussing the purpose of the model, what you can prove with it and why what you prove is important given the problem at hand. You are also required to show one or more worlds obtained by running your model)*

4.1 What we want to model?

In this section we want to test the correctness of some of the core components and features of the system to be.
In particular, we are interested in modeling the behavior of:

- A third party requests a specific user's data, and the user already granted the permission
- A third party requests a specific user's data, and the user rejects the permission
- A third party requests data about a group of users, and since the data is anonymizable the system responds with the anonymized data
- A third party requests data about a group of users, but the data is not anonymizable and the system doesn't send the requested data

- An elderly user's health status is beyond risk threshold, and AutomatedSOS dispatch an ambulance
- An athlete tries to enroll in a past run, without success
- An organizer schedules a run before today

(*Why are we modeling these things?)*

- Constraint more than 1000 users (Anonymization)
- (check unicity of usernames)
- Class Data: abstract (with ID and timestamp)
  Health and Location: extends abstract Data
- sig RequestHandler {
  request: thirdParty -> Data
  }
- Case: you can't enroll to a run already begun

5 Effort Spent

5.1 Piccinotti Diego

Description of the task	Hours
Purpose, Scope, Definition	5
Product Perspective	7
Product Functions	3.5
User Characteristics	1
Domain Assumptions	3
Functional Requirements	9
Non-functional Requirements	
Formal Analysis Using Alloy	0.5

5.2 Pietroni Umberto

Description of the task	Hours
Purpose, Scope, Definition	5
Product Perspective	8
Product Functions	2
User Characteristics	1
Domain Assumptions	3
Functional Requirements	11.5
Non-functional Requirements	
Formal Analysis Using Alloy	0.5

5.3 Rossi Loris

Description of the task	Hours
Purpose, Scope, Definition	6.5
Product Perspective	5
Product Functions	2
User Characteristics	2
Domain Assumptions	3.5
Functional Requirements	5
Non-functional Requirements	
Formal Analysis Using Alloy	3

6 References

WORD SHARED DOCUMENT