Software Engineering 2

TrackMe

Implementation and **T**esting **D**ocument

POLITECNICO
MILANO 1863

Romeo Paolo
Scotti Andrea
Staccone Francesco

January 11, 2019

Contents

1 Introduction

1.1 Purpose and Scope

The following document represents the Implementation and Testing Document for the TrackMe project. The purpose of this document is to provide a comprehensive overview of the implementation and testing activity for the development of the software application. The TrackMe application aims at be a Health Data Management and Run-Friendly Mobile App, useful for a quite wide range of users with different features and necessities.

1.2 Definitions, Acronyms, Abbreviations

1.2.1 Definitions

- **Framework**: is an abstraction in which software providing generic functionality can be selectively changed by additional user-written code.
- **Third party**: a company, association or, more in general, a public or private entity that uses TrackMe to acquire data of users or to organize running events;
- **User**:a person who uses TrackMe;
- **Individual**:equivalent to User;

1.2.2 Acronyms

DBMS	Data Base Management System
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
API	Application Program Interface
REST	REpresentational State Transfer
MVC	Model View Controller
JSON	JavaScript Object Notation
VAT	Value Added Tax

1.3 Reference Documents

- Design document
- RASD document
- Project assignment

1.4 Overview

The rest of the document is organized in this way:

- **Implemented requirements:** explains which functional requirements outlined in the RASD are accomplished, and how they are performed.
- **Adopted frameworks:** provides reasons about the implementation decisions taken in order to develop the application.
- **Source code structure:** explains and motivates how the source code is structured both in the front end and in the back end.
- **Testing:** provides the main testing cases applied to the the application

2 Implemented requirements

In this section we describe the implemented functionalities with reference to the requirements outlined in the RASD document.
The requirements with their reference number green are the ones fully available in the current implementation. Requirements in red are not fully implemented but the base for a further complete development is present.

2.1 Third Party Registration

- **R16)** The system must allow the third party to register to the application, by specifying its VAT registration number, name and password.

Database

The database stores the User information and credentials inside the table denominated 'users'.
Passwords are stored as salted hashes for security purposes.

Back-end

Front-end

The homepage of the client application for a Guest contains a registration section showing a form to fill with registration information. The client application validates the information, performs the request to the server and shows the response.

2.2 Individual Registration

- **R13)** The system must allow the individual to register to the application by selecting a password and providing his/her data, fiscal code included;

Database

The database stores the User information and credentials inside the table denominated 'users'.
Passwords are stored as salted hashes for security purposes.

Back-end

Front-end

The homepage of the client application for a Guest contains a registration section showing a form to fill with registration information. The client application validates the information, performs the request to the server and shows the response.

2.3 Third Party Login

- **R17)** The system must allow the third party to log in to the application by providing the combination of a VAT registration number and a password that match an account.

Database

Back-end

Front-end

The homepage of the application contains (if the User is not logged in) a login form. The client application validates the information, performs the request to the server and shows the response.

2.4 Individual Login

- **R14)** The system must allow the individual to log in to the application by providing the combination of a fiscal code and a password that matches an account.

Database

Back-end

Front-end

The homepage of the application contains (if the User is not logged in) a login form. The client application validates the information, performs the request to the server and shows the response.

2.5 Third Party Settings Management

- **R33)** The system must allow the Third party to change its password.

Database

Back-end

Front-end

2.6 Individual Settings Management

- **R31)** The system must allow the user to change his/her personal info.
- **R32)** The system must allow the individual to change his/her password.
- **R34)** The system must allow the user to connect a smartwatch or a similar device to its smartphone.

Database

Back-end

Front-end

2.7 Data Management

- **R11)** The system is optimized to send the data received from the mobile application to the third parties as soon as possible.
- **R9)** The third party is not allowed to access the users data until he/she accepts the request.
- **R27)** The system must be able to store data retrieved from registered users.
- **R3)** The system must be able to retrieve data from the smartwatches and similar devices;
- **R2)** The system must be able to provide to the third party the location and the health status of individuals;
- **R1)** The users must have given the consensus to the treatment of their information to the third party;

Database

Back-end

Front-end

2.8 AutomatedSOS service

- **R25)** The AutomatedSOS service must be enabled.
- **R18)** When the health status values go below the threshold, the system must send an SOS within 5 seconds;
- **R26)** The system must allow the user to enable/disable the AutomatedSOS service at any time.

Front-end

2.9 Individual Requests

- **R28)** The user must have an active subscription to stop it;
- **R29)** The system must be able to allow the user to unsubscribe to the third party and to stop the transmission of his/her data.
- **R7)** The system must be able to forward the requests from the third party to the user;
- **R8)** The system must save the preference of the user;

Database

Back-end

Front-end

2.10 Anonymous Requests

- **R5)** The system must be able to provide to the third party the health status of individuals in an anonymous way;
- **R4)** The groups must be composed at least by 1000 individuals;
- **R6)** The system must be able to aggregate the data of the individuals, as requested by the third party;

Database

Back-end

Front-end

3 Adopted Frameworks

3.1 Spring Boot

Spring is a framework which provides comprehensive infrastructure support for developing Java applications and Spring Boot is basically an extension of Spring which eliminated the boilerplate configurations required for setting up a Spring application. In other words, while the Spring framework focuses on providing flexibility to you, Spring Boot aims to shorten the code length and provide you with the easiest way to develop a web application. With annotation configuration and default codes, Spring Boot shortens the time involved in developing an application. It helps create a stand-alone application with less or almost zero-configuration.

