POLITECNICO
MILANO 1863

Software Engineering 2 - Mandatory Project

AY 2019/2020

# RASD

Version 1.1 – 9/12/19

Maserati Alessandro - 944593

# Table of Contents

# 1. Introduction

## 1.1 General Purpose

Safe Streets is a software application with the main aim of providing end users with the possibility to report to authorities traffic violations (in particular parking violations). Another aim of the system is allowing both end users and authorities to get information mining the received notifications.

The deepest purpose of the application is to provide authorities a way to know where road works (clearer traffic signals) or municipality interventions are necessary with the aim of improve street safety (in particular from the point of view of pedestrians and bikers).

A detailed description of the functionalities offered to end users and authorities is provided in Section 2.2.

## 1.2 Goals

- [G1] – Allow end users to report traffic violations.
  - [G1.1] – Allow end users to visualize all previous personal reports.
- [G2] – Allow end users and the authorities to visualise which are the streets with the highest frequency of violations.
  - [G2.1] – Allow both users to visualize specific “Traffic Violation Maps”, using only a part of the total amount of data.
- [G3] – Allow authorities to know the vehicles that commits one or more violations.
  - [G3.1] – Allow authorities to know the vehicles that commits violations, using only a part of the total amount of data.
- [G4] – Allow authorities to check and verify the consistency of traffic violations reported by end users.

## 1.3. Scope

Safe Streets is a software application thought to improve road safety. Using Safe Streets, common people can notify authorities when they see traffic violations only taking and sending one or more pictures of that and specifying the type of violation (e.g. abusive bike lane or sidewalk parking, double parking, parking in disabled reserved spaces…) from their mobile devices. Authorities, on the other side, after the run of an algorithm that reads the license plates from the pictures, can check the reported violations and delete from the "Report list" the incorrect ones (e.g. misunderstanding of traffic signals by the user, special permits, special events occurring in that areas…). This is done before data are saved into the database, to be sure that only the correct reports can modify the mined information (e.g. the "Traffic Violation Map").

Contacts between stakeholders and some authorities, in order to find some public institution to start the project with, have already been made before the drawing up of this document. Without this agreement, the development of the application can be useless, because of the absence of enabled people to verify the correctness of violation reports.

From their devices, both end users and authorities can also see which are the streets and areas with a high number of traffic violations. In addition, authorities are allowed to see a list of the vehicles (identified by the license plates) which commit traffic violations. These functions are also available exploiting a filtered research by type of violation, street or date (see Section 3.1 and 3.2).

### 1.3.1. World and shared phenomena

The following table illustrates some of the world and shared phenomena related to the use of Safe Streets, referring to the Jackson & Zave distinction.

| **Phenomenon** | **Shared or not** |
|---|---|
| End user sees a traffic violation | No |
| End user takes a picture of the violation | Yes |
| End user fills and sends the "Traffic violation form" | Yes |
| End user/Authority searches the highlighted streets | Yes |
| Authority wants to know the vehicles that committed the most violations | No |
| Authority investigates the list of vehicle plates | Yes |
| Authority checks a violation report | Yes |
| Authority deletes a certain violation report | Yes |

The 2nd phenomenon "*End user takes a picture of the violation"* is considered to be shared because the picture is taken opening the camera from the application (when the user chooses to report a traffic violation).

## 1.4. Definitions, acronyms and abbreviations

### 1.4.1. Definitions

- *Authority*: public institution related to street safety (e.g. municipality, local police).
- *End user*: people (unrelated with authorities) using Safe Streets application with the aim of report traffic violations and know the streets where the most violations occur.
- *Customer*: a "general" user of the SafeStreets service, can be an end user or an authority.
- *Traffic violation form*: it's the form that an end user must fill in the app on his device when he wants to notify a violation.
- *Traffic Violation Map*: it's a map, build exploiting a Map Service, that highlights the streets with different colours, depending on how many traffic violations have been notified there.

- *Violation report list*: the queue containing the violation reports which have not yet been checked by authorities. Obviously visible only to authorities.
- *My Reports*: section visible on End User mobile app in which all the reports made by that end user are shown, underlying the current state of each report (it can be Unchecked, Confirmed or Rejected depending on the decision of the authority).
- *Rules for a well-formed traffic violation report:* it's a list of rules and advices useful to those end users who wants to simplify the checking process by the authority who investigates their report.

### 1.4.2. Acronyms

- RASD: Requirement and Analysis Specification Document
- GPS: Global Positioning System

### 1.4.3. Abbreviations

- [Gn]: n-th goal
- [Dn]: n-th domain assumption
- [Rn]: n-th requirement

## 1.5. Revision History

- 10/11/19: Version 1.0
  - First Release
- 9/12/19: Version 1.1
  - Section 2.2.5: slight update to Checking a traffic violation report function.
  - Section 1.2: added goal [G1.1]
  - Section 3.2.4: added [R18] to Traceability matrix.
  - Section 3.2.4: slight changes to [R14].

- Section 1.4.1: added the *Traffic Violation Map* definition
- Section 3.1.1: added the *MyReports Section* mockup
- Section 3.1.1: slight changes to *Traffic Violation Map* and *Violation report detail* mockups

## 1.6. Reference Documents

- Specification Document "*SafeStreets Mandatory Project Assignment*"
- 29148 – 2011 – ISO/IEC/IEEE International Standard
- Alloy Guide: https://alloytools.org/quickguide

## 1.7. Document Structure

This RASD is composed by 6 macro sections:

**Section 1** is an introduction illustrating the general purpose, the goals and the scope (underlining the world and shared phenomena) of Safe Streets application. Furthermore, details about terminology, revision history and reference documents are provided.

**Section 2** is a more detailed overall of the application. The first subsection *Product perspective* includes class diagrams and state-charts. Furthermore, a summary of the major functions of the system and the description of the user's characteristics are shown. Finally, constraints and domain assumption are listed to have a complete view of the world domain of the application.

**Section 3** represents the main chapter of the document. It firstly contains some mockups of the app interfaces; then, the part of the document related to functional requirements is described in detail showing scenarios, use case diagrams, use cases, sequence diagrams and finally listing the requirements, linking them to goals and domain assumption. Lastly, non-functional requirements, design constraints and some attributes that the system shall have are illustrated.

**Section 4** contains the Alloy model, including some examples of worlds generated from it.

# 2. Overall description

## 2.1 Product perspective

SafeStreets is a software product to be used by a large number of people in order to collect much information to be integrated with those of the authorities. It is a stand-alone product and in future may be updated with several new functionalities. It is thought to be exploited via mobile devices' app.

