

POLITECNICO DI MILANO



Academic Year 2019/2020

# RASD

## **Requirements Analysis and Specification Document**

version 1.0 – 10/11/2019

Computer Science and Engineering  
Software Engineering 2

Matteo Falconi 945222 - Davide Galli 944940

# Contents

<b>1</b>	<b>Introduction .....</b>	<b>4</b>
1.1	Purpose .....	4
1.1.1	Project Description.....	4
1.1.2	Goals.....	4
1.2	Scope.....	5
1.2.1	World.....	5
1.2.2	Phenomena .....	5
1.3	Definitions, acronyms, abbreviations .....	6
1.3.1	Definitions .....	6
1.3.2	Acronyms.....	7
1.3.3	Abbreviations .....	7
1.4	Revision history.....	7
1.5	Reference Documents .....	7
1.6	Document Structure .....	8
<b>2</b>	<b>Overall Description.....</b>	<b>9</b>
2.1	Product prospective .....	9
2.2	Product functions.....	9
2.2.1	Profile Management .....	9
2.2.2	Data Collection .....	10
2.2.3	Data Inspection .....	10
2.2.4	Data Analysis .....	11
2.3	Users characteristics.....	11
2.4	Assumption, dependences, constrains .....	12
2.4.1	Domain Assumption .....	12
2.4.2	Dependences.....	12
2.4.3	Constrains.....	12
<b>3</b>	<b>Specific Requirements .....</b>	<b>14</b>
3.1	External interface requirements .....	14
3.1.1	User Interface.....	14
3.1.2	Hardware Interface .....	20
3.1.3	Software Interface.....	20
3.2	Functional Requirement .....	20
3.3	Scenarios .....	22
3.3.1	Double Parking Report .....	22
3.3.2	Reports Check .....	22
3.3.3	Violations Statistics Check.....	22
3.3.4	Municipality Suggestion .....	23
3.4	Use Cases.....	23

3.4.1	UML Modelling.....	30
<b>3.5</b>	<b>Performance requirements.....</b>	<b>32</b>
<b>3.6</b>	<b>Design constrains .....</b>	<b>32</b>
3.6.1	Standard compliance.....	32
3.6.2	Hardware limitations.....	32
3.6.3	Any other constraints.....	32
<b>3.7</b>	<b>Software system attributes.....</b>	<b>32</b>
3.7.1	Reliability.....	32
3.7.2	Availability .....	32
3.7.3	Security.....	32
3.7.4	Maintainability .....	32
3.7.5	Portability .....	33
<b>4</b>	<b>Alloy modelling.....</b>	<b>34</b>

# 1 Introduction

## 1.1 Purpose

### 1.1.1 Project Description

*SafeStreets* wants to develop a software-based service that allows individual basic users to report traffic violation. Those data consist in pictures of violation, type of violation, date, time and position. When a picture is upload, the system runs an algorithm in order to read the license plate. Finally, all those data are store in *SafeStreets*' databases.

The system allows also authorities registration, who can receive notifications about new violations in a certain area. When a notification occurs, an authority can reserve it taking charge of that violation.

Both basic users and authorities can access to collected data in order to analyze the streets and the relative safeness. However, a basic user can only access to anonymized data clusters, that give an idea of how many violations occur in each area; whereas authorities can also access to specific anonymized data.

Another service that *SafeStreets* wants to develop, in partnership with the municipalities, is an algorithm that can cross-reference data provided by the users with the accidents data provided by the municipalities, in order to identify unsafe areas and suggest possible interventions.

### 1.1.2 Goals

Basic users:

[G.BU1] Basic users can report traffic violations.

[G.BU2] Basic users can view a data clustering about violations that had occurred.

Authorities:

[G.A1] Authorities should choose to receive anonymous notifications in real time about new violations.

[G.A2] Authorities should reserve a violation.

[G.A3] Authorities can view both data clustering and specific data about violations that had occurred.

Municipalities:

[G.M1] Municipalities can identify potential unsafe zones and receive suggestions about possible solutions.

## 1.2 Scope

### 1.2.1 World

There are three main types of actors in our world: citizen, authorities and municipalities. Citizen are interested in reporting traffic violations and receiving information about violations in certain areas, authorities and municipalities are interested in exploiting the data gathered from the citizen: the firsts want to get notified when new violations occur in order to generate traffic tickets, the seconds want to identify unsafe zones and to receive possible solution.

*SafeStreets* is the service that acts as a bridge between these actors' needs.

system ← da definire meglio

### 1.2.2 Phenomena

Phenomena that occur in the world and that are related to the system application domain are:

- Traffic violations occur in a city;
- An authority makes traffic tickets;
- Authorities, users and municipalities are interested in analyzing violation data;
- Municipality wants to reduce the number of accidents.

The system shares also some events with the world in order to communicate with it. The phenomena that occur in the world and are observed by the machine are:

- A user registers and logs in filling the various form;
- A user fills the violation data and sends a new report;
- An authority manages notifications, enabling or disabling them;
- An authority searches a violation in which he/she is interested;
- An authority looks the details of a violation;
- An authority reserves a violation;
- App user views mined data on a map in his/her smartphone;
- Municipality studies unsafe zones;
- Municipality views safety report with suggestions for reducing accidents.

On the other hand, the aspects generated by the machine and observed by the world are:

- The system localizes app users on the map;
- The system uploads, receives and confirms data insert by users through an acknowledgement (login credentials, new violation reports, etc.);
- The system generates notifications about new violations;
- The system creates safety reports for the municipalities, with suggestions to reduce accidents;
- The system loads and renders graphically data to the user (violations list, detail of a violation, etc.)
- The system creates a map where are rendered the mined data and shows them to users.

### 1.3 Definitions, acronyms, abbreviations

#### 1.3.1 Definitions

<i>User</i>	Any kind of person who use the system (basic user, authority and municipality).
<i>App User</i>	Any kind of person who use the system through the App (basic user and authority).
<i>Basic user</i>	Citizen who can report a traffic violation and view a data clustering about violations that had occurred.
<i>Authority</i>	Recognized entity which can empower the law (ex. local police).
<i>Municipality</i>	Authority recognized by the State who hold the government in an area.
<i>Data clustering</i>	A set of anonymous data about violations group by location and type.
<i>Specific violation / violation data</i>	Information about a violation. Contains: photos, location, type of violation, license plate, date and time.
<i>Traffic violation</i>	Illegal action performs by any vehicle (ex. double parking, stopped on zebra cross).
<i>Accident</i>	Traffic violation result in an injury for at least one person.
<i>Unsafe zone</i>	Area of the city where accidents happens frequently.
<i>PC</i>	Generic system able to navigate through the internet.

*Integrated data*      *SafeStreets'* violation data united with municipality's accident data.

