RASD document

Version 0.6 – 21/10/2019

Giulio A. Abbo – 10538950

Gianmarco Accordi – 10587213

Massimiliano Bonetti – 10560496

# Introduction

## Purpose

In this document a description of the SafeStreets application – from now on referred to as the system – will be presented along with the analysis of its requirements and the assumptions taken.

### Problem overview

The system is addressed to two different entities: the subscribed user and the municipality; anonymous access is not permitted.

The system will allow its users to send reports (including a picture, time, date, position and type) about traffic violations. The gathered data will be elaborated (plate recognition if the plate number is not provided, street name) and used to show to users and the municipality the streets or areas with the highest frequency of violations; in addition the municipality will have access to a list of the plate numbers of the vehicles that committed the most violations and to suggestions for possible interventions.

The system will be able to collect data from the municipality about violations on the territory, crossing it with the data from user reports and using it as above.

The system will also provide a way for the municipality to access the data from user reports, to allow the municipality to generate traffic tickets; the data from the generated traffic tickets will be used for building statistics (on the person with most tickets and the trends in the issuing of tickets) accessible by the municipality. Care must be taken to ensure that the chain of custody is never broken.

### Goals

These are the goals of the SafeStreets system:

* G1: The System accepts reports by the users about the violations.
* G2: The System gives information about the violations to the Municipality.
* G3: The System suggests possible interventions to the Municipality.
* G4: Allow the Municipality to retrieve submitted violations.
* G5: The System gives statistics to the User about the violations.
* G6: The System gives statistics to the Municipality about the violations.
* G7: The System can retrieve the tickets issued by the Municipality.

## Scope

Here it is presented the initial distinction between events that belong to the world and those that belong to the machine (the system). The violations and the event user finds a violation are considered part of the world. Indeed, the machine is not aware of these events unless the user sends a report notifying the violation and its location, which are shared events. Likewise, are part of the shared phenomena the events in which the users or the municipality access the statistics and the municipality accesses the violations and the suggestions, the systems accesses data on violations from the municipality and vice versa. The machine side of the model contains the data memorized, the process of data analysis and the process of plate recognition.

Traffic violation

Municipality makes traffic tickets

User finds violation

Violation memorization

User memorization

Plate recognition

Data analysis

User notifies a violation

Municipality accesses violations

Suggest intervention

**WORLD**

**MACHINE**

Detecting violation’s position

User requests statistics

Access to traffic tickets from municipality

The world in which the system will work is modelled as follows: all the authorities that oversee the viability or can generate traffic tickets are considered as one for simplicity and are referred to as municipality. The municipality is not a mandatory agent, the system can work fine even without any.  
The user is a person that is subscribed to the system and is considered trustworthy for simplicity sake: he does not send false or wrong reports.

The system is considered to be supported by an organization of some kind, which handles the infrastructure necessary for the operativity and any contracts with the municipalities.

The data provided by the municipality is given for accurate and timely. It is taken for granted that the municipality will generate tickets when prompted (if the ticket service is active).

## Definitions, acronyms, abbreviations

### Definitions

* The System: it refers to the software to be and all its components
* User: the end user, a generic registered individual who can send reports and access statistics
* Municipality: the authority that oversees the viability in the area and generates tickets
* Violations: traffic violations punishable by law, e.g.: parking on bike lanes or reserved lots, double parking
* Report: an alert from a user about a traffic violation

### Acronyms

* API: Application Programming Interface
* GPS: Global Positioning System
* UI: User Interface
* S2B: Software to Be

### Abbreviations

* G*n*: *n*th goal
* D*n*: *n*th domain assumption
* R*n*: *n*th requirement

## Revision history

Text

## Document structure

The RASD document comprises six chapters:

**1. Introduction**:

**2. Overall Description:**

**3. Specific Requirements:**

**4. Formal Analysis using Alloy:** it contains the model, described using the formal language Alloy, of the most critical parts of the system and of the environment. This chapter contains also some worlds obtained from the model and the checks of some assertions.

**5. Effort Spent:** It contains the number of hours used by each member of the group to work on each part of the document.

**6. Reference Documents:** It contains the documents used to write this document.

# Overall description

## Product perspective

The system is divided into three products: an app for the user, a software for the municipality and a software for the system’s backend.

The notify violations is a shared phenomenon that is triggered by the User (that has to be registered in the system) or it can be retrieved by the services offered by the municipality , the violation includes the date, the time, one or more pictures with the main picture contains the License Plate that will be recognized by the system or inserted by the user, the position that can insert by the user or retrieved by the Location system of the device, the type of the violation and the Users that has notify the violation. The violation accessed by the service offered by the Municipality are verified, while the violations notify by the users need to be verified.