## UML Class diagram

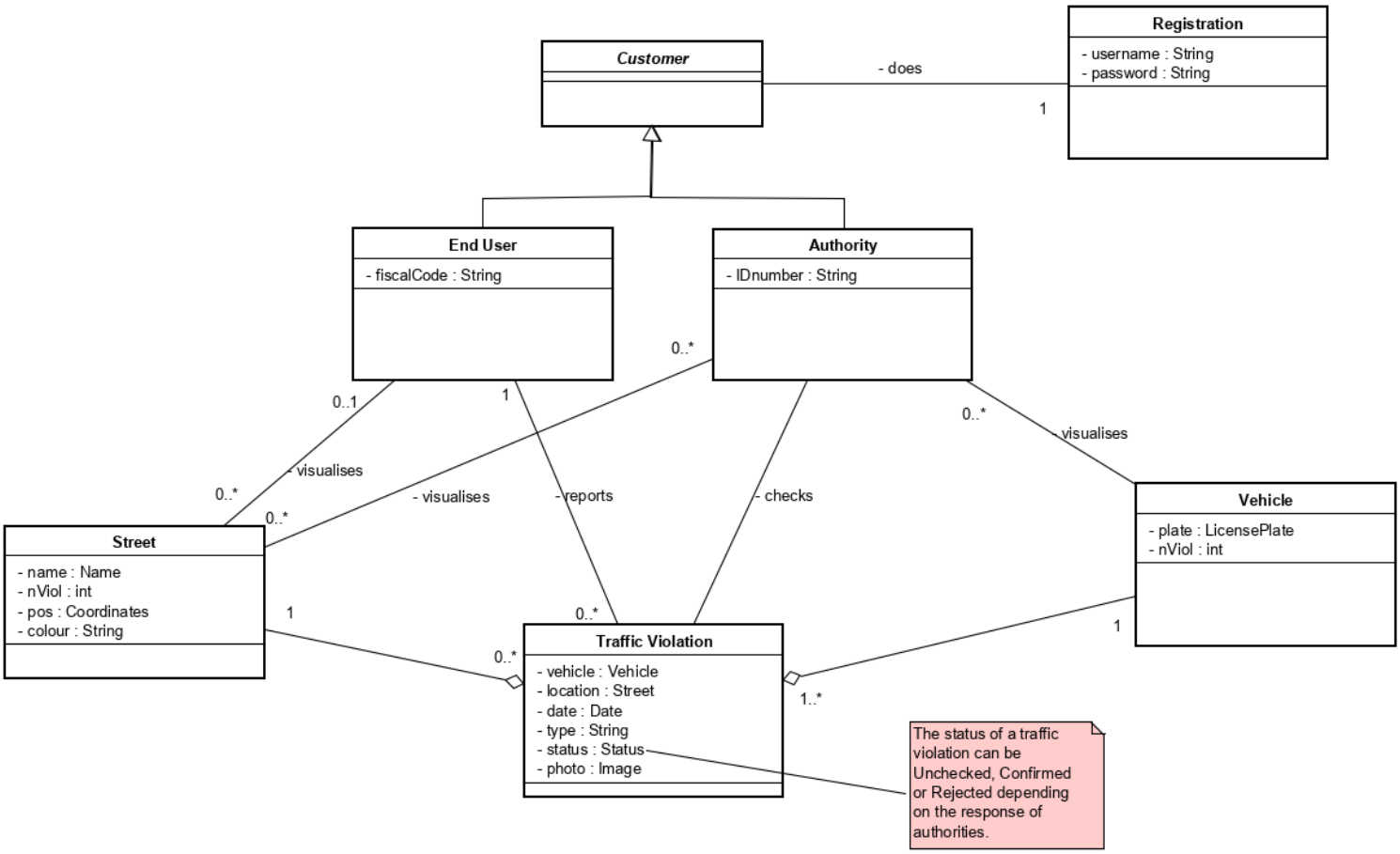



*Figure 1: Class Diagram*

State chart diagrams

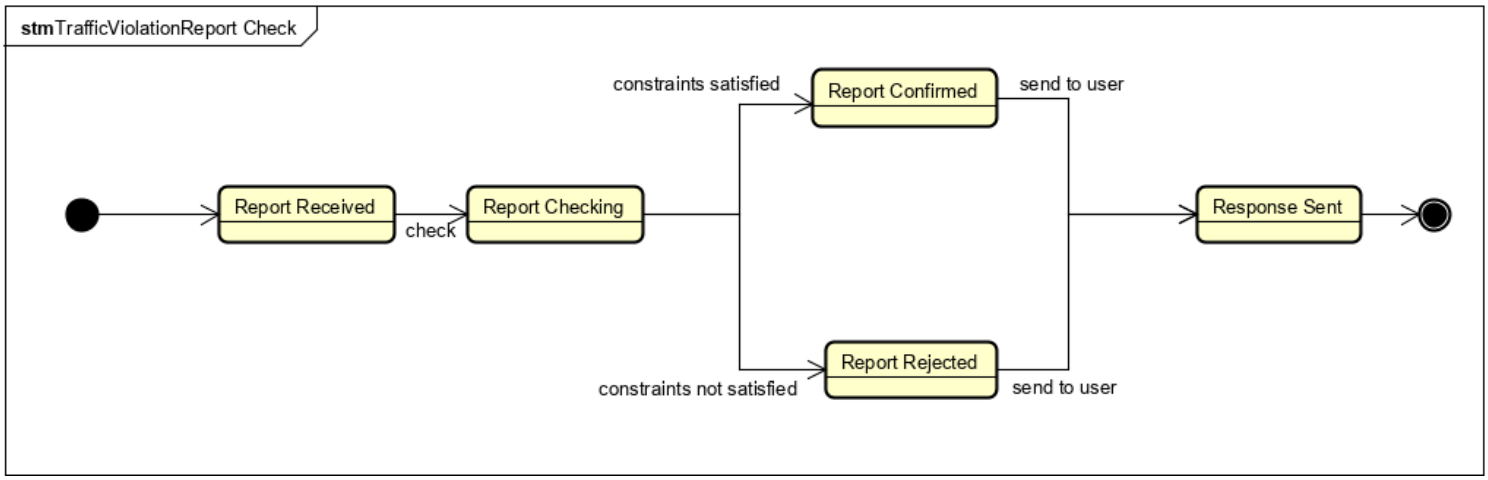



*Figure 2: Traffic Violation Report Check State Chart*

In this application, the concept of state can only be found in the Traffic Violation Report process. In particular, the report notified by an End User is firstly considered Unchecked, then, after the check process, it becomes Confirmed or Rejected, depending on the choice of the authority. Lastly, the response evaluation is sent back to the end user that made it.

## 2.2 Product functions

In this sub-section are listed and described in detail the functions that SafeStreets users must be allowed to exploit. The details of these functions can be deeply explored in Section 3.1.1, where the mockups of the application are shown.

### 2.2.1 Registration and login

This function is exploited by both end users and authorities. They must be able to register to the system, filling a registration form with personal data. While end users are identified by their fiscal codes, authorities must specify their ID number. After the registration, customers must be able to login with a username and a password, both chosen during the registration phase.

### 2.2.2 Reporting a traffic violation

When the end user sees a traffic violation, he must be able to notify it using the SafeStreets application. In particular, he has to fill the "Traffic violation form" taking one picture (or more, depending on the situation, in order to take every necessary detail) of the violation (some constraint for the validity of the report must be respected, e.g. the license plate and the background of the vehicle must be clearly visible, listed in the "Rules for a well-formed traffic violation report") and specifying the type of violation. The GPS position of the violation and the current date and hour are automatically reported by the system, exploiting a map service.

After the user sends the "Traffic violation form", the report goes into the queue of unchecked reports, visible only from authorities app.

### 2.2.3 Visualizing the highlighted streets

This function must be exploited by all users. The user must be able to see on a map which streets are places of the most violation. Different colours can be used, for instance a street where more than 10 traffic violations per day are notified is highlighted with red, another with only 3-5 violation per day with yellow and the ones with less than 2 violations per day are shown with green colour. Furthermore, the user is allowed to search on the map a particular street. Specifying some filters is also possible to generate the map created using only a part of the reported traffic violation (e.g. only the violations occurred on a specific date or the ones with a specific type). Also, this function is possible because of the exploiting of a map service.

### 2.2.4 Visualizing the most notified vehicles

The following function is reserved to authorities because of privacy rules. Authorities must be able to see which vehicles, identified by license plates, commits traffic violations. Using different filters, authorities are allowed to see the list of vehicles grouped by kind of violation committed, date or area, to mine information with the aim of organize police controls or road working to improve streets safety.

### 2.2.5 Checking a traffic violation report

Authorities must be allowed to check every violation reported by end user in order to delete the ones which are not clear enough, fake or already been reported. This can be done only clicking a button during the visualization of the report. After that, the response is notified to the end user who made the report. Furthermore, the verified or rejected reports are no more visible in the Traffic Violation Reports queue of all authorities.

