

Software Engineering 2 - Mandatory Project

AY 2019/2020

Immagine che contiene serviziodatavola

Descrizione generata automaticamente

RASD

Version 1.0 – 10/11/20

Maserati Alessandro - 944593

**Table of Contents**

Table of Contents

1. **Introduction**
   1. General Purpose
   2. Goals
   3. Scope
      1. World and shared phenomena
   4. Definitions, acronyms and abbreviations
      1. Definitions
      2. Acronyms
      3. Abbreviations
   5. Revision history
   6. Reference Documents
   7. Document Structure
2. **Overall Description**
   1. Product perspective
   2. Product functions
   3. User characteristics
   4. Assumptions, dependencies and constraints
3. **Specific Requirements**
   1. External Interface Requirements
      1. User Interfaces
      2. Communication Interfaces
   2. Functional Requirements
      1. Use Case Diagram
      2. End User Scenarios and Use Cases
      3. Authority Scenarios and Use Cases
   3. Performance Requirements
   4. Design Constraints
      1. Standard compliance
      2. Hardware limitations
   5. Software System Attributes
      1. Reliability
      2. Availability
      3. Security
      4. Maintainability
      5. Portability
4. **Formal Analysis using Alloy**
5. **Effort Spent**
6. **References**

**1. Introduction**

1.1 General Purpose

Safe Streets is a software application with the main aim of providing end users with the possibility to report to authorities traffic violations (in particular parking violations). Another aim of the system is allowing both end users and authorities to get information mining the received notifications.

The deepest purpose of the application is to provide authorities a way to know where road works (clearer traffic signals) or municipality interventions are necessary with the aim of improve street safety (in particular from the point of view of pedestrians and bikers).

A detailed description of the functionalities offered to end users and authorities is provided in Section 2.2.

1.2. Goals

* [G1] – Allow end users to report to the authorities traffic violations.
* [G2] – Allow end users and the authorities to visualise which are the streets with the highest frequency of violations. dividere
* [G3] – Allow authorities to know the vehicles that commits one or more violations. dividere
* [G4] – Allow authorities to delete incorrect violations reported by end users.

1.3. Scope

Safe Streets is a software application thought to improve road safety. Using Safe Streets, common people can notify authorities when they see traffic violations only taking and sending one or more pictures of that and specifying the type of violation (e.g. abusive bike lane or sidewalk parking, double parking, parking in disabled reserved spaces…) from their mobile devices. Authorities, on the other side, after the run of an algorithm that reads the license plates from the pictures, can check the reported violations and delete from the “Report list” the incorrect ones (e.g. misunderstanding of traffic signals by the user, special permits, special events occurring in that areas…). This is done before data are saved into the database, to be sure that only the correct reports can modify the mined information (e.g. the “Violation map”).

Contacts between stakeholders and some authorities, in order to find some public institution to start the project with, have already been made before the drawing up of this document (Section 2.4, constraint [C4]). Without this agreement, the development of the application can be useless, because of the absence of enabled people to verify the correctness of the violation reports.

From their devices, both end users and authorities can also see which are the streets and areas with a high number of traffic violations. In addition, authorities are allowed to see a list of the vehicles (identified by the license plates) which commit traffic violations. This functions are also available with exploiting a filtered research by type of violation, street or date (see Section 3.1 and 3.2).

1.3.1. World and shared phenomena

The following table illustrates some of the world and shared phenomena related to the use of Safe Streets, referring to the Jackson & Zave distinction.

|  |  |
| --- | --- |
| **Phenomenon** | **Shared or not** |
| End user sees a traffic violation | No |
| End user takes a picture of the violation | Yes |
| End user fills and sends the “Traffic violation form” | Yes |
| End user/Authority searches the highlighted streets | Yes |
| Authority wants to know the vehicles that committed the most violations | No |
| Authority investigates the list of vehicle plates | Yes |
| Authority checks a violation report | Yes |
| Authority deletes a certain violation report | Yes |

The 2nd phenomenon “*End user takes a picture of the violation”* is considered to be shared because the picture is taken opening the camera from the application (when the user chooses to report a traffic violation).

1.4. Definitions, acronyms and abbreviations

1.4.1. Definitions

* *Authority*: public institution related to street safety (e.g. municipality, local police).
* *End user*: people (unrelated with authorities) using Safe Streets application with the aim of report traffic violations and know the streets where the most violations occur.
* *Customer*: a “general” user of the SafeStreets service, can be an end user or an authority.
* *Traffic violation form*: it’s the form that an end user must fill in the app on his device when he wants to notify a violation.
* *Violation report list*: the queue containing the violation reports which have not yet been checked by authorities. Obviously visible only to authorities.
* *My Reports*: section visible on End User mobile app in which all the reports made by that end user are shown, underlying the current state of each report (it can be Unchecked, Confirmed or Rejected depending on the decision of the authority).

