DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Acceptance Test Deliverable (ATD)

TRACKME

Authors:

Avila, Diego

Schiatti, Laura

Virdi, Sukhpreet

903988

904738

904204

January 20th , 2019

Contents

List of Figures

Introduction

1.1 Purpose

This document is the Acceptance Test Plan (ATP) for **TrackMe** software. The acceptance test verifies that the system works as required and validates that the correct functionality has been delivered. The main purpose of this test is to evaluate the system's compliance with the requirements and verify if it is has met the required criteria for delivery to end users.

1.2 Analyzed project

Documents and source code regarding the project that was analyzed can be consulted from the repository **https://github.com/DavideRutigliano/FerranteMattaRutigliano**.

Authors:

- Davide Rutigliano

- Davide Matta

- Claudio Ferrante

1.2.1 Abbreviations

- $[REQ - n]$: n-functional requirement.

- $[TEST - n]$: n-test case.

1.3 Overview

This document describes the focuses of the validation activity and the results obtained. The below table describes how the activities under Acceptance Testing was formulated:

Activity	*Checkpoint*
Plan Acceptance Testing for TrackMe	*Preliminary Acceptance Test Scenario*
Identify Test Materials	*Preliminary Acceptance Test Data*
Establish Acceptance Test Environment	*Acceptance Test Environment Inventory*
Conduct Acceptance Test Readiness Review	*Draft Acceptance Test Plan* *Completed Test Readiness Checklist*
Execute Tests	*Acceptance Test Progress*
Complete Acceptance Testing	*Acceptance Test Summary*
Document Acceptance Testing	*Final Acceptance Test Document*

Table 1.1: Acceptance Testing Schedule

Test traceability is used to verify that functional requirements are accounted for, and tested in the delivered software. It provides traceability between the requirements and the test materials.

Regarding the preparation of environment preparatory activities involve constructing the actual acceptance test environment and installing the hardware and software, including the software to be tested, in order to ensure the configuration is correct. Changes need to be closely managed and controlled. Prior to an installation or configuration, the acceptance test team reviews a list of affected components and the planned installation or configuration activities. Deviations from the planned activities are recorded and reported to the acceptance test team. The acceptance test team begins acceptance testing by conducting an initial series of ad hoc, diagnostic tests designed to exercise the acceptance test environment and verify that major system software capabilities are available and functioning.

With the Acceptance Test Plan as the guiding document, components reviewed include, but are not limited to:

- Software components

- Database

- Environmental components

- Documentation

- Known problems and outstanding issues from the System Test

- Inventory of the acceptance test environment

If the status of one or more of the components is unsatisfactory, the parties identify corrective actions and schedule another ATD. If the status is acceptable, then testing will proceed.

Acceptance test execution is an iterative process that begins with the initial execution of the planned tests. If no defects are discovered, then the test procedure has "passed." If defects are discovered, they are documented.

Installation setup

The application was tested by means of a Linux machine, following the steps provided in the *Implementation and Testing Deliverable*.

2.1 Server Build and Installation

- **Database**

  1. Install PostgreSQL Database and pgAdmin

     ```
     $ sudo apt install postgresql
     $ sudo -u postgres psql postgres
     $ \password postgres
     $ sudo apt install pgadmin4
     ```

  2. Launch pgAdmin

     ```
     $ sudo pgadmin4
     ```

     (a) Select pgAdmin 4 > New pgAdmin 4 window
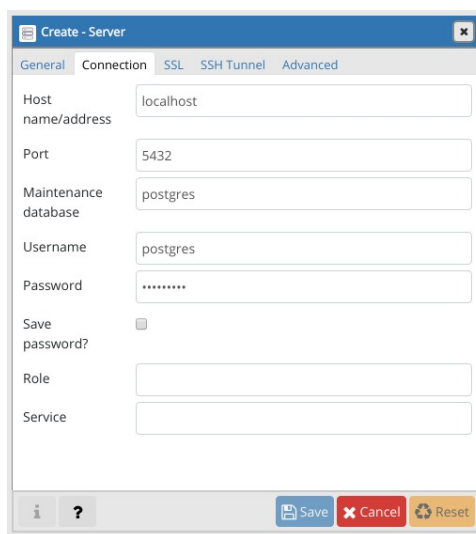   (b) Create and setup a new server [Fig. 2.1].