## 2.3 User characteristics

The actors of the application are the following:

- End users: “normal” people who is allowed to notify authorities when traffic violations occur. This users’ knowledge about road issues is assumed to be “limited” (errors in judgement due to special causes are possible). In order to ensure a proper service, as described in the previous section, violation reports received from these actors must be checked by authorities before their insertion into the SafeStreets database.
- Authorities: municipality, local police, traffic police or any other public institution registered to the system with the aim of mine information from the end users’ reports. Authorities are verified by the system and identified by a unique ID number.

## 2.4 Assumptions, dependencies and constraints

Domain assumptions:

[D1] – To all authorities registered to the system, taking decisions (approve or delete) about the correctness of the violation reports, is allowed.

[D2] – Every authority registered to the system is assumed to have the necessary “traffic rules” knowledge and to be able of taking decisions consulting vehicle and traffic authorities data (vehicles special permissions…).

[D3] – Each authority is uniquely identified by an ID Number (it must be used during the registration process).

[D4] – Every end users’ report is checked in less than 24 hours form the notification.

[D5] – The device used to run SafeStreets mobile app must be able to take pictures.

[D6] – The device used to run SafeStreets mobile app must be able to provide the GPS position of the end user.

[D7] - The device used to run SafeStreets mobile app must be connected to internet.

Notes:

[D1] is taken with the aim of simplify the problem and the drawing up of this document. Obviously, in real world, it can’t be possible because only a part of authorities is enabled to take decisions about traffic violation (e.g. local police or traffic police, not administrative authorities of municipality).

With regards to [D2], the information can be taken from vehicle registration authorities or from the municipality.

[D5], [D6], [D7] may seem like technological constraints. To simplify the requirements design process, they are listed as domain assumption.

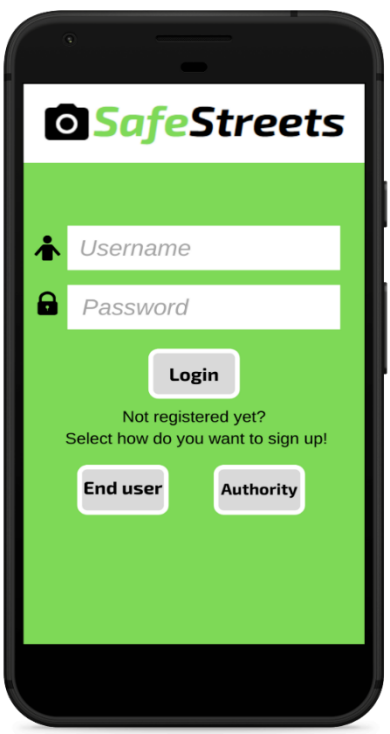
# 3. Specific Requirements

## 3.1 External Interface Requirements

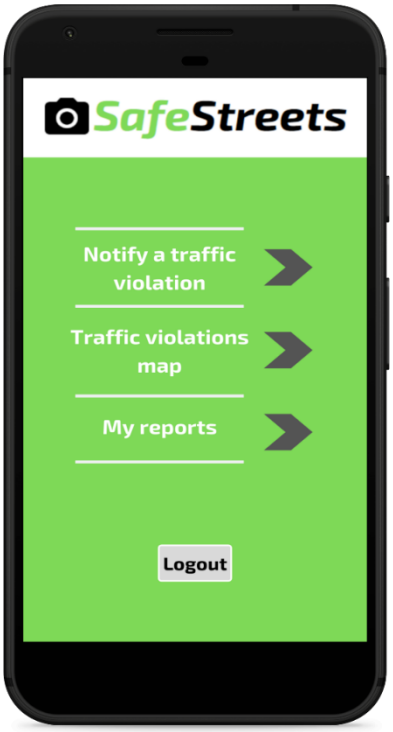
### 3.1.1 User Interfaces

The following mockups represent the main screenshots of the mobile application to be (illustrating both end users and authorities' functionalities):

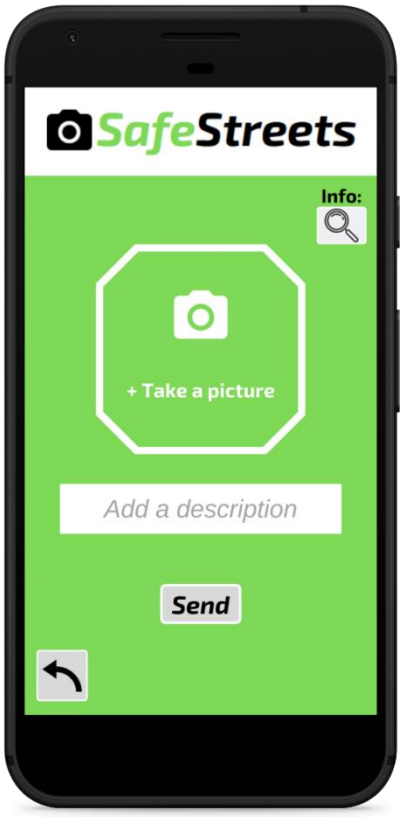
*Figure 3: SafeStreets Mobile App mockups*



