1 INTRODUCTION

1.1 Purpose

This document aims to serve as a basis for the organization of the project that will lead to the development of SafeStreets by presenting the requirements. Secondly, once the product has been completed, it will be used to evaluate the final result and judge its loyalty with respect to what is set in these pages. In conclusion, the document is intended for those who actually will contribute to the creation of the software (Project Managers, System and Requirement Analysts, Developers, Testers and so on) but also to users who will benefit from it.

1.1.1 Description of the given problem

Nowadays the number of cars on the streets of our cities is constantly increasing and for this reason it is necessary to find new solutions in order to guarantee that they can be parked legally. Nevertheless, it often happens, due to lack of parking zones or laziness of drivers, to find cars parked in an unacceptable way and the situation sometimes becomes difficult to manage by the authorities. In this context SafeStreets operates: the software permits to send some information like photos, date and time to authorities but this is not all. In addition, SafeStreets mixes the information given by its users with some data computed by the system in order to have a complete description of the violation, in fact the report is accompanied by the licence plate, the location and the type of violation. The retrieval of the first two data is managed by the software while the third is entered manually by the user and the result is saved in the database.
Moreover, the product has another interesting feature that gives the opportunity to users to navigate through data stored anonymously in the system visualizing correlations between places and the number of violations through maps, diagrams and other representations.
Obviously not everything is accessible by anyone but in some cases permission is required to access certain data.
In conclusion there is a very innovative function, possible thanks to the collaboration of the municipality, that allows SafeStreets to propose possible solutions to solve problems due to repeated violations that occur in certain areas of the city that for these reasons are dangerous.

1.1.2 Goals

- G1: Users must be able to visualize statistics regarding the zones where violations occur more frequently.
- G2: Users must be able to visualize statistics regarding the vehicles that commit the most violations.
- G3: Users must be able to visualize statistics regarding the days and time at which certain zones are more exposed to violations.
- G4: Users and authorities must be able to mine information collected by the system with a visibility level based on their role.
- G5: Municipality should receive solutions that suggest how to solve the problems of violations in problematic zones.
- G6: Users should be able to report violations anonymously.
- G7: Municipality should be able to have access to all the anonymous segnalations sent to SafeStreets.
- G8: Municipality should be able to provide its own data to SafeStreets.

1.2 Scope

SafeStreets is a service intended for common citizens who care about traffic rules and who want to play their part for the commonweal.
In the following we identify the machine as the system composed by the web app and the mobile app, while the world is being composed by drivers, municipality and the users of the app.

World phenomena

- Collection of municipality data into database
- Traffic violations

Machine phenomena

- Traffic plate recognition
- Position retrieval from GPS
- Collection of data into database
- Data anonymization
- Processing of data to be presented to users

Shared phenomena

- Violation report to SafeStreets using the mobile app via a picture
- Data mining and visualization by users
- Suggestions of interventions
- Linking SafeStreets and municipality data

1.3 Definitions, Acronyms, Abbreviations

Definitions

- Driver: the one who makes the violation.
- User: who benefits from the service offered by SafeStreets by sending photos of the violations or accessing the data saved in the system.
- Municipality: public authority who take advantage of the service offered by SafeStreets and eventually collaborate with it providing supplementary information.
- Authority: used as a synonym for municipality.
- Violation: corresponds to parking in a no-parking zone.
- Solution: report sent by SafeStreets to the municipality after an accurate evaluation of violations occurred in a certain area. Its aim is to reduce the number of these types of events.
- Zone: the basic unit in which it is possible to divide a city.
- Anonymous segnalation: segnalation saved in the database which presents no reference to the specific person who sent the same notification.

Acronyms

- RASD: Requirement Analysis and Specification Document
- DB: DataBase
- GPS: Global Positioning System
- OCR: Optical Character Recognition

Abbreviations

- Gn: n^{th} goal
- Rn: n^{th} requirement

- Dn: n^{th} domain assumption

1.4 REFERENCE DOCUMENTS

- Specification document: “SafeStreets Mandatory Project Assignment”
- Alloy documentation: http://alloy.lcs.mit.edu/alloy/documentation.html