Figure 2.1: Data4Help Create Server

     (c) Press Save.

3. Create and setup a new user [Fig. 2.2].

    (a) Right click on Login/Group Roles > create > Login/Group Role

    (b) In the tab *General*, set name = trackmeadmin.

    (c) In the tab *Definition*, set password = password.

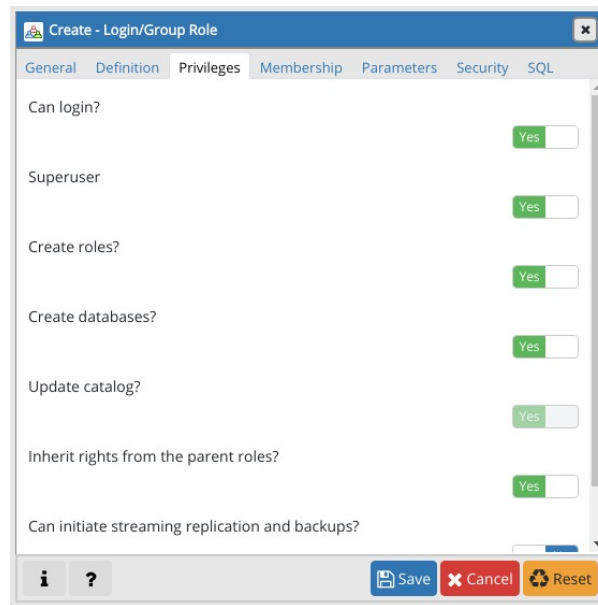   (d) In the tab *Privileges*, set Superuser = yes and Can login = yes.

Figure 2.2: Data4Help Setup User privileges

4. Create a new database

    (a) Right click in *Databases* > Create > Database

    (b) In the tab *General*, set name = trackme_db.

- **Build with Maven**

```
$ cd FerranteMattaRutigliano/DeliveryFolder/software/server
$ chmod +x mvnw
$ ./mvnw package
```

2.2   Client Build and Installation

- Build with Android Studio

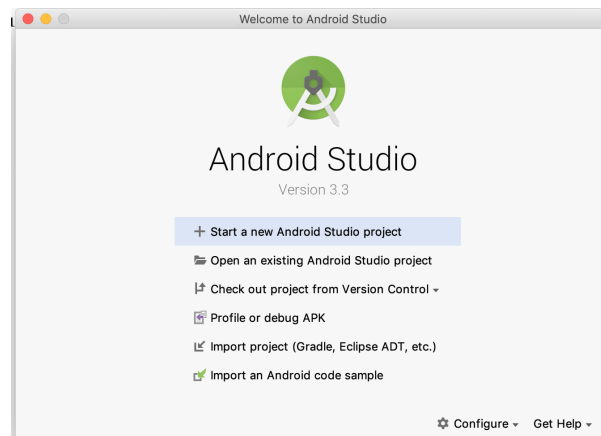   1. Download and install **Android Studio**

Figure 2.3: Android Studio

2. Select *Open an existing android studio project*

3. Open "ferrantemattarutigliano/software/client/build.gradle".

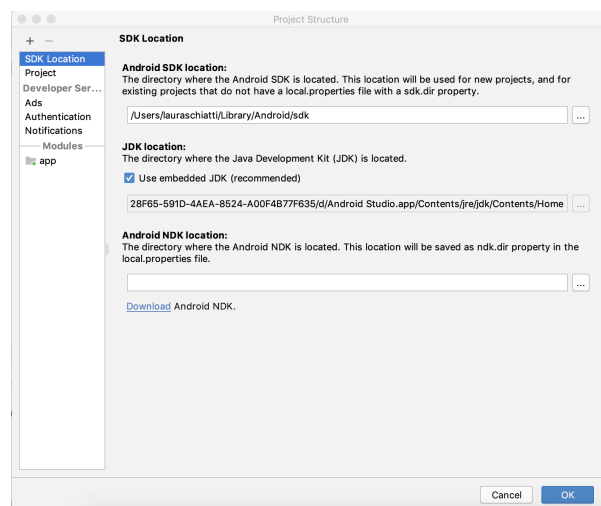4. Configure the project. Go to *Select file > project structure.*

