POLITECNICO
MILANO 1863

Computer science and engineering
Software engineering 2 - Project 2019/2020

SafeStreets

ACCEPTANCE TEST DERIVERABLE

Deliverable:	ATD
Title:	Acceptance Testing Document
Authors:	Maldini Pietro , Paone Angelo
Version:	1.0
Date:	21-January-2020
Download page:	https://github.com/pm390/MaldiniPaone
Copyright:	Copyright © 2019, Maldini Pietro, Paone Angelo – All rights reserved

Contents

1 Introduction

1.1 Purpose

We have analyzed the prototype of safestreef released by another group of students. So, in this document we want to describe:

- The identification of the project to be analyzed.
- The installation setup and problems occurred during the process.
- The acceptance tests cases we have considered and the result.
- Other additional aspects to consider of the quality of code and documentation.

1.2 Definitions, Acronyms

1.2.1 Definitions

- User: a person that use the software
- Violation: parking violations which can be notified by users
- App: application software

1.2.2 Acronyms

- RASD: Requirement Analysis and Specification Document.
- DD: Design Document
- ITD: Implementation and Testing Deliverable
- API: Application Programming Interface
- GPS: Global positioning system
- HTTP: HyperText Transfer Protocol
- HTTPS: HyperText Transfer Protocol over Secure Socket Layers

1.3 Revision History

- Version 1.0: First release

2 Project analyzed

The project analized can be found at the git repository : https://github.com/lethanity/PedrozoPerez.
To perform the acceptance tests we took as reference the following documents:

- Requirements Analysis and Specification Document : RASD3.pdf
- Design Document : DD2.pdf
- Implementation and Testing Document : ITD1.pdf

3 Setup

3.1 Back-end setup

In this phase of setup, it is required a software called docker that could be preinstalled. The installation process of this prerequisites should be explained because, during this configuration part we have found some complications: At first time, we've tried to install docker provided from the official web site but, the installation is interrupted, because Docker Desktop requires Windows 10 Pro or Enterprise version 15063 to run, therefore the versions of the operating system of our devices can't guest the software. At second time, after consulting of forum web site we have found on https://github.com/docker/toolbox/releases the installer that could by pass the prerequisites of the operating system version. We have downloaded DockerToolbox-19.03.1.exe and we've executed it and done the full installation setup. After that, we have run the Docker Quickstart Terminal which during his initial setup it have downloaded some files, when finished his initial phase we have copied the docker-compose.yml from package given by the other team (... -> DeliveryFolder -> implementation -> back) and pasted to the folder of Docker Toolbox (C: Program Files -> Docker Toolbox). Subsequently we have run the command "docker-compose up", on docker terminal, (indicated in the section 6 from ITD document) and when docker have finished his set up, we have tested on the browser the link given by the other team http://localhost:8080/auth/ping having a successful result. Immediately we have tried to connect the mobile app but without success and then we have created and executed a file.bat which use "NETSH PORTPROXY" command to create a bridge from virtual machine to subnet and then we have tried again to establish the connection with mobile app having a successful result.

3.2 Front-end setup

In this phase of setup, we have followed the instructions (indicated in the section 6 from ITD document) with a successful result about the installation of mobile app.

3.2.1 Setup conclusion

The ITD1.pdf has a lack of explanation abouth the installation process of the back-end, some instructions should be given in case of a non compatibility of the system with the prerequisite software which should be installed.

4 Acceptance tests

This acceptance tests are based on the use cases located inside the RASD document provided by the other team are respected.

