

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
   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 the end users to report to the authorities traffic violations.
* [G2] – Allow the end users and the authorities to visualise which are the streets with the highest frequency of violations.
* [G3] – Allow the authorities to know the vehicles that commits one or more violations.
* [G4] – Allow the 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. Without this agreement, the development of the application can be useless, because of the absence of someone enabled 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.

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 deletes a certain violation report | Yes |

The 2nd phenomenon “*End user takes a picture of the violation”* is shared because the picture is taken using 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.

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 the 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.

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. Also this function must be exploited with the aid 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 the 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.

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] -

[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).

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.

**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):

*Traffic violation map*



*End User violation form*



*Authority main menu*



*End User and Authority login interface*



*End User main menu*



*Authority most notified vehicle list interface*



*Authority reports check interface*



*Violation report detail*

3.1 Functional Requirements