AA 2019-2020

POLITECNICO
MILANO 1863

Computer Science and Engineering

Software Engineering 2

SAFESTREETS
RASD

Version 1.1 - 10/11/2019

Authors:

Simone De Vita
Fabio Fontana

Professor:

Elisabetta Di Nitto

1 INTRODUCTION

1.1 Purpose

This document aims to serve as a basis for the organization of the project that will lead to the development of SafeStreets by presenting the requirements. Secondly, once the product has been completed, it will be used to evaluate the final result and judge its loyalty with respect to what is set in these pages. In conclusion, the document is intended for those who actually will contribute to the creation of the software (Project Managers, System and Requirement Analysts, Developers, Testers and so on) but also to users who will benefit from it.

1.2 Scope

1.2.1 Description of the given problem

Nowadays the number of cars on the streets of our cities is constantly increasing and for this reason it is necessary to find new solutions in order to guarantee that they can be parked legally. Nevertheless, it often happens, due to lack of parking zones or laziness of drivers, to find cars parked in an unacceptable way and the situation sometimes becomes difficult to manage by the authorities. In this context SafeStreets operates: the software permits to share with authorities photos of parking violations but this is not all. In addition, SafeStreets mixes the information given by its users with some data computed by the system in order to have a complete description of the violation, in fact the report is accompanied by the licence plate, the location, the date and the type of violation. The retrieval of the first two data is managed by the software while the third is entered manually by the user and the result is saved in the database and can be visualized by municipality.

Moreover, the product has another interesting feature that gives the opportunity to users to navigate through data stored anonymously in the system visualizing correlations between places and the number of violations through maps, diagrams and other representations.

Obviously not everything is accessible by anyone but in some cases permission is required to access certain data.

In conclusion there is a very innovative function, possible thanks to the collaboration of the municipality, that allows SafeStreets to propose possible solutions to solve problems due to repeated violations that occur in certain areas of the city that for these reasons are considered dangerous.

1.2.2 Goals

- G1: Users and municipality must be able to visualize on a map the zones where violations occur more frequently.
- G2: Users and municipality must be able to visualize statistics regarding violations.
- G3: The municipality must be able to confirm reports or mark them as fake
- G4: Municipality must receive solutions that suggest how to solve the problems of violations in problematic zones.
- G5: Citizens must be able to report parking violations.
- G6: Municipality must be able to have access to all the anonymous segnalations sent to SafeStreets.
- G7: Each user must be able to visualize the list of segnalations sent by him.

1.2.3 The world, the machine and shared phenomena

Phenomeno	Shared	Who controls it
Collection of municipality data into a database	N	W
Traffic plate recognition	N	M
Violation reported to SafeStreets using the mobile app via a picture	Y	W
Traffic violations	N	W
Position retrieval from GPS	N	M
Data mining and visualization by users	Y	M
A user encounters a parking violation	N	W
Collection of data into database	N	M
Suggestions of interventions	Y	M
The municipality tries to solve the problem of parking violations by adopting some solutions.	N	W
Data anonymization	N	M
Linking SafeStreets and municipality data	Y	M

Retrieval of date and time of the segnalation	N	M
Processing of data to be presented to users	N	M

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- Driver: the one who makes the violation.
- User: who benefits from the service offered by SafeStreets by sending photos of the violations or accessing the data saved in the system.
- Municipality: public authority who take advantage of the service offered by SafeStreets and eventually collaborate with it providing supplementary information and receiving possible solutions to reduce the number of violations.
- Segnalation: report sent by a user. It is composed by some elements like a photo in which the violation figures, the type of violation, date, time and the street where the violation occurred.
- Authority: used as a synonym for municipality.
- Violation: corresponds to parking in an illegal way, there are many types of different parking violations: parking where it’s not permitted (on a crosswalk, near intersections, on sidewalks, on bus stops, on a handicapped zone without permissions, parking in time slots where it is not allowed to stop on that specific street, double parking,).
- Solution: report sent by SafeStreets to the municipality after an accurate evaluation of violations occurred in a certain area. Its aim is to reduce the number of these types of events.
- Zone: the basic unit in which it is possible to divide a city.
- Anonymous segnalation: segnalation saved in the database which presents no reference to the specific user who sent the same notification when the municipality visualizes it.
- Statistic data: representations of the data that SafeSteets stores. Users can visualize them.
- Coordinates: latitude and longitude of the position in which the user takes the photo that will attach to the report.
- Solution table: It corresponds to a map that assigns to each possible parking violation a solution to propose. It is regularly updated because the system evaluates the impact that certain solutions have had in specific zones of the city and whether these have actually reduced the number of violations.
- System: This word is used to refer to SafeStreets and its components such as the server that allows communications with users, the database and all the others that allow data to be computed and solutions to be offered to its users.

1.3.2 Acronyms

- RASD: Requirement Analysis and Specification Document
- DB: DataBase
- GPS: Global Positioning System
- OCR: Optical Character Recognition

1.3.3 Abbreviations

- Gn: n^{th} goal
- Rn: n^{th} requirement
- Dn: n^{th} domain assumption

1.4 Revision history

- V1.1: minor update. Paragraphs organized better in order to increase readability.

1.5 Reference documents

- Specification document: “SafeStreets Mandatory Project Assignment”
- Alloy documentation: http://alloy.lcs.mit.edu/alloy/documentation.html
- 830-1993 - IEEE Recommended Practice for Software Requirements Specifications: https://ieeexplore.ieee.org/document/392555

1.6 Document structure

This RASD is divided into five chapters and each one of these have a specific function that is briefly described below:
The first chapter gives an overview of the problem indicating what the document proposes to do describing in principle the operation of the software, listing the goals to be reached and finally clarifying some details that allow a better understanding. This is meant to be the entry point for developers and other technical people, while users can match their expectations with the presented goals.
The second chapter explains in an accurate way important details that permit to understand better the overall functioning of the system. Here the reader can also find the descriptions of the main functions of the software, the actors and the list of assumptions, dependencies and constraints to be considered. The latter ones serve as limitations for the developer’s options.
The third chapter contains very important details for the development team, in fact it describes the functioning of all the interfaces of the system. Moreover, it proceeds by defining the

functional requirements using the use case and sequence diagrams. Finally, non-functional requirements are listed through Performance Requirements, Design Constraints and Software System Attributes.

In the fourth chapter the results obtained through the formal analysis using Alloy are reported but first the modeling is justified specifying what it demonstrates and why it is important for this specific problem.

Finally, the fifth chapter shows schematically the effort spent for the realization of this document.

2 OVERALL DESCRIPTION

2.1 Product perspective

The product is not totally independent but it does not rely on any previous structure. It just cooperates with municipality when it shares with SafeStreets its data regarding accidents and solutions adopted in the past.

Moreover, SafeStreets exploits the services offered by an API which permits to retrieve, using GPS coordinates, the name of the street from where the user is sending the segnalation.

In addition, the system uses an OCR API that can transform the licence plate, which captured in the photo taken by the user, in a string easily manageable.

The service offered by SafeStreet is designed to be accessible both from a mobile app and through a website. The only difference is that from the smartphone the user can either send reports or consult data while in the web version only the latter option is accessible.

In order to have a better view of the system as a whole, the following is a very simple class diagram that represents a model of the application domain. Note that the classes do not report all the attributes that characterize them but only the main ones that favor understanding.

