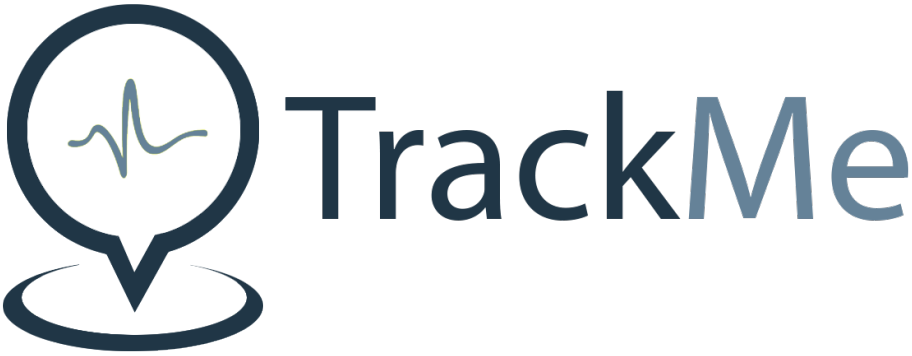
POLITECNICO
MILANO 1863

M.Sc. Computer Science and Engineering
Software Engineering 2 Project



Design Document

Gargano Jacopo Pio, Giannetti Cristian, Haag Federico

10 December 2018

GitHub Repository: https://github.com/federicohaag/GarganoGiannettiHaag

Version 1.0

Contents

Chapter 1

Introduction

1.1 Purpose

TrackMe wants to offer a service named "Data4Help" on top of which two services, named "AutomatedSOS" and "Track4Run", will be built.

Data4Help will be a system collecting data of *Users* through a *Smart wearable* connected to their smartphone. This data may be sent to *Third parties* after *User* consent. AutomatedSOS will constantly monitor *User data* to allow for immediate assistance. Track4Run will allow individuals to organize running competitions, to enroll in or watch one.

More details may be found in section 1.1 of [3].

In this document, the design of the system to be will be explained and analyzed in detail. This includes the identification of all the components and the design decisions regarding the structure and the patterns to be implemented. Moreover, implementation and testing will be discussed and planned.

1.2 Scope

TrackMe proposes to offer its system in a world in continuous technological evolution, where almost everyone owns a smartphone and is always connected to the Internet. By using Data4Help, *Users* will integrate their everyday activities with constant collection of their health data and position, through sensors of their *Smart wearable* and smartphone GPS. By adding *Third party Services* they may benefit of data monitoring and analysis with useful insights on their daily activities.

Users may enhance data collection by adding AutomatedSOS to their *Services*. This *Service*, by constantly monitoring their data, will allow for *Anomalous data*identification and immediate assistance for the *User in need*. This can be crucial for individuals with health issues and elders living alone. People fond of running have the possibility of enrolling in a *Run* listed in Track4Run. Their data, including position and health data, will be shown to *Spectators*, who can

watch the *Run*. The process of organizing running competitions will be carried out by *Organizers* through Track4Run.

More details may be found in section 1.2 of [3], including a detailed analysis of shared phenomena.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

User: registered individual of Data4Help who agreed on the acquisition and processing of their data (see *User data*).

User data: *User*'s health data and location acquired by Data4Help.

Third party: a company that is willing to access *User data* stored in TrackMe's database.

Service: application available for Data4Help *Users*, generally offered by a *Third party*.

Smart wearable: smart devices that can be worn on the body as accessories. These devices are required to have specific sensors for data acquisition. They must be connected to an external device, such as a smartphone.

Anomalous data: health data that is outside certain intervals identifying a *User* normal health condition.

Run: running competition registered on Track4Run.

1.3.2 Acronyms

GPS: Global Positioning Service

1.3.3 Abbreviations

G_n: n^{th} goal

R_n: n^{th} requirement

1.4 Revision History

1. Version 1.0 - 10^{th} December 2018

1.5 Reference Documents

- WORK IN PROGRESS

1.6 Document Structure

This document is divided into seven chapters.

