Politecnico di Milano
AA 2018-2019

Computer Science and Engineering

Software Engineering II

TRACKMe

Acceptance Test Document

Stefano Bagarin
Alessandra Pasini

Document version: 1.0
January 20, 2019

Contents

Section 1

Introduction

1.1 Scope

This document has the aim to analyze the Software Engineering II's project developed by *Giovanni Lupi & Davide Maffiola*.
The whole project, source code and documentation, can be found at the following link:
`https://github.com/dvdmff/LupiMaffiola`

In the next sections the installation and all functionalities developed will be analyzed.

1.2 References

This document is strictly based on the specification concerning the Implementation and Testing assignment for the Software Engineering II project, part of the course held by professor Matteo Rossi and Elisabetta Di Nitto at the Politecnico di Milano, A.Y 2018/2019. It refers to the project done by the assigned group.

Section 2

Installation Setup

We have first cloned the whole project and saved it in a chosen folder.

2.1 Server Installation

To install the server we have tried on Windows and we have followed the next steps:

- installation of MySQL 5.7, in particular, on this platform, we have used the workbanch and not the shell:
  - run the script data4help_db.sql;
  - create the user;
  - give all prermission to the user just created;
- download of WildFly-15.0.1- Final.zip:
  - unzip the package;
  - insertion of the new absolute path in which is it possible to find the WildFly files in the *default.config* file. This file can be found in *./DeliveryFolder/server*;
- installation of Python 3.7;
- installation of Apache Maven:
  - the chosen JDK version has been the 8th because, from the 9th, JAXB and other functionalities have been transformed in APIs and it would has been necessary to download them apart ;

    - the JAVA_HOME evironment variable has been changed with the chosen JDK version;
    - the MAVEN_HOME environment variable has been added;

- execution of the three commands to build up the server.
- check on browser if the server and the DB has been correctly started and connected.

Unfortunately we had some troubles in the MySQL user registration, by checking on browser we have seen that both the DB and the server were correctly working. The user was correctly created, we have tried to modify the way username and password where enclosed, but no changes has been seen. After lots of try we haven't been able to solve this problem.
Because of the fact that the group has worked on Linux we have tried on this platform. To install the server on Linux we have followed the next steps:

- installation of JDK-8 and Maven packages;
- installation of MySQL 5.7, in this case we have used the shell:
    - run the script data4help_db.sql by adding to the given command the whole path in which it was possible to find the file;
    - change the variables for password requirements;
    - create the user;
    - give all permission to the user just created;
- download of WildFly-15.0.1- Final.zip:
    - unzip the package;
    - insertion of the new absolute path in which is it possible to find the WildFly files in the *default.config* file. This file can be found in *./DeliveryFolder/server*;
- execution of the three commands to build up the server.
- check on browser if the server and the DB has been correctly started and connected.

This time the server worked in a correct way.

The group members of the project we had to analyze have been really helpful and fast in answering to our questions. They tried in every way to solve the problem we have found in the server installation.

2.2 Client Installation

To install theusers' client we have followed the next steps:

- connect the smartphone to the computer;
- add the APK in a reachable smartphone's folder;
- install it.

Everything worked immediately.

For what concern the Thrid Party client the link at which could be found the web page was missing in the installation instructions. We have received it as soon as we asked it to the group members.
Everything worked correctly.

Section 3

Acceptance Tests

3.1 Mobile App Acceptance Test

This section has the aim to explain all acceptance tests done on the mobile app, the reasons why those tests have been chosen and the obtained output.

3.1.1 Connection Settings

Test 1

To make the app work, the smartphone has been connected to the same wifi of the computer in order to use the local domain. This test has been done because that's the first prerequisite to correctly run the app.

Given: domain = local domain found by the command ifconfig -a
When: As soon as the user open the app
Expected then: the change is saved and the app go back at the login scene

The output is equal to the expected behaviour.

3.1.2 Registration

Test 1

Check if the registration is stopped if all fields are empty. This test has been done because the user must complete the registration form in order to obtain a personal area.

Given: all fileds in the registration form are empty

When: the guest wants to register
Expected then: the app doesn't allow the guest to register

The output is equal to the expected then, it appears a popup saying that all fields must be filled in order to register.