1.5 DOCUMENT STRUCTURE

This RASD is divided into five chapters and each one of these have a specific function that is briefly described below:
The first chapter gives an overview of the problem indicating what the document proposes to do describing in principle the operation of the software, listing the goals to be reached and finally clarifying some details that allow a better understanding. This is meant to be the entry point for developers and other technical people, while users can match their expectations with the presented goals.
The second chapter explains in an accurate way important details that permit to understand better the overall functioning of the system. Here the reader can also find the descriptions of the main functions of the software, the actors and the list of assumptions, dependencies and constraints to be considered. The latter ones serve as limitations for the developer’s options.
The third chapter contains very important details for the development team, in fact it describes the functioning of all the interfaces of the system. Moreover, it proceeds by defining the functional requirements using the use case and sequence diagrams. Finally, non-functional requirements are listed through Performance Requirements, Design Constraints and Software System Attributes.
In the fourth chapter the results obtained through the formal analysis using Alloy are reported but first the modeling is justified specifying what it demonstrates and why it is important for this specific problem.
Finally, the fifth chapter shows schematically the effort spent for the realization of this document.

2 OVERALL DESCRIPTION

2.1 Product perspective

The service offered by SafeStreet is designed to be accessible both from a mobile app and through a website. The only difference is that from the smartphone the user can either send reports or consult data while in the web version only the latter option will be accessible.
In order to let users send a report, SafeStreets will exploit the camera of the device integrating a feature which allows to take pictures only at the moment, preventing fake photos. The GPS of the device will be used to infer the street where the picture is taken and the licence plate is obtained through an OCR algorithm. Finally, the user is asked to select the type of the violation. All these data are then stored into the DB, whose access depends on the privilege of the user. SafeStreet can also exploit data of municipality generating statistics on which are the most problematic zones. Based on the collected data and on already tested solutions, SafeStreet will recommend the most appropriate intervention to face a certain type of violation in a given context.
In order to have a better view of the system as a whole, the following is a very simple class diagram that represents a model of the application domain. Note that the classes do not report all the attributes that characterize them but only the main ones that favor understanding.