First chapter In this chapter is summarized the project. This has been already introduced and described in RASD so here it is shown briefly only in terms of purpose and scope. Moreover, the chapter contains the definitions, acronyms and abbreviations needed to properly understand the sections following. All the documents used during the development of this project are listed at the end of the chapter.

Second chapter In this chapter is described a proposal of architecture able to deliver an implementation of the system to be described in the RASD. The architecture takes into consideration all the goals and requirements listed in the RASD. A particular attention is dedicated to the satisfaction of non functional requirements (e.g. reliability, performances). The architecture is described following a top-down approach: from overview to detailed components descriptions and interaction flows.

Third chapter In this chapter are listed and showed all the user interfaces of the system to be. The description is given with a particular attention for details and interaction between different interfaces.

Fourth chapter In this chapter requirements defined in the RASD are mapped to the design elements defined in in Chapter 2.

Fifth chapter In this chapter is presented the list of activities needed to implement the system to be, according to the architecture showed in Chapter 2 and according the user interfaces showed in Chapter 3. Activities are analyzed to identify eventual constraints that force certain activities to be executed before or after other else. A rough estimation and estimation is given.

Sixth chapter Effort spent by all team members is shown as the list of all activities done during the realization of this document.

Seventh chapter References of documents that this project was developed upon.

Chapter 2

Architectural Design

2.1 Overview

WORK IN PROGRESS

2.2 Component View

WORK IN PROGRESS

2.3 Deployment View

WORK IN PROGRESS

2.4 Runtime View

WORK IN PROGRESS

2.5 Component Interfaces

WORK IN PROGRESS

2.6 Selected Architectural Styles

WORK IN PROGRESS

2.7 Other Design Decisions

WORK IN PROGRESS

Chapter 3

User Interface Design

WORK IN PROGRESS

Chapter 4

Requirements Traceability

Description fare riferimenti a component diagrams

Data4Help

R_1 Unregistered individuals and companies must not be able to use Data4Help.

1. Authentication (User App and Data4Help)

R_2 At sign up, *User* must provide: first name, last name, SSN, gender, date of birth, email and password.

1. Authentication (User App and Data4Help)

R_3 At sign up, *Third party* must provide a company name.

1. Authentication (Data4Help)

R_4 At sign up, *User* must accept *Terms and conditions*, including the *Privacy statement*.

1. Authentication (User App and Data4Help)

R_5 At sign up, *Third party* must accept *Terms and conditions*.

1. Authentication (Data4Help)

R_6 Identify a *User* by their identifier.

1. Authentication (User App and Data4Help)

R_7 Query the database for a *User* by their identifier.

1. Authentication (User App and Data4Help)

R_8 Receive *User Data*.

1. Authentication (User App and Data4Help)

2. Data Collector
3. Data Sender
4. Data Receiver

R_9 Validate *User Data*.

1. Data Receiver

R_{10} Authenticate *User Data*.

1. Data Receiver

R_{11} Store collected *User Data* in a database.

1. Data Receiver
2. Data Manager

R_{12} Retrieve specific *User Data* by database querying based on *User* identification.

1. Data Manager

R_{13} Receive *Third party* data access request.

1. Request Manager

R_{14} Validate *Third party* data access request.

1. Request Manager

R_{15} Authenticate *Third party* data access request.

1. Request Manager

R_{16} Forward *User Data* access request to the specific *User*.

1. Request Manager
2. Authorization
3. Data Manager
4. Services Manager (Data4Help)

R_{17} Receive *User* consent approval or denial.

1. Services Manager (User App and Data4Help)

R_{18} Check if a specific *User* gave consent to a specific *Service*.

1. Authorization
2. Data Manager

R_{19} Send specific *User Data* to the requesting *Third party*.

1. Request Manager
2. Authorization
3. Data Manager

R_{20} Not send specific *User Data* to the requesting *Third party* if the specific *User* denied consent.

