

Software Engineering 2

**TrackMe**

**A**cceptance and **T**est **D**ocument



**POLITECNICO**  
**MILANO 1863**

Romeo Paolo  
Scotti Andrea  
Staccone Francesco

January 16, 2019

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose and Scope . . . . .	3
1.2	References . . . . .	3
1.3	Overview . . . . .	3
<b>2</b>	<b>Project</b>	<b>4</b>
2.1	Project info . . . . .	4
<b>3</b>	<b>Installation Setup</b>	<b>5</b>
<b>4</b>	<b>Acceptance Tests</b>	<b>7</b>
<b>5</b>	<b>Drawbacks</b>	<b>15</b>
<b>6</b>	<b>Conclusion</b>	<b>16</b>

# 1 Introduction

## 1.1 Purpose and Scope

This Document represents the Acceptance Test Document for the TrackMe project. The purpose of this document is to evaluate the system's compliance with respect to user needs and the functional requirements specified in the RASD.

## 1.2 References

For the acceptance tests we have considered the latest version of these reference documents:

- **RASD**: Requirements Analysis and Specification Document;
- **DD**: Design Document
- **ITD**: Implementation and Testing Document;

## 1.3 Overview

The rest of the document is organized in this way:

- **Project**: contains the identification of the project we have analysed;
- **Installation Setup**: explains what we did to install the prototype of the tested project as well as any problem or incoherency we have found following the provided documentation;
- **Acceptance Tests**: lists the acceptance test cases we have applied and the corresponding outcome.

## **2 Project**

### **2.1 Project info**

These are the details about the project we have tested:

#### **Authors**

- Andrea Constantin Scutariu 833370
- Carlo Pulvirenti 828459
- Sergio Piernario Placanica 916702

#### **Links**

- Git Repository: [link](#).

### 3 Installation Setup

A description of the installation process of the prototype is presented in the following subsections.

#### Downloaded files

Here follows a list of the tools needed to run the prototype:

- Node JS v-11.6.0
- npm v-6.5.0 (Node Package Manager)
- Flutter v-1.0.0
- Dart v-2.1.0
- JDK 8
- adb (Android Debug Bridge)
- kvm (Kernel-based Virtual Machine)
- Android Studio
- Android 8.0 Oreo (API 26)

#### Installation

As advised in the instructions, the installation has been tested in a linux environment (Ubuntu 16.04). The installation process can be divided into three parts:

- back-end server;
- smartphone app;
- wearable device;

The path to run the **back-end server** is simple and straightforward. It is just needed to fully execute step by step the instructions provided into sections 6.1 of the ITD document.

On the contrary, the instructions provided to install the smartphone application and the wearable device lack some useful information to be easily executed by someone who has never worked with Android before.

To run the **smartphone application** is required to use an Android smartphone

and to enable the debug mode on it. After having enabled the debug mode on the phone and having connected the phone to the notebook through USB, the execution of the instructions in the section 6.2 of the ITD document allows a correct installation of the application on the smartphone.

To run the **wearable application** is also required to install the tool kvm (Kernel-based Virtual Machine) and to enable the virtualization technology from the notebook's BIOS. Further, it is required to have a specific version of Android to run the application (in this case Android 8.0 (API 26)). Though, after the previous technical measures the sequence of instructions provided in section 6.3 of the ITD document works well.

## Conclusion

The installation instructions are generally correct even if not all the information is easy to get at first reading. They allow a user who has a minimum of confidence with the development of Android applications a quick configuration of the entire system.

However, in order for the instructions to be performed by a beginner it is advisable to integrate them with additional information on the configuration of the smartphone and the computer for a smoother execution of the whole process.

## 4 Acceptance Tests

### *Third Party*

The tests regarding the Third Party have been conducted through the use of the API development environment Postman, since the authors have not developed any interface for that.

The following tables include all the API requests we have tested for what concerns the Third Party side:

#### **Sign Up - POST**

Body	Expected outcome	Result
-Filled all the fields correctly -Select the company VAT -Select the company name -Select the company image -Select the company businessSector	Registration successful	Passed  Warning: the application also accepts VAT code of less than 11 characters. May be incoherent.
-Missing VAT or Company Name	Registration unsuccessful	Passed
-Already registered VAT	Registration unsuccessful	Passed

### Specific Request - POST

Body	Expected outcome	Result
-Filled all the fields correctly -Inserted existing SSN	Request registered	Passed
-Missing SSN	Request unsuccessful	Passed
-SSN already in charge of my previous request	Request unsuccessful	Passed

### Group Request - POST

Body	Expected outcome	Result
-Filters written correctly	Request registered	Passed
-No filters	Request registered	Passed

### List Specific Requests - GET

Body	Expected outcome	Result
-None	Specific requests retrieved	Passed



### Specific Request Data - GET

Body	Expected outcome	Result
-Id of a request sent by me (third party)	Specific request retrieved	Passed
-Id of a request sent by another company	Specific request not retrieved	Passed
-Id of a non existing request	Specific request not retrieved	Passed