Test	Sign up
Prerequisites	- The user has installed the application on their device. - The application is running
Event flow expected	1. The user presses “Sign up” button. 2. The user fills the fields with the required data. 3. The user presses the “Confirm” button. 4. The system saves the data.
Real event flow	Correspond with event flow expected
Data inserted	- Name: marco - Surname: polo - Username: m@rco - Email: m.polo@mail.com - Password: marcopolo - Repeat password: marcopolo
Output	- The user is successfully registered in the system. - The user is redirected to the login screen
Result	Success
Verify exception	Repeat the process with match exceptions. - The email is already registered in the system. The system warns the user that the email is already in use. Checked: m.polo@mail.com - The username is already registered in the system. The system warns the user that the username is already in use. Checked: m@rco - The user did not fill all the required fields. The system marks the empty fields for the user to fill. Checked: some fields haven’t been filled - The password does not meet the security requirements. The system asks the user to enter another password. Checked: marco

Test	Sign in
Prerequisites	- The application is running - The user is signed up.
Event flow expected	1. The user presses the “Sign in” button. 2. The user fills the “Email” and “Password” fields. 3. The user presses the “Sign in” button. 4. The system verifies the user credentials.
Real event flow	Point 1 isn’t respected: the application shows directly the sing in form
Data inserted	- Email: m.polo@mail.com - Password: marcopolo
Output	- The user is successfully signed into the system. - The user is redirected to the home screen.
Result	Success
Verify exception	Repeat the process with match exceptions. - The user enters a non-matching combination of “email” and “password”. The system shows a warning that “email” and “password” do not match. Checked: Email: m.polo@mail.com, Password: mircopoli - The user did not fill all the required fields. The system marks the empty fields for the user to fill. Checked: some fields haven’t been filled

Test	See profile
Prerequisites	- The application is running. - The user is signed in. - The user is in the home screen.
Event flow expected	1. The user presses the user icon button. 2. The system shows the user information.
Real event flow	Correspond with event flow expected
Output	- The user information is displayed to the user.
Result	Success

<table>
<tr><td>Test</td><td colspan="2">Edit User information</td></tr>
<tr><td>Prerequisites</td><td colspan="2">- The application is running.
- The user is signed in.
- The user is in the profile screen.</td></tr>
<tr><td>Event flow expected</td><td colspan="2">1. The user presses the “edit” button.
2. The user is redirected to the edit profile screen.
3. The user edits the fields they want to change.
4. The user presses the “confirm” button.
5. The system saves the data.</td></tr>
<tr><td>Real event flow</td><td colspan="2">Correspond with event flow expected</td></tr>
<tr><td rowspan="2">Data</td><td>Inserted</td><td>Before</td></tr>
<tr><td>- Name: giuseppino
- Surname: invernizzi
- Username: invernizzi
- Email: peppino@mail.com</td><td>- Name: giuseppe
- Surname: garibaldi
- Username: peppe
- Email: peppe@mail.com</td></tr>
<tr><td>Output</td><td colspan="2">- The user information is successfully updated.
- The user is redirected to the profile screen.</td></tr>
<tr><td>Result</td><td colspan="2">Success</td></tr>
<tr><td>Verify exception</td><td colspan="2">Repeat the process with match exceptions.
- The user did not fill all the required fields. The system marks the empty fields for the user to fill.
Checked: some fields haven’t been filled

- The email is already registered in the system. The system warns the user that the email is already in use.
Checked: m.polo@mail.com

- The username is already registered in the system. The system warns the user that the username is already in use.
Checked: m@rco</td></tr>
</table>

Test	Submit report
Prerequisites	- The application is running. - The user is signed in. - The user is in the home screen. - The user's GPS is active.
Event flow expected	1. The user presses the "Report a violation" button. 2. The user fills the fields with the required data. 3. The user presses the "Take photo" button. 4. The user takes a photo of the vehicle committing the violation. 5. The user repeats steps 3 and 4 as desired until the amount of photos reaches the limit. 6. The user presses the "Confirm" button. 7. The system prompts the user to select a photo where the license plate is clearly identifiable. 8. The user selects a photo. 9. The user presses the "Confirm" button. 10. The system submits the report
Real event flow	Correspond with event flow expected
Data inserted	- Violation type: Crash - License plate: BX405LF - Description: - Photo: figure 1, figure 2
Output	- The report is successfully submitted.
Result	Success
Verify exception	Repeat the process with match exceptions. - The user did not fill all the required fields. The system marks the empty fields for the user to fill. Checked: some fields haven't been filled - The user did not take a photo. The "Confirm" button is disabled. Checked: without photos