Figure 2.4: Configure Android Studio

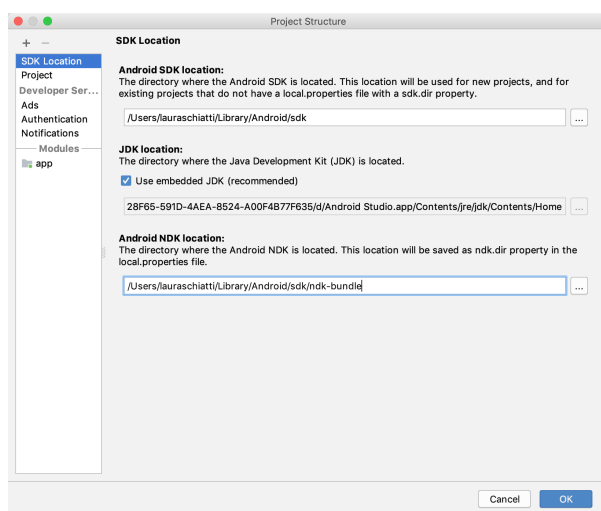5. Install **Android NDK** by clicking in the *Download* link (Fig. 2.3).

Figure 2.5: Install Android NDK

6. Run the application and select a deployment target. For testing a Samsung J8 was used instead of a virtual device 2.6.

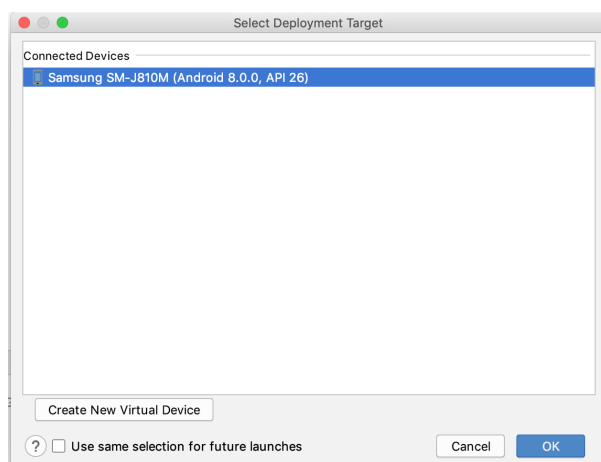
Figure 2.6: Choose Deployment Target
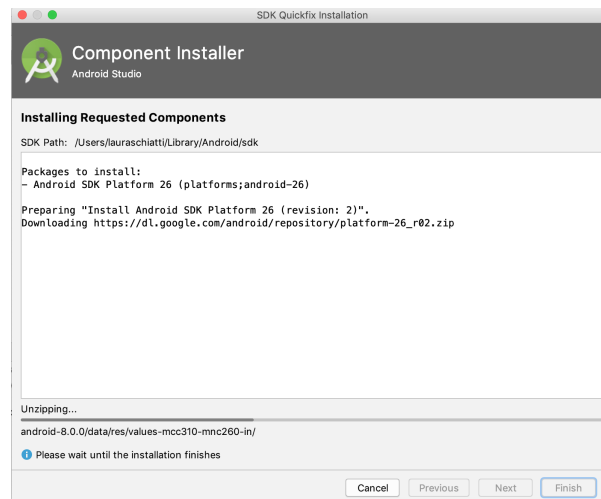
7. Download the corresponding SDK.