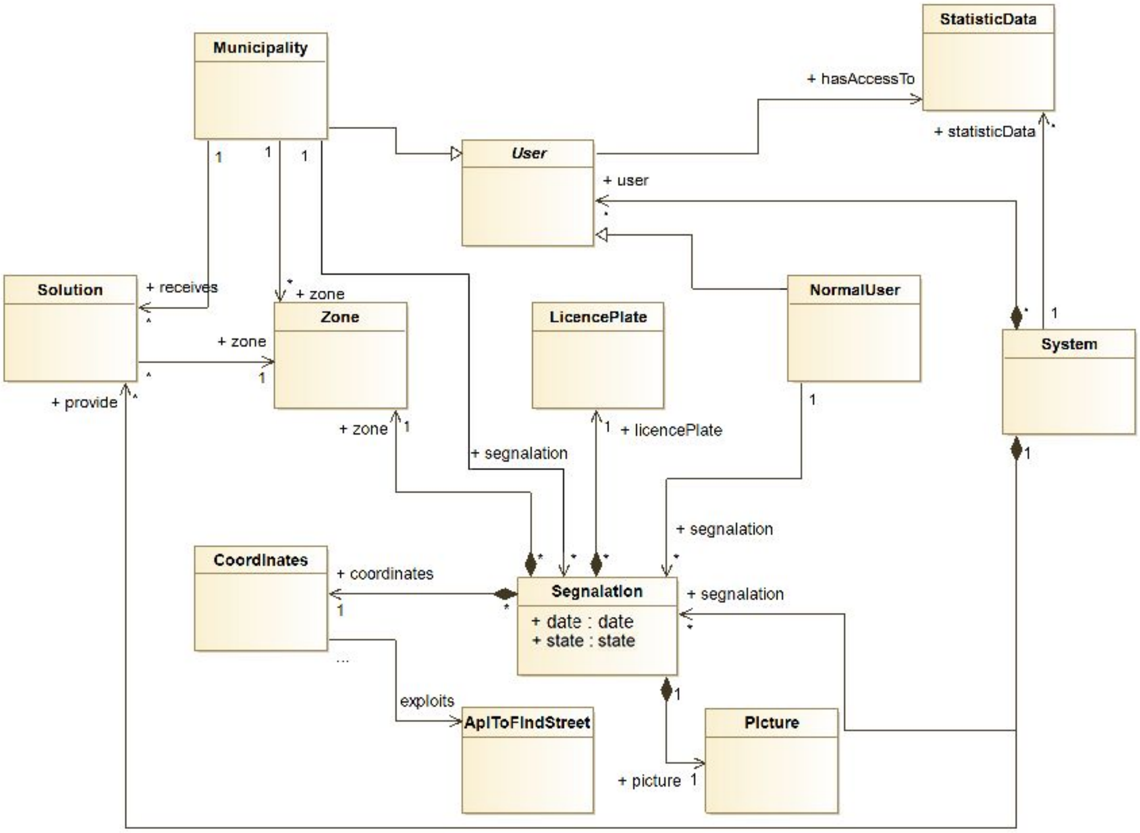


Figure 1 - Class diagram

In the class diagram over here the class “StatisticData” represents also the maps that can be visualized by users.

Below the reader can find some state diagrams which show how states change in particular operations that are really important for the system.

The first diagram represents the various states of a segnalation while the user fulfil its mandatory requests and SafeStreets compute the missing parts.

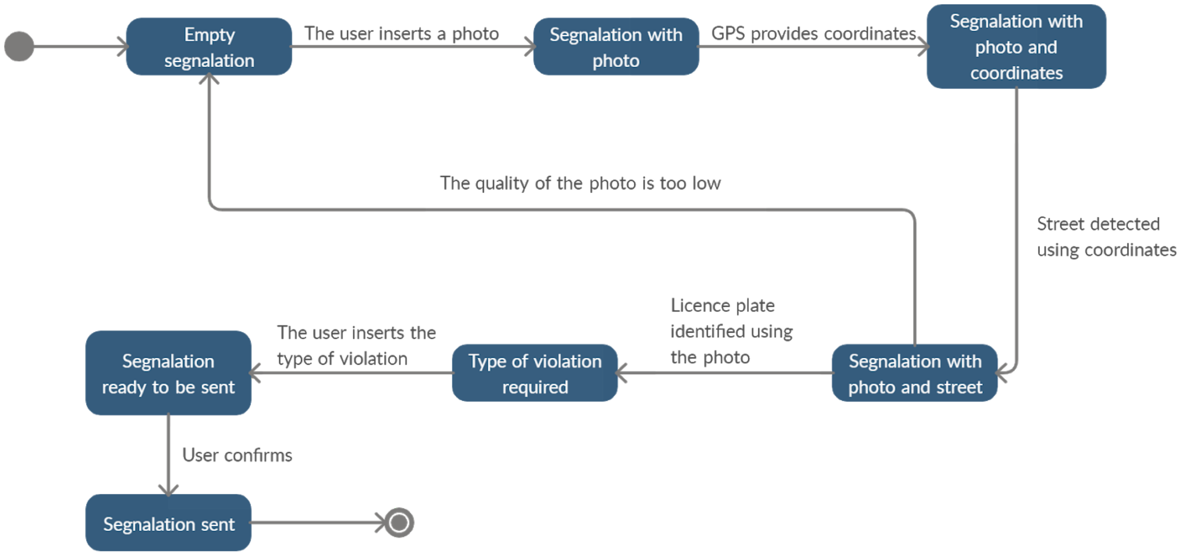

Empty segnalation
The user inserts a photo
Segnalation with photo
GPS provides coordinates
Segnalation with photo and coordinates
The quality of the photo is too low
Street detected using coordinates
Licence plate identified using the photo
Segnalation with photo and street
Type of violation required
The user inserts the type of violation
Segnalation ready to be sent
User confirms
Segnalation sent

Figure 2 - First state diagram

The second one proceeds in the description of the flow of the state of the segnalation once it is received by SafeStreets till the moment in which it is definitely stored in the database. While in the third and last case shown by state diagrams the reader can see what happens to a specific zone when SafeStreets receives a large number of segnalations indicating that many violations occur there.

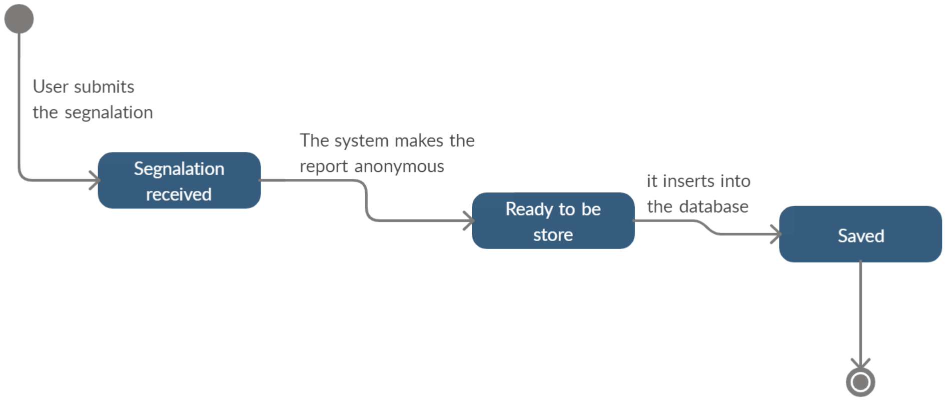

User submits the segnalation
Segnalation received
The system makes the report anonymous
Ready to be store
it inserts into the database
Saved