*Matricula / Personal ID*      A code able to identify uniquely an authority, stored in State's DB

### 1.3.2 Acronyms

*API*      Application Programming Interface

*GPS*      Global Positioning System

*S2B*      Software to Be

*UI*      User Interface

*IEEE*      Institute of Electrical and Electronics Engineers

DB      Database

### 1.3.3 Abbreviations

*[G.BU<sub>n</sub>]*      Basic users' n<sup>th</sup> goal;

*[G.A<sub>n</sub>]*      Authorities' n<sup>th</sup> goal;

*[G.M<sub>n</sub>]*      Municipalities' n<sup>th</sup> goal;

*[D.<sub>n</sub>]*      N<sup>th</sup> domain assumption;

*[R.<sub>n</sub>]*      N<sup>th</sup> requirement;

*[R.M<sub>n</sub>]*      Municipalities' n<sup>th</sup> requirement;

## 1.4 Revision history

Date	Version	Log
10/11/2019	v. 1	First RASD release

## 1.5 Reference Documents

Specification document: "SafeStreets Mandatory Project Assignment"

IEEE 830-1993 - IEEE Recommended Practice for Software Requirements Specifications

Di Nitto's course slide

## 1.6 Document Structure

According to the IEEE standards for requirement analysis documents, this document is composed into 5 sections, organised as follow:

Section 1 gives a short introduction to the project; giving a clear idea of who are the actors and what are the goals of the S2B;

Section 2 defines the main functions of the project, analysing the constraints and declaring the assumptions;

Section 3 is the most important part of the RASD: it

Sez 1. → Breve riassunto del progetto da sviluppare. Fornisce un'idea chiara degli attori, dei goal da raggiungere.

Sez 2. → Definisce le funzioni principali del progetto, I suoi vincoli (limitazioni fisiche o di progettazione), le assunzioni fatte

Sez 3. → Analisi di requisiti, scenari,

Sez 4. → Modello alloy sviluopato per provare la correttezza del progetto

Sez 5. → Tempo speso da ciascun partecipante



## 2 Overall Description

### 2.1 Product prospective

*SafeStreets* is a crowd-based service oriented to data acquisition and data analysis. Its software exploits basic user interaction to retrieve data from the world. Once the user registered to the service, he/she can submit a report, allowing the system to collect data and to analyze it, sorting the data acquired into clusters.

It is mandatory for the basic user to have a smartphone able to connect to the internet, taking photos and with a built in GPS.

Authorities are interested in data inspection: they can receive notification when a new report is submitted. In order to allow them to choose an area of interest in which to be notified, *SafeStreets* provides a user interface that is charge of composing their request, to make it understandable by the system.

Municipalities are interested in data analysis. They provide their accident DBs to *SafeStreets*, which cross-reference the data with their own data collected thanks to the users' report, in order to generate a Safety Report. Municipalities are required a PC with an internet connection.

### 2.2 Product functions

In the following section we present the major functions that our product will offer.

#### 2.2.1 Profile Management

The system will provide a registration form for new users. Each type of user will need to provide different information, as reported in the table below. Creating an account is mandatory in order to exploit the system's functionalities. The system must be able to distinguish accounts for basic users, authorities and municipalities, as it should offer different functionalities between the three of them.

The system will offer the ability to delete an account: the deletion of an account will be permanent and will imply the removal of data from *SafeStreets*' DB.

Account Type	Required Information	Additional Information
Basic User	<i>Name, surname, email address, password, date of birth</i>	<i>None</i>

Authority	<i>Name, district's name, personal ID, email, password</i>	<i>None</i>
Municipality	<i>Name of the municipality, referencing email, referencing name, referencing surname, referencing phone number</i>	<i>Comments</i>

*Table 1: Registration form information*

### 2.2.2 Data Collection

Data collection is exploited through the submission of reports by basic users: collected data includes photos of the violation taken by the user, type of the violation chosen among some proposed, a timestamp collected by the device and the location in which the report is submitted.

### 2.2.3 Data Inspection

*SafeStreets* relies on a database for data storage. Both basic users and authorities can access to the data stored in the database; but with a different granularity level, as explained in this table:

Account Type	Viewable Data
Basic Users	<i>They can only access to statistics' map, in which the violations are organised by type and by location (Figure 6).</i>
Authorities	<i>They can access both to statistics' map and to a specific violation: the former is as explained above, the latter contains detailed information about a violation in a certain area. (Figure 14).</i>

*Table 2: Data Inspection Granularity Level*

Authorities can also choose to receive a notification when a new violation is reported. Notifications can be enabled in the S2B's settings and they are set base on a geographical area: when a violation is reported in the area selected, a notification is sent to the authority. More than one notification can be set in different areas and only authorities can enable them.

#### 2.2.4 Data Analysis

This function is only available to municipalities that can provide access to their accidents' DB. Our system will extract the data and will compare it with our own data, gathered from the basic users, in order to map the unsafe areas with the most likely triggering violation. It will then generate a safety report, compiling the safety form as follows:

<i>Raw Data</i>	Here will be listed all the raw data collected from our DB in the area of interest; listing the number of possible violations report received during the period of time considered.
<i>Accidents' Analysis</i>	This part of the safety report will contain the data extracted from the municipalities' DB, regarding the number and the type of accidents that had happened in the area of interest.
<i>Possible Solution</i>	Finally, here we will list some possible solutions to the most common accidents observed by municipality' s data, based on the most common violations report observed by our data.

### 2.3 Users characteristics

The system interacts with the following actors:

<b>Basic User</b>	Person interested in the reporting violations system; he/she is required to create a basic user account in order to exploit <i>SafeStreets</i> ' functionalities. He/She can also view clustered data about violations in his/her area. Every basic user owns a mobile device able to connect to the internet and able to monitor GPS location.
<b>Authority</b>	Person interested in the viewing violations system; he/she is required to create an authority account in order to exploit <i>SafeStreets</i> ' functionalities. He/She can also view clustered data about violations in his/her area.

Every authority owns a mobile device able to connect to the internet and able to monitor GPS location.

**Municipally** Entity interested in safety reports system; it is required to contact *SafeStreets* through the web-page in order to exploit *SafeStreets*' functionalities.

Every municipality owns a digital database about accidents and a PC with internet access.

## 2.4 Assumption, dependences, constraints

### 2.4.1 Domain Assumption

- [D.1] All traffic violations data are correctly encoded.
- [D.2] Every license plate is unique and identify uniquely a vehicle.
- [D.3] Every authority's Personal ID is unique and identify uniquely an authority.
- [D.4] An authority reserving a violation will take care of it.
- [D.5] Basic users send only pictures about violations.
- [D.6] Municipality partner has a digital database about accidents.
- [D.7] Accident data are provided by location.
- [D.8] Exist a State's database where are stored all the authority matriculas.

### 2.4.2 Dependences

SafeStreets relies on:

- Geo-location services, to access users' location.
- Maps provider API, to show maps on application.
- Image scanning algorithm, for reading data from violation's photos.
- Municipality's API, in order to access accident's DB.
- State's DB, in order to access authorities' matriculas and verify their identity.

### 2.4.3 Constraints

#### Regulation policies

The system will have to ask for users' permission in order to retrieve and use their positions without storing them.

Email addresses won't be used for commercial uses.

#### Hardware limitation

In order to work properly the system requires:

- EDGE/3G/4G/5G connection;
- iOS or Android smartphone;
- GPS/Glonass/Galileo service.

Municipalities also require:

- Modern browser, we recommend Chrome.

