



POLITECNICO DI MILANO

MASTER'S DEGREE IN
COMPUTER SCIENCE AND ENGINEERING

SOFTWARE ENGINEERING 2

TrackMe Design Document

Authors

Alberto ARCHETTI
Fabio CARMINATI

Reference professor

Elisabetta DI NITTO

v.0 - November 24, 2018

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Scope	2
1.2.1	World	2
1.2.2	Shared phenomena	2
1.3	Definitions	3
1.4	Acronyms and abbreviations	3
1.5	Revision history	4
1.6	Document structure	4
2	Architectural Design	5
2.1	Overview	5
2.2	Component view	5
2.3	Deployment view	5
2.4	Runtime view	5
2.5	Component interfaces	5
2.6	Selected architectural styles and patterns	5
2.7	Other design decisions	5
3	User Interface Design	6
4	Requirements Traceability	8
5	Implementation, Integration and Test Plan	9
6	Effort spent	10

1 Introduction

1.1 Purpose

TrackMe wants to develop a software-based service that allows individual users to collect health data, called **Data4Help**. This data can be retrieved from the system and visualized according to different filters by a user interface.

The system allows third parties registration. Third parties can request access to users' collected data in two ways:

Single user data After a third party makes a request to the system for a single user data sharing, by providing user's fiscal code, the system asks the user for authorization; if positively provided, the third party is granted access to the user's data

Anonymous group data Third parties can be interested in big amounts of data, but not in who are the people providing it; the system, once the request is sent by the third party, checks if the data can be effectively anonymized (it must find at least 1000 people that can provide data matching the third party request's filters) and, if positively evaluated, grants access to the anonymized data to the third party

Third parties can subscribe to new data and receive it as soon as it is collected by the system.

Another service that TrackMe wants to develop is **AutomatedSOS**, built on **Data4Help**. This service analyzes users' data and calls a SOS whenever data exceeds the basic health parameters. For this particular purpose, system performance will be a critical aspect to be taken into account, because even the slightest delay matters in critical health situations.

We will list the project **goals**, described in the RASD document:

- G.U1 Users can collect, store and manage their health data
- G.U2 Users can choose to have their health monitored; if their health is critical, an ambulance will be dispatched
- G.T1 Third parties can ask single users for their health data sharing
- G.T2 Third parties can request access to anonymized data that comes from groups of people
- G.T3 Third parties can subscribe to new data and receive it as soon as it is produced

1.2 Scope

1.2.1 World

Our *world* is composed of two main types of actors: *users* and *third parties*. Users are interested in monitoring their health parameters and third parties are interested in developing services or researches that exploit data gathered from the users. **Data4Help** is the service that acts as a bridge between these actors' needs.

Phenomena that occur in the *world* and are related to our application domain are

- physical conditions of the users
- third parties' projects, researches and interests
- ambulances dispatched by the SOS system

These phenomena exist in the *world*, but cannot be observed directly by our system.

1.2.2 Shared phenomena

In order to communicate with the *world*, our system needs to share some aspects with it. We will list the aspects controlled by the world, but observable by the machine:

S.1 physical parameters of the users, gathered through sensors on wearable devices

S.2 third parties requests to the system for the data they need

S.3 users' location, acquired through GPS signals

On the other hand, the aspects that occur in the machine, but are observable by the world are

S.4 interfaces that organize the gathered data that can be filtered according to time or type of data

S.5 messages for the SOS system, that are sent in case of critical health of a user

S.6 payment requests

1.3 Definitions

Data Quantitative variables concerning health parameters

Aggregate data See *DataSet*

Anonymous data *data entry* that doesn't contain information about the user from which it was produced; a *data set* is said to be anonymized if it contains only anonymous *data entries* and its cardinality is greater or equal than 1000

Data entry Tuple that corresponds to the user's parameters in a particular moment

Data set Set of *data entries*; depending on the context, it can identify a set of entries all belonging to a single user or or a set of anonymous entries belonging to more that 1000 users; a *data set*, among all *data* that the system is storing, can be identified and constructed according to the filters of a third party request

Request Third parties can ask the system for some data sharing through requests; requests are encoded through filling a form; the system, provided that the request is satisfiable, grants the third party access to the requested data

Third party Actor interested in collecting data from a single user or from an anonymous group of users

Threshold Numerical values related to a particular health parameter; they act as boundaries between the domain of critical health status and normal health status

User Actor interested in his/her health data collecting and managing; a user can also be interested in automating SOS calls whenever his health status becomes critical

Some of these definitions may already be present and further explained in the RASD document.

1.4 Acronyms and abbreviations

API Application Programming Interface

Data Whenever the context refers to generic groups of *data entries*, the terms *data* and *data set* are interchangeable

System Software product that TrackMe wants to develop; can be interchanged with *S2B*

S2B Software To Be

1.5 Revision history

Version	Log
v.0	DD first draft

1.6 Document structure

This document describes architecture and design of **Data4Help** and **AutomatedSOS** systems. The description will start with a top-down approach, in order to make the reader familiar with the overall structure; a bottom-up approach will then be adopted, in order to describe components in a isolated way. This document is divided in

- Section 1: brief introduction on the project to be developed in order to make this document self-contained
- Section 2: describes the high-level architecture (high-level components, their interaction, runtime views and architextural decisions)
- Section 3: provide an overview on how the user interfaces will look like
- Section 4: mapping between software requirements, described in the RASD document, and design elements
- Section 5: identifies the order in which subcomponents will be implemented, integrated and tested
- Section 6: list of the work sessions that drove this document's development, ordered by date, as the hour counter of effort spent by each group member