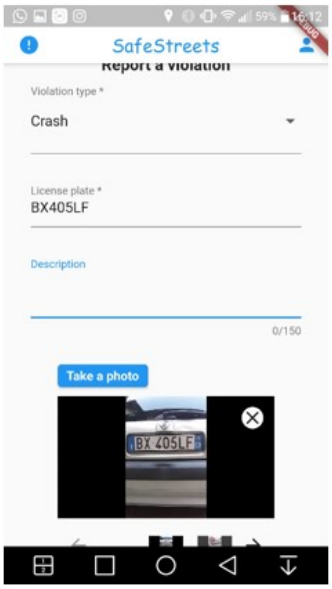

Figure 1

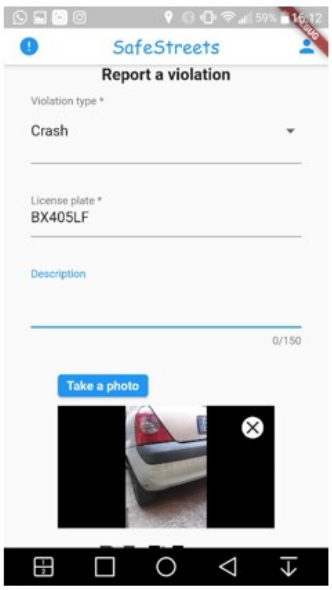

Figure 2

Test	See reports map
Prerequisites	- The application is running. - The user is signed in. - The user is in the home screen.
Event flow expected	1. The user presses the “Reports map” button. 2. The user fills the “from”, “to”, “type” and "search" fields. 3. The user presses the “filter” button. 4. The system shows an interactable map with reports that match the filter.
Real event flow	Correspond with event flow expected, but the form is precompiled with violation type empty, and then the system automatically shows all reports for each type (figure 3)
Data inserted	- From: 21/1/2020 - To: 21/1/2020 - Violation type: Crash
Output	- The system shows the reports map.
Result	Success
Verify exception	Repeat the process with match exceptions. - No reports matching the filter were found. The system shows the empty map. Checked: Parking (Figure 4)

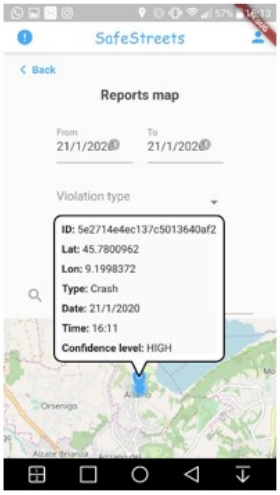

Figure 3 violation type empty

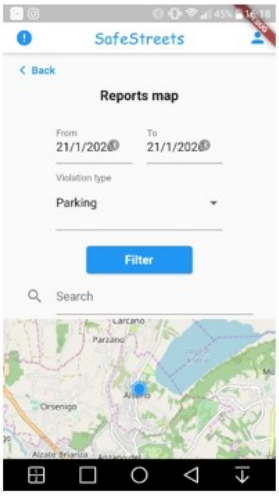

Figure 4

5 Other aspects to consider

This section presents an analysis of the usability of the project. This section should be considered as suggestions for a further release, some ideas and hints to increase the user experience.

5.1 Mobile Application Testing

The application is well structured and provides a good environment to test out the functionalities of the system. Some functions may be changed to simplify testing :

- Save last connected ip address: when closing the application the ip gets deleted and the default ip is automatically inserted and used to connect to the server. This slows down testing of behaviour of the application for example testing the behaviour of closing it and reopening it. If the mobile phone is set so that application in background are stopped the ip must be reinserted to make the application communicate with the right ip.
- Make the Camera choosable: when testing the application having a fixed camera can cause unexpected and undesirable problems. On some phones the primary camera is the internal camera. This caused our testing phase of report making really difficult since we had to make clear photos of the cars and license plates. That was quite a difficult task since internal camera's makes photos with less quality and are usually focused on taking a photo of a person not of a car which can be really big and this reduces too the accuracy of the photo making the testing almost impossible since most photos gets rejected.