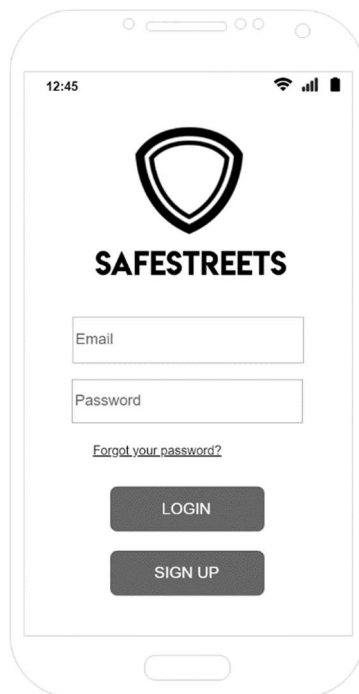
## 3 Specific Requirements

### 3.1 External interface requirements

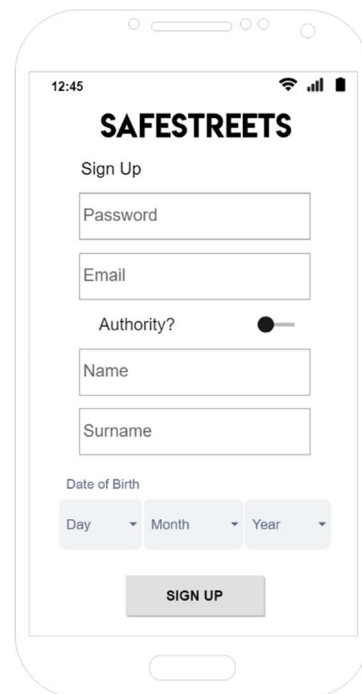
#### 3.1.1 User Interface

We will present the mockups of SafeStreets, for all our target users. Forms fields and maps are presented only for illustrative purposes as they may change during development and be different in the final product.

These mockups are intended only to give an idea of what the graphical interface of our system will be like.



