

# SafeStreets

Marco Premi, Fabrizio Siciliano, Giuseppe Taddeo

October 21, 2019

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Scope . . . . .	4
1.3	Definitions, Acronyms, Abbreviations . . . . .	4
1.3.1	Definitions . . . . .	4
1.3.2	Acronym . . . . .	4
1.3.3	Abbreviations . . . . .	4
1.4	Revision History . . . . .	4
1.5	Reference Documents . . . . .	4
1.6	Document Structure . . . . .	4
<b>2</b>	<b>Overall Description</b>	<b>4</b>
2.1	Product Perspective . . . . .	4
2.2	Product Functions . . . . .	4
2.3	User characteristics . . . . .	4
2.4	Assumptions, dependencies and constraints . . . . .	4
2.4.1	Assumptions . . . . .	4
2.4.2	Dependencies . . . . .	4
2.4.3	Constraints . . . . .	4
<b>3</b>	<b>Specific Requirements</b>	<b>4</b>
3.1	External Interface Requirements . . . . .	4
3.1.1	User Interfaces . . . . .	4
3.1.2	Hardware Interfaces . . . . .	4
3.1.3	Software Interfaces . . . . .	4
3.1.4	Communication Interfaces . . . . .	4
3.2	Scenarios . . . . .	4
3.3	Functional requirements . . . . .	4
3.4	Use Cases . . . . .	4
3.4.1	User use cases . . . . .	4
3.4.2	Third party use cases . . . . .	4
3.5	Performance requirements . . . . .	4
3.6	Design Constraints . . . . .	4

<i>CONTENTS</i>	2
3.6.1 Standars compliance . . . . .	4
3.6.2 Hardware Limitations . . . . .	4
3.6.3 Any Other Contrain . . . . .	4
3.7 Software System Attributes . . . . .	4
3.7.1 Reliability . . . . .	4
3.7.2 Availability . . . . .	4
3.7.3 Security . . . . .	4
3.7.4 Mantainability . . . . .	4
3.7.5 Portability . . . . .	4
<b>4 Formal Analysis using Alloy</b>	<b>4</b>
<b>5 Effort spent</b>	<b>4</b>
<b>6 References</b>	<b>4</b>

# 1 Introduction

## 1.1 Purpose

**General Purpose** The purpose of this document is to correctly analyze all requirements, goals and actions needed in order to correctly develop SafeStreets.

SafeStreets is a crowd-sourced application that intends to provide users with the possibility to notify authorities when traffic violations occur (eg. traffic violations) and will help to maintain stability and order within the streets. A reporting system will also be available to police offices and municipal employees in order to allow them to analyze (and take actions accordingly) different areas of the city and assess which areas have the most violations committed in.

The core application will focus on storing useful traffic violation data provided by users, mainly with the help of input forms and hard evidence such as images. At any violation input, SafeStreets will also store useful metadata such as date and time the violation was retrieved, geolocate where it is and update a city wide map highlighting the areas where violations happen.

In addition to that, with the help of third parties (eg. municipality), SafeStreets will be able to retrieve the data given by such party and cross-reference them with its own data retrieved by users. By doing so, it will be possible to identify unsafe areas, assess which kind of problems happen more frequently and suggest possible intervention. It will also be possible for third parties (municipality and police officers) to automatically generate traffic tickets. This will be happening in order to cross-reference all available data in order to build statistics such as the most (or less) egregious offenders or the effectiveness of the SafeStreets initiative

**Goals** Follows a list of all goals that will be reached with the SafeStreets initiative.

- G1: The system must allow all kind of users (both third parties and civilians) to correctly input (with hard evidence) traffic violations around the city;
- G2: The system must autonomously retrieve metadata from hard evidence useful to report and to all users;
- G3: The system must allow to create different clearance levels in order to offer different reporting systems to different kind of users;
- G4: The system will provide an efficient reporting system in order to highlight different violation categories through all parts of the city where the initiative is active;
-

## 1.2 Scope

## 1.3 Definitions, Acronyms, Abbreviations

### 1.3.1 Definitions

### 1.3.2 Acronym

### 1.3.3 Abbreviations

## 1.4 Revision History

## 1.5 Reference Documents

## 1.6 Document Structure

# 2 Overall Description

## 2.1 Product Perspective

## 2.2 Product Functions

## 2.3 User characteristics

## 2.4 Assumptions, dependencies and constraints

### 2.4.1 Assumptions

### 2.4.2 Dependencies

### 2.4.3 Constraints

# 3 Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

### 3.1.2 Hardware Interfaces

### 3.1.3 Software Interfaces

### 3.1.4 Communication Interfaces

## 3.2 Scenarios

## 3.3 Functional requirements

## 3.4 Use Cases

### 3.4.1 User use cases

### 3.4.2 Third party use cases

## 3.5 Performance requirements

## 3.6 Design Constraints

### 3.6.1 Standards compliance

### 3.6.2 Hardware Limitations

### 3.6.3 Any Other Constraint

## 3.7 Software System Attributes

### 3.7.1 Reliability

### 3.7.2 Availability

### 3.7.3 Security

### 3.7.4 Maintainability