An additional point is that the interface is really good looking at first glance but seems to be tested only on a certain specific type of screens. Because some screens shows some really bad looking overflow messages for email addresses of accounts which were already available in the database provided.

5.2 Mobile Application for Users

Here we present some ideas and hints to improve user experience for final users and some improvements to the functionalities to make them more clear and usable:

- Making the License plate input field Auto Capitalizing : The application for making a report requires the user to manually input the license plate of a violation. The input is not accepted if letters are in lower case, but since the input is a normal text input a normal phone keyboard capitalizes the first letter and puts the other letters in lowercare. This can slow down the creation of a report , the input field may set programmatically the capitalization of the letters inside the license plate. Another way to handle this problem could be just modifying to uppercase all the letters in the license plate server side or client side indifferently.
- Taking a photo with a camera of choice : The possibility of choosing the camera to take a photo on a phone is required to make the process as easy as possibile for all possible users on the most different devices as possible.
- Check overflow behaviour of components : A fixed behaviour should be decided for overflowing elements and texts. Making buttons proportional to the screen size and making text size also proportional to the size may be a possible choice, making scrollbars to see overflowing elements is also a possible behaviour to handle overflows.
- Top Left Alert button : the button which gives the possibility to review a photo and increase reliability of a report should be visible only if there is a real report to be reviewed. Otherwise for a user which doesn't know how it works may see it and click it several time getting no feedback from the application. Another possibile way to avoid this unexplained button may be setting a

default screen pop-up telling the user that there are no reports to be reviewed. Explaining this way the meaning of the component in the user interface.

- Expect the unexpected : On the report map screen there is a map where the violations are shown with their id , their date and type. In this screen it is possible using a menu to choose to filter violations. The filtering used properly works really well. it is possible to delete the starting or ending date of the filtering . If a user deletes one date of the two and presses filter the screen shows a loading icon on bottom left and after several minutes of loading no responses seems to arrive from the server. Filter should be disabled if both dates are not available or a default date should be decided for both fields. For example one week before as default starting date and the current date as default final date. Another unexpected behaviour is that after changing filter options and setting a certain violation type the map reload the violations but if there is another change in the violation type and filter is clicked there is no loading and no changes occur on the map, the changes happens only when the map is moved. The filter button should be enabled again after a filter option is changed, this functionality may be useful for municipalities, one municipality may want to know how different violations are located on their territory. Not having the possibility to filter without moving the map can make the experience tedious and less intuitive.

- Don't make the user learn , give them what they already know : On the report map it is not possible to directly move in direction north-south because vertical scrolling moves a menu on the top of the map. This behaviour is really unexpected by the user and user must learn that to move north-south they must first move a little east-west and keep moving on the screen vertically to actually move in the desired direction. Final users can be really difficult to please and making them learn new behaviours which are not compatible with behaviours they already are familiars with can really make the difference from a used and a not used application. To fix this issue a button may be placed on top of the screen to make the menu appear or as an alternative an area which is easily recognizable may be put on top and dragging this area makes the menu appear and disappear. This way the map itself remains untouched while the experience of the user increases and a more natural interaction with the map is possible.

5.3 Interfaces

The design of the interfaces in the application is quite clear and there are no distractions from the actual activity of the user which makes the experience of the user linear and creates less space for unexpected behaviours.
There is unity between the various components in the screens so every interface has its compoents arranged in a clear way and makes the interactions clear to the user.
Functionalities are showed on the main screen with a scaling hierarchy ,showing in big the main functionalities which are making reports and accessing the map with the reports and givin less importance to functionalities like reviewing a report and modifying account.
All the interfaces are have its components placed in such a way to preserve a balance in the screens which improves the user experience.

