POLITECNICO
MILANO 1863

Academic year 2019-2020
Software Engineering 2 Project

SafeStreets

ATD

Acceptance Test Document
(testing the implementation of Morreale, Maddes, Innocente)

Version 1.0

Authors:
Marcer Andrea - 941276
Marchisciana Matteo - 945878
Motta Dennis - 940064

Professor:
Rossi Matteo

1. Introduction

1.1. Purpose

The purpose of this document is to test and validate the project assigned. In particular, referring to the RASD, the DD and the ITD of the project, we have derived the requirements and the goals and gathered apposite test cases in order to check if they are granted by the implementation.

The members of the group we tested are:

- Morreale Federico: 945258
- Maddes Evandro: 945642
- Innocente Federico: 870726

Their repository is available at: https://github.com/fedy97/MorrealeMaddesInnocente

1.2. Document structure

Section 1: The purpose of this document.

Section 2: Describes how we have installed their implementation.

Section 3: Test cases we have performed and the evaluation of requirements and goals.

Section 4: Comments on the quality of documentation, code and application.

Section 5: Effort spent working on this document by group members.

2. Installation setup

We followed the instructions given in the ITD and they provided everything we needed to know to compile, test and run their implementation.
We didn’t encounter any problems during this phase.

2.1. Dependency installation

Their implementation of SafeStreets depends on Flutter which is a framework that allows developers to build apps for multiple operating system with only a single code base. So the first thing we needed to do was to install Flutter on our machines: they provided a link to the official documentation of Flutter that explained everything we needed.
Then they also provided the link for the official Intellij plugin for Flutter. This plugin allows the IDE to interact with Flutter and to take advantage of the Dart programming language.

2.2. Backend installation

No backend installation was needed since it was managed by the team as a serverless component, so their instance of the backend was already running on the cloud.

2.3. Frontend installation

For the frontend we needed to compile the code on our machines with Intellij, equipped with Flutter, and deploy the executable on our mobile devices running both Android or iOS.
As usual with the Apple ecosystem a computer running macOS was needed to deploy the executable on a device running iOS.
For Android, upon request, they provided also the option to install directly the .apk file.

3. Acceptance Test Cases

3.1. Test cases

Here we illustrate the tests we have executed on the Android application to verify the requirements.

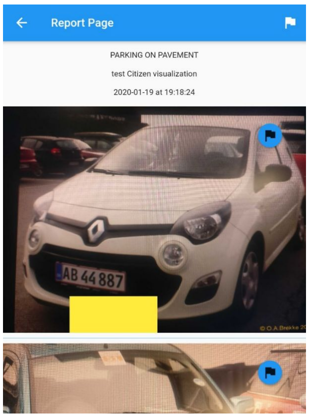
Test ID	**A_1**
Test title	Signup and login of a citizen
Actions	In this test, we try to sign up. First, an already registered email is inserted in input. Then, a new email is used in order to register and the authority ID is left blank. After registering, we reinstalled the application and attempted to log in.
Result	The application correctly refuses to sign us up with the already registered email. Then, it registers the account, only after “accept terms” is checked. The authentication works as well.
Notes	Requirements granted: R1, R2, R4, R5, R6, R9

Test ID	**A_2**
Test title	Signup and login of authorities
Actions	In this test, we try to sign up. A new email is used in order to register and a random ID is used to fill “Authority ID”. After registering, we reinstalled the application and attempted to log in.
Result	The application correctly registers the authority account. Indeed more statistics and sensitive information can be visualized.
Notes	Requirements granted: R1, R2, R3, R4, R5, R6, R9

Test ID	**A_3**
Test title	Reporting a violation
Actions	In this test, we attempt to report a violation. Initially, we try to report an empty violation, by sending it right after it is created. Then we added a random picture, not containing a license plate or a vehicle. We added a description as well, that will serve us later. We also specified the type of violation, choosing it from the dropdown menu.
Result	We verified that sending an empty violation report is not allowed, as the application tells to upload a photo first. The report with a photo attached was submitted as expected. After returning to the initial screen, we are able to see a marker on the map representing the violation that we just submitted.
Notes	Requirements granted: R10, R16, R17, R18, R19, R21

Test ID	**A_4**
Test title	Reporting the same violation
Actions	This is a follow-up to the test **A_3**. Right after reporting the violation, we try to submit another violation in the same place, with the same violation type and within a reasonable time interval.
Result	Upon attempting to submit the second report, an alert shows up saying “this report may already exist, do you want to upload it anyway?”. We can either click “Close” or “Ok”, and we selected the latter. The report shows up as a different violation on the map.
Notes	Requirements granted: - R26: the requirement implies that, upon attempting to submit the report, the application should have shown us the report submitted in test **A_3**. The system however did not show the report. It is not clear from the requirement if we should still be able to submit the second report.