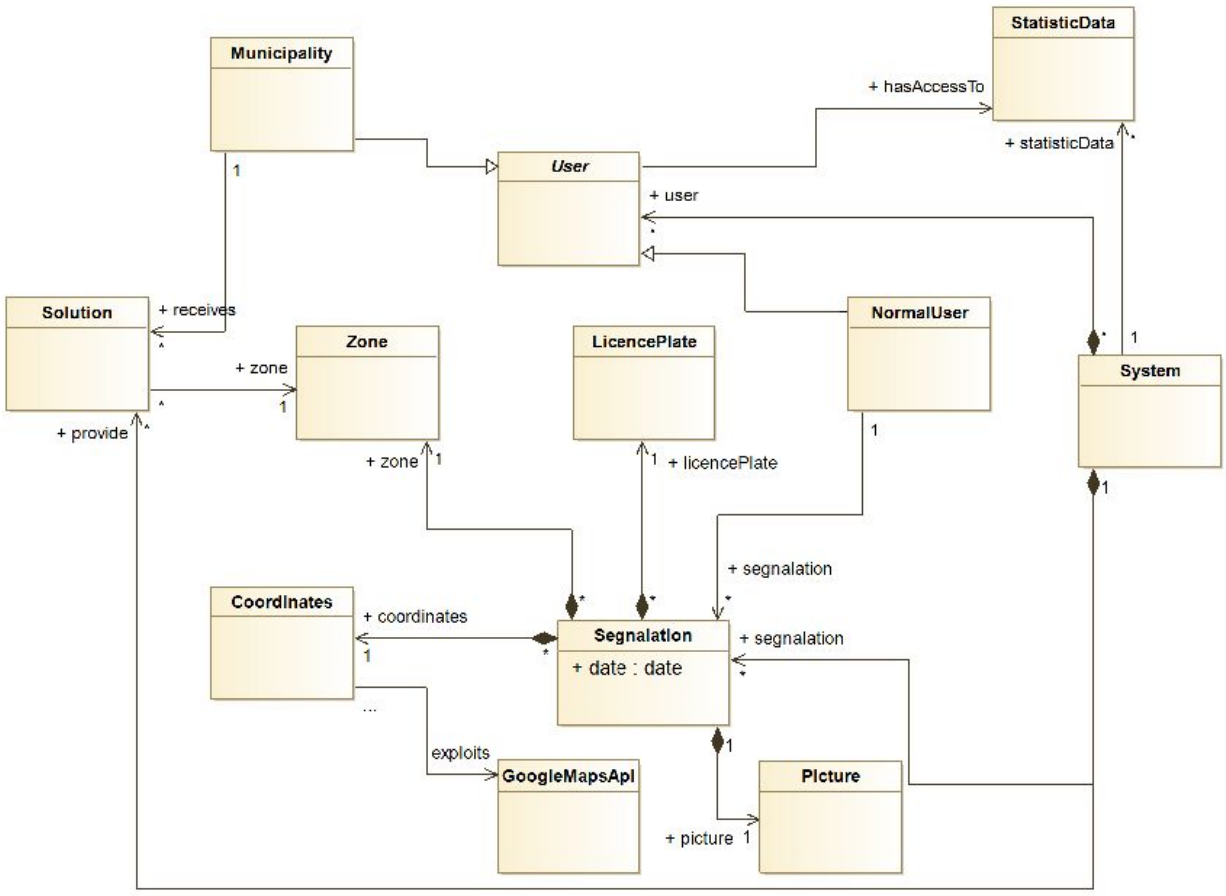

Figure 1 - Class diagram

Below the reader can find some state diagrams which show how states change in particular operations that are really important for the system.
The first diagram represents the various states of a segnalation while the user fulfil its mandatory requests and SafeStreets compute the missing parts.

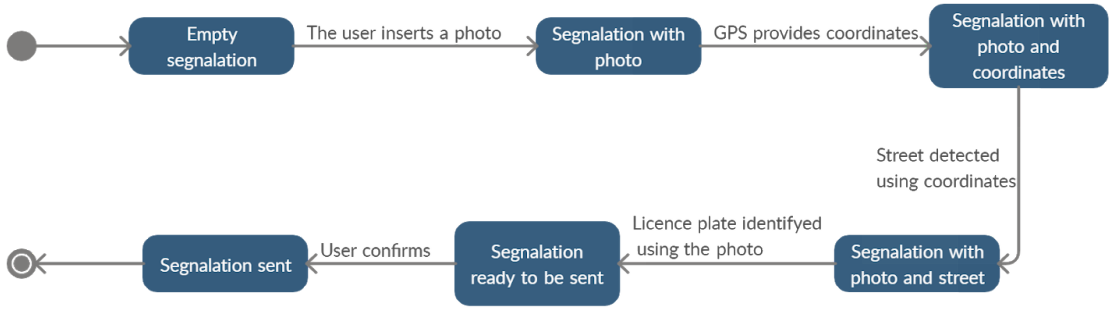

Figure 2 - First state diagram

The second one proceeds in the description of the flow of the state of the segnalation once it is received by SafeStreets till the moment in which it is definitely stored in the database.

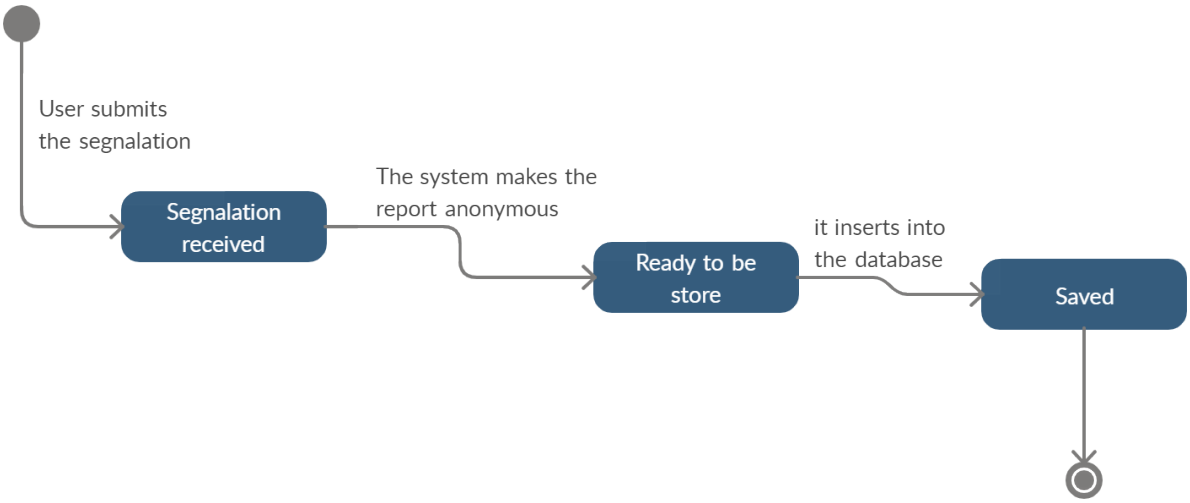

Figure 3 - Second state diagram

In the third and last case shown by state diagrams the reader can see what happens to a specific zone when SafeStreets receives a large number of segnalations indicating that many violations occur there.