Figure 3 - Second state diagram

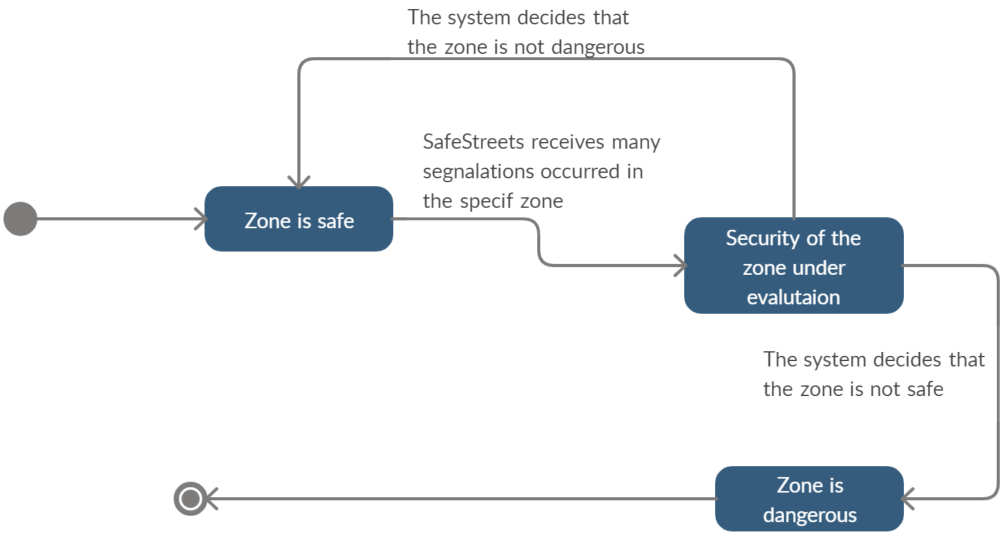


Figure 4 - Third state diagram

In the last state diagram the reader can understand how the status change that characterizes each signal occurs: when it is sent to SafeStreets by the user it is "not verified"; in the event that the authorities actually verify the validity or otherwise of the report, they can be marked as fake or verified. In fact, if the municipality cooperates with SafeStreets, only the latter contribute to the updating of statistics and maps of that specific area while the fake ones are deleted from the database. Finally, it is important to specify that the segnalations imported through communication with municipality has the state “verified” since their creation.

Figure 5 - Third state diagram

2.2 Product Functions

Based on the presented goals, we can identify three product functions that the system is going to provide. It’s worth mentioning that SafeStreets can do its best when the municipality cooperates with it.

2.2.1 Collection of segnalations

This is the crucial function of SafeStreets: the system lets normal users send segnalations about violations and these can be accessed by the municipality users, if they are registered and cooperate with SafeSteets, associated to zones which segnalations refer to. Even if the municipality doesn't work with SafeStreets, authorities will be notified via email. The only data the user is asked to enter is the type of violation and a photo which captures the car parked. The GPS of the device will be used to infer the street where the picture is taken, the licence plate is obtained through an OCR algorithm while the date and time is obtained from the system. All the segnalations will only store the email of the author in order to permit each user to visualize all the segnalations he sent in the past. On the other hand, when municipality users visualize all the segnalations occurred in their area of competence, the email of the author will not be shown. This ensures that the DB is freely explorable by everyone with the right permission. SafeStreets must also ensure that no fake data are sent by users, it achieves this by allowing to take pictures only by an embedded camera feature who makes impossible to modify the photo or to upload already existing photos from the gallery.

2.2.2 Visualizing statistics

This product function is in charge of mining the DB in order to provide insightful statistics to better understand how much spread out are certain types of violations. In particular, it will be possible to visualize on a map the zones more prone to parking violations highlighting them with different colors that represent different degrees of frequency.

Moreover, SafeStreets will provide specific data for each zone that show in which days the number of violations is higher in number and in which hour of the same days or which is the most frequent violation. Finally, it is important to clarify the following: if a user wants to visualize statistics or maps of an area where municipality doesn't cooperate with SafeStreets then the system will show him statistics computed using not verified segnalations. On the other hand, if the municipality works with SafeStreets, in other words if it is registered, then these statistics and maps will be computed using only the reports that presents a verified state. In this way it is possible to avoid that a normal user can falsify the statistics by sending a large number of fake reports. It is also important to clarify that the accidents that the municipality may share with SafeStreets are verified.

2.2.3 Offering solutions

This product function must be implemented if and only if the Municipality cooperates with SafeStreets. In this case SafeStreets will cross these data with its own. Exploiting the interventions Municipality applied in the past, the system will recommend possible solutions to fight violations and prevent accidents. Based on the impact that the proposed solution had, SafeStreets will use this feedback to suggest better and better solutions in the future. The procedure that leads to the proposal of a solution to the municipality works in this way: the system realizes that a solution for a certain area must be proposed when certain types of parking violations occur regularly there and their number is higher than a threshold value. At this point, SafeStreets actually verifies which are the specific violations that characterize the zone, then consults the solution table and check which solution corresponds to the specific accident in the current state of the table. Then, the system prepares the solution to propose to the municipality by attaching the area and the type of accident to which reference is made. Finally, SafeStreets sends the solution to the interessed municipality user.

2.3 User Characteristics

Here we better define who the users of the service are and what characterizes them.
There are two types of user:

- Normal user: citizens can sign up in SafeStreets and become normal users. After the registration, each citizen interested on getting a grasp of the situation of the traffic violations in its territory can visualize the maps and statistics offered by SafeStreets. However, more active citizens can take part in collecting data by providing segnalations.
- Municipality user: a particular subclass of user is the one represented by municipality, they can't send segnalations but they can provide their own data in order to receive solutions to avoid accidents and violations. Moreover, these special users have access to the list of all the reports referring to the zones under their competence sent by normal user, in this way they can use them in the way they prefer. In addition, municipality user can mark a segnalation as fake or verified changing its state. Obviously during the registration it is not immediate to sign up an account with these additional permissions: those who actually represent the authorities or work for them must explicitly make a request to SafeStreets during the registration. They also must specify which is their area of competence, in this way they can have access to the data of their interest and not to data of other cities. Then the system verifies that the request is legal and if everything is ok the municipality user will be activated. In any case each municipality can't have more than one user, so it must be shared between the users that represent the same municipality, and the registration of the account implies a cooperation with SafeStreets.

2.4 Assumptions, dependencies and constraints

We assume that the following domain assumptions hold:

- D1: The user's smartphone GPS will provide a position with an error within 20 meters.
- D2: The picture is taken with a camera that has at least 5 Mpx to allow the OCR algorithm to retrieve the licence plate.
- D3: The user's Internet Connection works properly when he interacts with SafeStreets.
- D4: The data provided by Municipality are correct.
- D5: SafeStreets has a mapping that associates every possible type of violation with a possible solution to fight the specific problem.

- D6: SafeStreets has access to the GPS and camera of the device that is running the app.
- D7: The municipality stores in a DB all the parking violations verified by authorities and the solutions adopted to face them.
- D8: SafeStreets knows an email address for each not registered municipality to which the system refers when it notifies a segnalation.
- D9: SafeStreets receives instructions from each municipality on how to divide the zones.

SafeStreets rely on three external elements:

1. External library that implements the OCR algorithm, to be chosen by the development team.
2. An API to retrieve the location.
3. Eventually the data that the municipality shares with SafeStreets: they consist of solutions adopted in the past to face a high occurrence of parking violations and a list of reports of parking violations verified by the authorities.