Test ID	**A_5**
Test title	Giving feedback
Actions	In this test we explore the feedback functionality of the application. To verify the requirements, we log in with five different accounts to report the same picture and the same violation.
Result	We found that an account cannot report the same picture and report more than once, and that is as it should be. Upon giving feedback to the picture with the fifth account, the picture was removed from the report. The report however was still visible. Upon giving feedback to the report, the report was removed from the map.
Notes	Requirements granted: R23, R24 Although this was not a requirement, we also verified that a report (or a picture), upon receiving five different feedback, was removed, as it was written in the RASD.

Test ID	**A_6**
Test title	Citizen violations visualization
Actions	Here we try to test the requirements about visualizing violations, picture, hiding sensitive information, ecc. A violation is reported. Three pictures are taken: the first two contain only one license plate, the third contains multiple license plates. The report is sent and then visualized.
Result	The report is visualized on the map and it contains all the information: the description, the timestamp, the violation type and the three pictures. Unfortunately the plates were not covered correctly and we were able to read them clearly.  Report Page PARKING ON PAVEMENT test Citizen visualization 2020-01-19 at 19:18:24
Notes	Requirements granted: R7, R8, R10 R12 and R20 are clearly not granted. After contacting the developers they explained that this depends on the screen resolution of the devices and they were looking for a fix.

Test ID	**A_7**
Test title	Authority violations visualization
Actions	We logged with an authority account and visualized the report done in test **A_6**.
Result	All the information about the report are correctly visualized, with no attempts at obscuring information.
Notes	Requirements granted: R7, R8, R10, R11

Test ID	**A_8**
Test title	Report deletion
Actions	We created a report on January 16th and opened the “My Reports” section on the 19th.
Result	On the 19th we were able to delete the report pressing on the bin icon.
Notes	Requirements granted: - R14 is not completely granted since a report can be deleted even after a day has passed.

Test ID	**A_9**
Test title	Fining a violation
Actions	We logged with an authority account, opened a report and marked it as fined. On another device logged with a different authority account we tried to fine the same report.
Result	On the first account by pressing the exclamation mark on the top right of the screen the report was marked as fined. As we tried to fine the same report with the second account the report was correctly marked as already fined. Moreover now the marker on the map has become red.
Notes	Requirements granted: R7, R27

Test ID	**A_10**
Test title	Violation querying
Actions	We logged with an authority account and performed a complex query and then repeated the same query with a citizen account
Result	With the authority account we filled all the fields in order to retrieve the **A_6** test’s report. We were able to correctly visualize both the report and its data. The same query is performed on a citizen account and the report is correctly retrieved.
Notes	Requirements granted: R7, R8, R11, R28

3.2. Unit and integration tests

3.2.1. Unit tests

Following the same specification used by the team in their ITD we will present here the result of the unit tests also run by us on our machines.

3.2.1.1. Unit tests “Build statistics”

Number of unit tests for this feature: 10.

Test ID	**Result**
“Effectiveness: One report uploaded and not fined”	**OK**
“Effectiveness: One report uploaded and fined“	**OK**
“Effectiveness: Two reports uploaded: one fined and one not“	**OK**
“Daily reports: Zero violation uploaded today“	**OK**
“Daily reports: One violation uploaded today“	**OK**
“Daily reports: One violation uploaded today and one yesterday“	**OK**
“Most committed violation: Zero violation uploaded, no most committed violation“	**OK**
“Most committed violation: One violation uploaded is the most committed“	**OK**
“Most committed violation: Three violations uploaded of the same type are the most committed“	**OK**
“Most committed violation: Two violations uploaded of the same type and one of another type, the two ones are the most committed“	**OK**

3.2.1.2. Unit tests “Violation query”

Number of unit tests for this feature: 16.

Test ID	Result
“One report respects all the bounds of the query“	OK
“Two report respects the bounds of the query (no bounds on city)“	OK
“Two report respects the bounds of the query (no bounds on violation)“	OK
“Two report respects the bounds of the query (no bounds on timeTo)“	OK
“Two report respects the bounds of the query (no bounds on timeFrom)“	OK
“Two report respects the bounds of the query (no bounds on timeFrom and timeTo)“	OK
“Two report respects the bounds of the query (no bounds on violation and timeTo)“	OK
“Two report respects the bounds of the query (no bounds on violation and timeFrom)“	OK
“Two report respects the bounds of the query (no bounds on violation and timeFrom and timeTo)“	OK
“Two report respects the bounds of the query (no bounds on city and timeTo)“	OK
“Two report respects the bounds of the query (no bounds on city and timeFrom)“	OK
“Two report respects the bounds of the query (no bounds on city and timeFrom and timeTo)“	OK
“Two report respects the bounds of the query (no bounds on city and violation)“	OK
“Two report respects the bounds of the query (no bounds on city and violation and timeTo)“	OK
“Two report respects the bounds of the query (no bounds on city and violation and timeFrom)“	OK
“Two report respects the bounds of the query (no bounds on city and violation and timeFrom and timeTo)“	OK

3.2.1.3. Unit tests “Report visualization on map”

Number of unit tests for this feature: 6.