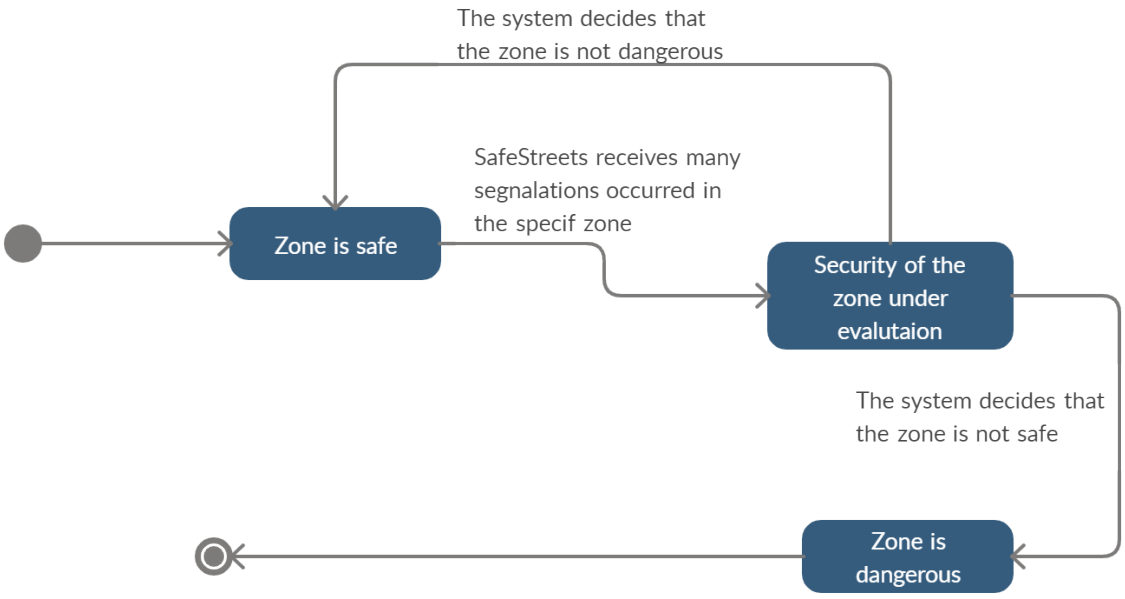

Figure 4 - Third state diagram

2.2 Product Functions

Based on the presented goals, we can identify three product functions that the system is going to provide. It's worth mentioning that SafeStreets can do its best when the municipality cooperates with it.

2.2.1 Collection of user data

This is the crucial function of SafeStreets: the system lets a user make a segnalation about a violation. The only data the user is asked to enter is the type of the violation, while the location, date and the licence plate are obtained by the system. All the segnalations will only store these information and any user sensitive data will not be saved. This ensures that the DB is freely explorable so that everyone with the right permissions can download the dataset and work with it to extract whatever information they care about. SafeStreets must also ensure that no fake data are sent by users, it achieves this by allowing to take pictures only by an embedded camera feature who makes impossible to modify the photo or to upload already existing photos from the gallery. Picture's metadata must also be checked to verify if the location matches the one provided by the GPS.

2.2.2 Visualizing statistics

This product function is in charge of mining the DB in order to provide insightful statistics to better understand how much spread out are certain types of violations. In particular, it will be possible to visualize on a map the zones more prone to parking violations highlighting them with different colors that represent different degrees of frequency.
Moreover, SafeStreets will provide specific data for each zone that show in which days the number of violations in higher and in which hour of the same days. A ranking concerning the vehicles which committed the most violations will also be available.

2.2.3 Offering solutions

This product function must be implemented if and only if the Municipality will allow users to retrieve information about accidents. In this case SafeStreets will cross these data with its own. Exploiting the interventions Municipality applied in the past, the system will recommend possible solutions to fight violations and prevent accidents. Based on the impact that the proposed solution had, SafeStreets will use this feedback to suggest better and better solutions in the future.

2.3 User Characteristics

Here we better define what the users of the service are.
A portion of the collected data is available to everyone, so each citizen interested on getting a grasp of the situation of the traffic violations in its territory can enjoy the statistics computed by SafeStreets. However, more active citizens can take part in collecting data by providing segnalations. A particular subclass of user is the one represented by Municipality, who in addition can provide its own data in order to get offered solutions to avoid accidents and violations.