### List Group Requests - GET

Body	Expected outcome	Result
-None	Group requests retrieved	Passed

### Group Request Data - GET

Body	Expected outcome	Result
-Id of a request sent by me (third party)	Group request retrieved	Passed
-Id of a request sent by another company	Group request not retrieved	Passed
-Id of a non existing request	Group request not retrieved	Passed

### Subscribe Specific Request - POST

Body	Expected outcome	Result
-Request Id and forwardingLink inserted correctly	Subscription registered	Passed
-Pending request Id	Subscription not registered	Passed
-Id of a request not sent by me (third party)	Subscription not registered	Passed
-Not existing request Id	Subscription not registered	Passed
-ForwardingLink not valid	Subscription not registered	Passed

### Unsubscribe Specific Request - POST

Body	Expected outcome	Result
-Request Id inserted correctly	Subscription registered	Passed
-Id of a request not sent by me (third party)	Subscription not registered	Passed
-Not existing request Id	Subscription not registered	Passed

### Subscribe Group Request - POST

Body	Expected outcome	Result
-Request Id and forwardingLink inserted correctly	Subscription registered	Passed
-ForwardingLink not valid	Subscription not registered	Passed
-Id of a request not sent by me (third party)	Subscription not registered	Passed
-Not existing request Id	Subscription not registered	Passed

### Unsubscribe Group Request - POST

Body	Expected outcome	Result
-Request Id inserted correctly	Subscription registered	Passed
-Id of a request not sent by me (third party)	Subscription not registered	Passed
-Not existing request Id	Subscription not registered	Passed

### ***Individual***

Tests regarding the Individual have been conducted through the use of the smart-phone application properly installed on an Android smartphone and the wearable application simulated on a fictitious smartwatch on Android Studio.

The following tables include all the requests we have tested for what concerns the Individual side:

#### **Sign Up - POST**

Action	Expected outcome	Result
- Filled all the fields correctly	Registration successful	Passed
- SSN not long 16 characters	Registration unsuccessful	Passed
- Already registered SSN or email	Registration unsuccessful	Passed

#### **Login - POST**

Action	Expected outcome	Result
- Correct credentials	Login successful	Passed
- Missing or wrong email	Login unsuccessful	Passed
- Missing or wrong password	Login unsuccessful	Passed

### Sign Up - POST

Action	Expected outcome	Result
- Click on add device, select a device from the list of available ones and specify a name	Device connected	Passed

### Sign Up - POST

Action	Expected outcome	Result
- Swipe right on the registered device	Device disconnected	Passed

**Limitation:** After having disconnected a device from a user, it's possible to register the same device to another user but the second user cannot see the device as connected and the device doesn't work anymore.

### Respond to a request - POST

Action	Expected outcome	Result
- Swipe left on the request in the requests section	Request accepted	Passed
- Swipe right on the request in the requests section	Request refused	Passed

## **Automated SOS**

It's possible to correctly simulate an emergency by simply changing the value of the parameters in the emulated smartwatch. After a few seconds from the parameter change, an SOS is raised and the smartphone application displays a page to communicate the user that an ambulance is coming. So the functionality itself works well.

Though, after the communication of the incoming ambulance, the user cannot do any other action on the application and it's impossible to continue to use it.

## 5 Drawbacks

The main drawback of this propotype is the impossibility to continue to use the smartphone application after the raising of an SOS, even if the server continues to work. Even if the smartwatch simulator is a good idea, it makes really hard to connect and test simultaneously multiple individuals.

Another weakness of this project is that, although it's a prototype, several core functionalities are missing. For example, since the third party, as a client, has not been implemented, there is no way to see aggregate properties about an anonymous group. Moreover it's impossible to recognize a difference between a get of past data and a get of new data. Another missing functionality, not explicitly required but important for an application that manage sensitive data, is the possibility change the password and other personal data.

To conclude this section, it has to be noticed that anonymity is not really guaranteed for group requests. Let's focus on a particular case: a group request for all people with age between 22 and 23 years with subscription to new data activated. In this case let's imagine that at the moment of the request there are 900 people of 23 years old and 100 people of 22 years old and let's suppose that no people among 22-23 years old will sign up to the service in the next years. After a year, the group will change and it will be composed by only the 100 people that has changed their age from 22 to 23 years old (meanwhile the other 900 people are out of this group because they are 24 years old). It should be expected that no more data are retrieved from this group after a year, but this is not the outcome of this application.

## 6 Conclusion

The application is working as expected and, aside for the few issues presented in the Drawbacks section, all the features described in the ITD are present. The user interface is clear and well organized and the user can easily and intuitively perform the few operations that are now accessible. Future updates should focus on the implementation of the parts needed to make the user able to continue to use the application after an SOS is sent and the change of credentials for the user.