Figure 2.7: Install SDK

2.3   External Device App Build and Installation

It was possible to run the device emulator. But it was not possible to run both emulators at the same time and make them communicate. The main problem was due to the next command, it was not working

```
$ adb -d forward tcp :5601 tcp :5601
```

Acceptance test cases

3.1 Motivation

The system requirements are categorized according to their importance regarding the scope of the problem and the main functionalities to be implemented in order to have a minimum viable prototype as **HIGH**, **MEDIUM** and **LOW**.

All requirements defined as **MEDIUM** and **HIGH** were tested and their outcomes. Even if *registration* (for both third parties and individuals) was not consider relevant, it was also assessed in order to have available users for testing.

3.2 Tested features

3.2.1 Data4Help

[REQ-1] A customer not signed-on must be able to begin the Individual's registration process to TrackMe providing a username, a password and its organization data. [**LOW**]

    [TEST-1] : Register an individual for the first time [Fig. 3.1].

          ∗ **Inputs:** username=Individuo, ssn=344918500

          ∗ **Outcome:** OK. Individual registered successfully.

          ∗ **Result:** Passed.

Figure 3.1: Data4Help Register Individual

[TEST-2] : Register an individual with an already registered ssn.

* **Inputs:** username=individual2, ssn=344918500
* **Outcome:** Error. Since ssn is duplicated, the registration was not allowed.
* **Result:** Passed.
  - **Possible bugs**: Birth date field is not validated

[REQ-2] A customer not signed-on must be able to begin the Third Party's registration process to TrackMe providing a username, a password and its organization data. [**LOW**]

[TEST-3] : Register third party for the first time [Fig. 3.2].

* **Inputs:** username=Tp1
* **Outcome:** At first, time out warning but then we redo the request, and everything was fine.
* **Result:** Conditionally Passed.

Figure 3.2: Data4Help Register Third Party

[TEST-4] : Register a Third party with an already registered vat.

* **Inputs:** username=Tp1
* **Outcome:** Error. The vat was duplicated, registration not allowed.
* **Result:** Passed.

[REQ-3] The system must provide a log-in interface for already registered users, not previously signed into the application. [**MEDIUM**]

[TEST-5] : Login user with valid credentials [Fig. 3.2].

* **Inputs:** Password=1234567890, Username=Tp1
* **Outcome:** OK. User Logged in successfully.
* **Result:** Passed.

[TEST-6] : Login with invalid credentials

* **Inputs:** Username=Tp1, Password=12341231
* **Outcome:** Login Error.
* **Result:** Passed.

[REQ-4] TrackMe must provide to users the possibility to change their username. [**LOW**]

[REQ-5] TrackMe must provide to registered users the possibility to change their password. [**LOW**]

[REQ-6] The system must provide the possibility to the user of logging out. [**LOW**]

[REQ-7] The system must provide the possibility to change Individual's personal data. [**LOW**]

[REQ-8] The system must provide the possibility to change Third Party's organization data. [**LOW**]

[REQ-9] Data4help must allow the Third-Parties to send a request to a particular individual, provided his SSN/FC. [**HIGH**]

[TEST-7] : Send request to individual [Fig. 3.3].

* **Inputs:** SSN 344918500
* **Outcome:** Ok. Request was sent to individual successfully.
* **Result:** Passed.

– **Possible Bugs:** a third party can send 2 requests to the same user, the user can accept one of them and reject the other, and the DB will be in an inconsistent state.
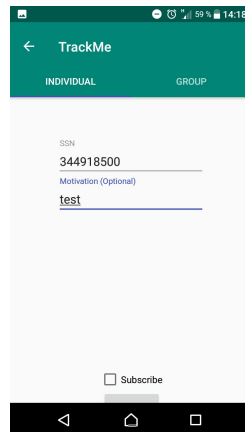
Figure 3.3: Data4Help Request Individual

[TEST-8] : Send request with an invalid ssn

       ∗ **Inputs:** SSN 1112121a (invalid)

       ∗ **Outcome:** Error. Invalid SSN.

       ∗ **Result:** Passed.

[REQ-10] Data4help must allow the Third-Parties to generated a request for a group of individuals. [**HIGH**]

[TEST-9] : Request for group of individuals [Fig. 3.4].

