SAMUELE MOSCATELLI, ANDREA POZZOLI

SAFESTREETS - RASD

Version 1

22nd October 2019

SAFESTREETS

Software Engineering 2 Project

Andrea Pozzoli and Samuele Moscatelli

SafeStreets RASD

Software Engineering 2 project

Politecnico di Milano

TITLEBACK

This document was written with LaTeX

CONTACTS

☞ pozzoliandrea97@gmail.com

☞ sem.mosca@gmail.com

CONTENTS

1 INTRODUCTION

1.1 PURPOSE

This RASD (Requirement Analysis and Specification Document) document aims to give a complete and rigorous description of the system "SafeStreets". In particular, this will be done through a deep analysis of the customer's needs, so identifying the main goals of the project, describing the functional and non-functional requirements and then proposing a solution able to fulfil all of them. This document is addressed to clients, but not only, in fact it is meant to be used as a contractual basis to which all the members of the team designed to create "SafeStreets" have to refer.

1.2 SCOPE

1.2.1 Description of the problem

SafeStreets is a crowd-source application that aims to give users the opportunity and responsibility of contributing to road regulation, with more attention to parking violations. In particular, the system offers functionalities which allow users reporting infringements, describing when, where and how they have been perpetrated. In order to certify the reporting, the user has also to attach a picture of the violation, making sure to include the license plate of the vehicle, so that the system can have a guarantee of the truthfulness of the information and at the same time identify the transgressor. So, for example, if a citizen, while walking through via Golgi, sees a car with license plate XXX parked in the middle of the bike lane, he can open SafeStreets, take a photo of the vehicle location, insert the date and the time and then send the report to the system. Once received the information, the application reads the license plate from the picture and store it together with the other data provided by the user. So, in the example previously shown, SafeStreets would memorize a parking violation of the type "car on bike lane" in via Golgi, with license plate XXX.

Citizens are not the only type of users of the application, also authorities, in fact, can use it in different ways. In first place, they can consult SafeStreet in order to retrieve information about the situation of streets which are under their jurisdiction. The system, in addition to providing individual violations, in fact, once received one of them, elaborates it and combines it with the other data already stored, creating statistics on which streets are characterised by the highest number of infringements, or on which vehicles commit violations most frequently, or, again, on which type of violation is most perpetrated. Secondly, authorities can also collaborate with SafeStreets with the aim of making roads safer through prevention. In particular, if mu-

nicipality provides information about accidents that occur on its territory, the system can merge this data with those coming from violations and in this way identify the most dangerous areas, such that interventions can be suggested. (manca forse da dire che anche i cittadini possono vedere i dati elaborati)

Phenomena	**Shared**	**Who controls**
User sees a traffic violation	NO	World
User wants to notify authorities	NO	World
User launches the application and logs in	NO	World
SafeStreets software is loaded checking the role of the user (end user or authority)	NO	Machine
User inserts picture, date, time, position, type of violation	YES	World
SafeStreets asks to the user to insert again some wrong data	YES	Machine
User sends data	YES	World
SafeStreets receives a picture	NO	Machine
SafeStreets runs the algorithm to read the license plate	NO	Machine
The algorithm can't read the license plate so SafeStreets asks to the user to insert it manually	YES	Machine
User inserts the license plate manually	YES	World
SafeStreets creates a new object Violation and stores the retrieved information	NO	Machine
SafeStreets calculates the statistics	NO	Machine
User or authority mines the information	YES	World
SafeStreets receives data about accidents from municipality	YES	World
SafeStreets identifies unsafe areas	NO	Machine
SafeStreets suggests interventions	YES	Machine
SafeStreets goes out of service	YES	Machine

1.3 DEFINITIONS, ACRONYMS, ABBREVIATIONS

1.3.1 Definitions

- **End user:** The end user is a person that sees a traffic violation and wants to notify the authorities about it by using SafeStreets application. He can't see the violations sent by other users. He can see the statistics. End user is one of the two user types of the system.
- **Authority:** Authority is the second type of user and usually he is a police man or a employee of the municipality. He does not send violation data to the system. He cam see the violations sent by the end users. He can access to the statistics.