1. Request Manager
2. Authorization
3. Data Manager

R_{21} *Third party* must be able to set specific constraints to define a group of *Users*: age, gender, residence.

1. Request Manager

R_{22} Check how many *Users* requested *Group Data* refers to.

1. Anonymizer

R_{23} Properly anonymize *Group Data*.

1. Anonymizer

R_{24} Send *Group Data* to the requesting *Third party*.

1. Anonymizer
2. Request Manager

R_{25} Not send *Group Data* if the group it refers to is made up of less than 1000 *Users*.

1. Anonymizer
2. Request Manager

R_{26} Receive *Third party* subscription request.

1. Request Manager

R_{27} Validate *Third party* subscription request.

1. Request Manager

R_{28} Authenticate *Third party* subscription request.

1. Request Manager

R_{29} Automatically send new data to subscribed authorized *Third parties* as soon as they are produced.

1. Data Receiver

2. Data Manager
3. Services Manager (Data4Help)
4. Authorization

R_{30} Allow *Users* to subscribe to *Services*.

1. Services Manager (Data4Help)
2. Data Manager

R_{31} Allow *Users* to unsubscribe from *Services*.

1. Services Manager (Data4Help)
2. Data Manager

R_{32} Send a specific *User* all their data stored, if requested by them.

1. Services Manager (Data4Help)
2. Data Manager

R_{33} Delete a *User* specific data, if requested by them.

1. Services Manager (Data4Help)
2. Data Manager

R_{34} Allow *Users* to request all their data stored by TrackMe at any time.

1. Services Manager (User App and Data4Help)

R_{35} Allow *Users* to request the deletion of all their data stored by TrackMe at any time.

1. Services Manager (User App and Data4Help)

AutomatedSOS

R_{36} Receive *User Data* from Data4Help.

1. Data Receiver (AutomatedSOS)

R_{37} Compare *User Data* against certain thresholds.

1. Data Analyzer

R_{38} Call local emergency services providing necessary *User Data* of *User in need*.

1. Local Emergency Services Caller

R_{39} *User* must be able to reactivate AutomatedSOS monitoring.

1. Services Manager (User App and Data4Help)
2. User Monitor spiegare interface

Track4Run

R_{40} Receive *User Data* from Data4Help.

1. Data Receiver (Track4Run)

R_{41} At sign up, *Organizers* must provide: first name, last name, email and password.

1. Authentication

R_{42} At sign up, *Organizers* must accept *Terms and conditions*.

1. Authentication

R_{43} Allow *Organizers* to create a *Run*, defining: name, path, date, maximum number of *Participants* and enrollment closure date.

1. Run Manager

R_{44} Provide *Users* enrollment for an existing *Run*.

1. Run Manager

R_{45} Prevent a *User* from enrolling in a *Run* if the maximum number of *Participants* was already reached.

1. Run Manager

R_{46} Prevent a *User* from enrolling in a *Run* if it already started or finished.

1. Run Manager

R_{47} Prevent a *User* from enrolling in a *Run* if enrollment is closed.

1. Run Manager

R_{48} Show a *Run* by displaying the position of *Participants* on a map.

1. Run Manager

R_{49} Identify a *Run* by its identifier.

1. Run Manager

R_{50} Query the database for a *Run* given its identifier.

1. Run Manager

Chapter 5

Implementation, Integration and Test Plan

WORK IN PROGRESS

Chapter 6

Effort Spent

Gargano Jacopo Pio Total hours of work:

- effort spent

Giannetti Cristian Total hours of work:

- effort spent

Haag Federico Total hours of work:

- effort spent

Chapter 7

References

1 E. Di Nitto. *Lecture Slides*. Politecnico di Milano.

2 E. Di Nitto. *Mandatory Project Assignment AY 2018-2019*. Politecnico di Milano.

3 J. Gargano, C. Giannetti, F. Haag. *Requirement Analysis and Specification Document*. Politecnico di Milano.

- references