The SafeStreet system has also to perform Data Analysis on the streets and vehicle by highlighting the highest frequency violations, that can be useful for the User and the Municipality, that will have different level of granularity: so the user will see the aggregate Data( for example he can see the street with the highest violations), while the Municipality can see also who committed the violations.

Another Shared Phenomena is triggered by the Data Analysis that by looking at the violation frequency can suggest the road that need some intervention.

The next picture contains the class diagram representing the conceptual model of the application domain.

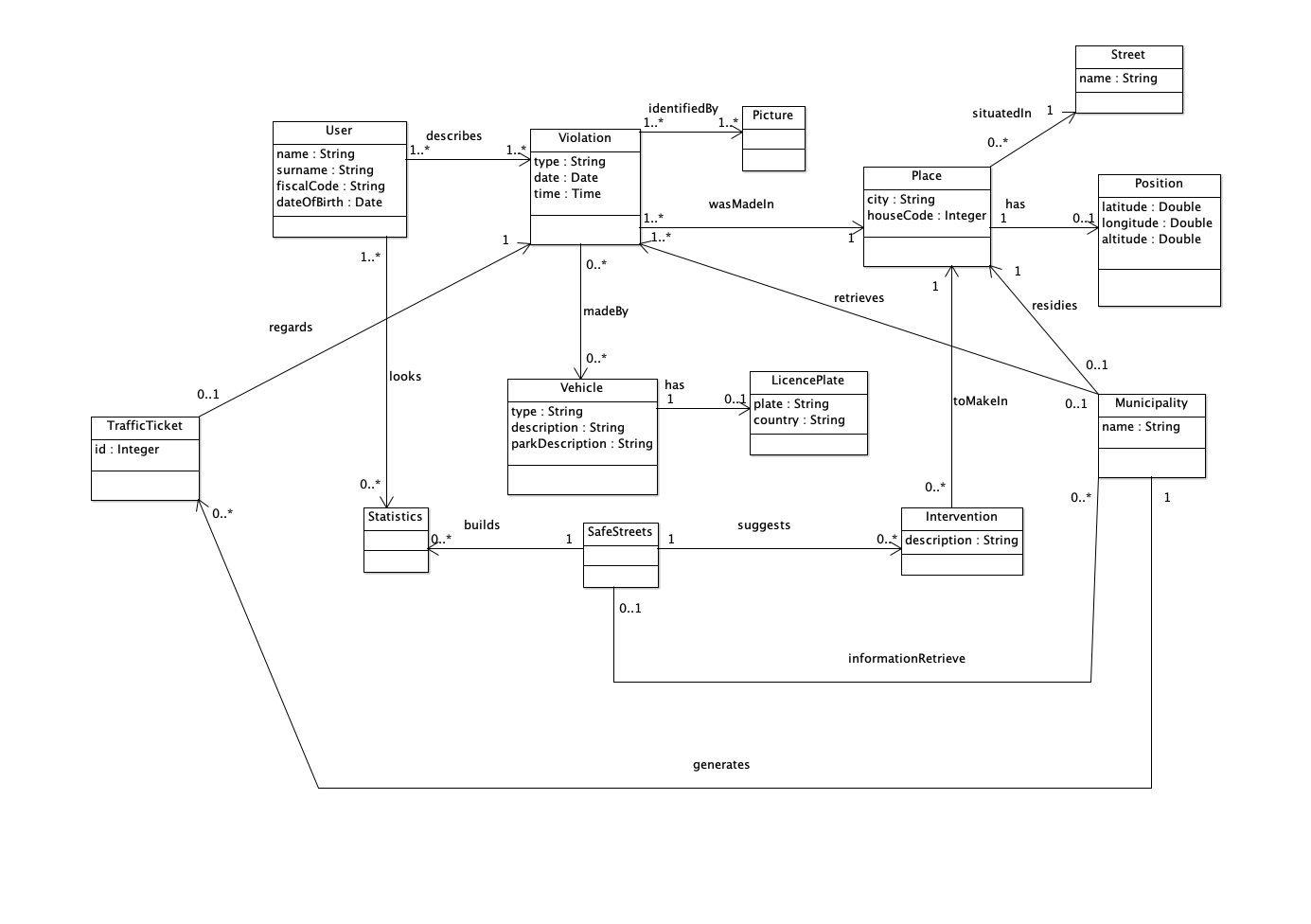


Immagine che contiene screenshot

Descrizione generata automaticamente

## Product functions

The requirements of the SafeStreet system are:

- R1: The notifications about the violations are correctly stored.

