SAMUELE MOSCATELLI, ANDREA POZZOLI

SAFESTREETS - DD

Version 1

10th November 2019

SAFESTREETS

Software Engineering 2 Project

Andrea Pozzoli and Samuele Moscatelli

SafeStreets DD

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 DD (Design Document) document aims to continue with the work done in the RASD document, this time going much more in details in the definition of the system SafeStreets, describing the high-level architecture and the technical aspects that characterize it. In particular, the computational components and the interactions among these components are presented and explained accurately, clearing up their role, their behaviour and their implementation. The audience to whom the document is addressed is represented by the development team, which will rely on the principles exposed in this document in order to implement the system.

1.2 SCOPE

SafeStreets is a crowd-source project which has three different targets of people:

- The citizens (referred to as End Users)
- The Authorities
- The municipalities

The core idea is that of giving people the opportunity of participate in making their cities safer from the point of view of traffic regulation. In particular, the primary aim of the project is to give citizens a concrete tool that can grant them the possibility of signal a traffic violation to authorities. For this reason the End User is the key category for the correct functioning of the application. In more details, SafeStreets software allows citizens to report at any moment a traffic violation that is being perpetrated in front of them by specifying the type of the infringement, the description of it and also attaching a picture that depicts the vehicle, being sure to include also the license plate, so that the system can have a guarantee of the truthfulness of the information and at the same time identify the transgressor. Then the application automatically detects the time and the date of the report, together with the position from which it has been sent. So, for example, if a citizen, while walking through via Golgi on 30th October 2019, 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 type of violation and a description 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 and the position, date and time. So, in the example previously shown, SafeStreets would memorize a parking violation of the

type "car on bike lane" in via Golgi, on 30th October 2019, with license plate XXX, and a brief description of the situation. All the traffic violations sent by an end user are not lost, in fact every end user can see on the application his past contribu- tion to the traffic regulation. An end user cannot see the traffic violations sent by the other users.

Once received the information, SafeStreets firstly has to make it available for reading by all the authorities, also notifying those to whom is assigned the area in which the violation has occurred. The authorities notified can also decide to go and check directly the infringement, so, in order to avoid a concentration of them on the same event, they can also warn the other authorities of it. Another functionality offered to this type of users, as to all the other, is the possibility of mining some information from the system. In particular, SafeStreets has to elaborate the data received and combine them together in order to calculate traffic violation statistics. The possible statistics are: which streets are characterised by the highest number of infringements, in which moment of the day there are more violations, which type of violation is most perpetrated. The statistics regarding traffic violations can be accessed by all the three types of user. Instead data regarding a specific traffic violation is visible only by authorities.

Another type of user is municipality user, who can collaborate with SafeStreets with the aim of making roads under his jurisdiction safer through prevention. In particular, municipality provides information about accidents that has occurred and occur on its territory to the system. The system then can merge this data with those coming from violations and in this way it can identify the most dangerous areas, and at the same time suggest the best interventions that can be applied to make them safer.

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 three user types of the system.
- **Authority:** Authority is the second type of user and usually he is a police man. He does not send violation data to the system. He can see the violations sent by the end users. He can access to the statistics.
- **Municipality:** Municipality represents mayor and municipal employees of a city which decide to collaborate with SafeStreets and it is the third type of user. Municipality sends data about accidents to SafeStreets in order to allow it to cross them with violations data and find out the unsafe areas on its territory. He also can see 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.

- **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, the type of violation and a description of it. 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 days and times at which there are more violations, the most common types of violations.
- **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. An unsafe area can be only in one municipality jurisdiction and two different municipality jurisdiction cannot have the same unsafe area.
- **Intervention:** After having discovered some unsafe areas, SafeStreets suggests to municipality some actions (interventions) to do in order to make these areas safer.
- **Accident:** an accident is data that municipality sends to SafeStreets in order to find the unsafe areas and suggest interventions. It describes a dangerous situation that occurred in one of the streets or areas under the municipality jurisdiction.

1.3.2 Acronyms

- RASD – Requirement Analysis and Specification Document
- API - Application Programming Interface
- GPS - Global Positioning System
- RMI - Remote Method Invocation
- UML - Unified Modeling Language

1.3.3 Abbreviations

- Gn: n-goal.
- Dn: n-domain assumption.
- Rn: n-functional requirement.
- rn: n-row of the matrix.

1.4 REVISION HISTORY

This is the first version of the document. The date of release of this document is 9th December 2019.

1.5 REFERENCE DOCUMENTS

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 also presents definitions, acronyms and abbreviations that will be used in the document.

The second chapter represents the core of the document. The first section is a formal overview of the application, so here specific styles used are identified and exposed. Then the document get more into architectural details of the system, showing and explaining the architectural components and their distribution with respect to the each other and to the hardware resources. In order to do so, appropriate UML diagrams (component diagram, class diagram and deployment diagram) are presented and outspread, giving a static view of the application architecture. Then also a dynamic view is given through the use of sequence diagrams, so that to show in a more detailed way the behaviour of the components and the interactions among them, defining in this way how the system responds to the external invocations. Subsequently, the attention is focused on the interfaces offered by the previously presented components, going more in detail in the explanation of the methods exposed by them. Finally, patterns used in the architecture and other design decisions are shown.

The third chapter gives a precise and polish idea of the user interface design, displaying how they actually are implemented at the implementation level.

The forth chapter represents requirements traceability, so requirements are allocated to the introduced components, highlighting which of them are involved in the satisfaction of which goals.

The chapter number five involves the planning of implementation, integration and testing phases, referring to what is expressed in chapter two, and so relating the introduced components with the way in which they will be implemented and the way in which the system will be integrated and tested.

Finally chapters number six and seven concern respectively the effort spent by Andrea Pozzoli and Samuele Moscatelli in order to complete this document and the references consulted during the development of the project.

2 ARCHITECTURAL DESIGN

3 USER INTERFACE DESIGN

4 REQUIREMENTS TRACEABILITY

5 IMPLEMENTATION, INTEGRATION AND TEST PLAN

6 EFFORT SPENT

6.1 SAMUELE MOSCATELLI

- 14/11/2019: 2h

Total: 2h

6.2 ANDREA POZZOLI

- 11/11/2019: 1h

Total: 1h

7 REFERENCES