Test ID	Result
“No violation uploaded, no violation on the map“	OK
“Two violations not fined today, two violations on the map“	OK
“Two reports not fined yesterday, zero violation on the map“	OK
“Report uploaded yesterday, zero violations on the map“	OK
“Two reports, only one today, one violation on the map“	OK
“Two reports on same position, two reports on the map“	OK

3.2.1.4. Unit tests “Add a new report”

Number of unit tests for this feature: 3.

Test ID	Result
“Similar report doesn’t exist“	OK
“Report with same location and day“	OK
“Report with same location and violation“	OK

3.2.1.5. Unit tests “User feedback and action”

Number of unit tests for this feature: 3.

Test ID	Result
“Add the user to the list of violation feedback sent“	OK
“Add a feedback to a picture“	OK
“Delete one picture from the report to be sent“	OK

3.2.2. Widget tests

This type of test was used by the team to test the UI of their implementation in an automatic way. Here it is reported the results of these tests when run on our machines.

3.2.2.1. Widget tests “Login page”

Number of widget tests for “Login page”: 3.

Test ID	**Result**
“Email, password and button are found“	**OK**
“It validates empty email and password“	**OK**
“It calls sign in method when email and password is entered“	**OK**

3.2.2.2. Widget tests “Sign up page”

Number of widget tests for “Sign up page”: 3.

Test ID	**Result**
“Email, password, confirm password and button are found“	**OK**
“It validates empty fields“	**OK**
“It calls sign up method when email and password is entered“	**OK**

3.2.3. Integration tests

Number of integration tests: 8.

Test ID	Result
“The user tries to login with an unregistered email and password“	OK
“The user tries to register himself without accepting terms and conditions“	OK
“The user registers himself with a new email and password”	OK
“The user tries to upload a report without any image“	OK
“The user checks that there are no reports associated with his account“	OK
“The user queries all the reports uploaded in Milan“	OK
“The user gives a feedback to the first report queried“	OK
“The user tries to give a second feedback to the first report queried“	OK

3.3. Summary for requirements and goals

GREEN: Granted
ORANGE: Not completely granted
RED: Not granted
BLUE: Not testable
GRAY: Not implemented

3.3.1. Requirements

R1. **Unregistered user cannot use SafeStreets.**
Granted: unregistered users can only sign up.

R2. **In signing up, users must provide an email and a password.**
Granted: the system asks for them when you sign up.

R3. **In signing up, an authority must also provide an ID.**
Granted: the app has an appropriate field called “ID authority” in which to insert the ID.

R4. **In signing up, users must accept terms & conditions.**
Granted: the user must check the box “accept terms” before signing up.

R5. **SafeStreets identifies a user by its email.**
Granted: the system refuses to register an account with an email already in use.

R6. **SafeStreets collects user data in its database.**
Granted: the system recognized the email and password of a previously registered account also on different devices, so they are reasonably saved on a database.

R7. **SafeStreets stores all reports in its database.**
Granted: the user can visualize all his reports and that of other people on the map, so they are reasonably saved on a database.

R8. **SafeStreets retrieves reports from its database.**
Granted: the user can visualize his reports and that of other people on the map.

R9. **SafeStreets authenticates a user when tries to log in.**
Granted: the system recognized the email and password of a previously registered account.

R10. **Users can view the reports in the map.**
Granted: the user can visualize his reports on the map.

R11. **Authorities can see all the sensitive information contained in a report.**
Granted: the system shows not-blurred images for authority accounts.

R12. **Citizens cannot see sensitive information in a report.**
Not completely granted: the system sometimes fails to obscure sensitive information, such as license plates (see also R20).

R13. **Users can view unsafe areas in the map.**
Not implemented.

R14. **Users can delete reports they submitted within a day.**
Not completely granted: you can delete reports, but you can delete them even after a day has passed by.

R15. **Users can edit reports they submitted within a day.**
Not implemented.

R16. **Users can upload valid reports.**
Granted: the app allows to upload only valid reports.

R17. **A report must contain at least one image.**
Granted: the app does not allow you to send a report without at least one image.

R18. **A report must indicate the type of violation.**
Granted: there is a dropdown menu from which a violation type can be selected to select the type of violation.

R19. **A report is valid if and only if R17 and R18 are satisfied.**
Not testable: this is not a requirement since it is not something the system can enforce.