- R2: User must be registered to notify a violation.

- R3: The user can view only the most unsafe areas, the streets with the highest number of violations and the effectiveness of the system.

- R4: The system accepts the violations only of its competence area. ???

- R5: Violations registered by the Municipality can be retrieved by the system.

- R6: The system must avoid the manipulation of the violations.

- R7: The System allows the Municipality to query the violations.

- R8: Only the Municipality can access the violations.

- R9: The User can only access aggregated data. ???

- R10: The system accepts notification from the user also if the municipality is not registered.

- R15: The system must calculate the most unsafe areas.

- R16: The system must calculate the streets with the highest number of violations.

- R17: The system must calculate the effectiveness of the service.

- R18: The system must calculate the vehicles that commit the most violations.

- R19: The municipality can view all the statistics calculated by the system and all the violations.

- R20: Every month the system notifies the municipality with the most unsafe streets and the vehicles that have committed the most violations.

## User characteristics

Anything that is relevant to clarify their needs.

## Assumptions, dependencies and constraints

The domain assumptions of the SafeStreet system are:

- D1: Trust the notification made by the user. (?)

- D2: The plates are correctly recognized.

- D3: The positions are correctly retrieved.

- D4: The Identity card is correctly verified. (?)

- D7: The chain of custody of the violation is never broken if and only if the information about the violation is never altered.

- D9: The Municipality possesses only real violations.

# Specific requirements

## External interface requirements

### User interfaces

Text

### Hardware interfaces

It requires the camera of the smartphone

### Software interfaces

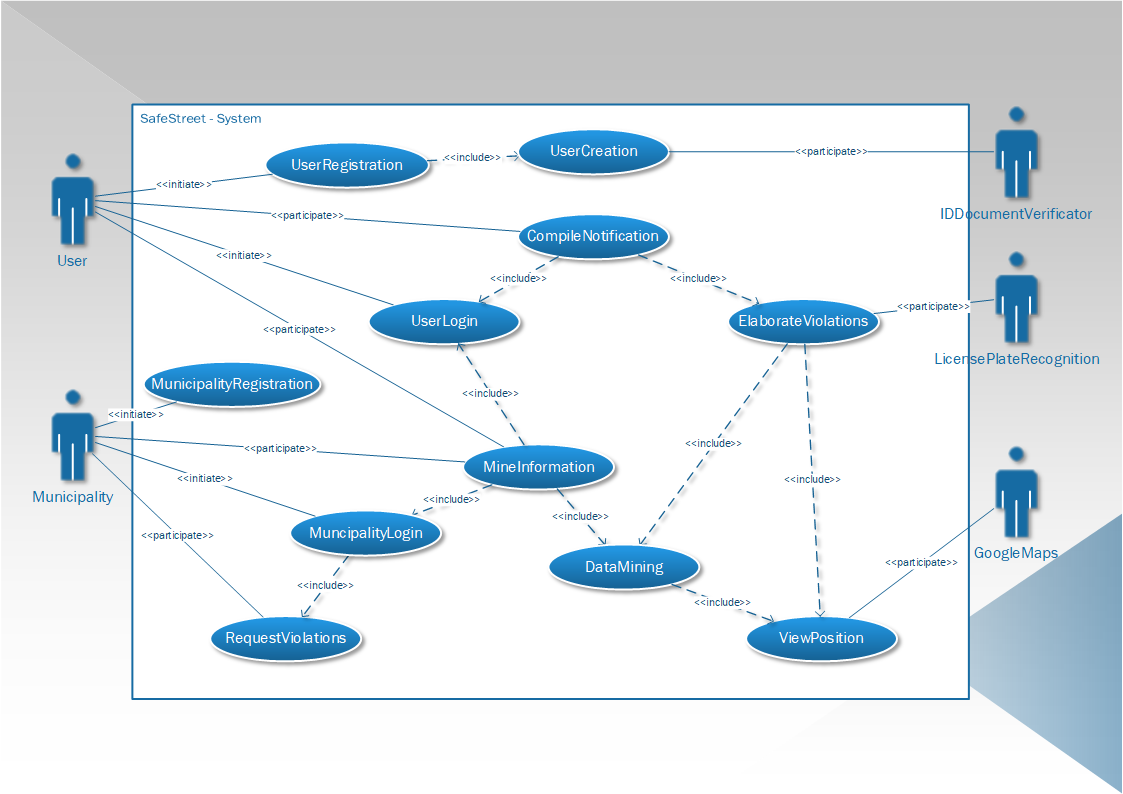
Google Maps  
Traffic Plate recognition  
(Identity card recognition?)

