Front Page Test

AmorosiniCasaliFioravanti

October 2019

Contents

Introduction

This document represents the Requirement Analysis and Specification Document (RASD) for SafeStreets: a crowd-sourced application that intends to provide users with the possibility to notify authorities when traffic violations and accidents occur. Our purpose is to supply a description of the system in terms of its functional and non functional requirements, listing all of its goals, discussing the constraints and the limits of the software, and indicating the typical use cases that will occur after the release. This document is addressed to the stakeholders, who will evaluate the correctness of the assumptions and decisions contained in it, and to the developers who will have to implement the requirements.

1.1 Purpose

SafeStreets is a crowd-sourced application that intends to provide users with the possibility to notify authorities when traffic violations and accidents occur. In order to report a traffic violation, users have to compile a report containing a picture of the violation, the date, time, position, and type of violation. SafeStreets stores all the information provided by users, completing it with suitable metadata (timestamps, sender's GPS if available, etc.). In particular, in case the user had not provided any information regarding the license plate of the car breaking the traffic rules, SafeStreets runs an algorithm to read it from the submitted picture. In addition, the application allows both end users and authorities to mine the information that has been received: this function allows users to perform useful statistics

1.1.1 Goals

- [G.1] The System allows the Users to access the functionalities of the application from different locations and devices.
- [G.2] The System allows the Users to authenticate themselves either as Authority or Citizen.
  - [G.2.1]The System offers different levels of visibility to different roles.

- [G.3] The System allows the Citizens to document traffic violations to Authorities by compiling a Report.
- [G.4] The System stores the reports provided by the Citizens.
  - [G.4.1] If the Citizen has not provided any information about the license plate, the System runs an algorithm to read it from the submitted picture.
- [G.6] The System allows the Citizens to add testimony of an accident they have seen.
- [G.7] The System elaborates information to detect unsafe areas
  - [G.7.1]If the System collects common witnesses for the same problem, can suggests possible interventions to the Authority.
- [8] The System ensure the chain of custody of the informations coming from the Citizens.
- [9] The System can builds statistics on violations using the stored informantion about issued tickets.

1.2 Scope

According to the *World* and *Machine* paradigm we can distinguish the *Machine*, that is the portion of the system to be developed, from the *World*, that is the portion of the real-world affected by the machine. In this way, we can classify the phenomena in three different types: World, Machine and *Shared* phenomena, where the latter type of phenomena can be controlled by the world and affected by the machine or controlled by the machine and affected by the world.
In this context, the main relevant phenomena are organized as in the following table.

World	Machine	Shared
Traffic violation: circumstances in which someone doesn't respect the traffic rules	**DBMS queries**: operation performed to retrieve/store data	**Registration/Login**: an user can sign up to the application or log in if is already registered.
Incident:event that caused damage to things or people.	**API queries**: request for third-part services.	**Send of report**: someone who send the description of the traffic violation.
•	**use of Hash**: compute the hash function to ensure that information is never altered.	**Build statistics**: compute statistics to found unsafe area, to see which veichles commit multiple infractions and also to see how the trends of issued tickets change.
		Safe interventions: the application give suggestion for possible safe interventions (e.g add a barrier).

Table 1.1:

Overall Description

Specific Requirements

3.1 Functional Requirements

- [G.4] The Report Manager handles the submissions:
  - [G.4.1] Before accepting any Report, the Report Manager ensures that it contains a picture of the violation, the date, time, position, and type of violation.
  - [G.4.2] The Report Manager notifies all logged in Authorities whenever a new Report is accepted.
  - [G.4.3] The Report Manager must ensure that no more than one Authority has taken in charge the same Report.
  - [G.4.4] The Report Manager notifies Citizens whenever one of their Reports has been taken in charge by an Authority.
- [G.9] The System can accedes to the stored traffic tickets with a permission from the local municipality.