*Figure 1: Basic User and  
Authority Log In*



*Figure 2: Basic User Sign Up*

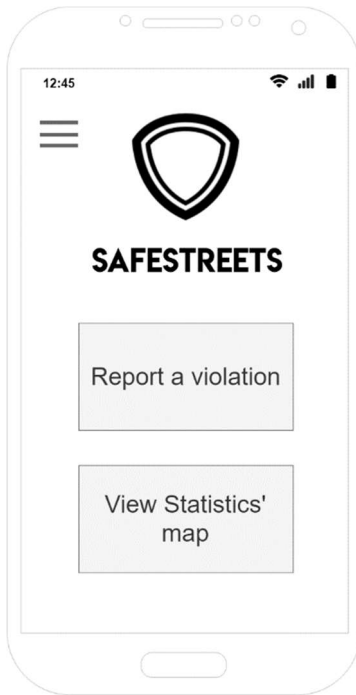


Figure 3: Basic User Main Page

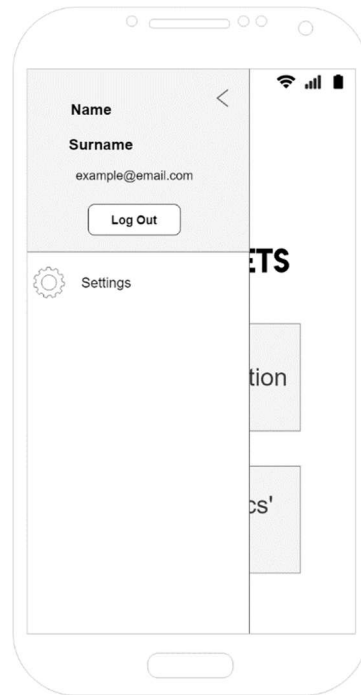


Figure 4: Basic User Side Menu

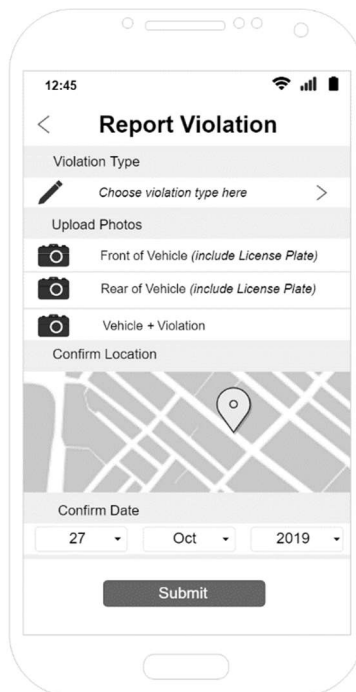


Figure 5: Basic User Report Screen

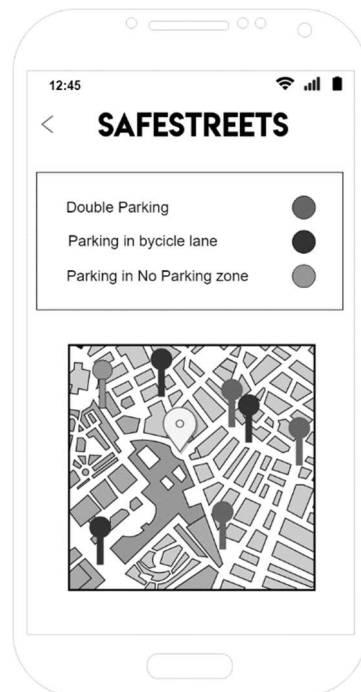


Figure 6: Basic User and Authority Map Screen

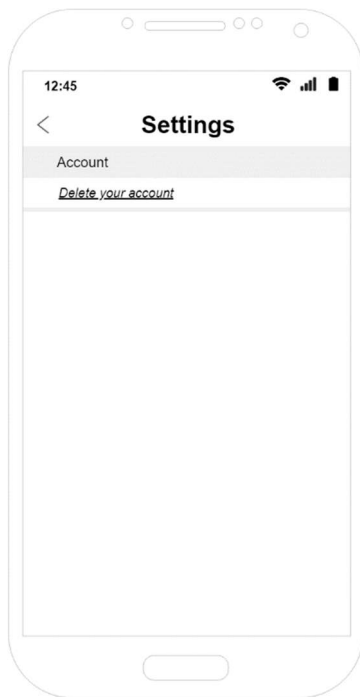


Figure 7: Basic User Setting

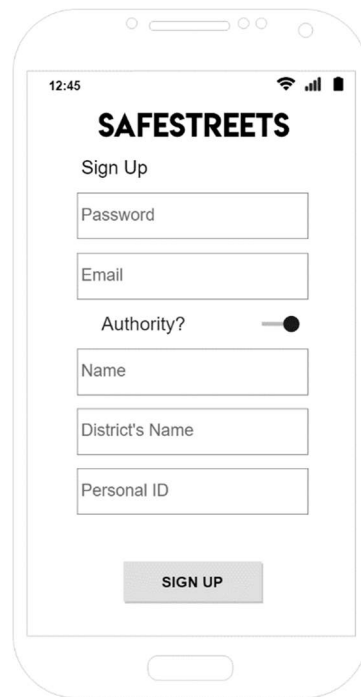


Figure 8: Authority Sign Up



Figure 9: Authority Main Screen

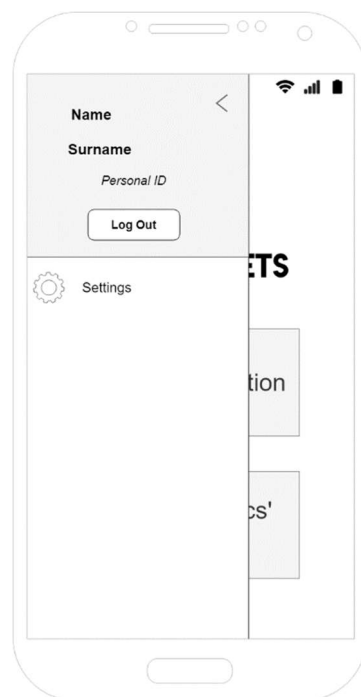


Figure 10: Authority Side Screen



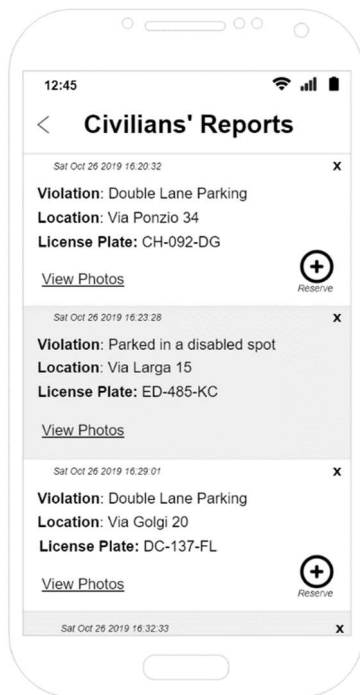


Figure 11: Authority Report Screen

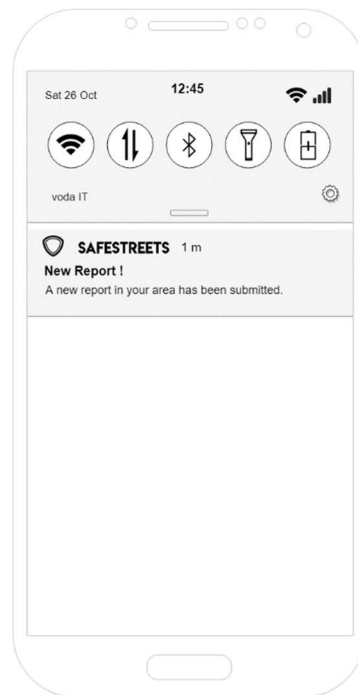


Figure 12: Authority Report Push Notification

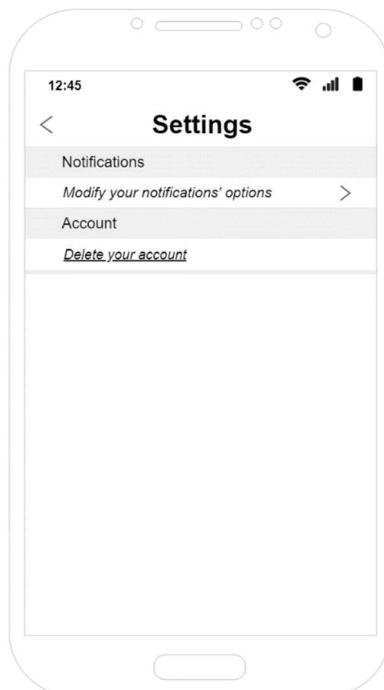


Figure 13: Authority Settings

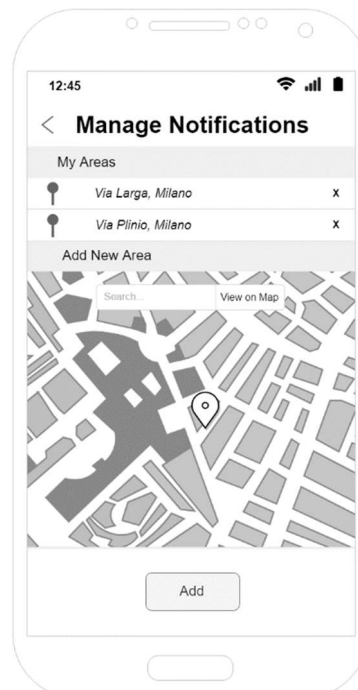
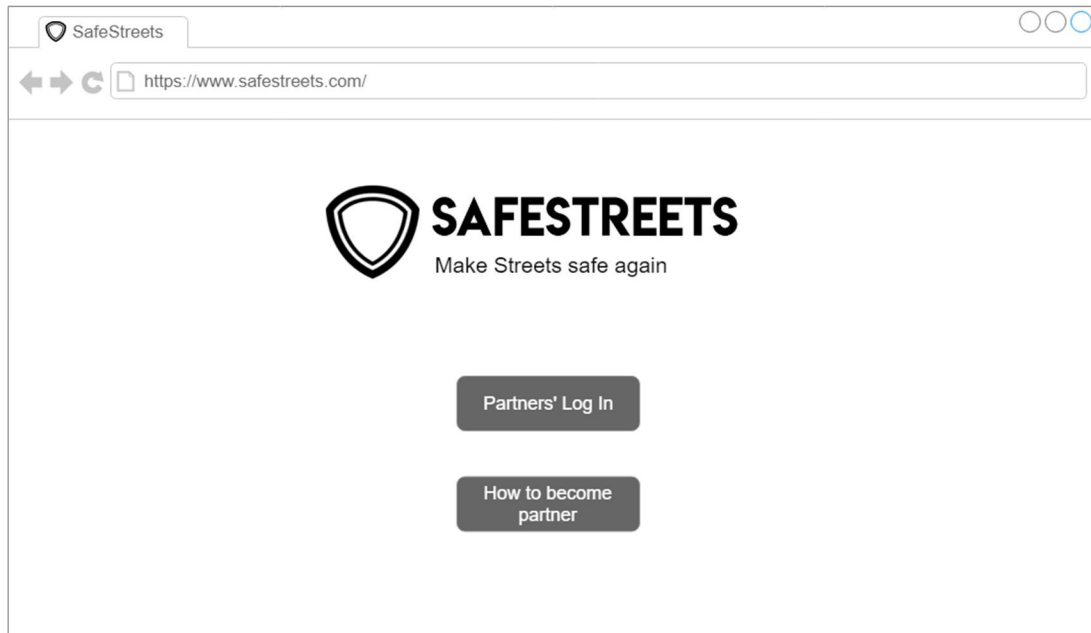
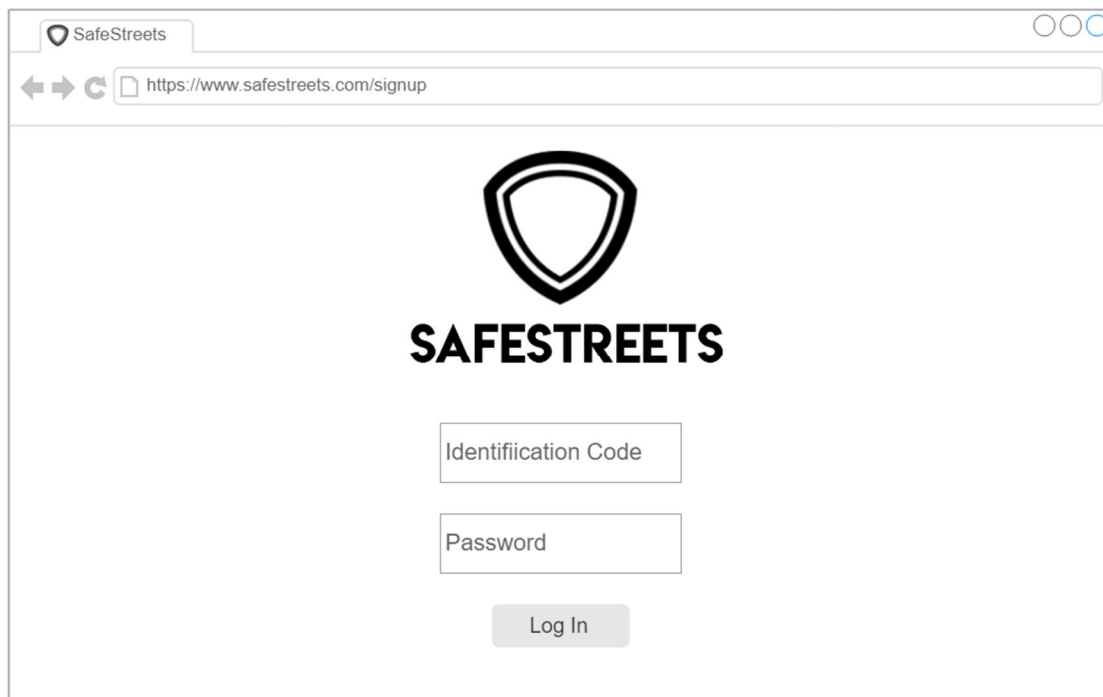