2 Architectural Design

2.1 Overview

2.2 Component view

2.3 Deployment view

2.4 Runtime view

2.5 Component interfaces

2.6 Selected architectural styles and patterns

2.7 Other design decisions

3 User Interface Design

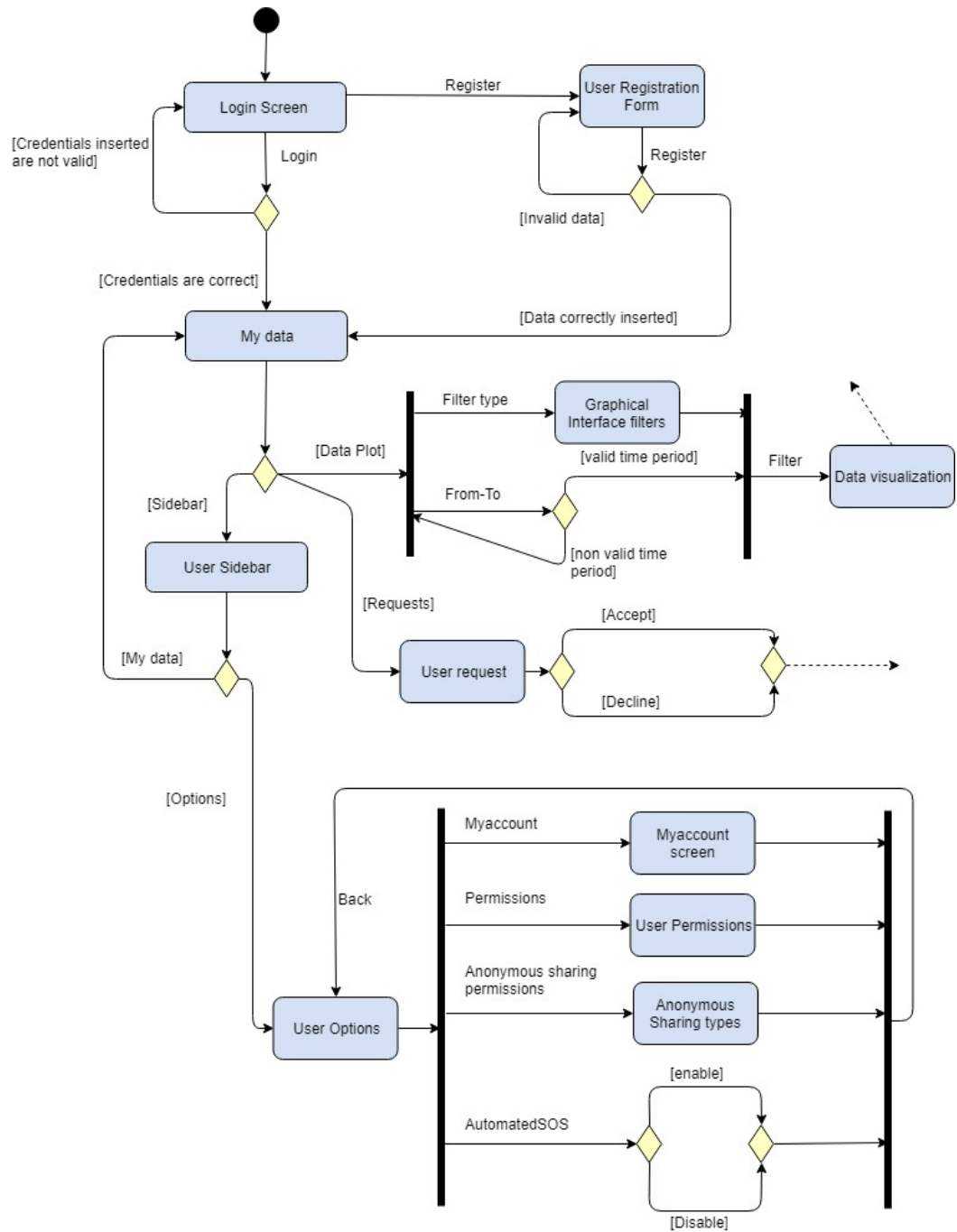


Figure 1: User account

4 Requirements Traceability

5 Implementation, Integration and Test Plan

6 Effort spent

Date	Archetti Alberto	Carminati Fabio	Activity
12/11/2018	1	1	Introduction sketch

References

- [1] Mandatory Project Assignment AY 2018-2019
- [2] ISO/IEC/IEEE 29148 - Systems and software engineering - Life cycle processes - Requirements engineering
- [3] Collection and Processing of Data from Wrist Wearable Devices in Heterogeneous and Multiple-User Scenarios
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5038811/>
- [4] Wearable Devices in Medical Internet of Things: Scientific Research and Commercially Available Devices
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5334130/>
- [5] Google Fit API
<https://developers.google.com/fit/overview>
- [6] PayPal API
<https://developer.paypal.com/docs/>
- [7] RapidSOS Emergency API
<https://info.rapidsos.com/blog/product-spotlight-rapidsos-emergency-api>
- [8] Slides of the course by Prof. Di Nitto
<https://beep.metid.polimi.it/>
- [9] L^AT_EX templates
<http://www.latextemplates.com/>
- [10] Draw.io
<https://www.draw.io/>