Politecnico di Milano

SAFESTREETS

POLITECNICO
MILANO 1863

Samuele Meta - Stiven Metaj

Supervisor: Matteo Rossi

SOTTOTITOLO OLEEEEEEEE

Department of Computer Science and Engineering

October 23, 2019

Contents

1 Introduction

The following Requirement Analysis and Specification Document (RASD) aims at illustrating a complete overview of the project *SafeStreets*, providing a baseline for its planning and development. It guides the reader in understanding the specifics of the application domain and the relative System in terms of functional requirements, non functional requirements and constraints. It details how, according to these, the System interacts with the external world, showing concrete use case scenarios. A more comprehensive description of the most relevant features will be modelled with the use of the Alloy language. This document is addressed to all the stakeholders - such as users, system and requirement analysts, project managers, software developers and testers - who will evaluate the correctness of the assumptions and of the decisions contained in it.

1.1 Purpose

SafeStreets is a new crowd-sourced mobile application that allows Citizens to notify Authorities about traffic violations, with particular emphasis on parking contraventions. In fact, every day is possible to encounter minor traffic infringements that affect the driving experience, whether it's a double parking or an unduly occupied parking lot reserved for disables. With *SafeStreets*, Citizens can actively participate in road monitoring, partially compensating for the impossible ubiquity of patrols.
To do so, *SafeStreets* requires the User to create an account as Citizen - providing personal credentials such as name, surname, National Insurance Number (NIN) - and to verify it sending a picture of a valid document. From this moment on, the System will allow the Citizen to send photos of traffic violations, enriched by the relative description, position, date and time. At any time, the application allows the Citizen to visualize the history of the reported violations since it's registered. In case of error or incorrect information, the System allows the Citizen to revoke the alert produced.
Futhermore, the System offers the possibility to sign up as an Authority, once filled the registration form and verified the institutional identity through the relative burocracy. *SafeStreets* will allow Authorities to have a complete overview of the incoming signalations and to choose if to accept or reject them.
Since the System stores all the violations and their metadata, *SafeStreets* is able to process them in order to extract meaningful insights and statistics. The System allows both Citizens and Authorities to access them, with a different visibility level according to the role.
Moreover, if the municipality offers a service that allows to retrieve the information about the accidents occured in the covered area, *SafeStreets* will cross the two knowledge bases to

identify potentially unsafe areas and suggest possible interventions.
The above description can be summarized as follows, in a list of goals:

- [**G.1**] The System allows the User to register to the application either as *Citizen* or as *Authority*, providing an identification code and a password.
- [**G.2**] The System allows the User to report a traffic violation by sending a photo of it and the relative date and position.
- [**G.3**] The System allows the User to find out the streets with highest frequency of violations.
- [**G.4**] The System allows the Authority to find out vehicles that commit the most violations.
- [**G.4**] The System allows the Authority to cross the information in order to identify potentially unsafe areas.
- [**G.5**] The System suggests the Authority possible solutions to the problem.
- [**G.6**] The System allows the User to see the reported violations.
- [**G.7**] The System allows the User to retire a violation.
- [**G.8**] The System completes the report with metadata
- [**G.9**] The System suggests the Authority possible solutions to the problem.
- [**G.10**] The System analyzes plate, model quality of photo
- [**G.11**] The System checks if photo is good (security).
- [**G.11**] The System allows the Authority to accept or refuse a violation.

1.2 Scope

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

1.3.2 Acronyms

1.3.3 Abbreviations

1.4 Revision History

1.5 Reference Documents

1.6 Document Structure

2 Overall Description

2.1 Product Perspective

2.1.1 Use Case Templates

2.2 Product Functions

OLE

2.3 User Characteristics

OLE

2.4 Assumptions, Dependencies and Constraints

OLE

3 Specific Requirements

OLE

3.1 External Interface Requirements

OLE

3.1.1 User Interfaces

ole

3.1.2 Hardware Interfaces

ole

3.1.3 Software Interfaces

ole

3.1.4 Communication Interfaces

ole

3.2 Functional Requirements

OLEE

3.2.1 Use Cases

oleole

3.2.2 Sequence and Activity Diagrams

oleole

3.2.3 Mapping on Requirements

oleole

3.3 Performance Requirements

OLE

3.4 Design Constraints

OLE

3.4.1 Standards Compliance

ole

3.4.2 Hardware Limitations

ole

3.4.3 Any Other Constraint

ole

3.5 Software System Attributes

OLE

3.5.1 Reliability

oleole

3.5.2 Availability

oleole

3.5.3 Security

oleole

3.5.4 Maintainability

oleole

3.5.5 Portability

oleole

4 Formal Analysis using Alloy

4.1 CODICE SCHIFOSO

OLE

4.2 SCREEN DEL CODICE SCHIFOSO CHE RUNNO E FUNZIONA TUTTO STRABENE

OLE

5 Effort Spent