Figure 14: Authority Manage Notifications



*Figure 15: Municipality Main Page*



*Figure 16: Municipality Log In*

SafeStreets

https://www.safestreets.com/signup

**SAFESTREETS**  
Are you a *municipality*?  
Contact Us for becoming our Partner.

Name of the Municipality \*

Referencing Email \*

Referencing Name and Surname \*

Referencing Phone's Number \*

Comments

Submit

Figure 17: Municipality Sign Up

SafeStreets

https://www.safestreets.com/FH1HU1BTQVB3P75JEDHK

**SAFESTREETS**  
Safety Report on

**Raw Data**

- Double Parking:
- Invalid Park:
- Parked on Zebra Cross:
- Other:

**Accidents' Analysis**

**Possible Solutions**

Figure 18: Safety Report's Sample

### 3.1.2 Hardware Interface

The system doesn't provide any hardware interface.

### 3.1.3 Software Interface

The system has two interfaces to interconnect with the external DBMS, the former one to acquire accident data from municipality, the latter to access the DB where authorities' matriculas are stored.

Although, there are no functionalities which can be used by an external system, so the S2B will not provide any API.

## 3.2 Functional Requirement

- [R.1] The system shall allow users registration.
- [R.2] The system shall distinguish basic users and authorities accounts.
- [R.3] The system shall allow authority registration only to real authority.
  - a. The system shall be able to access the State's DB where authorities' matriculas are store.
- [R.4] The system shall guarantee unique account for each email registered.
- [R.5] The system shall allow users to access their account only if they provide correct username and password.
- [R.6] The system shall allow users to access *SafeStreets*' functionalities only after the login.
- [R.7] The system shall allow users to delete their account.
  - a. Once an account is deleted, the system shall deny access.
- [R.8] The system shall allow a basic user to upload violation data. Moreover:
  - a. The system shall let the user to choose the type of violation between a predefined list, without allow them to add a custom violation type.
  - b. The system shall read automatically the license plate from the photos uploaded.
  - c. The system shall add automatically the location of a new report through the GPS information.
  - d. The system shall deny reporting a violation with uncompleted data.
- [R.9] The system shall store internally the data.
- [R.10] Once data is store, the system shall not erase it.
- [R.11] The system shall never show the basic user who reported a particular violation.

- [R.12] The system shall allow authority to choose to receive push notifications regarding new violations affecting a given geographical area.
- [R.13] When a violation is reported, the system shall be able to generate a real time push notification to all authorities which have enable them in the area where the violation occurred.
- [R.14] The system shall allow authority to access and view all data about a violation, without contradicting [R.11].
- [R.15] The system shall not allow a basic user to access all the data about a violation.
- [R.16] The system shall allow an authority to reserve a violation. Moreover:
- a. Once a violation is reserved, the system shall not allow other authorities to reserve it.
  - b. The system shall allow authorities to know which violations have been reserved.
- [R.17] The system shall be able to classify violation data according to violation type and geographical location.
- [R.18] The system shall provide to basic users and authorities an interface able to render mined data graphically, allowing showing geographical area and violation type on a map.
- [R.19] The system shall provide to authorities an interface where are listed all the violation.

**Municipality requirements:**

- [R.M1] The system shall allow municipalities to contact *SafeStreets* in order to make a partnership.
- [R.M2] The system shall be able to access to accident data of the municipality using the API provided.
- [R.M3] The system shall be able to integrate accident data of the municipality with *SafeStreets*' DB matching the location.
- [R.M4] The system shall be able to mine the integrated data in order to find unsafe zones.
- [R.M5] The system shall provide to municipality an interface able to render mined data graphically in order to highlight the unsafe zones.
- [R.M6] The system shall be able to find correlations between accidents and violations.

[R.M7] Once identified correlations between accidents and violations, the system shall be able to generate a report with suggestions to reduce them.

### 3.3 Scenarios

#### 3.3.1 Double Parking Report

Mia is a 35 years old mother of 2 little kids, one of 5 and one of 7. She is currently unemployed and spends her time taking care of her sons. She loves watching them play outside, but unfortunately, she lives in a busy neighborhood, with vehicles always double parked. In order to avoid her children to being runover by a car that is trying to move through all the vehicles parked, she downloads the *SafeStreets* app. The system allows her to create an account after compiling the sign-up form (UC.1).

The next day, while playing with her kids, she notices a car double parked before a narrow turn: she launches *SafeStreets* app, compiles a report and after checking that all the information are correct, she submits it (UC.5).

#### 3.3.2 Reports Check

Johnny is 62 years old policeman and although his age, he is really into technology. During his career he has always tried to feel useful for the citizen and now that he is near to retirement, he prefers to do easy tasks as parking fines. Recently, thanks to his grandson Jordi, he discovered the *SafeStreets* system and started to use it.

During his patrols, usually he receives from five to seven notifications, about some violations that have been reported through the system. He opens the notifications and watches the photos: if the vehicle is indeed in violations, and if the report isn't too far from him, he locks it and drive to the location indicated and fine the car (UC.6).

#### 3.3.3 Violations Statistics Check

Lana, 23 years old Mathematics' university student, is passionate about statistics. She heard from a colleague that *SafeStreets* app allows her to check the areas around her in order to view statistics about violations that had occurred.

Interested about that new app, she downloaded it and created a new account. Then, she fills the log in form with her credentials, and press the “*Log In*” button (UC.4).

Now, every time Lana is in a new city, she opens *SafeStreets* app, clicks on the “*View Statistics' map*” button and has fun checking which violations are the most common (UC.7).

### 3.3.4 Municipality Suggestion

Sofie, 44 years old, is a municipality's employee in the road safety department. She noticed that, in the last months, the numbers of accidents have been grown dangerously, and she can't understand why. Luckily, her manager forwards her a commercial email about a new system, *SafeStreets*, that is able to cross-reference the violation data that the new system gathers from its users with the already present municipally accidents DB's data, in order to generate a safety report. Sofie, intrigued by the offer, decides to try it. She goes with her PC to *SafeStreets* webpage and contact them through the "Contact Us" form (UC.3).