2.4 Assumptions, dependencies and constraints

We assume that the following domain assumptions hold:

- D1: The user's smartphone GPS will provide a position with an error within 20 meters.
- D2: The picture taken by the user is clear enough to allow the OCR algorithm to retrieve the licence plate.
- D3: The user's Internet Connection works properly during the upload of the segnalation.
- D4: The data provided by Municipality are correct.
- D5: SafeStreets has a mapping that associates every possible type of violation with a possible solution to fight the specific problem.
- D6: Each user has a specific username that is unique.
- D7: SafeStreets has access to the GPS of the device that is running the app.

SafeStreets rely on two external tools:

1. External library that implements the OCR algorithm, to be chosen by the development team.
2. GoogleMaps API, to retrieve the location.

3 SPECIFIC REQUIREMENTS

3.1 External Interface Requirements

3.1.1 User Interfaces

The following mockups give a suggestion on how the mobile and web app should look like.

Figure 5 - Home Page

Figure 6 - Segnalation report

Figure 7 - Type of violation

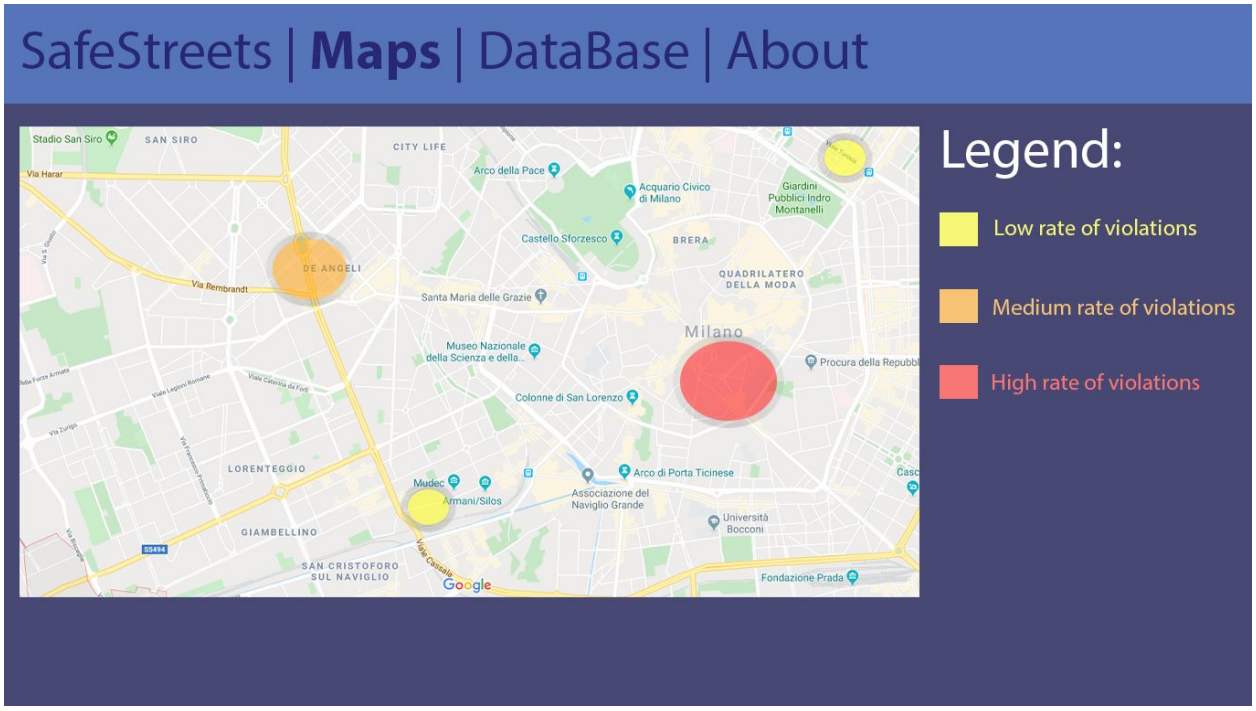

Figure 8 - Web App visualizing statistics

3.1.2 Hardware Interfaces

This application doesn't have hardware interfaces, however it requires to the device of the user to have an internet connection to access and visualize data, and a smartphone equipped with GPS to send segnalations.

3.1.3 Software Interfaces

The application doesn't provide any API to external applications.
It provides access to the DB, though, and it does so according to the permission each user has.
GoogleMaps API is used to retrieve the location and the name of the street.
An external library will provide the OCR algorithm to read the licence plate.

3.1.4 Communication Interfaces

HOURS OF WORK

De Vita Simone

15/10/2019: 3h
17/10/2019: 30min
18/10/2019: 45min
20/10/2019: 5h

Fabio Fontana

15/10/2019: 3h
17/10/2019: 45min
18/10/2019: 1h 30min
19/10/2019: 4h