- **System:** The system is a synonymous of SafeStreets. The system receives data form the end users, elaborates and stores data, shows data to the authorities, calculates statistics and unsafe areas, suggests interventions.
- **Municipality:** Municipality is a city that decides to collaborate with SafeStreets. Municipality sends data about accidents to SafeStreets in order to find the unsafe areas.
- **Traffic violation:** Data sent by a end user is called traffic violation or only violation. A traffic violation is composed by a license plate (taken from a picture or a text inserted by the user), date, time, GPS position and the type of violation. Example of violations can be vehicles parked on the stripes or in places reserved to people with disabilities, double parking, parking in no parking places.
- **Statistics:** The statistics are some information calculated by SafeStreets in order to highlight the streets with a high number of violations, the license plates linked to many violations, the best end users, some percentages related to traffic tickets.
- **Unsafe Area:** Municipality sends to SafeStreets all the accidents occurred in a city. The streets and the areas in which there is a high number of accidents are called unsafe areas.
- **Intervention:** After having discovered some unsafe areas, SafeStreets suggests to municipality some actions (interventions) to do in order to make these areas safer.

1.3.2 Acronyms

- RASD – Requirement Analysis and Specification Document
- API - Application Programming Interface
- GPS - Global Positioning System
- UA - Unsafe Area
- TV - Traffic Violation

1.3.3 Abbreviations

- Gn: n-goal.
- Dn: n-domain assumption.
- Rn: n-functional requirement.

1.4 REVISION HISTORY

This is the first version of the document. The date of release of this document is 10th November 2019.

1.5 REFERENCE DOCUMENTS

- SafeStreets Mandatory Project Assignment
- Software Engineering 2 course slides

1.6 DOCUMENT STRUCTURE

The first chapter is a brief introduction to the document. It describes the purpose and the scope of SafeStreets software. In order to understand better the document, the first part presents also definitions, acronyms and abbreviations that will be used in the document.

The second chapter enters more in detail in the description of the project SafeStreets. In the first section we presents the shared phenomena and explains the domain model through a class diagram and some state-chart diagrams. Then we define the most important requirements that the software has to respect. They are divided in three subsections, each subsection presents a functionality of the system. In the third section instead we specifies which are the users of the system and which are their needs and the functionalities that they can use. Finally, there is a section in which there are the assumed domains and other constraints.

In the third chapter we focus our attention on specific requirements. First of all, we describes the requirements related to external interfaces, dividing them in different categories. Then, we mapped all the requirements and explained them through different UML diagrams. In fact, we defined the use case diagrams for each type of user, the use cases for each functionality and the sequence diagrams for explaining better the order of the operations and the relationships between users. After that, we presents some performance requirements in a textual way. (continua)

Alloy

The last two chapters present the effort spent by Andrea Pozzoli and Samuele Moscatelli in order to complete this document. The effort is measured by the hours of work. Finally there are the references that we consulted during the development of the project.

2 OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

2.2 PRODUCT FUNCTIONS

2.2.1 Traffic violations reporting

This function represents the core of the application, because, in fact, without it all the other functionalities offered by SafeStreets would be useless. The application, in particular, must allow any end user to easily report a violation in the instant in which he sees it. Going in further details, SafeStreets mobile app firstly has to allow citizens to take a picture of the violation, remembering them to include in the photo also the license plate of the transgressor. Then it has to let user select a traffic violation type and also insert a brief description of the situation, so that they can express all the inconveniences originated by the abuse. While providing the information, the application software must check for the correctness of them. In particular, it must make sure of the legibility of the license plate from the picture, and, if it is not so, SafeStreet must warn the user, making him choose between taking another photo or writing the license plate number himself. Once provided all the information, the citizen can send the violation data, and in this instant the application detects from the user device the date, the time and also the position in which the violation has been sent. In case that it is not able to detect one or some of this data SafeStreet has to notify again the user so that he can fix the problem (for example enabling geolocation). Once received the complete information, SafeStreet dispatching software searches for the nearest authority users to the place of the violation and notifies them of the occurrence of it, showing all the details. After receiving it they can decide to go there to solve the problem, so the software system allows them to warn of that their colleagues who have been notified of the same violation, so that not too many authorities are going to be busy to solve the same problem. Authority users can also request the list of all the reported violations, and read the details of each one.