3 SPECIFIC REQUIREMENTS

3.1 External Interface Requirements

3.1.1 User Interfaces

The following mockups give a suggestion on how the mobile and web app should look like.

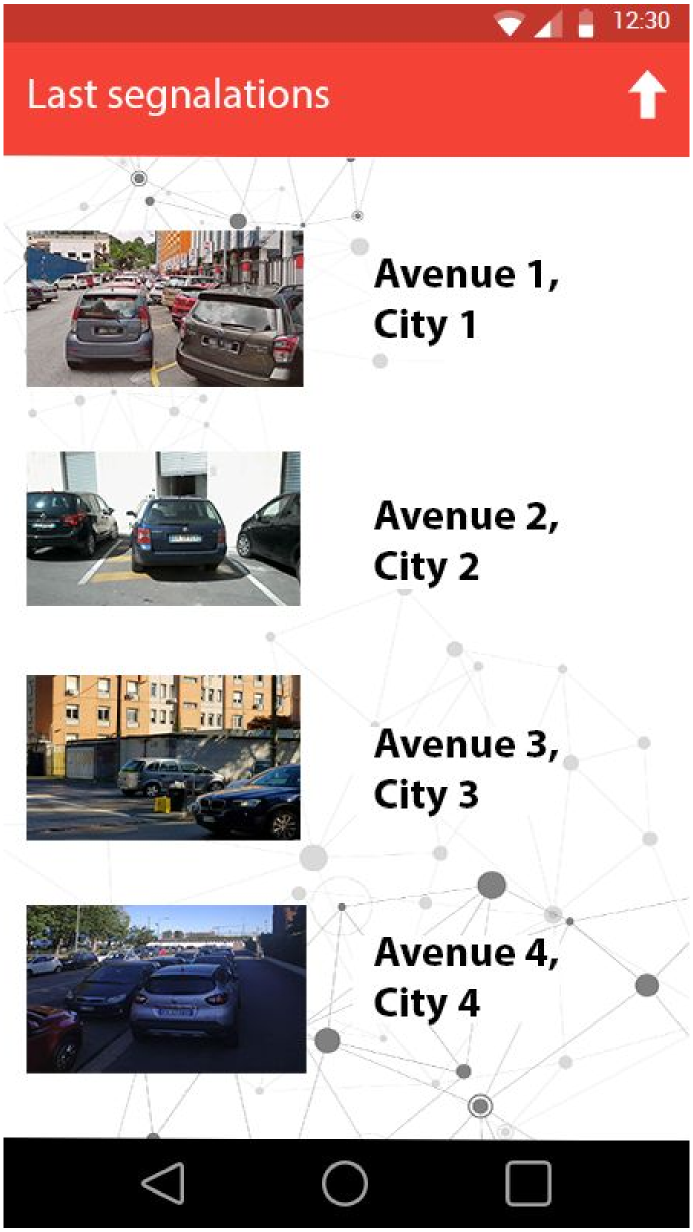


Figure 6 - Home Page

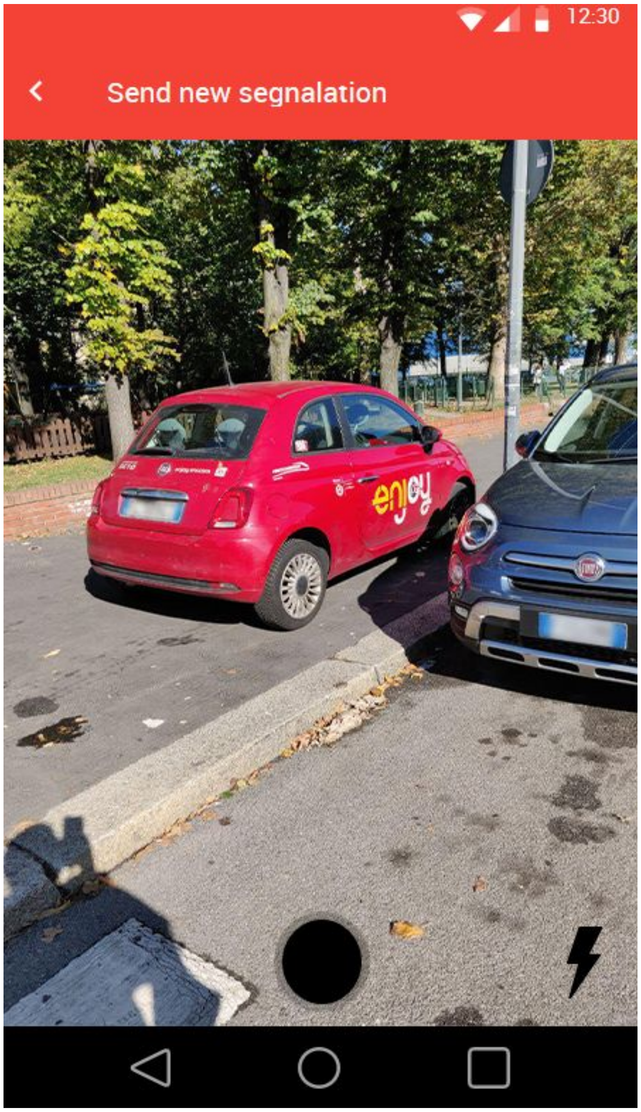