       ∗ **Inputs:** -

       ∗ **Outcome:** The system cannot anonymize the data (minimum amount of individuals is not satisfied), and error is displayed.

       ∗ **Result:** Manual testing was not possible.
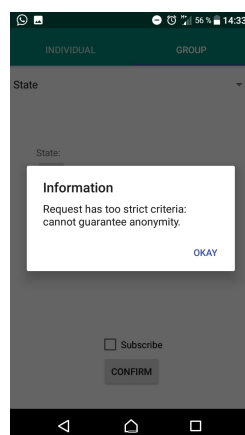

Figure 3.4: Data4Help Request Group Individual

[REQ-11] The system must be able to anonymize users' requested data. [**HIGH**]

[TEST-10] : Anonymize data.

     ∗ **Inputs:** -

     ∗ **Outcome:** cannot be tested because there is no enough data.

     ∗ **Result:** Testing was not possible.

[REQ-12] Data4help must allow the Third-Parties to choose if subscribe to new data associated with a particular Individual. [**MEDIUM**]

[TEST-11] : Subscribe to new data of an individual while requesting for data [Fig. 3.5].

     ∗ **Inputs:** SSN 344918500

     ∗ **Outcome:** If the request is accepted by the individual, the subscription is created automatically.

     ∗ **Result:** Passed.

    – **Possible Bugs:** even if the third party is subscribed to the individual, he does not receive the data as soon as it is produced. To do it, he has to look for the accepted request and click on the individual to visualize a chart.
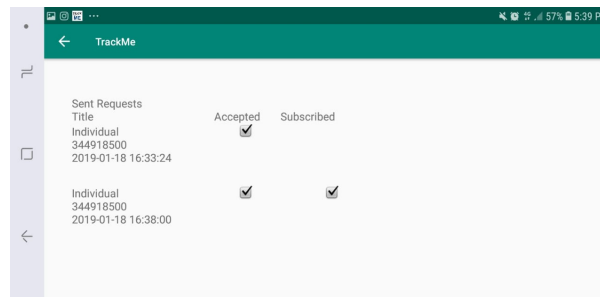

Figure 3.5: Data4Help data subscription

[REQ-13] The system must be able to periodically query the database in order to get new data as soon as they are produced. [**MEDIUM**]

[TEST-12] : Run the query for data periodically.

     ∗ **Inputs:** -

     ∗ **Outcome:** None of the provided documents state the meaning of periodically and there is no way of testing it.

     ∗ **Result:** Testing was not possible.

[REQ-14] Data4help must allow the Third Party to see a list of sent requests and related Individual's data. [**LOW**]

[REQ-15] Data4help must allow the Individual to see a list of the received requests from Third Parties. [**LOW**]

[REQ-16] Data4help must allow the Individual to accept or reject a request from the list. [**MEDIUM**]

    [TEST-13] : Accept or Reject the request [Fig. 3.6].

        ∗ **Inputs:** password=1234567890, username=Tp1

        ∗ **Outcome:** Ok. The status is updated and the request is removed from the list (no update options).
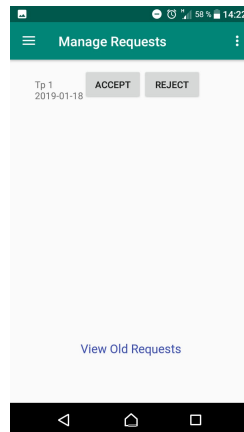
        ∗ **Result:** Passed.

Figure 3.6: Data4Help Accept/Reject Request

[REQ-17] Data4help must allow the Individual to connect an external device through BT or NFC. [**HIGH**]

3.2.2 Track4Run

- **Runs management**

[REQ-24] Track4Run must allow the organizer to create a run with a date, time, duration and a path. [**HIGH**]

    [TEST-14] : Create a run with valid data set [Fig. 3.7].

        ∗ **Inputs:** Set title, Date, time and path.