After few days, she receives a confirmation email from *SafeStreets* with her credentials. After setting up the connections between the DB and *SafeStreets* itself, she requests a safety report: in a couple of hours she receives another email from *SafeStreets* with a permalink to the webpage where the safety report is present (UC.8).

## 3.4 Use Cases

For each use case we don't report the exception about connection error in order to be concise. This exception is common to all use cases and it is managed in the same way: the system notifies the issue to the user and the flow of event restart from the previous point.

<b>ID</b>	UC.1
<b>Name</b>	Register of basic user
<b>Actors</b>	Basic User
<b>Entry Condition</b>	<i>SafeStreets</i> app downloaded on basic user's smartphone
<b>Flow of Events</b>	<ol style="list-style-type: none"><li>1. Basic user clicks on <i>SafeStreets</i> app's icon entering in the Log In page. (Figure 1)</li><li>2. Basic user taps on Sign-Up button.</li><li>3. Basic user fills the Sign-Up form without checking Authority flag. (Figure 2)</li><li>4. Basic user confirms his/her data clicking on Sign-Up button.</li><li>5. The system checks the validity of data.</li><li>6. The system creates a basic user account.</li><li>7. The system returns on the Log In page.</li></ol>

<b>Exit Condition</b>	Basic user's account has been successfully created and added to the system database.
<b>Exceptions</b>	<p>5.* The field email is already taken, or date of birth is an invalid set, or a field is not filled.</p> <p>The system notifies the issue to the user and the Flow of Events returns to 3, erasing invalid fields.</p>
<b>Special Requirements</b>	

<b>ID</b>	UC.2
<b>Name</b>	Register of authority
<b>Actors</b>	Authority
<b>Entry Condition</b>	<i>SafeStreets</i> app downloaded on authority's smartphone
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>1. Authority clicks on <i>SafeStreets</i> app's icon entering in the Log In page. (Figure 1)</li> <li>2. Authority taps on Sign-Up button.</li> <li>3. Authority fills the Sign-Up form checking Authority flag. (Figure 8)</li> <li>4. Authority confirms his/her data clicking on Sign-Up button.</li> <li>5. The system checks the validity of data, it tries to match the Personal ID and the District Name with the ones on the State's DB.</li> <li>6. The system creates an authority account.</li> <li>7. The system returns on the Log In page.</li> </ol>
<b>Exit Condition</b>	Authority's account has been successfully created and added to the system database.
<b>Exceptions</b>	<p>5.* The Personal ID and the District Name do not correspond in the state's DB or the Personal ID is already linked to another account, or a field is not filled.</p> <p>The system notifies the issue to the user and the Flow of Events returns to 3, erasing invalid fields.</p> <p>5.** The access to the State's DB is denied.</p> <p>The system notifies the issue to the user and the Flow of Events goes to 7.</p>



<b>Special Requirements</b>	
-----------------------------	--

<b>ID</b>	<b>UC.3</b>
<b>Name</b>	Contact of municipality
<b>Actors</b>	Municipality employee, <i>SafeStreets</i> employee
<b>Entry Condition</b>	Browser open on municipality employee computer
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>1. Employee navigates to <i>SafeStreets</i>' website.</li> <li>2. Employee clicks on How to partner with us entering in Sign Up page. (Figure 17)</li> <li>3. Employee fills the form.</li> <li>4. Employee clicks on Contact Us.</li> <li>5. The system forwards the request to a <i>SafeStreets</i> employee.</li> <li>6. A <i>SafeStreets</i> employee reviews the request and accepts the partnership.</li> <li>7. A <i>SafeStreets</i> employee sends the credential for logging in.</li> </ol>
<b>Exit Condition</b>	Municipality's account has been successfully created and added to the system database.
<b>Exceptions</b>	<p>5.* A <i>SafeStreets</i> employee reviews the request and refuses the partnership.</p> <p>The system notifies the issue to the user.</p>
<b>Special Requirements</b>	

<b>ID</b>	<b>UC.4</b>
<b>Name</b>	App Log In
<b>Actors</b>	Basic user / authority
<b>Entry Condition</b>	<i>SafeStreets</i> app is open on basic user's or authority's smartphone.
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>1. User fills the email and password field. (Figure1)</li> <li>2. User press the Log In button.</li> </ol>
<b>Exit Condition</b>	User is successfully logged in the system and can exploit all the system services; the graphical interfaces moves to the default screen. (Figure 3 / Figure 9)

<b>Exceptions</b>	<p>2.* The system discovers that field email is invalid or that field password doesn't correspond to the one paired with the email.</p> <p>The system notifies the issue to the user and the Flow of Events returns to 1.</p>
<b>Special Requirements</b>	

<b>ID</b>	UC.5
<b>Name</b>	Report traffic violation
<b>Actors</b>	Basic user
<b>Entry Condition</b>	<p>Basic user has logged in.</p> <p>Basic user has <i>SafeStreets</i> app opened on the default page. (Figure 3)</p>
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>1. Basic user taps Report a Violation button.</li> <li>2. The app responds by presenting a form to the basic user. (Figure 5)</li> <li>3. Basic user fills the required fields: choses the type of violation in a list, takes some photo of the vehicle and checks if the location, the data and the <b>time</b> are correct.</li> <li>4. Basic user presses the submit button.</li> <li>5. The system receives the report and reads the vehicle license plate from the uploaded photos.</li> <li>6. The system notifies authorities.</li> </ol>
<b>Exit Condition</b>	The new violation is store and basic user has received an acknowledgment.
<b>Exceptions</b>	<p>3.* Basic user finds out some errors between location, data or time, he/she corrects them selecting new parameters from a list.</p> <p>The Flow of Events continuous to 4.</p> <p>4.* There are some fields that are incomplete, the system notifies the issue.</p> <p>The Flow of Events restarts from 3.</p>

	6.* There aren't any authority that has enabled notifications in the area where the violation occurred. This event is skipped.
<b>Special Requirements</b>	

<b>ID</b>	<b>UC.6</b>
<b>Name</b>	Reserve traffic violation
<b>Actors</b>	Authority
<b>Entry Condition</b>	Authority has logged in. Authority has received at least one notification (Figure 12).
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>1. Authority taps on notification.</li> <li>2. The system responds loading the Report Screen page. (Figure 11)</li> <li>3. Authority chooses one of the available reports.</li> <li>4. Authority taps on <i>View Photos</i>.</li> <li>5. Authority taps on Reserve button.</li> <li>6. The system receives the reservation and links the report with the authority.</li> </ol>
<b>Variations</b>	Point 4 of Flow of Events is optional, so it can be skipped and go from 3 to 5 directly.
<b>Exit Condition</b>	The report turns into grey and the report is reserved for the authority.
<b>Exceptions</b>	<p>5.* Authority clicks on Reserve but meanwhile the report he/she want to reserve has already been reserved by someone else.</p> <p>The issue is notified to the users and the Flow of Events terminates without verifying the exit condition.</p>
<b>Special Requirements</b>	