Figure 7 - Segnalation report

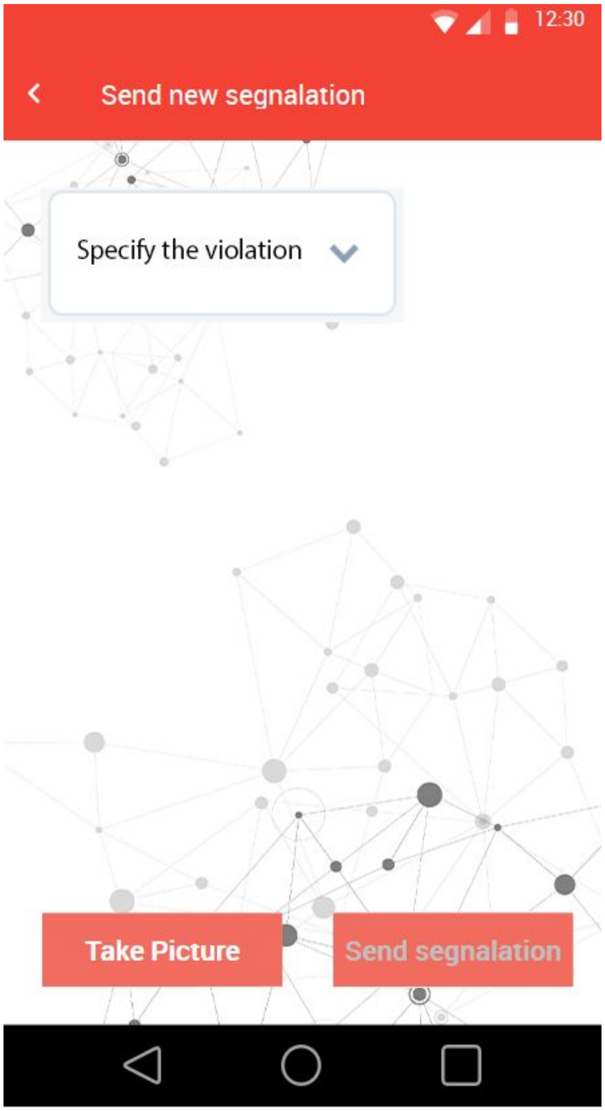


Figure 8 - Type of violation

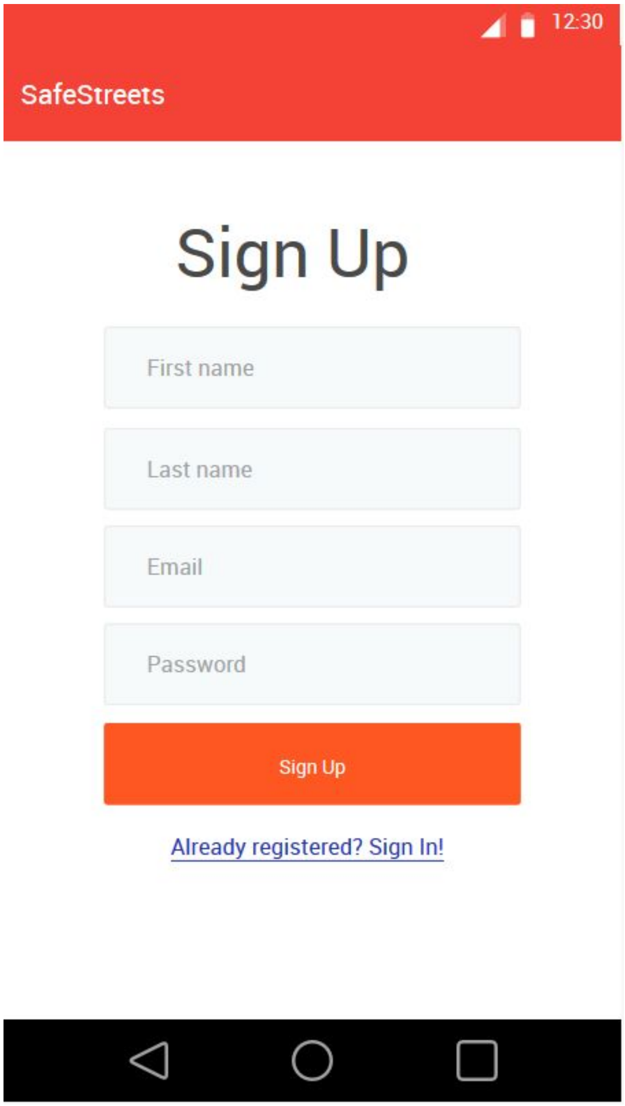


Figure 9 - SignUp

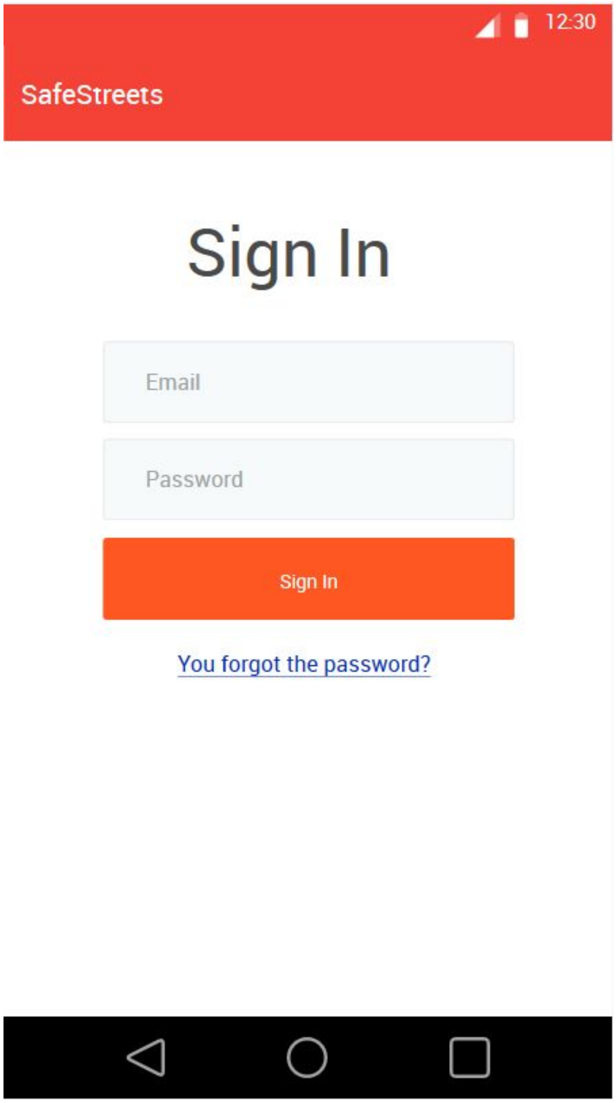


Figure 10 - SignIn

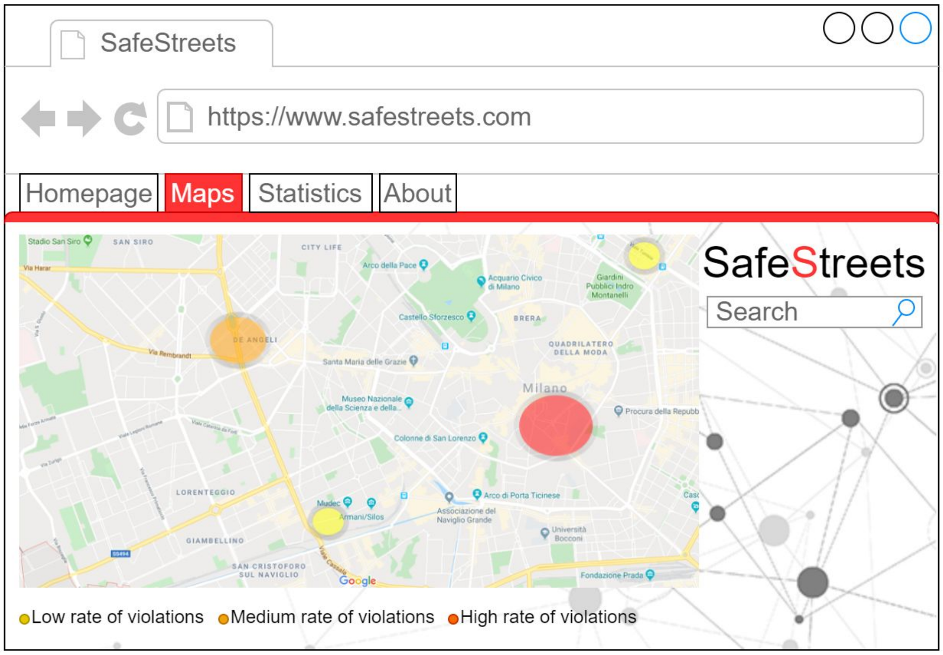
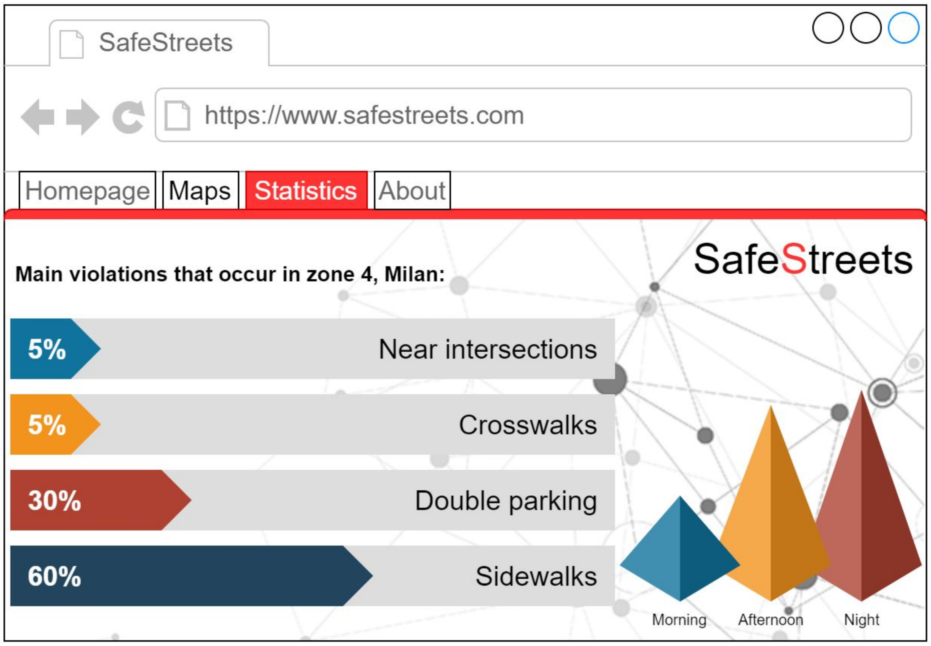