        ∗ **Outcome:** Ok. Path was created successfully.
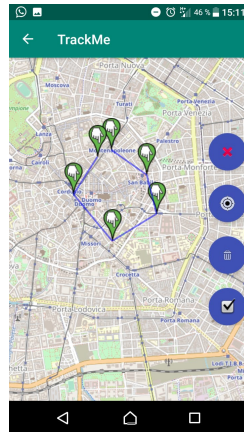
        ∗ **Result:** Passed.

Figure 3.7: Track4Run Create path

[TEST-15] : Create a run with invalid fields

* **Inputs:** Select only 2 points
* **Outcome:** Path with less than 2 points are not valid. All fields are validated.
* **Result:** Passed.

[REQ-25] The system must provide an interface that allows the user to define a path on an interactive map. [**MEDIUM**]

[TEST-16] : Set path on an interactive map [Fig. 3.7].

* **Result:** Passed (same as above).

[REQ-26] Track4Run must allow the organizer to start a run previously created. [**MEDIUM**]

[TEST-17] : Start a run

* **Inputs:** Select a run to start from the active run list
* **Outcome:** the organizer can check the list of runs he created and start any of them at any time (the system does not check whether it is about to take place or not)
* **Result:** Fail.
  – **Possible Bugs:** the only way of "stopping" a run is deleting it.

[TEST-18] : Start a run with less than 2 athletes

* **Inputs:** Select a run in which there are less than 2 athletes participating
* **Outcome:** it was possible, considering domain assumption 18 (A -18).
* **Result:** Passed.

[TEST-19] : Start a run that start in future dates

* **Inputs:** Select run whose start data is in future.
* **Outcome:** Possible to start the run, but cannot be "watched" by the spectators.
* **Result:** Fail.

[REQ-27] Track4Run must allow the organizer to delete a run previously created. [**LOW**]

[REQ-28] Track4Run must allow the Athlete to enroll to an already existing run. [**HIGH**]

[TEST-20] : Enroll a run [Fig. 3.8].

* **Inputs:** Select a run to to enroll from active run list
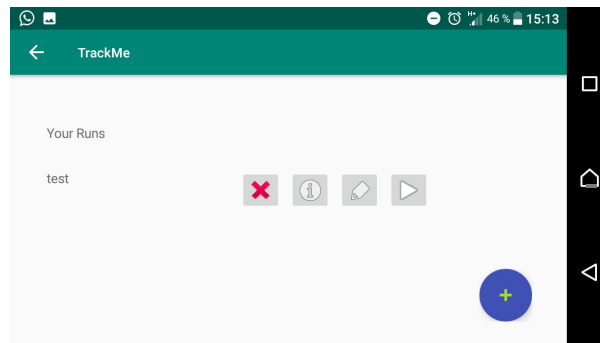* **Outcome:** Ok. User is allowed to enroll into the run.
* **Result:** Passed.

Figure 3.8: Track4Run Enroll Run

[TEST-21] : Enroll to a run that has already started

* **Inputs:** Select a run to enroll which has already been started.
* **Outcome:** the user can enroll even if the run has started.
* **Result:** Fail.

[REQ-29] Track4Run must allow the Athlete to unroll a run. [**LOW**]

[REQ-30] Track4Run must allow the Spectator to see the position of the Athletes on a map during a run.[**HIGH**]

[TEST-22] : Spectators can see the position of athletes during a run.

* **Inputs:** press watch, if the run wasn't previously started by the organizer an error message is displayed.
* **Outcome:** We can visualize the map with the path but since there's no data, it is possible to see the position of the athletes.
* **Result:** Testing with real data was not possible.

[TEST-23] : See position of athletes in a a run that hasn't started is not possible.

        ∗ **Inputs:**   Select a run which has already been started.

        ∗ **Outcome:**   No real data was available.

        ∗ **Result:**   Testing was not possible.

Additional annotations

Any additional point you want to raise on the quality of documentation and code.

References

- Requirement Analysis and Specification Document.pdf. Version 1.0 - 10.11.2018.

- Design Document.pdf. Version 1.0 - 10.12.2018.

- Implementation and Testing Document.pdf. Version 1.0 - 13.01.2019.