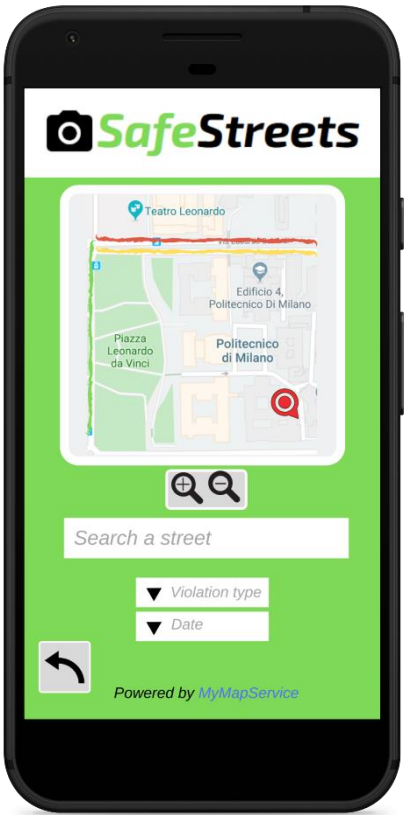

*End User and Authority login*



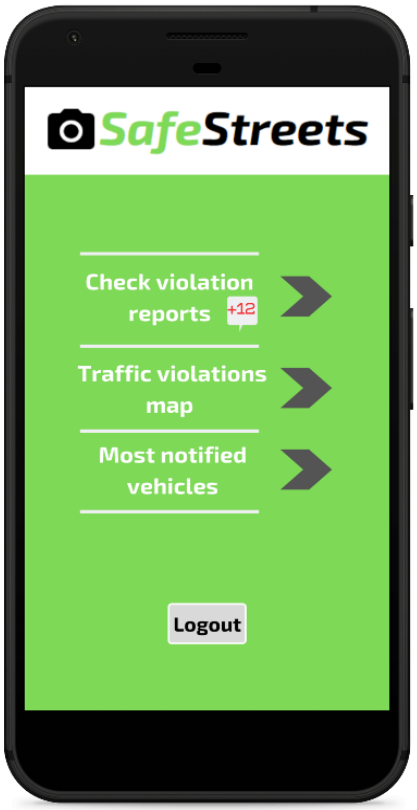

*End User main menu*



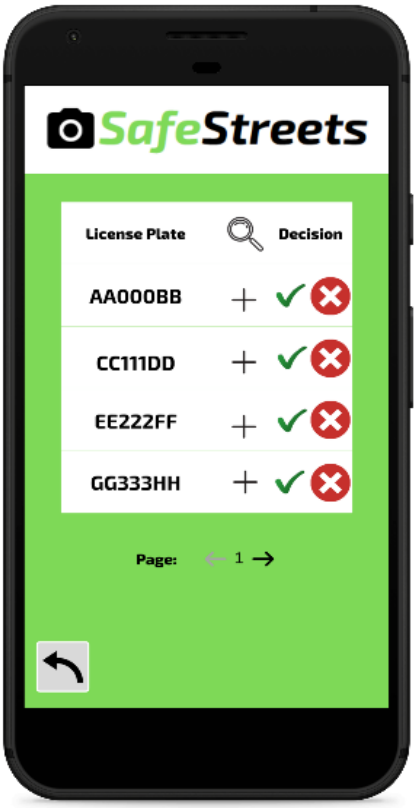

*End User violation form*



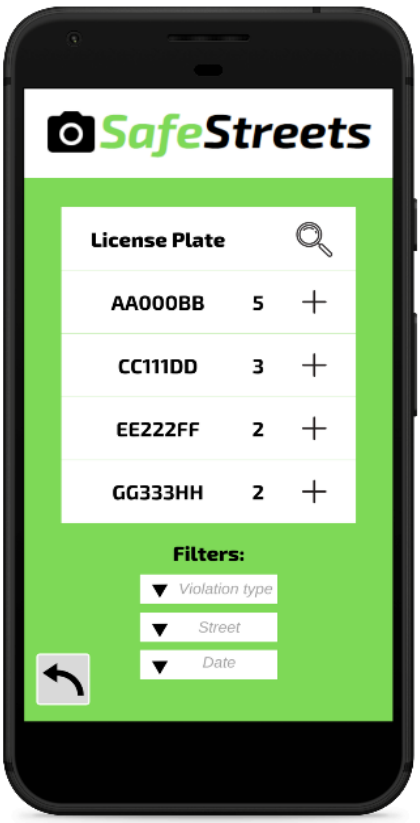

*Traffic violation map*



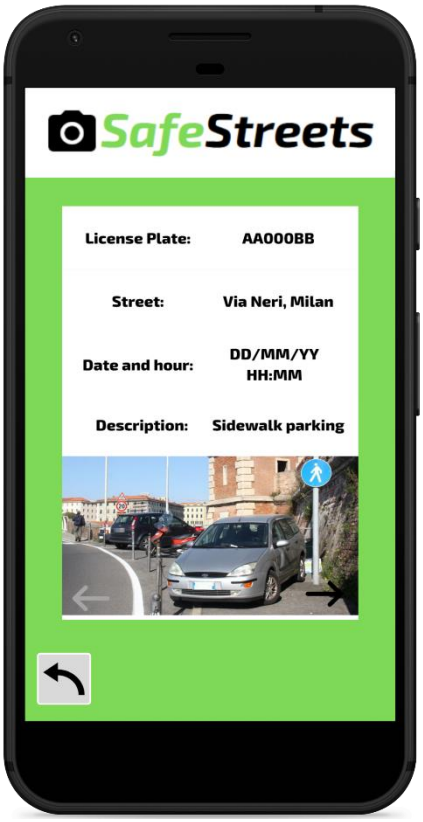

*Authority main menu*



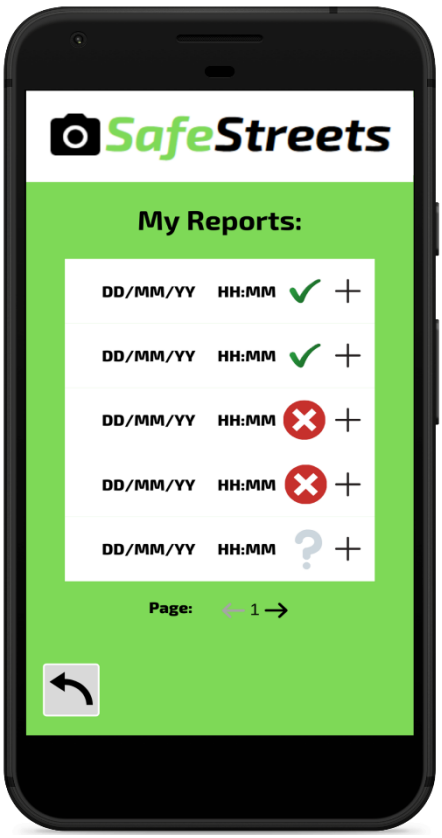

*Authority reports check*




*Authority most notified vehicle*




*Violation report detail*




*MyReports Section*

## 3.2 Functional Requirements

### 3.2.1 Use Case Diagram

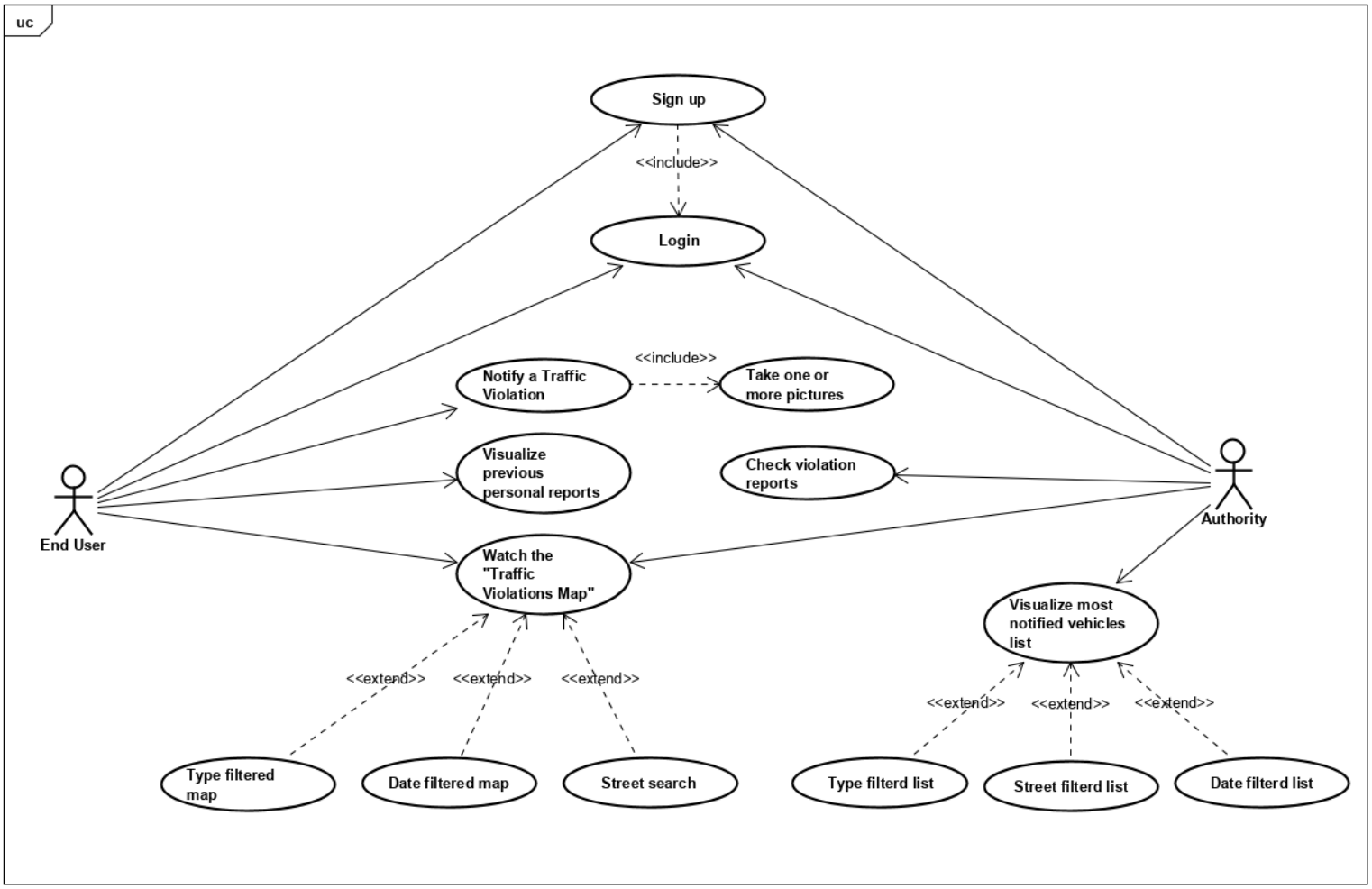



*Figure 4: Use case diagram*

The <<include>> arcs from each functionality (e.g. Notify a traffic violation report, Check violation reports…) to the Login functionality are omitted to simplify the readability of the diagram.