1.4.2. Acronyms

* RASD: Requirement and Analysis Specification Document
* GPS: Global Positioning System

1.4.3. Abbreviations

* [Gn]: n-th goal
* [Dn]: n-th domain assumption
* [Cn]: n-th constraint
* [Rn]: n-th requirement

1.5. Revision History

* 10/11/19: Version 1.0
  + First Release

1.6. Reference Documents

* Specification Document “*SafeStreets mandatory Project Assignment*”

1.7. Document Structure

This RASD is composed by 6 macro sections:

**Section 1** is an introduction illustrating the general purpose, the goals and the scope (underlining the world and shared phenomena) of Safe Streets application.

**Section 2** is a more detailed overall of the application. The first subsection *Product perspective* includes further details of the shared phenomena, class diagrams and state-charts. Furthermore, a summary of the major functions of the system and the description of the user’s characteristics are shown. Finally, constraints and domain assumption are listed to have a complete view of the world domain of the application.

**Section 3** is

**Section 4** is

**Section 5** gives information about the time spent to realize the entire document.

**2. Overall description**

2.1 Product perspective

- vincoli delle foto, magari qualcosa sul processo di checking

UML Class diagram

State chart diagrams

2.2 Product functions

In this sub-section are listed and described in detail the functions that SafeStreets users must be allowed to exploit. The details of these functions can be deeply explored in Section 3.1.1, where the mockups of the app are shown.

2.2.1 Registration and login

This function is exploited by both end users and authorities. They must be able to register to the system, filling a registration form with personal data. While end users are identified by their fiscal codes, authorities must specify their ID number. After the registration, customers must be able to login with a username and a password, both chosen during the registration phase.

2.2.2 Reporting a traffic violation

When the end user sees a traffic violation, he must be able to notify it using the SafeStreets application. In particular, he has to fill the “Traffic violation form” taking one (or more pictures, depending on the situation, in order to take every necessary detail) of the violation (some constraint for the validity of the report must be respected, e.g. the license plate and the background of the vehicle must be clearly visible) and specifying the type of violation. The GPS position of the violation and the current date and hour are automatically reported by the system, exploiting a map service.

After the user sends the “Traffic violation form”, the report goes into the queue of unchecked reports, visible only to authorities app.

2.2.3 Visualizing the highlighted streets

This function must be exploited by all users. The user must be able to see on a map which streets are places of the most violation. Different colours can be used, for instance a street where more than 10 traffic violations per day are notified is highlighted with red, another with only 3-5 violation with yellow. Furthermore, the user is allowed to search on the map a particular street. Specifying some filters is also possible to generate the map created using only a part of the reported traffic violation (e.g. only the violations occurred on a specific date or the ones with a specific type). Also, this function is possible because of the exploiting of a map service.

2.2.4 Visualizing the most notified vehicles

The following function is reserved to authorities because of privacy rules. Authorities must be able to see which vehicles, identified by license plates, commits traffic violations. Using different filters, authorities are allowed to see the list of vehicles grouped by kind of violation committed, date or area, to mine information with the aim of organize police controls or road working to improve streets safety.

2.2.5 Deleting a traffic violation instance

Authorities must be allowed to check every violation reported by end user in order to delete the ones who don’t satisfy certain legal constraints (described in Section …). This can be done only clicking a button during the visualization of the report. After that, the response is notified to the end user who made the report.

2.3 User characteristics

The actors of the application are the following:

* End users: “normal” people who is allowed to notify authorities when traffic violations occur. This users’ knowledge about road issues is assumed to be “limited” (errors in judgement due to special causes are possible). In order to ensure a proper service, as described in the previous section, violation reports received from these actors must be checked by authorities before their insertion into the SafeStreets database.
* Authorities: municipality, local police, traffic police or any other public institution registered to the system with the aim of mine information from the end users’ reports. Authorities are verified by the system and identified by a unique ID number.

2.4 Assumptions, dependencies and constraints

Domain assumptions:

[D1] – To all authorities registered to the system, taking decisions (approve or delete) about the correctness of the violation reports is allowed.

[D2] – Every authority registered to the system is assumed to have the necessary “traffic rules” knowledge and to be able of taking decisions consulting vehicle and traffic authorities data (vehicles special permissions…)

[D3] – Each authority is uniquely identified by an ID Number (it must be used during the registration process).

[D4] – Every end users’ report is checked in less than 24 hours form the notification.