### Communication interfaces

Text

## Functional requirements

**Use Case**

****

**introduction**

**User Registration**

**Actors**: User, ID Document Verificator.  
**Entry conditions**: The User wants to registrate himself inside the service.  
**Flow of events**:

1. The Users sends the request of Registration
2. The Users starts to fill the registration forms
   1. A picture of the User is registered
   2. The User inserts his generality
   3. The User provides also his email and the password he will use
   4. The document of the user is inserted
3. The user then sends the form compiled and the system will take care of it
4. The system validates the user identity and verify if another user already exists with the same generality
5. The system contacts a Document Verificator in order to find out if the document provided by the User is valid.

**Exit conditions:** The User has been correctly registered.  
**Exceptions:** Errors occurs if the Identity Document provided by the user cannot be validated by the Document Verificator, or if there exists another User with the same generality.

**User Login**

**Actors**: User  
**Entry conditions**: The User wants to access his account   
**Flow of events**:

1. The User fills the form with his generality and tries to sign in
2. The System receives this request and search the requested account
3. The System verify if the password is correct for the requested account
4. The System responds to the Client and give it the possibility of managing his account or to send some violations notifications

**Exit conditions:** The User has correctly sign in in his account  
**Exceptions:** If the User provide an email that doesn’t appears inside the User List, if the provided password is wrong, also if the User is already logged in.

**Violation Notification**

**Actors**: User, Google Maps, License Plate Recognition  
**Entry conditions**: The User wants to notify a Violation   
**Flow of events**:

1. The User sign in the System
2. The User starts to fill a form for the Violations notification
3. The System asks the User for a picture of the violation with the license plate of the vehicle
4. The User can select a picture from the gallery of make a shoot from his camera
5. The System receives the picture inserted by the User and ask the User to fill some required options
   1. The System asks the User to input the Position of the Violations
   2. The User can use the position retrieved through his smartphone or he can insert the position manually
   3. The System then ask the User to select the violation type
   4. The User insert the violation type
   5. Then the System ask the user to fill some options that are not mandatory
   6. The User responds with the non-mandatory options
6. The System than elaborates the violation
7. The System sends the pictures to the Licence Plate recognition services
8. The System elaborate the acquired position through the Google Maps services
9. The System store the violation, and it informs the User about the success of the operation

**Exit conditions:** Violation has been correctly registered  
**Exceptions:** The License Plate Recognition services cannot understand the License Plate ID, the Google Maps services cannot retrieve the position, the User cannot sign in the system.

**Analysis Request**

**Actors**: User, Google Maps, Municipality  
**Entry conditions**: The User or the Municipality ask for an Analysis of the Data  
**Flow of events**:

1. The User or the Municipality sign in
2. The User or the Municipality ask for an analysis of the Data
3. The System receives the request and verifies the right of access of the requested query
4. The System than retrieves the information from his knowledge and asks also the Municipality for their knowledge about the violations
5. The System from the retrieved data mines the information
6. The System elaborates a graphical representation of the data using the map provided by Google Maps
7. The System sends the result to the User or the Municipality
8. The result is displayed to the terminal of the User or of the Municipality

**Exit conditions:** The data correctly visualized by the User or the Municipality  
**Exceptions:** The System cannot understand the analysis request, the System cannot retrieve enough data for the Data Analysis, the System cannot access his knowledge of the violations or the knowledge of the Municipality, the system cannot interpreted the data that it has retrieved (for example Google Maps cannot understand the position), the User or the Municipality has the wrong right access for the requested data analysis.

**Violations Request**

**Actors**: Municipality, Google Maps.  
**Entry conditions**: The Municipality wants to retrieve the violations notified to the System.  
**Flow of events**:

1. The Municipality sign in
2. The Municipality asks for an update of the latest notified violations to the System or it asks to retrieve some violations
3. The System elaborate the request and retrieve all the latest violations concerning the Municipality, and he verifies the right access to the violations
4. The System sends the latest violations to the Municipality
5. The Municipality access the update sends by the System

**Exit conditions:** The Municipality get the latest violations concerning its authority  
**Exceptions:** The Municipality cannot sign in, the municipality requests some violations that don’t concern the Municipality’s authority.