Test 2

Check if the registration is stopped if some fields are empty. This test has been done because the user must fill the whole registration form in order to obtain a personal area.

Given: some fields in the registration form are empty, some other have been filled
When: the guest wants to register
Expected then: the app doesn't allow the guest to register

The output is equal to the expected then, it appears a popup saying that all fields must be filled in order to register. This test has been done lots of times in order to evaluate if all fields are necessary for the registration or if some of them can be omitted. It has resulted that all fields are necessary in order to obtain a personal area.

Test 3

Check if the registration is stopped if the SSN is written in a wrong way. This test has been done because the user must provide a possible real SSN; this is important because a third party may required this user data and the way to ask them is by the SSN.

Given1: SSN != from format xxx-xx-xxxx
Given2: SSN with a correct format but with numbers and letters
When: the guest wants to register
Expected then: the app doesn't allow the guest to register

In both cases the output is equal to the expected then, it appears a popup saying that the SSN is not correct.

Test 4

Check if the registration is stopped if the written SSN is already associated to an existing profile. This test has been done because the user must provide a possible real SSN and it must be unique inside the DB.

Given: SSN = one SSN already in DB
When: the guest wants to register
Expected then: the app doesn't allow the guest to register

The output is equal to the expected then, it appears a popup saying that the SSN has already been registered.

Test 5

The user fills all fields with correct values. This is necessary in order to create a new personal area.

Given: All fields have been correctly filled
When: the guest wants to register
Expected then: the app register the user.

The output is equal to the expected then, it appears a popup saying that the new user can now login.

Test 6

The IP hasn't been set in a correct way. This Test has been done to check if the app behave in a proper way when it is not possible to connect to the server (BAD URL)

Given: The ip is wrong
When: the guest wants to register
Expected then: the app doesn't allow the guest to register

The output is equal to the expected then, it appears a popup saying that there is no active connection between the client and the server.

3.1.3 Login

Test 1

Check if the login is stopped if all fields are empty. This test has been done because the user must complete the login form in order to access to his/her personal area.

Given: all fileds in the login form are empty
When: the guest wants to login
Expected then: the app doesn't allow the guest to login

The output is equal to the expected then, it appears a popup saying that all fields must be filled in order to login.

Test 2

Check if the login is stopped if some fields are empty. This test has been done because the user must fill the whole login form in order to access to his/her personal area.

Given: some fields in the login form are empty, some other have been filled
When: the guest wants to login
Expected then: the app doesn't allow the guest to login

The output is equal to the expected then, it appears a popup saying that all fields must be filled in order to login.

Test 3

Check if the login is stopped if the SSN is written in a wrong way or is different from any stored in DB. This test has been done because the user has provided a possible real SSN during the registration and the one insert in the login must be equal to any stored in DB.

Given1: SSN != from format xxx-xx-xxxx
Given2: SSN with a correct format but with numbers and letters
Given3: SSN != from any SSN stored in DB
When: the guest wants to login
Expected then: the app doesn't allow the guest to login

In all cases the output is equal to the expected then, it appears a popup saying that the SSN format is not correct or that it is not in DB.

Test 4

Check if the login is stopped if the password is different from the one associated to the given correct SSN. This test has been done in order to verify if the authentication works in a correct way.

Given: password != stored password realted to correct SSN
When: the guest wants to login
Expected then: the app doesn't allow the guest to login

The output is equal to the expected then, it appears a popup saying that the password is not correct.

Test 5

The IP hasn't been set in a correct way. This Test has been done to check if the app behave in a proper way when it is not possible to connect to the server (BAD URL)

Given: wrong IP
When: the guest wants to login
Expected then: the app doesn't allow the guest to login

The output is equal to the expected then, it appears a popup saying that there is no active connection between the client and the server.

Test 6

The user fills all fields with correct values. This is necessary in order to acces to the personal area.

Given: All fields have been correctly filled
When: the guest wants to login
Expected then: the app login the user

The output is equal to the expected then, the scene changes and the user access to his/her personal area.