SafeStreets
https://www.safestreets.com
Homepage
Maps
Statistics
About
SafeStreets
Search
● Low rate of violations
● Medium rate of violations
● High rate of violations

Figure 11 - Web App visualizing a map which represents the rate of violations



Main violations that occur in zone 4, Milan:

5%	Near intersections
5%	Crosswalks
30%	Double parking
60%	Sidewalks

Figure 12 - Web App visualizing statistics

3.1.2 Hardware Interfaces

This application doesn't have hardware interfaces, however it requires that the device of the user must have an internet connection to access and visualize data, and a smartphone equipped with GPS and a camera to send segnalations.

3.1.3 Software Interfaces

The application doesn’t provide any API to external applications.
It provides access to the DB depending on the permission each user has and interacts with the database of authorities. In fact municipality, as mentioned before, can share with SafeStreets its data in order to allow the system to compute its statistics on a wider base and improve the quality of the solution table. After this, SafeStreets starts to research solutions in order to provide them to the municipality. The data that the municipality sends to SafeStreets may contain reports of parking violations, the confirmation that the administration has applied a solution proposed by the system, solutions that in the past have been applied to face a large number of violations. An API is used to retrieve the location and the name of the street. An external library will provide the OCR algorithm to read the licence plate.

3.1.4 Communication Interfaces

The system sends and receives data over HTTPS.

3.2 Functional Requirements

3.2.1 Normal user

Scenarios

Scenario 1

Francesco is a diseabled person who recently obtained the driving license and for this reason he has the right to park in parking lots reserved for disabled people. This is very important for him because walking for long distances is often a problem since he uses crutches. Luckily, some of the parking lots in front of the building where he lives are reserved for those like him who have the special permission shown on the dashboard of the car, in this way Francesco is able to go home without any problem. But unfortunately his neighbor, although she has no health problems and absolutely no particular permission, insists on parking her car in the parking lots of the disabled and sometimes because of this Francesco is forced to park elsewhere and this causes him many difficulties. In order to try to solve the problem, Francesco downloaded SafeStreets on his smartphone and whenever this unpleasant situation occurs he sends a report using the mobile app hoping that the authorities can intervene. Indeed, because of the numerous reports received for the frequent recurrence of the same violation the authorities decided to intensify controls in his neighborhood and actually they gave a ticket to the rude lady who finally stopped parking in illegal way.

Scenario 2

Federico is a civil engineering student very fascinated by the management of roads within cities and in his free time he loves studying the various choices that administrations adopt to solve numerous problems and sometimes tries to send letters to his municipality where he presents cases he studied with several statistics regarding the same ones and proposes real solutions. Usually Federico, to start his research or to prove that a particular problem is real, uses the maps and statistics that SafeStreets makes available to all registered users. In fact through the first he chooses the area he will study, while thanks to the latter he has a clearer vision of what really happens within each area. For this reason SafeStreets is a very important tool.