[D1] is taken with the aim of simplify the problem and the drawing up of this document. Obviously, in real world, it can’t be possible because only a part of authorities is enabled to take decisions about traffic violation (e.g. local police or traffic police, not administrative authorities of municipality).

With regards to [D2], the information can be taken from vehicle registration authorities or from the municipality.

Constraints:

[C1] – The device used to run SafeStreets mobile app must be able to take pictures.

[C2] – The device used to run SafeStreets mobile app must be able to provide the GPS position of the end user.

[C3] - The device used to run SafeStreets mobile app must be connected to internet.

[C4] – During the initial phase of the development of this software application, stakeholders have concluded agreements with some authorities, who agreed to register to the system during the launch of the app.

**3. Specific Requirements**

3.1 External Interface Requirements

3.1.1 User Interfaces

The following mockups represent the main screenshots of the mobile application to be (illustrating both end users and authorities’ functionalities):



*End User and Authority login interface*



*End User main menu*

*Traffic violation map*



*End User violation form*



*Authority reports check interface*



*Authority main menu*



*Violation report detail*



*Authority most notified vehicle list interface*

3.2 Functional Requirements

3.2.1 Use Case Diagram

3.2.2 End User Scenarios and Use Cases

**Scenarios**

1. Arthur is a Politecnico di Milano’s professor and he is also a biker. He loves riding his bike every morning going to work. During the new academic year, Bob has noticed that many vehicles are parked in the middle of bike lanes. Tired of the slalom between cars that he must do every morning, he decides to use SafeStreets app to notify authorities all the violations he sees, with the hope of more frequent traffic police controls.
2. Polly is an old woman’s granddaughter and she loves too much her granny. Unfortunately, her granny is disabled, so, every Thursday she has to take her to the supermarket. Reached her destination, she sees that the only place in front of the supermarket reserved for people with disabilities is filled by a vehicle not showing any disability badge. She can’t tolerate such a lack of respect, so she takes her smartphone and send to authorities a picture of the violation.
3. John, Arthur’s brother, loves cycling too. He wants to know how many traffic violations occur in the route from home to his new work office. Arthur told him that SafeStreets app allows users to know exactly what John is in search of. He downloads the app on his smartphone and fortunately discover that all the streets in his route are safe.
4. Finn is a regular user of SafeStreets service. During the last week, he has notified five traffic violations. Now, he wants to know whether his reports have been checked and approved from authorities. From his mobile phone, he opens the SafeStreets app, he selects his personal area available in My Reports page and find that all his reports have been checked. Good job Finn!

**Use Cases**

|  |  |
| --- | --- |
| **Name** | Sign Up |
| **Actor** | End User |
| **Entry conditions** | The End User has opened SafeStreets app on his mobile device. |
| **Events flow** | 1. The user chooses the “End User Sign Up” option. 2. The user fills the Registration Form with all the required information. 3. The user chooses the confirmation option. 4. The system saves the data. |
| **Exit conditions** | The user is registered and now he can login to the system using the specified credentials. |
| **Exceptions** | * The user doesn’t fill all the mandatory fields. * The user was already registered (the fiscal code or the email inserted are already in the database).   In all these cases the system notifies an Error Message and returns to the Sign-Up page. |

|  |  |
| --- | --- |
| **Name** | Login |
| **Actor** | End User |
| **Entry conditions** | 1. The user has opened SafeStreets app on his mobile device. 2. The End User is already registered to the system. |
| **Events flow** | 1. The user chooses the “Login” option. 2. The user fills the login form with his email and password. 3. The user chooses the confirmation option. |
| **Exit conditions** | The user is logged in and can exploit SafeStreets functionalities. |
| **Exceptions** | 1. The user enters the wrong email. 2. The user enters the wrong password.   In both cases, the system warns the user and suggests to re-insert the wrong fields. |

|  |  |
| --- | --- |
| **Name** | Report a Traffic Violation |
| **Actor** | End User |
| **Entry conditions** | 1. The user has opened SafeStreets app on his mobile device. 2. The End User is already logged into the system. |
| **Events flow** | 1. The End User chooses the “Report Traffic Violation” option. 2. The End User chooses the “Take a picture” option. 3. The End User takes one or more pictures exploiting his device’s camera. 4. The End User enters the description of the traffic violation. 5. The End User chooses the Send option. |
| **Exit conditions** | The traffic violation is registered, and authorities can check it. The Traffic violation is also visible in user’s “My Reports” section. |
| **Exceptions** | The End User tries to send the report having not taken any picture or entered the description. An error is notified by the system. |

