RASD document

Version 0.2 – 14/10/2019

Giulio A. Abbo – 10538950

Gianmarco Accordi – 10587213

Massimiliano Bonetti – 10560496

# Introduction

## Purpose

These are the goals of the SafeStreets system:

- G1: Give information about violations and identify potentially unsafe areas.

- G2: Accept notification about the violations.

- G3: Suggest possible interventions.

- G4: Allow the Municipality to retrieve submitted violations.

## Scope

Analysis of the world and the shared phenomena.

## Definitions, acronyms, abbreviations

Text

## Revision history

Text

## Reference documents

Text

## Document structure

Text

# Overall description

## Product perspective

Details on the shared phenomena and a domain model (class and states diagrams).

The SafeStreets system is designed as a Progressive Web App, <https://developers.google.com/web/progressive-web-apps>

The world in which the SafeStreets system will work is modelled as follows.

Traffic violations

Municipality

User

Violations storage

Users storage

License plates storage

Plate recognition

Data analysis

Retrieve violations from SafeStreets

Notify violation by User

Access violations by Municipality

User registration

Suggest intervention

**WORLD**

**MACHINE**

The notify violations is a shared phenomena 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.

## Product functions

The requirements of the SafeStreet system are:

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

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

- R3: Different actors have different levels of visibility.

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

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

- R6: The system needs to avoid the manipulation of the 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 (?)

# Specific requirements

## External interface requirements

### User interfaces

Text

### Hardware interfaces

Text

### Software interfaces

Text

### Communication interfaces

Text

## Functional requirements

**Use CaseA close up of a map

Description automatically generated**

**User Registration**

**Actors**: User, 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 start to fill the registration forms
   1. A picture of the User is registered
   2. The User insert his generality
   3. The User provide 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 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 starts to fill a form for the Violations notification

**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, and also if the User is already logged in.

## Performance requirements

Text

## Design constraints

### Standard compliance

Text

### 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.

# Effort spent

Text

# References

Text