3.1.4 Requests Management

Test 1

Check if a third party pending request appears in the Pending request scene. This is a crucial aspect because it is foundamental for the correct working of the web application.

Given: third party request without subscription
When: the user is in the pending request area
Expected then1: the request appears correctly in the scene and gives the possibility to be accepted or to be denied if no subscribed request has been accepted before
Expected then2: the request doesn't appear if a subscribed request has been accepted before

The output is equal to the expected one when the request is the first one from that specific third party. If the user has already accepted an other request from this third party, the request doesn't appear and it's automatically accepted. This behavior is partially different from the expected one: a request should be automatically accepted only when a subscribed request has been accepted before.
Reference to *[R5.1.4]*.

Test 2

Check if a third party pending subscribed request appears in the Pending request scene. This is a crucial aspect because it is foundamental for the correct working of the web application.

Given: third party request with subscription
When: the user is in the pending request area
Expected then1: the request appears correctly in the scene and gives the possibility to be accepted or to be denied if no subscribed request has been accepted before
Expected then2: the request doesn't appear if a subscribed request has been accepted before

The output is equal to the expected one when the request is the first one from that specific third party. If the user has already accepted an other request from this third party, the request doesn't appear and it's automatically accepted. This behavior is partially different from the expected one: a request should be automatically accepted only when a subscribed request has been accepted before.
Reference to *[R5.1.4]*.

Test 3

Check if a third party pending request can be accepted. This is a crucial aspect because it is foundamental for the correct working of the web app.

Given: third party request without subscription in the pending request area
When: the user wants to accept the request
Expected then: the request disappears from the pending request scene and the Third Party can see the data of the user obtained till that moment

The output is equal to the expected one.

Test 4

Check if a third party pending request can be refused. This is a crucial aspect because it is foundamental for the correct working of the web application.

Given: third party request without subscription in the pending request area
When: the user wants to refuse the request
Expected then: the request disappears from the pending request scene and the Third Party can't see the data of the user

The output is equal to the expected one.

Test 5

Check if a third party subscribed pending request can be refused. This is a crucial aspect because it is foundamental for the correct working of the web application.

Given: third party request with subscription in the pending request area
When: the user wants to refuse the request
Expected then: the request disappears from the pending request scene and the Third Party can't see the data

The output is equal to the expected one.

Test 6

Check if a third party subscribed pending request can be accepted. This is a crucial aspect because it is foundamental for the correct working of the web application.

Given: third party request with subscription in the pending request area
When: the user wants to accept the request
Expected then: the request disappears from the pending request scene and appears in the subscribers scene and the Third Party can see the data obtained till that moment and new data will be sent to it with the chosen frequency

The output is equal to the expected one.

Test 7

Check if a third party subscribed request in the subscribers scene can be removed. This is a crucial aspect because it is foundamental for the correct working of the web application.

Given: third party request with subscription in the subscribers scene
When: the user wants to stop the current subscription
Expected then: the request disappears from the subscriber scene and no more data is sent to the Third Party

The output is equal to the expected one.

3.1.5 Logout

Test1

Check if the logout is done in a correct way. This test is done in order to check if this functionality has been well developed.
Given: user logged in his/her personal area
When: the user wants to logout
Expected then: the personal area is closed and it will be necessary to login again in order to access again to the personal area

The output is equal to the expected one.

3.2 Web Client Acceptance Test

This section has the aim to explain all acceptance tests done on the web client, the reasons why those tests have been chosen and the obtained output.

3.2.1 Registration

Test 1

Check if the registration is stopped if some fields are empty. This test has been done because the Third Party must complete the registration form in order to obtain a personal area.

Given: some fileds in the registration form are empty
When: the guest wants to register
Expected then: the web page doesn't allow the guest to register

The output is equal to the expected one.

Test 2

Check if the registration is stopped if the VVT is already associated to an existing profile. This test has been done because the third party must provide a possible real VVT and it must be unique inside the DB.

Given: VVT = one VVT already in DB
When: the guest wants to register

Expected then: the app doesn't allow the guest to register

The output is equal to the expected one.

Test 3

The user fills all fields with correct values. This is necessary in order to create a new personal area.