2.2.2 Statistics elaboration

Another fundamental service offered by SafeStreets is the possibility of mining traffic violations information from the data sent by the end users. In more details, each time the system receives a violation, this is analyzed, and some information are extracted, elaborated and used to create statistics reports. In particular, SafeStreets calculates the places with the highest number of infringements, the most common type of infringements and the period of time in which they are more frequent. This can be very useful for

all the types of users, in fact end users can be aware of the most dangerous areas and try to avoid them, while authorities and municipality can exploit this data to improve road regulations by enforcing controls where violations are more frequent. For this reasons, SafeStreets must keep statistics always updated and provide them to each user who request them.

2.2.3 Unsafe area detection and intervention suggestions

In parallel with the elaboration of the statistics, SafeStreets must offer a function, this time dedicated only to municipality (but somehow linked also to citizens), that allows to insert data regarding accidents occurred in their region, such that the system can cross them with information related to traffic violations received by end users and provide a report on the safeness condition of the territory under the jurisdiction of that municipality. At the same time, SafeStreets has to analyze the elaborated data itself and by evaluating which type of events are more common in a specific area, it has to suggest the best interventions that can be done to make that area safer. In further details, the system first checks the type of the violations that frequently occur in a definite place and then, for those which happen really often, it also takes under consideration the description of them. In this way, if the problem mostly affects a particular category of citizens, SafeStreets can provide a more precise recommendation. As an example, if there is a specific bike lane on which drivers often park, the cyclists that uses to go biking there can report it each time this happens describing the situation, so that SafeStreets notices that and reports it to the municipality, with the suggestion of adding a barrier between the bike lane and the part of the road dedicated to motorized vehicles.

2.3 USER CHARACTERISTICS

2.3.1 End User

The end user is a person that can notify the authorities about a traffic violation. He must have a smartphone with an internet connection, GPS and a camera. The end user must install on his smartphone SafeStreets application and register in it. He must keep GPS and internet connection active when he uses the application. Every time the end user sees a traffic violation he can starts a new traffic violation report. The end user can see the statistics regarding traffic violations. He cannot see the traffic violations sent by other end users. He cannot see the unsafe areas and the suggested interventions. The end user is not notify when there is a new traffic violation near him.

2.3.2 Authority

The authority is a person that can see the traffic violation sent by end users. He must have a device with internet connection. The authority must install

SafeStreets software on his device and register in it. During the registration, it is necessary to verify that he is really an authority. He cannot create a new traffic violation report. He cannot be also an end user. The authority can see the statistics regarding traffic violations. He cannot see the unsafe areas and the suggested interventions. Authority must always keep GPS and internet connection active, also when he is not using SafeStreets software. He can be notified if a new traffic violation sent by an end user is near him. Authority can notify other authorities that he is going to verify if a traffic violation is true or not.

2.3.3 Municipality user

The municipality user represents a municipality that collaborate with SafeStreets. He must send to SafeStreets data regarding accidents that occur in his jurisdiction. Municipality can see the unsafe areas calculated by SafeStreets in his jurisdiction. He can also see the possible interventions suggested by SafeStreets in his jurisdiction. Municipality cannot see unsafe areas and suggested interventions that are not in his jurisdiction. He cannot be an end user. He cannot be an authority. Municipality must have a device with internet connection. He must install SafeStreets software on his device and register in it. During the registration, it is necessary to verify that he is really a municipality user. He cannot create a new traffic violation report. He cannot see traffic violations sent by end user. Municipality is not notify when there is a new traffic violation near him. He can see the statistics regarding traffic violations.

2.4 ASSUMPTIONS, DEPENDENCIES AND CONSTRAINTS

D1 A person (end user) knows the traffic rules.

D2 A person (end user) knows that he can notify the authority if there is a traffic violation.

D3 A person (end user) has a phone with a camera, internet connection and GPS, he knows SafeStreets app and he has SafeStreets app on his phone.