Why Spring Boot?

- To ease the Java-based applications Development, Unit Test and Integration Test Process.
- To reduce Development, Unit Test and Integration Test time by providing some defaults.
- To increase Productivity.

Why Spring?

- Spring is a light weight framework and It minimally invasive development with POJO.
- Spring achieves the loose coupling through dependency injection and interface based programming.
- Spring supports declarative programming through aspects and common conventions.

Inversion of Control

Inversion of Control is a principle in software engineering by which the control of objects or portions of a program is transferred to a container or framework. It's most often used in the context of object-oriented programming.

By contrast with traditional programming, in which our custom code makes calls to a library, IoC enables a framework to take control of the flow of a program and make calls to our custom code. To enable this, frameworks use abstractions with additional behavior built in. If we want to add our

own behavior, we need to extend the classes of the framework or plugin our own classes.

The advantages of this architecture are:

- decoupling the execution of a task from its implementation
- making it easier to switch between different implementations
- greater modularity of a program
- greater ease in testing a program by isolating a component or mocking its dependencies and allowing components to communicate through contracts

Inversion of Control can be achieved through various mechanisms such as: Strategy design pattern, Service Locator pattern, Factory pattern, and Dependency Injection (DI).

3.2 AngularJS

Designed by Google, AngularJS is an open-source framework that addresses the challenges of web development processes, it is a framework for internet applications, it is one of the most powerful front-end frameworks. Angular framework allows the use of HTML as the template language and allows the extension of HTML's syntax to express application's components in a brief and clear manner. It can automatically synchronize with models and views. Angular could literally be described as a hybrid HTML editable version that creates Apps.

Why AngularJS?

- Promotes Code Reusability: with Angular, developers can reuse the codes that were previously used in other different application, thus promoting code reusability and making Angular a unique and outstanding framework.
- Faster application development: Angular makes everything from development to testing and maintenance quite fast and quick. The MVC (model view controller) architecture assures this.
- Gives Developers full Control: developers can test, construct, inject and do almost anything with this framework, this is due to the fact that directives offer a free hand to experiment with HTML and attributes.

- Data Binding: is easy with Angular, data binding happens with ease.
- Time-Saving: all that Angular requires you to do is split your web app into multiple MVC(model view controller) components. Once that is done, Angular automatically takes over and performs the rest of the functions. It saves you from the trouble of writing another code.

3.3 Ionic

Ionic is a complete open-source SDK for hybrid mobile app originally built on top of AngularJS and Apache Cordova. Ionic provides tools and services for developing hybrid mobile apps using Web technologies like CSS, HTML5, and Sass. Apps can be built with these Web technologies and then distributed through native app stores to be installed on devices by leveraging Cordova. Ionic Framework is designed to work and display beautifully out-of-the-box across all platforms. It's a library of UI Components, which are reusable elements that serve as the building blocks for an application. Ionic Components are built with web standards using HTML, CSS, and JavaScript. Though the components are pre-built, they're designed from the ground up to be highly customizable so apps can make each component their own, allowing each app to have its own look and feel. More specifically, Ionic components can be easily themed to globally change appearance across an entire app.

Why Ionic?

- Ionic is completely free and open source.
- Ionic is built on Angular.
- Ionic is a really "Native Like" framework.
- Ionic has a beautiful default UI that is easy to customise.
- It has a lot of tools and services.
- Ionic integrates easily with native functionalities.

3.4 Drawbacks

Since the drawing up of the Design Document, we have been asking ourself if a hybrid client (with no local storage) for the third party could be a good choice or not. Now we can say that this choice requires few effort from the

client point of view but it really reduces the performances of the server because it doesn't allow the server to exploit the caching capability of the client. For simplicity, since we are developing a prototype, we have decided to continues with this choice but we have structured the third party front end code, in such a way to easily move this web browser app in a native desktop app, for Windows, MacOs, Linux. Of course this operation will require a change in the Design Document and in this document. By choosing AngularJs as a front-end framework it will be much easier to do this transformation. Also the back-end will require just few changes, in particular the Storage Controller needs to retrieve data with timestamp greater than the timestamp of the last cached data on the client, and the ThirdParty Controller needs to be able to retrieve the timestamp of the last cached data on the client with REST API, call the proper method on the Storage Controller and then send the new data to the client.

Electron

Electron is a framework for creating desktop applications with all the emerging technologies including JavaScript, HTML and CSS. Basically, Electron framework takes care of the hard parts so that you can focus on the core of the application and revolutionize its design. Designed as an open-source framework, Electron combines the best web technologies and is a cross-platform – meaning that it is easily compatible with Mac, Windows and Linux.

Why Electron?

- HTML, CSS, JS: Of course this is the most important point. It is amazing that you can now build Desktop Apps using these languages as it is very easy to learn and use them.
- Electron Apps Are Similar To Web Apps: Part of what makes Electron Apps a good alternative to a native desktop app is the fact that Electron apps behave like Web Apps. What sets them apart is that Web Apps can only download files to the computer's file system but Electron Apps can access the file system and can also read and write data.
- Chromium: Electron uses Chromium engine for rendering UI. This means that you can get several benefits from this like Developer Tools, Storage Access, etc.

4 Source Code Structure

4.1 Back End

4.2 Front End

5 Testing

6 Installation Instructions

7 Source Code Structure

7.1 Romeo Paolo

7.2 Scotti Andrea

7.3 Staccone Francesco