**Municipality Registration**

**Actors**: Municipality  
**Entry conditions**: The Municipality wants to perform a registration  
**Flow of events**:

1. The Municipality request a registration operation
2. The System asks the Municipality to insert his Contract Code
3. The System verify the contract code of the Municipality
4. The System asks the Municipality to setup an authentication method
5. The Municipality setups an authentication method
6. The System verifies the authentication method
7. The System memorize the authentication method choose by the Municipality
8. The Municipality is informed of the operation’s success

**Exit conditions:** The Municipality has been correctly registered and can access the services of the System  
**Exceptions:** The Municipality provides a wrong Contract Code, the System cannot verify the authentication method,  
the Municipality provide a wrong setup for the authentication method.

In this section we indicate for each goal the requirements and the domain assumption to ensure the correlated goal:

* **G1: The System accepts notifications about the violations.**
  + R1: The notifications about the violations are correctly stored.
  + R6: The system must avoid the manipulation of the violations.
  + R10: The system accepts notification from the user also if the municipality is not registered.
  + D2: The plates are correctly recognized.
  + D3: The positions are correctly retrieved.
  + D7: The chain of custody of the violation is never broken if and only if the information about the violation is never altered.
* **G2: The System gives information about the violations to the Municipality.**
  + R1: The notifications about the violations are correctly stored.
  + R6: The system must avoid the manipulation of the violations.
  + R7: The System allows the Municipality to query the violations.
  + R8: Only the Municipality can access the violations.
  + D7: The chain of custody of the violation is never broken if and only if the information about the violation is never altered.
* **G3: The System suggests possible interventions to the Municipality.**
  + R1: The notifications about the violations are correctly stored.
  + R6: The system must avoid the manipulation of the violations.
  + R15: The system must calculate the most unsafe areas.
  + R16: The system must calculate the streets with the highest number of violations.
  + R17: The system must calculate the effectiveness of the service.
  + R18: The system must calculate the vehicles that commit the most violations.
  + R19: The municipality can view all the statistics calculated by the system and all the violations.
  + D7: The chain of custody of the violation is never broken if and only if the information about the violation is never altered.
* **G4: Allow the Municipality to retrieve submitted violations.**
  + R1: The notifications about the violations are correctly stored.
  + R6: The system must avoid the manipulation of the violations.
  + R19: The municipality can view all the statistics calculated by the system and all the violations.
  + R20: Every month the system notifies the municipality with the most unsafe streets and the vehicles that have committed the most violations.
  + D7: The chain of custody of the violation is never broken if and only if the information about the violation is never altered.
* **G5: The System gives statistics to the User about the violations.**
  + R1: The notifications about the violations are correctly stored.
  + R3: The user can view only the most unsafe areas, the streets with the highest number of violations and the effectiveness of the system.
  + R6: The system must avoid the manipulation of the violations.
  + R15: The system must calculate the most unsafe areas.
  + R16: The system must calculate the streets with the highest number of violations.
  + R17: The system must calculate the effectiveness of the service.
  + D7: The chain of custody of the violation is never broken if and only if the information about the violation is never altered.
* **G6: The System gives statistics to the Municipality about the violations.**
  + R1: The notifications about the violations are correctly stored.
  + R6: The system must avoid the manipulation of the violations.
  + R15: The system must calculate the most unsafe areas.
  + R16: The system must calculate the streets with the highest number of violations.
  + R17: The system must calculate the effectiveness of the service.
  + R18: The system must calculate the vehicles that commit the most violations.
  + R19: The municipality can view all the statistics calculated by the system and all the violations.
  + D7: The chain of custody of the violation is never broken if and only if the information about the violation is never altered.
* **G7: The System can retrieve the tickets issued by the Municipality** 
  + R5: Violations registered by the Municipality can be retrieved by the system.
  + R6: The system must avoid the manipulation of the violations.
  + D7: The chain of custody of the violation is never broken if and only if the information about the violation is never altered.
  + D9: The Municipality possesses only real violations.

For User we intend the person who has been registered and has made the login.  
For Municipality we intend that Municipality which has been registered and has made the login.

## Performance requirements

The software should be used without waiting times, apart from when the application is uploading the violations or when is downloading the statistics from the server, whose time depends on the internet connection.

## Design constraints

### Standard compliance

Don’t broke the chain of custody.

### Hardware limitations

Text

### Any other constraint

Text

## Software system attributes

### Reliability

Text

### Availability

Text

### Security

Text

### Maintainability

Text

### Portability

Text

# Formal analysis using ALLOY

This section should include a brief presentation of the main objectives driving the formal modelling activity, as well as a description of the model itself, what can be proved with it, and why what is proved is important given the problem at hand. To show the soundness and correctness of the model, this section can show some world obtained by running it, and/or the results of the checks performed on meaningful assertions.

Proof that the chain of custody is never altered.

# Effort spent

Text

# References

* Specification document: “Mandatory Project Assignment AY 2019-2020”