Use case diagram

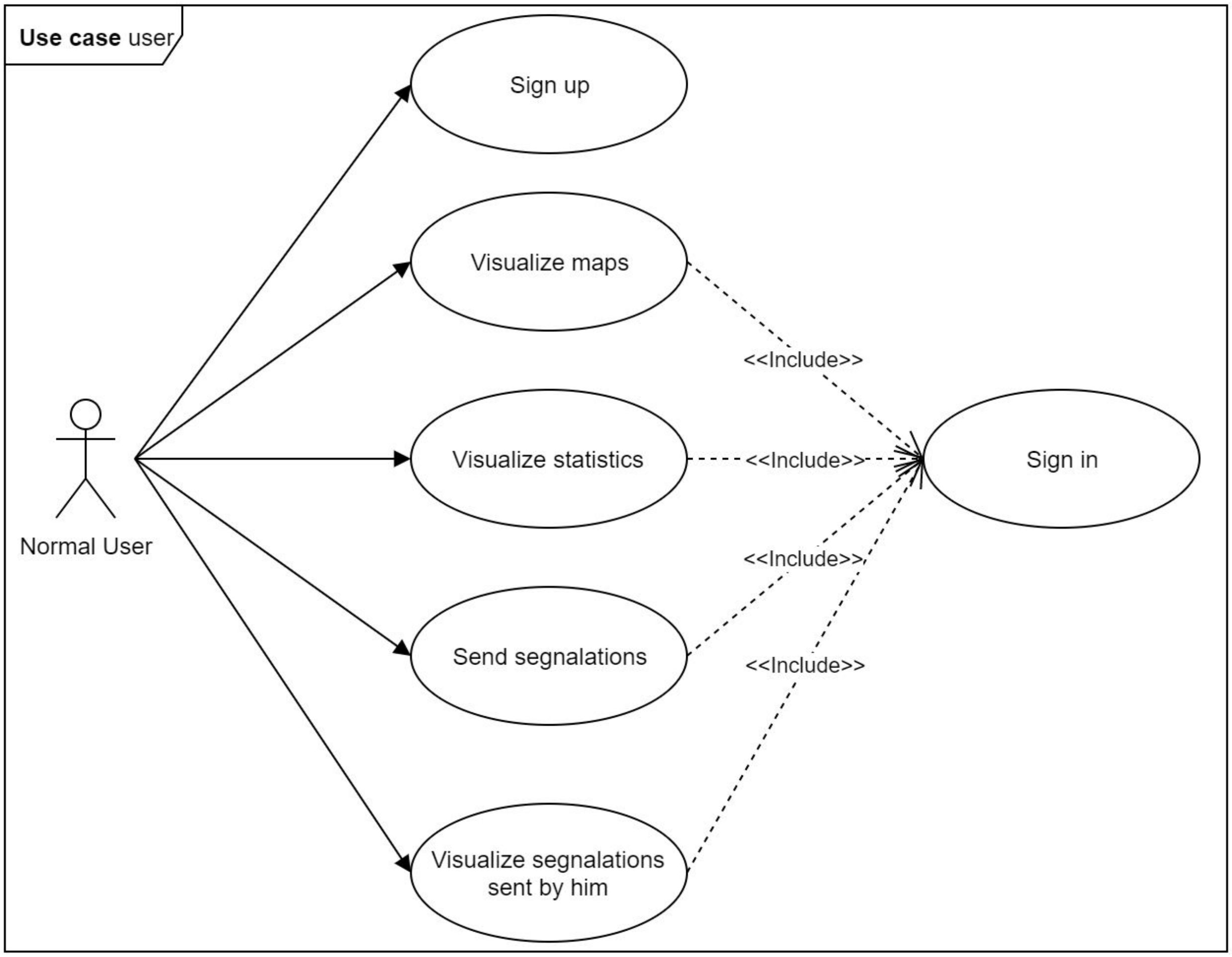


Figure 13 - Use case diagram for user

Use cases

Name	Send new segnalation
Actor	Normal User
Entry condition	The user is already logged in. The user has opened the mobile app and in the home page tapped the upload button in the top right corner.
Events flow	1. The user specify the type of the violation via the combobox. 2. The user presses the “take picture” button. 3. The user take the picture via the SafeStreets integrated camera feature. 4. The user confirms his segnalation by pressing the now activated button “send segnalation”. 5. Safestreets analyzes the picture to obtain the licence plate. 6. SafeStreets obtains the location using the GPS. 7. SafeStreets saves the data into the DB together with the current date and time and sends the segnalation to the municipality via email if the corresponding municipality user doesn’t exist.
Exit condition	The segnalation has been sent successfully to SafeStreets and authorities are notified.
Exceptions	1. The OCR algorithm couldn’t read the licence plate. In this case the user is asked to take the picture again.

Name	Sign Up
Actor	Normal User / Municipality User
Entry condition	The user opens the SafeStreets app or web site for the first time or the user doesn’t have

	an account.
Events flow	1. The SignUp page shows up. 2. The user chooses if he wants to become a normal user or a municipality user. 3. The user fills all the fields. 4. The user presses the “Sign Up” button. 5. If the user had decided to become a municipality user then SafeStreets checks that he has the requirements to become one of them. 6. SafeStreets stores the new user into the DB and send to the user a verification email. 7. The user verifies his account.
Exit condition	The user is registered in the database of SafeStreets.
Exceptions	1. A user with the same email is already registered. SafeStreets will show up the “SignIn” page. 2. The user doesn’t fill all the fields correctly, the system asks him to verify the data inserted. 3. The user does not verify his account within a week since registration. Then the system removes its profile from the DB. 4. The user tries to register a municipality user but it already exist one so the the system notifies him.

Name	Sign In
Actor	Normal User / Municipality User
Entry condition	The user has successfully signed up.
Events flow	1. The Sign Up page shows up. 2. The user clicks on “Already registered? Sign In!”. 3. The SignIn page shows up. 4. The user fills all the fields.

	5. The user presses the “SignIn” button. 6. The HomePage is displayed.
Exit condition	The user is signed in.
Exceptions	1. No match Email/Password has been found in the DB. An error message is shown and the user can try again. 2. The user has forgotten the password, he can click on the appropriate button to retrieve it.

Name	Visualize statistics
Actor	Normal User / Municipality user
Entry condition	The user has opened the SafeStreets App and he is already logged in.
Events flow	1. The user enters in the area where he can visualize statistics. 2. The user chooses which type of statistics he wants to visualize and the zone of interest. 3. The system retrieves what the user asked and it shows up.
Exit condition	The user successfully views the statistics.
Exceptions	1. SafeStreets doesn’t have enough data to compute statistics regarding the zone chosen by the user and shows an error message.

Name	Visualize maps
Actor	Normal User / Municipality user
Entry condition	The user has opened the Web App and he is already logged in.
Events flow	1. The user enters in the area where he can visualize maps.

	2. The user chooses the area that he wants to visualize. 3. The requested map shows up.
Exit condition	The user successfully views maps.
Exceptions	1. SafeStreets can't find the requested zone and asks the user to insert a different input.

Sequence diagram

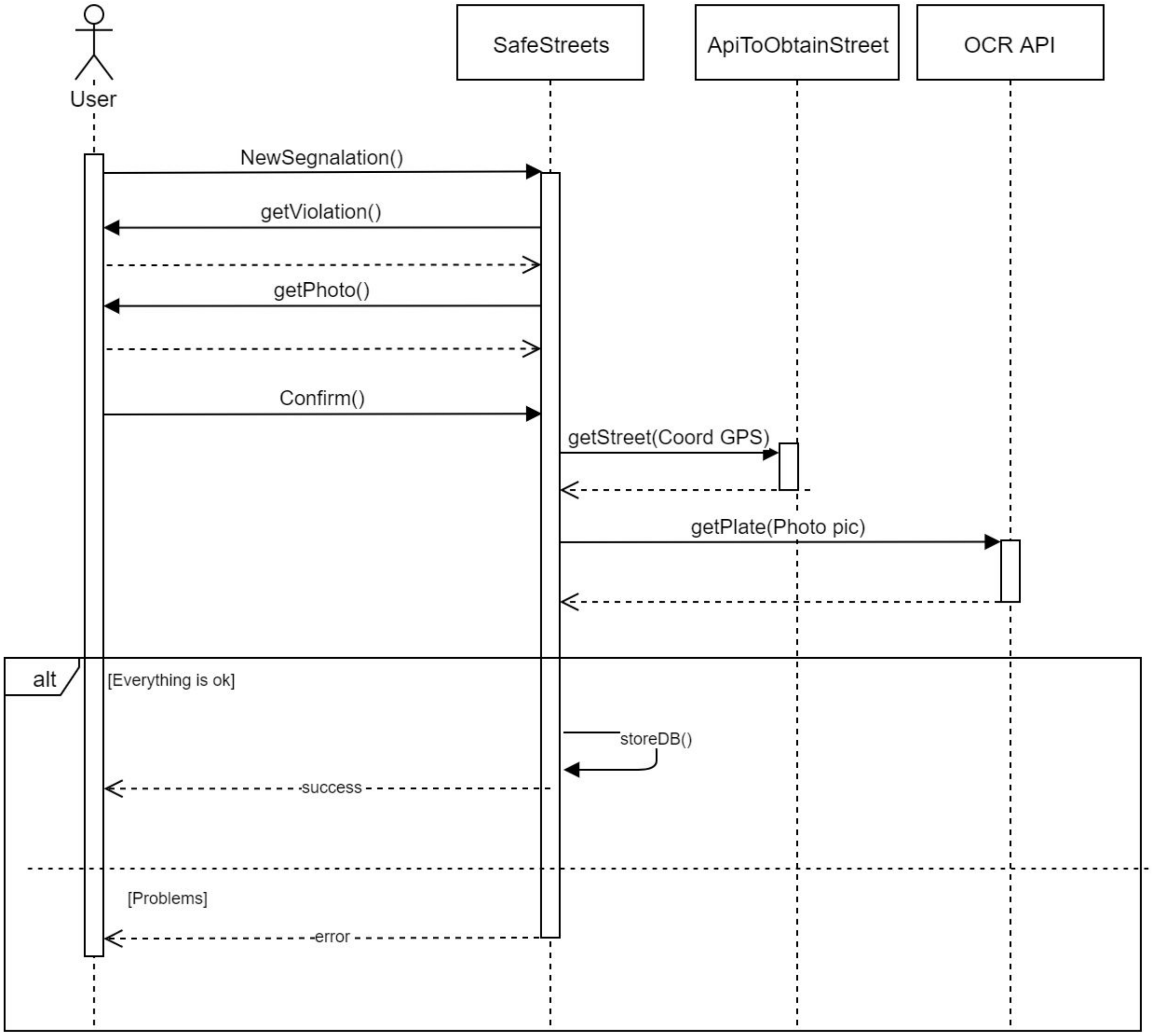


Figure 14 -Sequence diagram: send new segnalation

3.2.2 Municipality user

Scenarios

Scenario 3

Marco works as a traffic policeman in a small town where every Thursday morning there is a market in the main square. Because of this event, many citizens go to the center of the town every week and it often happens that they park outside the parking areas because they are full or too far from the desired destination. Because of what has just been described, SafeStreets receives many reports of parking violations coming from the zones near the city center. Marco, who has access to the municipality user of his town, on that day checks the list of reports received from the areas of his competence and hangs around that streets to fine those vehicles and all the others he finds.