|  |  |
| --- | --- |
| **Name** | Search a street on the “Traffic Violation Map” |
| **Actor** | End User |
| **Entry conditions** | 1. The user has opened SafeStreets app on his mobile device. 2. The End User is already logged into the system. |
| **Events flow** | 1. The End User chooses the “Traffic Violation Map” option. 2. The End User enters a street name. 3. The End User chooses the “Search” option. |
| **Exit conditions** | The street searched by the End User is visualized on the map. |
| **Exceptions** | The street name entered doesn’t exists. In this case a Not Found Error is notified. |

|  |  |
| --- | --- |
| **Name** | Visualize personal violation reports |
| **Actor** | End User |
| **Entry conditions** | 1. The user has opened SafeStreets app on his mobile device. 2. The End User is already logged into the system. |
| **Events flow** | 1. The End User chooses the “My Reports” option. |
| **Exit conditions** | The list of the user’s previous reports is visualized. |
| **Exceptions** | The End User hasn’t notified a traffic violation yet. The user is informed of this with a warning. |

**Sequence Diagrams**

3.2.3 Authority Scenarios and Use Cases

**Scenarios**

1. Oswald is an employee of Milano’s municipality. His boss told him that a significant amount of money is available for some new road works with the aim of improve the streets safety of the municipality. He also entrusted Oswald to select some streets which streets need an improvement of traffic signals for the safety of bikers. Oswald opens SafeStreets app and searches on the available Traffic Violation Map the streets in which the most of bike lanes illegal car parks occur. Then, he notifies his boss the result of his research.
2. Thomas is a diligent traffic policeman. He must organize the control routes of his colleagues for the next week. So, he searches on the Traffic Violations Map of SafeStreets application where the most of illegal car-parking (specifying this type of violation) are reported with the aim of generate many parking tickets. Now, Thomas exactly know where to send his colleagues.
3. The Milano’s traffic police must deliver a statistical survey about the vehicles that commits illegal disabled reserved car parks. Grace, the survey manager, uses SafeStreets application in order to merge the information mined from the traffic tickets generated in the last year with the information taken from SafeStreets end users. In particular, she uses the available list of vehicles license plates filtered by type of violation.
4. Michael, a municipality employee, is the responsible organizer of a gastronomic event that took place last week in the city of Milan. This event was the cause of many complaints from people who lives in that area. Michael wants to know whether the events causes some traffic disease in order to be prepared for the next year edition. He uses SafeStreets search function and he discovers that more than forty different vehicles were notified in the specified street in the date of the event with cause: illegal car parking on bike lanes and sidewalks.

**Use Cases**

The Use Cases related to the registration and login processes are not shown because they are identical to those related to the End User, except for the fields to fill during registration phase (more information to verify the identity are required) and for the field required to login (not the email but the authority ID Number).

|  |  |
| --- | --- |
| **Name** | Traffic violation report check |
| **Actor** | Authority |
| **Entry conditions** | The Authority is logged into the system. |
| **Events flow** | 1. The Authority chooses the “Check violation reports”. 2. The Authority chooses a report to check from the list. 3. The Authority chooses whether to confirm or reject the report. |
| **Exit conditions** | The selected report is confirmed and stored into the database or rejected. Now, on the user’s application, the choice of the authority is visible. |
| **Exceptions** | No exceptions. |

|  |  |
| --- | --- |
| **Name** | Search for the violations with a specific type (on the map) |
| **Actor** | Authority |
| **Entry conditions** | The Authority is logged into the system. |
| **Events flow** | 1. The Authority chooses the “Visualize traffic violations map” option. 2. The Authority digits the specific type of violation in the “Type box”. 3. The Authority chooses the “Search” option. |
| **Exit conditions** | The “Traffic violation map”, related only to the specified type of violation, is shown. |
| **Exceptions** | Violations with the specified type of violation are not present in the system, a Not Found Error is notified. |

|  |  |
| --- | --- |
| **Name** | Search for the traffic violations committed by a specific vehicle |
| **Actor** | Authority |
| **Entry conditions** | The Authority is logged into the system. |
| **Events flow** | 1. The Authority chooses the “Most notified vehicles” option. 2. The Authority digits the specific license plates in the corresponding search box. 3. The Authority chooses the “Search” option. |
| **Exit conditions** | The reports related to the specified vehicle are shown in a list. |
| **Exceptions** | The specified vehicle was never the subject of a traffic violation report, a Not Found Error is notified. |

|  |  |
| --- | --- |
| **Name** |  |
| **Actor** | Authority |
| **Entry conditions** |  |
| **Events flow** |  |
| **Exit conditions** |  |
| **Exceptions** |  |

**Sequence Diagrams**