Given: All fields have been correctly filled
When: the guest wants to register
Expected then: the web page register the Third Party.

The output is equal to the expected one, the Third Party can now login.

3.2.2 Login

Test 1

Check if the login is stopped if some fields are empty. This test has been done because the Third Party must complete the login form in order to access to its personal area.

Given: some fileds in the login form are empty
When: the guest wants to login
Expected then: the app doesn't allow the guest to login

The output is equal to the expected one.

Test 2

Check if the login is stopped if the VVT is different from any stored in DB. This test has been done because the Third Party has provided a possible real VVT during the registration and the one insert in the login must be equal to any stored in DB.

Given: VVT != from any VVT stored in DB
When: the guest wants to login
Expected then: the app doesn't allow the guest to login

The output is equal to the expected one.

Test 3

Check if the login is stopped if the password is different from the one associated to the given correct VVT. This test has been done in order to verify if the authentication works in a correct way.

Given: password != stored password realted to correct VVT
When: the guest wants to login
Expected then: the app doesn't allow the guest to login

The output is equal to the expected one.

Test 4

The Third Party fills all fields with correct values. This is necessary in order to acces to the personal area.

Given: All fields have been correctly filled
When: the guest wants to login
Expected then: the app login the Third Party

The output is equal to the expected one, the scene changes and the Third Party access to its personal area.

3.2.3 Personal Request

Test 1

Check if a personal request without or with subscription to a wrong SSN is stopped. This test has been done in order to see if the software does some checks on the SSN before creating the request.

Given: SSN format != xxxxxxxxx format
When: the Third Party wants to send a personal request
Expected then: the web page doesn't allow it

The output is equal to the expected one, the web page shows a popup saying that the request is malformed.

Test 2

Check if a personal request without or with subscription to a not existing SSN is stopped. This test has been done in order to see if the request isn't created when there is any SSN equal to the given one in the DB.

Given: SSN format = xxxxxxxxx format
When: the Third Party wants to send a personal request and the SSN ins't in the DB
Expected then: the web page doesn't allow it

The output is equal to the expected one.

Test 3

Check if a personal request with subscription to an existing and corret SSN is stopped if the frequency isn't set. This test has been done in order to see if it is necessary to set a frequency in obtaining new data.

Given: request with frequency not set
When: the Third Party wants to send a personal request with subscription and no frequency
Expected then: the web page doesn't allow it

The output is equal to the expected one.

Test 4

Check if a personal request without subscription to an existing and corret SSN goes well. This test has been done in order to see if the request is created when everything is correct.

Given: SSN format = xxxxxxxxx format
When: the Third Party wants to send a personal request without subscription and the SSN is in the DB
Expected then1: if there isn't any request with subscription related to this specific SSN, the request appears in the main page with the pending

attribute until the user answer
Expected then2: if there is a request with subscription related to this specific SSN, the request appears in the main page with the accepted attributed. The request is ready to let the Third Party download the data collected till that moment.

The output is equal to the expected one in the first case, partially to the second one because of what already said in the first and second tests of Request Manager in user's acceptance test section.
Reference to *[R5.1.4]*.
In the first case, if the request is accepted, data are retrieved and showed to the Third Party.

Test 5

Check if a personal request with subscription to an existing and corret SSN goes well. This test has been done in order to see if the request is created when everything is correct.

Given: correct request
When: the Third Party wants to send a correct personal request with subscription
Expected then1: a new line appears in the subscribed page. If there isn't any request with subscription related to this specific SSN, the request appears in the main page with the pending attribute until the user answer.
Expected then2: a new line appears in the subscribed page. If there is a request with subscription related to this specific SSN, the request should appear in the main page with the accepted attribute. Data is sent at the specified frequency.

The output is equal to the expected one in the first case, but partially to the second one because of what already said in the first and second tests of Request Manager in user's acceptance test section.
Reference to *[R5.1.4]*.
In the first case, if the request is accepted, data is sent and showed to the Third Party at the specified frequency.

Test 6

Check if it is possible to unsubscribe to a personal request with subscription.