<b>ID</b>	<b>UC.7</b>
<b>Name</b>	Exploit mined data
<b>Actors</b>	User
<b>Entry Condition</b>	User already has logged in and he/she is on the default page.
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>1. User clicks on View Statistics Map button.</li> <li>2. The system loads the Map Screen (Figure 6).</li> </ol>

<b>Exit Condition</b>	True
<b>Exceptions</b>	
<b>Special Requirements</b>	

<b>ID</b>	UC.8
<b>Name</b>	Take suggestion
<b>Actors</b>	Municipality's employee
<b>Entry Condition</b>	Municipality's employee has <i>SafeStreets</i> ' site open.
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>1. Municipality's employee clicks on "<i>Partners Log In</i>" button.</li> <li>2. Municipality's employee logs in with his/her municipality's credentials.</li> <li>3. The system automatically provides to municipality's employee the latest safety report, composed by a set of raw data, accident's analysis and a possible solution (Figure 18).</li> </ol>
<b>Exit Condition</b>	Municipality's employee knows how to improve streets safety.
<b>Exceptions</b>	<p>2.* The system discovers that field Identification Code is invalid or that field password doesn't correspond to the one paired with the Identification Code.</p> <p>The system notifies the issue to the user and the Flow of Events returns to 1.</p>
<b>Special Requirements</b>	

In the following traceability matrix, we are mapping, for each use case, which goals directly illustrate, and which requirement are related to.

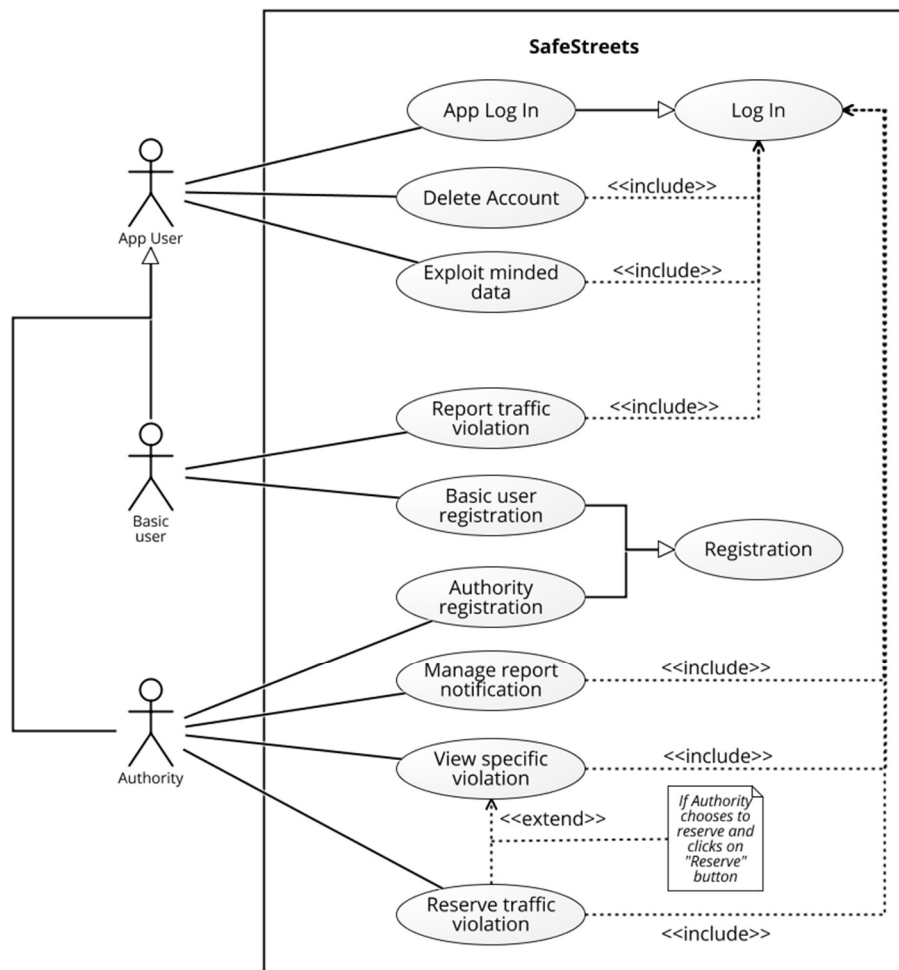
Only major requirements are listed for each use case, in order to improve readability and reduce redundancy (i.e. R.6 is not reported even though it is necessary for all use cases).

Use Case ID	Goal ID	Requirement ID
UC.1	/	R.1, R.2, R.4

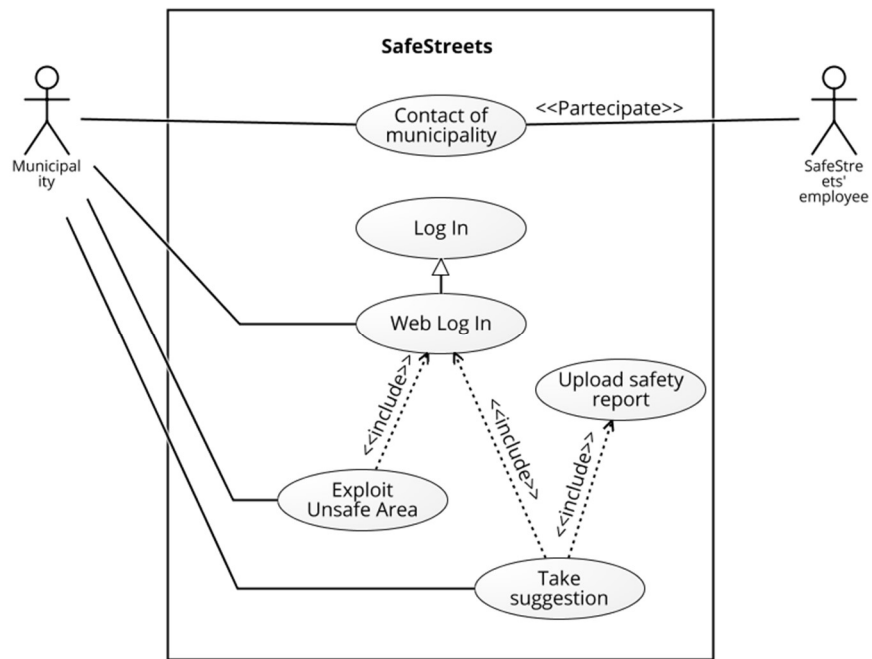
UC.2	/	R.1, R.2, R.3, R4
UC.3	/	R.M1, R4
UC.4	/	R.5
UC.5	G.BU1	R.8, R.9, R.13
UC.6	G.A1, G.A2	R.12, R.13, R.14, R.16, R.19
UC.7	G.BU2, G.A3	R.17, R.18
UC.8	G.M1	R.M6, R.M7