R20. **Images containing sensitive information (like license plates) must be blurred by the system. (For now, you need a high resolution of the device, still to be fixed)**
Not completely granted: the system sometimes fails to blur license plates. The developers explained that this depends on the resolution of the screen and that they were looking for a solution.

R21. **The system must notify the user if the report submitted is not valid.**
Granted: upon attempting to submit a report, the system blocks you and tells you to attach a photo.

R22. **For each type of statistics there must be at least user from which SafeStreets takes data.**
Not testable: we weren't able to find a way to test this requirement.

R23. **A user can access the feedback area for each report.**
Granted: a user can see a flag (used to give feedback) on the report and on each picture of the report.

R24. **A user can select negative feedback for each report.**
Granted: upon pressing of a flag button, the app sends the feedback.

R25. **Authorities can see suggestions for possible interventions.**
Not implemented.

R26. **During adding a new report, if there is a report of same type, in the same position and at the same date, SafeStreets shows the report stored to the user if the violation is the same.**
Not completely granted: upon attempting to send a report of the same type, position and date, the app tells you that the report may already exist, and if you want to upload it anyway. So the app doesn't show the report stored but only tells that there is a report that could be referring to the same violation.

R27. **Authorities can mark a report as fined.**
Granted: upon pressing the "fine" button, the system notifies that the reports was fined.

R28. **User can query SafeStreets to achieve a specific report.**
Granted: you can use the functionality "Query Report".

R29. **SafeStreets retrieves incidents from municipality database.**
Not implemented.

3.3.2. Goals

[G1] The application will allow users to upload pictures of the traffic violations, including pictures, date, time, classification, and optionally a textual description.
Requirements: R1, R2, R3, R4, R5, R6, R9, R14, R15, R16, R17, R18, R19, R20, R21, R26

[G2] The application will allow users to view a map of the violations registered in the last day.
Requirements: R1, R2, R3, R4, R5, R6, R7, R8, R10, R11, R12, R17, R18, R28

[G3] The application will allow users to see the violations uploaded by other users; however, citizens won't be able to see sensitive pictures, while authorities will be allowed.
Requirements: R1, R2, R3, R4, R5, R6, R7, R8, R11, R12, R17, R18, R19, R20

[G4] Users will be able to query the system to get all the reported violations that correspond to some temporal, geographical or categorical parameters.
Requirements: **R5**, **R6**, **R7**, **R8**, R12, **R17**, **R18**, **R19**, **R22**, R29

[G5] Users can give feedback about violations uploaded by other users to SafeStreets.
Requirements: **R1**, **R2**, **R3**, **R4**, **R5**, **R6**, **R7**, **R8**, **R23**, **R22**, **R24**

[G6] If SafeStreets can get the information about accidents by the municipality, it will give the possibility to merge them with its data to identify potentially unsafe areas.
Requirements: **R7**, **R8**, R13, **R16**, **R17**, **R18**, **R19**, **R22**, R29

[G7] If SafeStreets can get the information about accidents by the municipality, it will give the possibility to merge them with its data to suggest possible interventions.
Requirements: **R5**, **R6**, **R7**, **R8**, **R17**, **R18**, **R19**, R20, **R22**, R25, R29

[G8] The system will use the information about violations, accidents and fines that will collect by both users and authorities to build statistics.
Requirements: **R5**, **R6**, **R7**, **R8**, **R17**, **R18**, **R19**, **R23**, **R24**, **R22**, **R27**, R29

4. Additional Notes

Positive remarks we want to point out:

- All their Dart codebase is well commented, especially the code of the managers where all the main features are implemented.
  In fact, in the classes of the managers there were comments about what each class does, comments about what each method does and comments for some specific lines of code.
- The team was active and helpful in answering any concerns we had.

Other notes about small things we think could have been done better:

- We think the UI in some parts is not really user friendly: that is the “flag” buttons, for which it isn’t immediately evident that they mean to be used for reporting violations that are not correct. Adding a dialog after pressing the button would make it more clear for example. The same remark goes for the “fine” button.

5. Effort spent

Matteo Marchisciana		
Date	Task(s)	Time
16/01	Test cases	2 hr
18/01	Test cases	1 hr
19/01	Test cases	1 hr
21/01	Introduction and final review	1 hr
Total:		5 hr

Andrea Marcer		
Date	Task(s)	Time
15/01	Document creation and structuring	2 hr
17/01	Structuring	1 hr
19/01	Test cases	1 hr 30 min
20/01	Test cases	1 hr 30 min
21/01	Introduction and final review	1 hr
Total:		7 hr

Dennis Motta		
Date	Task(s)	Time
18/01	Unit tests	1 hr 0 min
20/01	Unit tests; Widget tests; Integration tests	2 hr 15 min
21/01	Installation setup; Additional notes; Final review	3 hr 0 min
Total:		6 hr 15 min