Given: a personal request subscribed in the Subscribe page
When: the Third Party wants to unsubscribe
Expected then: the web page allows it, the request disappear from the Subscribe page and no more data is sent to the Third Party.

The output is equal to the expected one.

3.2.4 Anonymous Request

Test 1

Check if an anonymous request without subscription and without filters is stopped. This test has been done in order to check if an anonymous request without filters can't be created and showed in the main page as said in the ITD.

Given: anonymous request without subscription and without filters
When: the Third Party wants to send an anonymous request
Expected then: the web page doesn't allow it and do not create the request

The output is equal to the expected one, the request isn't created and doesn't appear in the main page.

Test 2

Check if an anonymous request with subscription and without filters is stopped.

Given: anonymous request with subscription and without filters
When: the Third Party wants to send an anonymous request
Expected then: the web page doesn't allow it, the request is created but it doesn't require data.

The output is equal to the expected one, the request is created, put in both main and subscribed page, but data aren't retrieved.

Test 3

Check if an anonymous request without subscription and with some wrong filters is stopped.

Given: anonymous request without subscription and wrong filters
When: the Third Party wants to send an anonymous request
Expected then: the web page doesn't allow it, the request isn't created.

The output is equal to the expected one, the request isn't created and doesn't appear in the main page.

Test 4

Check if an anonymous request with subscription and with some wrong filters is stopped.

Given: anonymous request with subscription and wrong filters
When: the Third Party wants to send an anonymous request
Expected then: the web page doesn't allow it, the request is created but it doesn't get data.

The output is equal to the expected one, the request is created, put in both main and subscribed page, but data isn't retrieved.

Test 5

Check if an anonymous request without subscription and with correct filters works.

Given: anonymous request without subscription and correct filters
When: the Third Party wants to send an anonymous request
Expected then1: the web page allows it ; the request is created and sends all data obtained till that moment if the number of the individuals is higher than 1000.
Expected then2: the web page allows it ; the request is created, but it doesn't send data if the number of the individuals is less than 1000.

The output is always equal to the "Expected then2": it hasn't been possible to test if data are correctly send in case the number of individual is higher than the threshold bacause it is to high for testing. Refears to *[R5.2.1]*.

Test 6

Check if an anonymous request with subscription and with correct filters works.

Given: anonymous request with subscription and correct filters
When: the Third Party wants to send an anonymous request
Expected then1: the web page allows it ; the request is created and start sending data at the chosen frequency if the number of the individuals is higher than 1000.
Expected then2: the web page allows it ; the request is created, but it doesn't send data if the number of the individuals is less than 1000.

The output is always equal to the "Expected then2": it hasn't been possible to test if data are correctly send in case the number of individual is higher than the threshold bacause it is to high for testing.
Reference to *[R5.2.2]*.

Test 7

Check if it is possible to unsubscribe to an anonymous subscribed request.

Given: an anonymous request subscribed in the Subscribe page
When: the Third Party wants to unsubscribe
Expected then: the web page allows it and the request disappear from the Subscribe page

The output is equal to the expected one.

Section 4

Observation

We have observed that the app sends data to the server even when the GPS is OFF. The expected behaviour is different, in fact the app should be closed and data shouldn't be send to the server. The location (both longitude and latitude) is set equal to zero.
When the GPS is turned on the location starts to be send in a correct way and it is set to the correct value.
We have also observed that once an anonymous request with subscription is created new pending requests starts to appear in the main page.

Appendix A

Effort Spend

A.1 Efford Spent

The major part of the document has been produced working togheder and that's the reason way there is not a precise division of hours per sections and per group component.
The following is an approximate extimation of the number of hours of work for each group member:

- Alessandra Pasini: ~2 days;
- Stefano Bagarin: ~2 days;

A.2 Bibliography

- AA 2018/2019 Software Engineering 2 - *Requirements Analysis and Specification Document* - Giovanni Lupi, Davide Maffiola
- AA 2018/2019 Software Engineering 2 - *Design Document* -Giovanni Lupi, Davide Maffiola
- AA 2018/2019 Software Engineering 2 - *Implementation and Testing Document* -Giovanni Lupi, Davide Maffiola