### 3.2.2 End User Scenarios and Use Cases

**Scenarios**

1. Arthur is a Politecnico di Milano's professor and he is also a biker. He loves riding his bike every morning going to work. During the new academic year, Bob has noticed that many vehicles are parked in the middle of bike lanes. Tired of the slalom between cars that he must do every morning, he decides

to use SafeStreets app to notify authorities all the violations he sees, with the hope of more frequent traffic police controls.

2. Polly is an old woman's granddaughter and she loves too much her granny. Unfortunately, her granny is disabled, so, every Thursday she has to take her to the supermarket. Reached her destination, she sees that the only place in front of the supermarket reserved for people with disabilities is filled by a vehicle not showing any disability badge. She can't tolerate such a lack of respect, so she takes her smartphone and send to authorities a picture of the violation.
3. John, Arthur's brother, loves cycling too. He wants to know how many traffic violations occur in the route from home to his new work office. Arthur told him that SafeStreets app allows users to know exactly what John is in search of. He downloads the app on his smartphone and fortunately discover that all the streets in his route are safe.
4. Finn is a regular user of SafeStreets service. During the last week, he has notified five traffic violations. Now, he wants to know whether his reports have been checked and approved from authorities. From his mobile phone, he opens the SafeStreets app, he selects his personal area available in My Reports page and find that all his reports have been checked. Good job Finn!

## Use Cases

| | |
|---|---|
| **Name** | 1. Sign Up |
| **Actor** | End User |
| **Entry conditions** | The End User has opened SafeStreets app on his mobile device. |
| **Events flow** | 1. The user chooses the "End User Sign Up" option.<br>2. The user fills the Registration Form with all the required information.<br>3. The user chooses the confirmation option. |

| | |
|---|---|
| | 4. The system saves the data. |
| **Exit conditions** | The user is registered and now he can login to the system using the specified credentials. |
| **Exceptions** | • The user doesn't fill all the mandatory fields.<br>• The user was already registered (the fiscal code or the email inserted are already in the database).<br>In all these cases the system notifies an Error Message and returns to the Sign-Up page. |

| | |
|---|---|
| **Name** | 2. Login |
| **Actor** | End User |
| **Entry conditions** | 1. The user has opened SafeStreets app on his mobile device.<br>2. The End User is already registered to the system. |
| **Events flow** | 1. The user chooses the "Login" option.<br>2. The user fills the login form with his email and password.<br>3. The user chooses the confirmation option. |
| **Exit conditions** | The user is logged in and can exploit SafeStreets functionalities. |
| **Exceptions** | 1. The user enters the wrong email.<br>2. The user enters the wrong password.<br>In both cases, the system warns the user and suggests to re-insert the wrong fields. |

| | |
|---|---|
| **Name** | 3. Report a Traffic Violation |
| **Actor** | End User |

| | |
|---|---|
| **Entry conditions** | 1. The user has opened SafeStreets app on his mobile device.<br>2. The End User is already logged into the system. |
| **Events flow** | 1. The End User chooses the “Report Traffic Violation” option.<br>2. The End User chooses the “Take a picture” option.<br>3. The End User takes one or more pictures exploiting his device’s camera.<br>4. The End User enters the description of the traffic violation.<br>5. The End User chooses the Send option. |
| **Exit conditions** | The traffic violation is registered, and authorities can check it. The Traffic violation is also visible in user’s “My Reports” section. |
| **Exceptions** | The End User tries to send the report having not taken any picture or entered the description. An error is notified by the system. |

| | |
|---|---|
| **Name** | 4. Visualizing the “Traffic Violation Map” and focusing on a specific street |
| **Actor** | End User |
| **Entry conditions** | 1. The user has opened SafeStreets app on his mobile device.<br>2. The End User is already logged into the system. |
| **Events flow** | 1. The End User chooses the “Traffic Violation Map” option.<br>2. The End User enters a street name.<br>3. The End User chooses the “Search” option. |
| **Exit conditions** | The street searched by the End User is visualized on the map. |

| | |
|---|---|
| **Exceptions** | The street name entered doesn't exists. In this case a Not Found Error is notified. |

| | |
|---|---|
| **Name** | 5. Visualize personal violation reports |
| **Actor** | End User |
| **Entry conditions** | 1. The user has opened SafeStreets app on his mobile device.<br>2. The End User is already logged into the system. |
| **Events flow** | 1. The End User chooses the "My Reports" option. |
| **Exit conditions** | The list of the user's previous reports is visualized, including the check process decision (if already taken). |
| **Exceptions** | The End User hasn't notified a traffic violation yet. The user is informed of this with a warning. |

### 3.2.3 Authority Scenarios and Use Cases

**Scenarios**

1. Oswald is an employee of Milano's municipality. His boss told him that a significant amount of money is available for some new road works with the aim of improve the streets safety of the municipality. He also entrusted Oswald to select some streets which streets need an improvement of traffic signals for the safety of bikers. Oswald opens SafeStreets app and searches on the available Traffic Violation Map the streets in which the most of bike lanes illegal car parks occur. Then, he notifies his boss the result of his research.
2. Thomas is a diligent traffic policeman. He must organize the control routes of his colleagues for the next week. So, he searches on the Traffic Violations Map of SafeStreets application where the most of illegal car-parking (specifying this type of violation) are reported with the aim of generate

many parking tickets. Now, Thomas exactly know where to send his colleagues.

3. The Milano's traffic police must deliver a statistical survey about the vehicles that commits illegal disabled reserved car parks. Grace, the survey manager, uses SafeStreets application in order to merge the information mined from the traffic tickets generated in the last year with the information taken from SafeStreets end users. In particular, she uses the available list of vehicles license plates filtered by type of violation.
4. Michael, a municipality employee, is the responsible organizer of a gastronomic event that took place last week in the city of Milan. This event was the cause of many complaints from people who lives in that area. Michael wants to know whether the events causes some traffic disease in order to be prepared for the next year edition. He uses SafeStreets search function and he discovers that more than forty different vehicles were notified in the specified street in the date of the event with cause: illegal car parking on bike lanes and sidewalks.

**Use Cases**

The Use Cases related to the registration and login processes are not shown because they are identical to those related to the End User, except for the fields to fill during registration phase (more information to verify the identity are required) and for the field required to login (not the email but the authority ID Number).