D4 A user of SafeStreets is identified by the application.

D5 It is possible to verify that an authority is really such (through a governmental code for example).

D6 An authority have a device on which there is SafeStreets software.

D7 The authority device has an internet connection and GPS active.

D8 A municipality has data about accidents occurred in its jurisdiction stored.

D9 It is possible to verify that a municipality is really such (through a governmental code for example).

D10 A municipality has a device with SafeStreets software.

D11 There is internet access around the place where violation occurred.

D12 An authority user is able to reach the position of a violations when notified.

D13 Users are fair with each other.

D14 Users send data about a violation from the position where the violation occurred.

D15 Users send data about a violation when the violation occurred.

All traffic violations created by end users are sent to SafeStreets through internet connection. The position of a traffic violation is taken automatically from the GPS of the end user's smartphone. Date and time are taken automatically, too. The notifications received by the authorities are sent through internet connection. Authorities need GPS because they receive notifications only if they are near a new traffic violation. Authorities and Municipality must demonstrate their role during the registration, maybe trough a governmental code, in order to avoid that common citizens register themselves as authority or municipality. Accidents data, unsafe areas and suggested interventions are sent through internet connection. End users must have a phone with a camera because they have to take a picture of the license plate of the car that commits the traffic violation. A published traffic violation report cannot be removed. A traffic violation report is shown to authorities in an anonymous way. When an end user inserts data regarding a traffic violation he has to choose a type of traffic violation among different predefined types and then he can give a description of the situation.

3 SPECIFIC REQUIREMENTS

3.1 EXTERNAL INTERFACE REQUIREMENTS

3.1.1 User Interfaces

End user: application with a way to insert a new traffic violation.
Authority: application/software with a way to see all the traffic violations, a way to notify other authorities, a way to be notified.
Municipality: software with a way to see unsafe area and suggested interventions, a way to send accidents.
All users: software with a way to see statistics.

3.1.2 Hardware Interfaces

For end users: smartphone with internet connection, GPS and camera.
For authorities: device with internet connection and GPS.
For Municipality: device with internet connection.
For the System: a large database and several servers; parallel operations.
Modern browser.

3.1.3 Software Interfaces

Maybe interfaces to map applications and to photo-camera tools.
Notification on smartphone or on computer. Push notification service.
Date and Time.

3.1.4 Communication Interfaces

HTTP RMI TCP-IP

3.2 FUNCTIONAL REQUIREMENTS

3.3 PERFORMANCE REQUIREMENTS

3.4 DESIGN CONSTRAINTS

3.4.1 Standards compliance

3.4.2 Hardware limitations

3.4.3 Any other constraint

3.5 SOFTWARE SYSTEM ATTRIBUTES

3.5.1 Reliability

The system is active 24/7 service. It is possible to have some deviations or concessions.

3.5.2 Availability

In order to guarantee high degree of availability, system of redundant servers may be considered. The system can have some little period of updating.

3.5.3 Security

User data are stored in a secure way. Anonymity is guaranteed. Other users can see traffic violations published by other users. The license plate are not shared. Authority position is private, none can know it. Accidents from municipality are not visible from other users and other municipalities. Unsafe area are not visible. authority and municipality must demonstrate their role.

3.5.4 Maintainability

We use design patterns for having a better maintainability. In the future the application will be easy and cheap to modify and fix.

3.5.5 Portability

The application runs on smartphone, tablet and computer, with different operative systems, different measures and different characteristics. So it has to be implemented in a way that respects the portability.

4 FORMAL ANALYSIS USING ALLOY

5 EFFORT SPENT

5.1 SAMUELE MOSCATELLI

- 10/10/2019: 1h
- 12/10/2019: 3h
- 18/10/2019: 2h 30min
- 19/10/2019: 7h
- 20/10/2019: 6h 30min

5.2 ANDREA POZZOLI

- 11/10/2019: 1h
- 12/10/2019: 3h
- 18/10/2019: 2h 30min
- 19/10/2019: 5h
- 20/10/2019: 4h 30min
- 22/10/2019: 5h