*Table 3: Traceability Matrix*

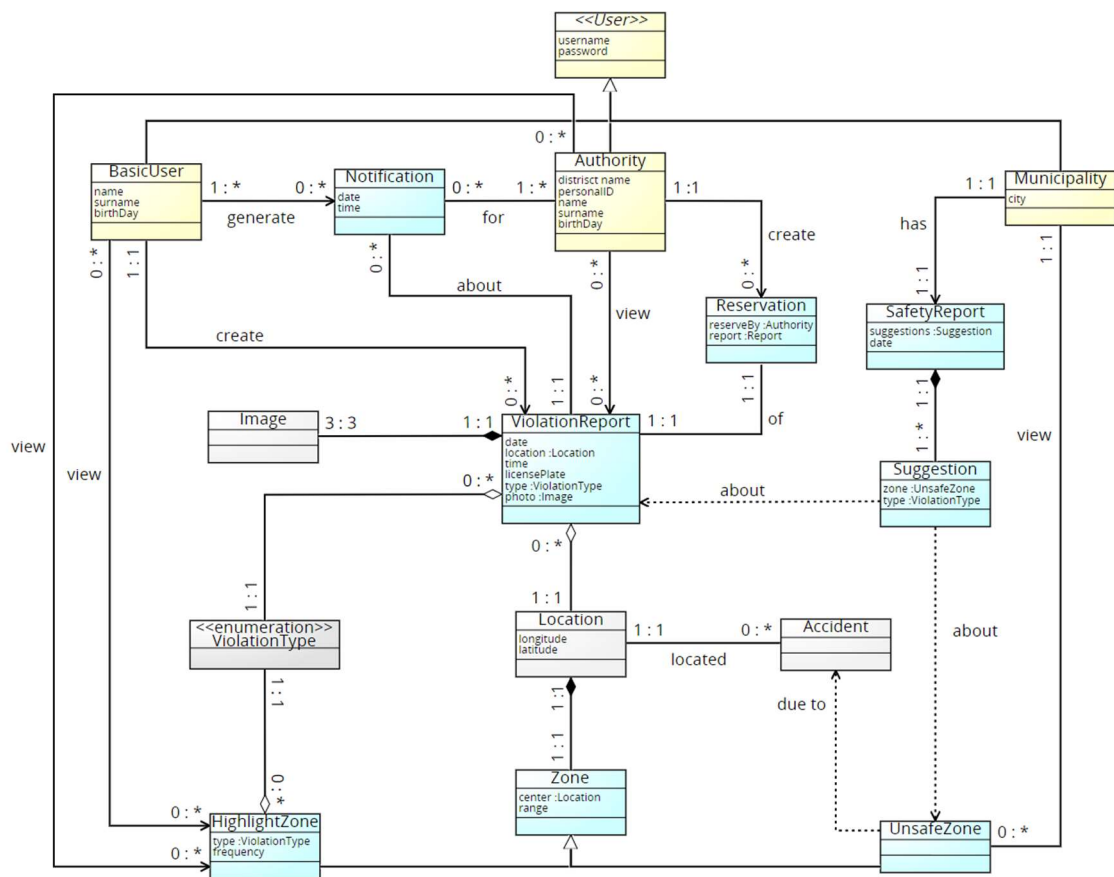
### 3.4.1 UML Modelling



UML 1: Use Case Diagram of the app



UML 2: Use Case Diagram for the web



UML 3: Class Diagram

### 3.5 Performance requirements

//todo

### 3.6 Design constraints

#### 3.6.1 Standard compliance

//todo

#### 3.6.2 Hardware limitations

The smartphone app version of *SafeStreets* requires Android/iOS operating system, geo-localization system, internet connectivity and a built-in camera to the smartphone on which it is installed, in order to exploit its basic functionalities.

For the municipalities it is also required a modern browser, we recommend Chrome.

#### 3.6.3 Any other constraints

There aren't any other constraints.

### 3.7 Software system attributes

#### 3.7.1 Reliability

//todo

#### 3.7.2 Availability

The system is expected to be available 99.99% of the time. In case of failure, it shall be able to recover with in a brief amount of time.

#### 3.7.3 Security

The system shall store securely users' credentials and data collected into its database.

The system shall not allow authorities to retrieve any personal information (including, but not limited to name, surname) about the basic users whose reports are being submitted into *SafeStreets*' database.

#### 3.7.4 Maintainability

//todo



### **3.7.5 Portability**

The system shall run on most Android/iOS devices, and on most modern browser.

## 4 Alloy modelling

```
//TODO

open util/boolean

sig Location {}

sig Username {}

abstract sig User {
    username: one Username,
    position: one Location
}

sig BasicUser extends User {}

sig Authority extends User {
    //Notification area are simplified as a set of Locations
    notifArea: set Location,
    //Notification are simplified as a set of Reports
    notifReceived: set Report
}

abstract sig ViolationType{ }

sig DoubleParking extends ViolationType{}

sig ZebraCrossParking extends ViolationType{}

sig NoParking extends ViolationType{}

sig Report {
    type: one ViolationType,
    position: one Location,
    /* True if this report is reserved by an Authority,
    false other way */
    reserved: one Bool
} {
    /* Report can be reserved if and only if an Authority
    has reserved it */
    reserved = True iff
        this in Authority.(S2B.reservation)
}

/**
S2B with all its data
*/
sig S2B {
    // violation report made by a Basic User
    violationReport: BasicUser -> set Report,
    // Reservation on a report by an Authority
    reservation: Authority -> set Report
}
```

```

} {
  // No violationType outside the system
  Report.type= ViolationType
  // No report outside the system
  BasicUser.violationReport = Report
  // No two BasicUser for same Report
  no r: Report | #violationReport.r != 1
  // No two Authority for same report reservation
  no r: Report | #reservation.r > 1
}

fact UniqueUsername {
  no disj u, u' : User | u.username = u'.username
}

/**
  Authority a reserve the report r
  s is the updated system
  s' is the old system
*/
pred reserve[s, s': S2B, a: Authority, r: Report] {
  s.violationReport = s'.violationReport
  // r can't have been already reserved
  r not in Authority.(s'.reservation)
  s.reservation = s'.reservation + a->r
  r.reserved = True
}

/**
  BasicUser u report a new violation of type vt
  s is the updated system
  s' is the old system
*/
pred addReport[s, s': S2B, u: BasicUser, r: Report, vt:
ViolationType] {
  s.reservation = s'.reservation
  r.position = u.position
  r.reserved = False
  r.type = vt
  s.violationReport = s'.violationReport + u->r
  notify[r]
}

/**
  Notify each authority who has enabled notification in
  the zone of the report
*/
pred notify[r: Report] {
  all a, a': Authority |
    a.username = a'.username and
    r.position in a.notifArea => (
      a.notifArea = a'.notifArea and
      a.notifReceived = a'.notifReceived + r )
}

```

```

}

/**
Verify that each time a BasicUser reports a violation,
all the interested Authorities are notified
*/
assert pushNotification {
  all u: BasicUser, s, s': S2B, r: Report, vt:
  ViolationType |
    all a: Authority |
      u.position in a.notifArea and addReport[s,s',u,r,vt]
      => r in a.notifReceived
}

check pushNotification
run reserve
run addReport
run {#S2B=1} for 3

```

Date	Falconi	Galli	Theme
23/10	1	2	Problem analysis
24/10	3	1.5	Goals, Definitions
25/10	3	1.5	UI Design, 2.4 paragraph
26/10	3	3	UI Design, Requirement
28/10	1	1	Use Cases
29/10	1.5		UI
30/10	2	3	Use Cases, Scenarios
1/11	2		<u>Section 2</u>
2/11	2	4	Section 3 UMLs Alloy
3/11		1.5	<u>Alloy</u>