Furthermore, Use Case 4 is related also to authorities.

| **Name** | 6. Traffic violation report check |
|---|---|
| **Actor** | Authority |
| **Entry conditions** | The Authority is logged into the system. |
| **Events flow** | 1. The Authority chooses the "Check violation reports".<br>2. The Authority chooses a report to check from the list.<br>3. The Authority chooses whether to confirm or reject the report. |

| | |
|---|---|
| **Exit conditions** | The selected report is confirmed and stored into the database or rejected. Now, on the user's application, the choice of the authority is visible. |
| **Exceptions** | No exceptions. |

| | |
|---|---|
| **Name** | 7. Search for the violations with a specific type (on the map) |
| **Actor** | Authority |
| **Entry conditions** | The Authority is logged into the system. |
| **Events flow** | 1. The Authority chooses the "Visualize traffic violations map" option.<br>2. The Authority digits the specific type of violation in the "Type box".<br>3. The Authority chooses the "Search" option. |
| **Exit conditions** | The "Traffic violation map", related only to the specified type of violation, is shown. |
| **Exceptions** | Violations with the specified type of violation are not present in the system, a Not Found Error is notified. |

| | |
|---|---|
| **Name** | 8. Search for the traffic violations committed by a specific vehicle |
| **Actor** | Authority |
| **Entry conditions** | The Authority is logged into the system. |
| **Events flow** | 1. The Authority chooses the "Most notified vehicles" option.<br>2. The Authority digits the specific license plates in the corresponding search box.<br>3. The Authority chooses the "Search" option. |
| **Exit conditions** | The reports related to the specified vehicle are shown in a list. |

| | |
|---|---|
| **Exceptions** | The specified vehicle was never the subject of a traffic violation report, a Not Found Error is notified. |

## Sequence Diagrams

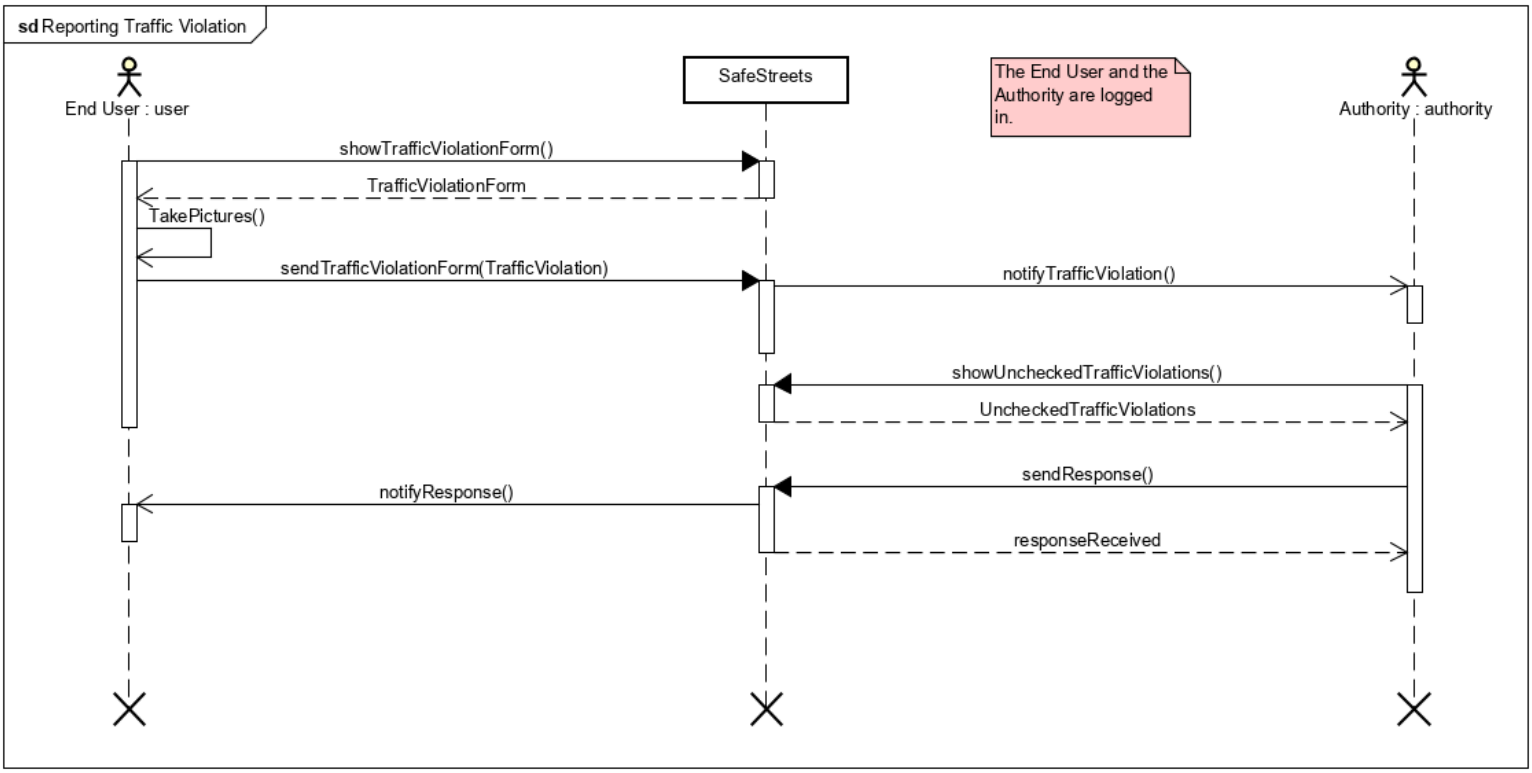



*Figure 5: Reporting Traffic Violation Process Sequence Diagram*

The only function which involves both type of user is Reporting Traffic Violation. So, the sequence diagram represented in figure is the only one shown in this document. Other sequence diagrams are shown in the DD, highlighting the interactions between specific design components.

### 3.2.4 Requirements

In this section, the requirements, that ensures the satisfaction of the goals in the context of the domain assumptions, are shown. After the statement of each goal, the corresponding requirements are listed.

**[G1] – Allow end users to report traffic violations. (including G1.1)**

- [R1] – End Users must be able to register to the system, providing personal information (and identified by Fiscal Code).
- [R2] – Registered End Users must be able to login, using their credentials.
- [R3] – End Users must be able to take pictures of traffic violations, opening their device's camera from the app.
- [R4] – End Users must be able to specify a textual description of the traffic violations.
- [R5] – The system must offer the possibility of being informed of the "Rules for a well-formed traffic violation report" to the user.
- [R6] – The system is able to read the notified vehicles' license plate from the reported pictures.
- [R18] – The possibility to visualise all personal previous reports, with the corresponding authority check decision, must be offered to End Users.

- [D5] – The device used to run SafeStreets mobile app must be able to take pictures.
- [D6] – The device used to run SafeStreets mobile app must be able to provide the GPS position of the end user.
- [D7] - The device used to run SafeStreets mobile app must be connected to internet.

**[G2] – Allow end users and authorities to visualize which are the streets with the highest frequency of violations. (including G2.1)**

- [R1] – An End User must be able to register to the system, providing personal information (identified by Fiscal Code).
- [R2] – Registered End Users must be able to login, using their credentials.
- [R7] – Authorities must be able to register to the system, providing personal information (identified by ID Number).

- [R8] – Registered Authorities must be able to login, using their credentials.
- [R9] – The system must be able to show a map reporting the number of violations that have occurred in every street. This must be made highlighting the streets with different colours.
- [R10] – Users are allowed to filter the data, by date and type of violation, which are used to "build" the map.
- [R11] – Users are allowed to ask for information about a specific street and the system must show the corresponding map.
- [D6] – The device used to run SafeStreets mobile app must be able to provide the GPS position of the end user.
- [D7] - The device used to run SafeStreets mobile app must be connected to internet.

**[G3] – Allow authorities to know the vehicles that commits one or more violations. (including G3.1)**

- [R7] – Authorities must be able to register to the system, providing personal information (identified ID Number).
- [R8] – Registered Authorities must be able to login, using their credentials.
- [R12] – The system must be able to show (only to authorities) the list of vehicles that have been reported one or more time in a traffic violation report.
- [R13] – Authorities are allowed to filter the list of notified vehicles by date, street and type of violation.
- [D3] – Each authority is uniquely identified by an ID Number.
- [D7] - The device used to run SafeStreets mobile app must be connected to internet.

**[G4] – Allow authorities to check and verify the consistency of traffic violations reported by end users.**

- [R7] – Authorities must be able to register to the system, providing personal information (identified ID Number).

- [R8] – Registered Authorities must be able to login, using their credentials.
- [R14] – Authorities must be able to know everything (pictures, license plate, date, street, textual description) about the reports made by end users.
- [R15] – Authorities are allowed to consult and investigate the list of unchecked traffic violation reports notified by end users.
- [R16] – Authorities are allowed to take decisions about any traffic violation report, after consulting it.
- [R17] - The system must notify that response to the related user.

- [D1] – To all authorities registered to the system, taking decisions (approve or delete) about the correctness of the violation reports, is allowed.
- [D2] – Every authority registered to the system is assumed to have the necessary "traffic rules" knowledge and to be able of taking decisions consulting vehicle and traffic authorities data (vehicles special permissions…).
- [D3] – Each authority is uniquely identified by an ID Number.
- [D4] – Every end users' report is checked in less than 24 hours form the notification.
- [D7] - The device used to run SafeStreets mobile app must be connected to internet.

Notes:

The details about [R9] are described in Section 2.2.3.

The decisions mentioned in [R16] are:

- confirm the selected traffic violation report and store it into the database
- reject the selected traffic violation report.

In both cases, the response is notified to the end user that take that report.

### Traceability matrix

| Requirement | Goals | Use cases | Comments |
|---|---|---|---|
| [R1] | [G1], [G2] | U.C. 1 | |
| [R2] | [G1], [G2] | U.C. 2 | |
| [R3] | [G1] | U.C. 3 | |
| [R4] | [G1] | U.C. 3 | |
| [R5] | [G1] | U.C. 3 | |
| [R6] | [G1] | U.C. 3 | |
| [R7] | [G2], [G3], [G4] | U.C. 1 | |
| [R8] | [G2], [G3], [G4] | U.C. 2 | |
| [R9] | [G2] | U.C. 4, U.C. 7 | U.C. 7 is also valid for authorities |
| [R10] | [G2] | U.C. 4, U.C. 7 | U.C. 7 is also valid for authorities |
| [R11] | [G2] | U.C. 4, U.C. 7 | U.C. 7 is also valid for authorities |
| [R12] | [G3] | U.C. 8 | |
| [R13] | [G3] | U.C. 8 | |
| [R14] | [G4] | U.C. 6 | |
| [R15] | [G4] | U.C. 6 | |
| [R16] | [G4] | U.C. 6 | |
| [R17] | [G4] | U.C. 6 | |
| [R17] | [G1] | U.C. 5 | |
| [R18] | [G1] | U.C. 5 | |

## 3.3 Performance requirements

The system has to be able to respond to a possibly great number of simultaneous requests, all the reports must be correctly stored, pending for an authority check.

## 3.4 Design constraints

### 3.4.1 Hardware limitations

As specified in Section 2.4, the devices used to run SafeStreets application must have an internet connection. In addition, regarding to the function of reporting traffic violations, they also must have a camera and the GPS, in order to provide the violations' position.

## 3.5 Software system attributes

### 3.5.1 Availability

The system doesn't offer critical functionalities. So, it must have an availability of 99.9%, because some functionalities, for instance the traffic violations reporting one, must be available when the end users want to notify a violation occurrence.

### 3.5.2 Security

With regards to sign up and login processes, the system must be safe and reliable. Vehicles data must be accessible only to authorities, end users mustn't be allowed to see the details form reports made by other end users.

### 3.5.3 Maintainability

The development of the application must be done in a way that guarantees easy fixing operations and future updates with new functionalities (offered to both kind of customers). Design patterns will be used to facilitate code development and integration between the various system components.

# 4. Formal Analysis using Alloy

```
open util/time

--Signatures
sig Username {}
sig Password {}
sig FiscalCode{}
sig IDNumber{}
sig LicensePlate{}
sig StreetName{}

sig GPSPosition{--coordinates are scaled to simplify the problem
    x: one Int,
    y: one Int
}{(x <= 3 and x>=-3) and (y>=-6 and y<=6)}

sig Registration{
    username: one Username,
    psw: one Password
}
abstract sig Customer{
    registration : one Registration
}
sig EndUser extends Customer{
    fc: one FiscalCode,
    reports: set TrafficViolationReport
}
sig Authority extends Customer{
    idNumber: one IDNumber,
    checkedReports: set TrafficViolationReport
}
sig Vehicle {
    plate: one LicensePlate,
    reports: set TrafficViolationReport
}
sig Street{
    name: one StreetName,
    coordinates: set GPSPosition,
    reports: set TrafficViolationReport
}
```

```
sig TrafficViolationReport{
   id : one Int,
   vehicle: one Vehicle,
   pos: one GPSPosition,
   reporter: one EndUser,
   status: ReportStatus one-> Time
}{id>0}

abstract sig ReportStatus{}
sig Confirmed extends ReportStatus{}
sig Rejected extends ReportStatus{}
sig Unchecked extends ReportStatus{}

--Facts

--Every customer has a unique username
fact uniqueUsernames{
   no disj c1, c2: Customer | c1.registration.username=c2.registration.username
}

--Every End User has a unique FiscalCode
fact uniqueFiscalCode{
   no disj eu1,eu2: EndUser |eu1.fc=eu2.fc
}
--Every Authority has a unique ID Number
fact uniqueIDNumber{
   no disj a1,a2: Authority|a1.idNumber=a2.idNumber
}
```

```
--Every vehicle has a unique license plate
fact uniqueLicensePlates{
   no disj v1, v2: Vehicle | v1.plate=v2.plate
}
--Every TrafficViolationReport has a unique ID number
fact uniqueNumber{
   no disj tvr1, tvr2: TrafficViolationReport | tvr1.id=tvr2.id
}

--The Status of a traffic violation report is firstly Unchecked,
--then (after the checking process) it becomes Confirmed or Rejected
fact staticReportStatus{
   all tvr: TrafficViolationReport| one t':Time|tvr.status.t' = Unchecked
   all tvr: TrafficViolationReport, t: Time|
      (tvr.status.t = Confirmed  => all t': Time| gte[t',t] => tvr.status.t' = Confirmed)
   and
      (tvr.status.t = Rejected  =>all t': Time| gte[t',t] => tvr.status.t' = Rejected)
}
--all vehicles have the traffic violation reports which involve it in the set of reports
fact vehicleReports{
   all v: Vehicle, tvr :TrafficViolationReport| (tvr.vehicle =v <=> tvr in v.reports)
}

--all reporter have in their set of traffic violations, the traffic violation made by themselves
fact vehicleReports{
   all eu: EndUser, tvr :TrafficViolationReport| (tvr.reporter =eu <=> tvr in eu.reports)
}

--streets have a disjointed set of GPSPositions
fact disjStreets{
   some  gps1: GPSPosition| no disj s1, s2: Street|gps1 in s1.coordinates and gps1 in s2.coordinates
}

--all GPSPosition in a street
fact gpsPositionInAStreet{
   all gps: GPSPosition, s: Street| gps in s.coordinates
}

--if GPS coordinates are the position of a report and are part of a street, that street has the report in her set of reports
fact streetCoherence{
   all s: Street, gps: GPSPosition, tvr: TrafficViolationReport| (gps = tvr.pos and gps in s.coordinates) <=> tvr in s.reports
}

--each report is checked by one and only one authority
fact noDifferentAuthoritiesCheckTheSameReport{
   no disj a1, a2: Authority, tvr:TrafficViolationReport| (tvr in a1.checkedReports) and (tvr in a2.checkedReports)
}
```

# Worlds

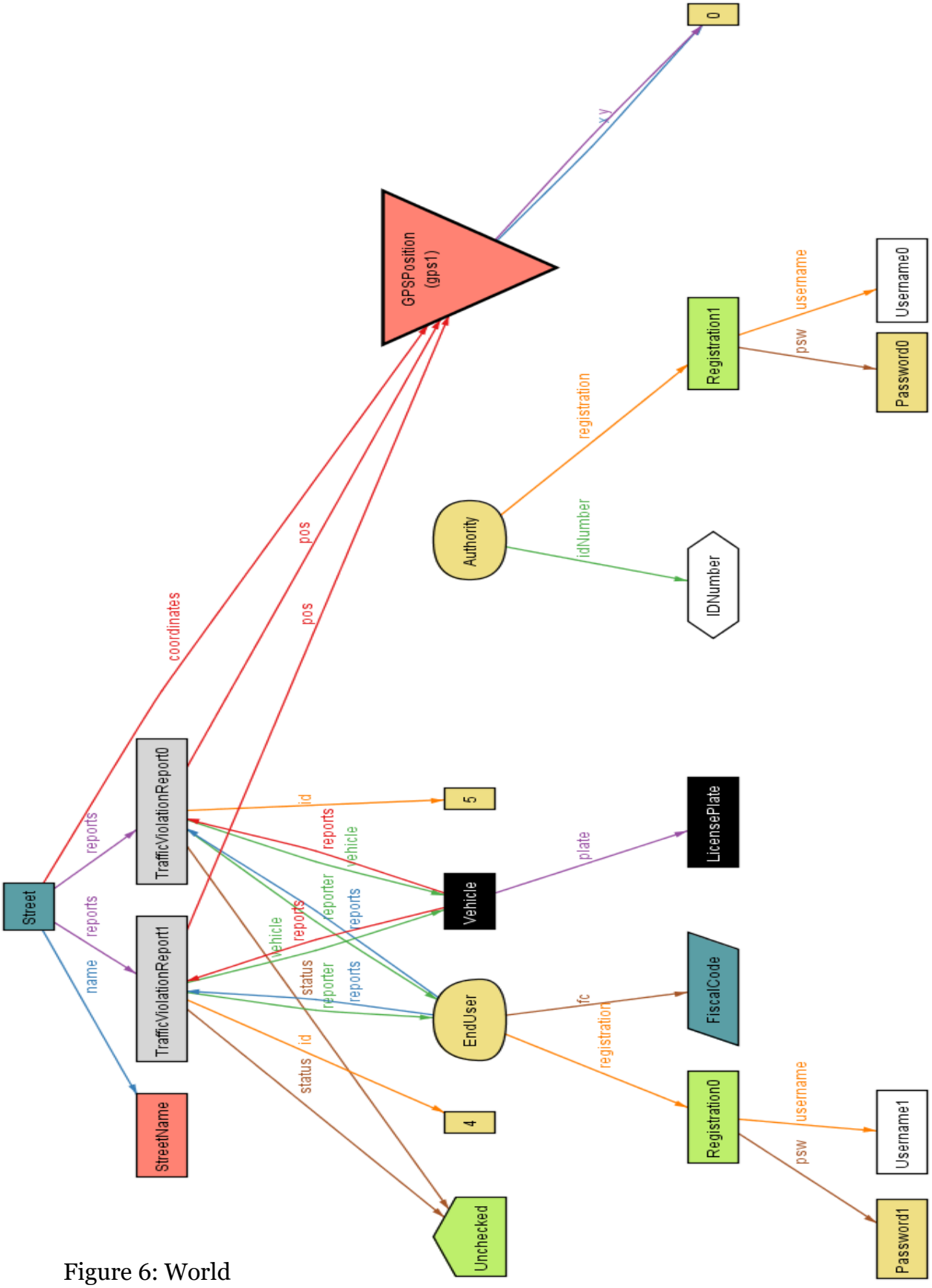



Figure 6: World

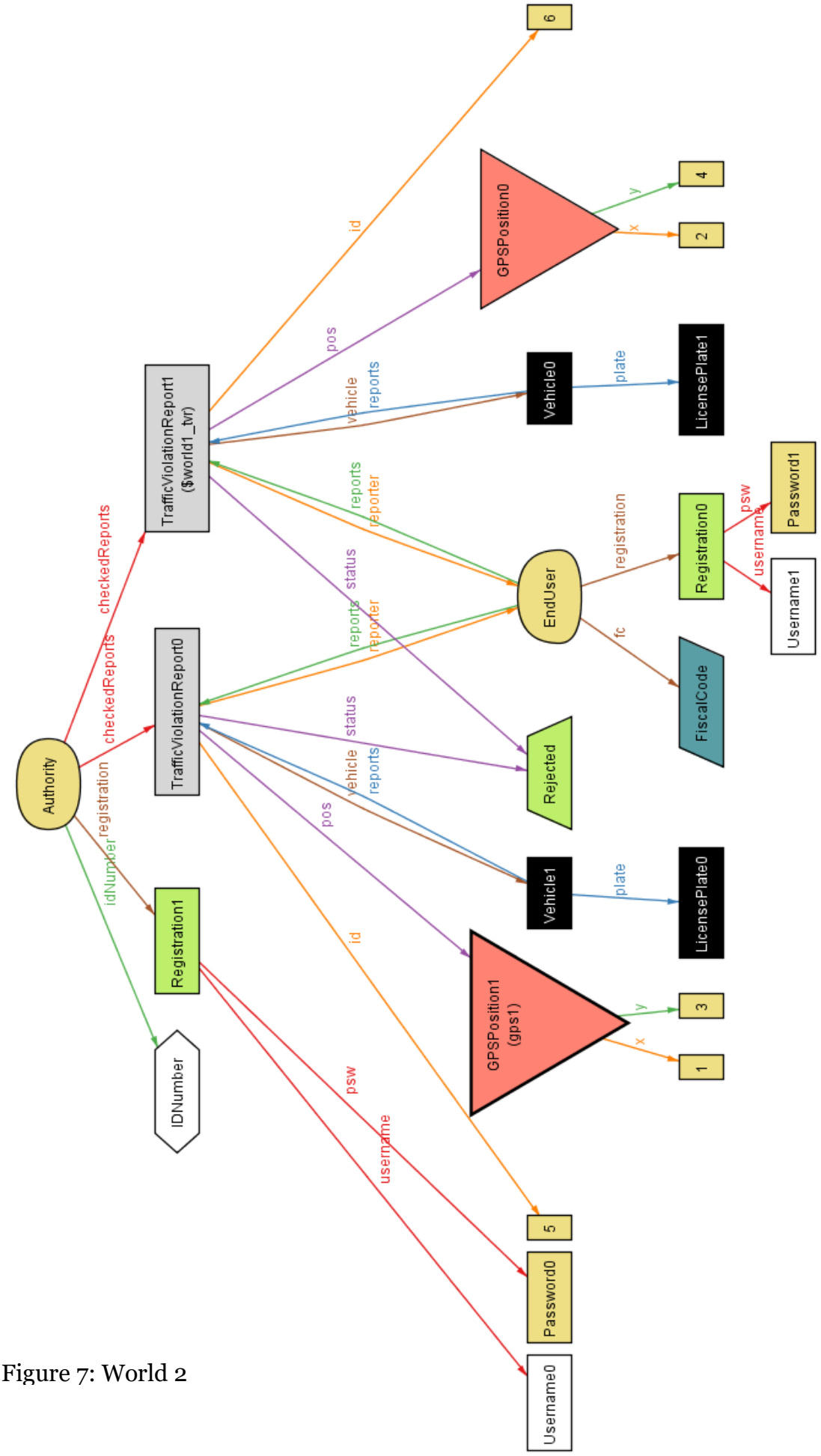



Figure 7: World 2