Scenario 4

Monica is an employee of the municipality of her city and she is responsible for the administration of the roads and manages the problems related to it. She has often solved problems due to the occurrence of a high number of parking violations in different areas adopting many solutions, notifying each time SafeStreets of the remedy adopted, but this is not always possible since it is not easy to have continuous control of each area of the city. It happens in fact that one day SafeStreets realizes that in the area near the elementary school of the city often occur too many parking violations related to the fact that vehicles are parked over sidewalks and after checking the solution table notifies the municipality advising to install some poles to separate the sidewalk from the street as it has been noticed in the past that this is the solution that permits to obtain the best result. Monica, since she has access to the municipality account of her city, receives this notification and put into practice the proposed solution, effectively obtaining a remarkable decrease in the number of violations.

Use case diagram

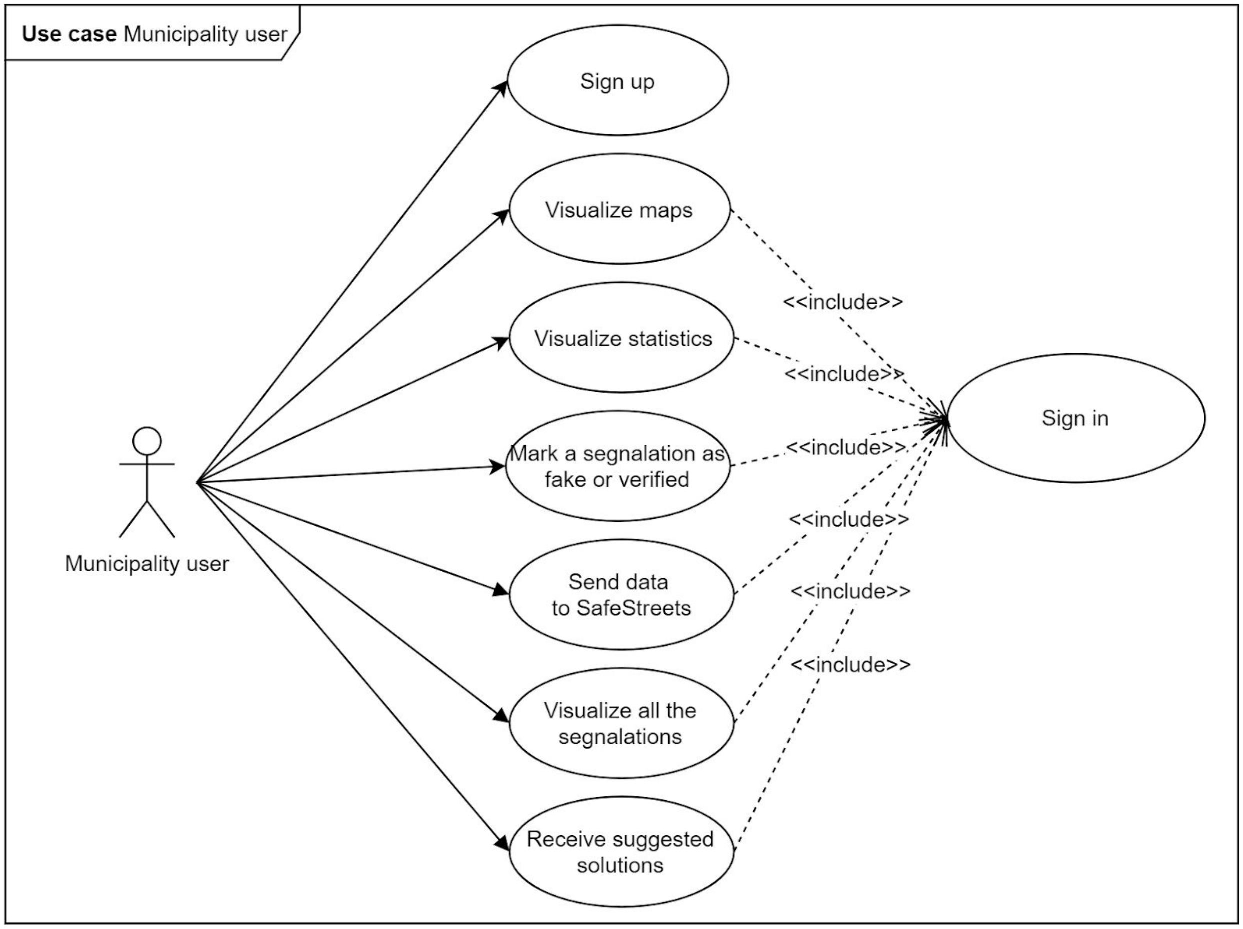


Figure 15 - Use case diagram for municipality user

Use cases

Name	Receive suggested solutions
Actor	Municipality User
Entry condition	SafeStreets recognizes that a particular kind of violation occurs frequently in a given zone.
Events flow	1. SafeStreets checks the solution table looking for that kind of violation. 2. SafeStreets extracts the subset of possible interventions to tackle the problem.

	3. SafeStreets sends the possible solutions to the interessed municipality users mentioning the violations to which the suggestions refer and the zone where the solution should be applied. 4. If the Municipality apply the solution, SafeStreets will update its solution table based on the results the solution obtained.
Exit condition	The Municipality receives the possible solutions proposed by SafeStreets.
Exceptions	//
Special Requirement	SafeStreets must provide its solution to the Municipality within a day since the problem is recognized.

Name	Visualize all the segnalations
Actor	Municipality User
Entry condition	The user is already logged in.
Events flow	1. The user is in the homepage and chooses to visualize all the segnalations of the zones of his city. 2. A list of all the segnalations is shown. They are sorted by date and zone. 3. The user may want to know more info about a specific segnalation so he clicks on the desiderd one and in this way he can see all the data of the report except the email of the author.
Exit condition	The Municipality user visualize all the segnalations of his city.
Exceptions	//
Special Requirement	SafeStreets must provide all the segnalations of at least the last 3 months.

Name	Mark a segnalation as fake or verified
Actor	Municipality User
Entry condition	The user is already logged in. He went to check a report and found out if it was real or fake.
Events flow	1. The user is in the homepage and chooses to visualize all the segnalations of the zones of his city. 2. A list of all the segnalations is shown. They are sorted by date and zone. 3. The municipality user looks for the report whose status he wants to change. 4. The municipality user selects the new status of the report and confirms. 5. If the new status is “verified” then SafeStreets will use the segnalation for maps and statistics, if the new state is “fake” then the report is deleted.
Exit condition	The status of the segnalation is changed
Exceptions	//

Sequence diagram

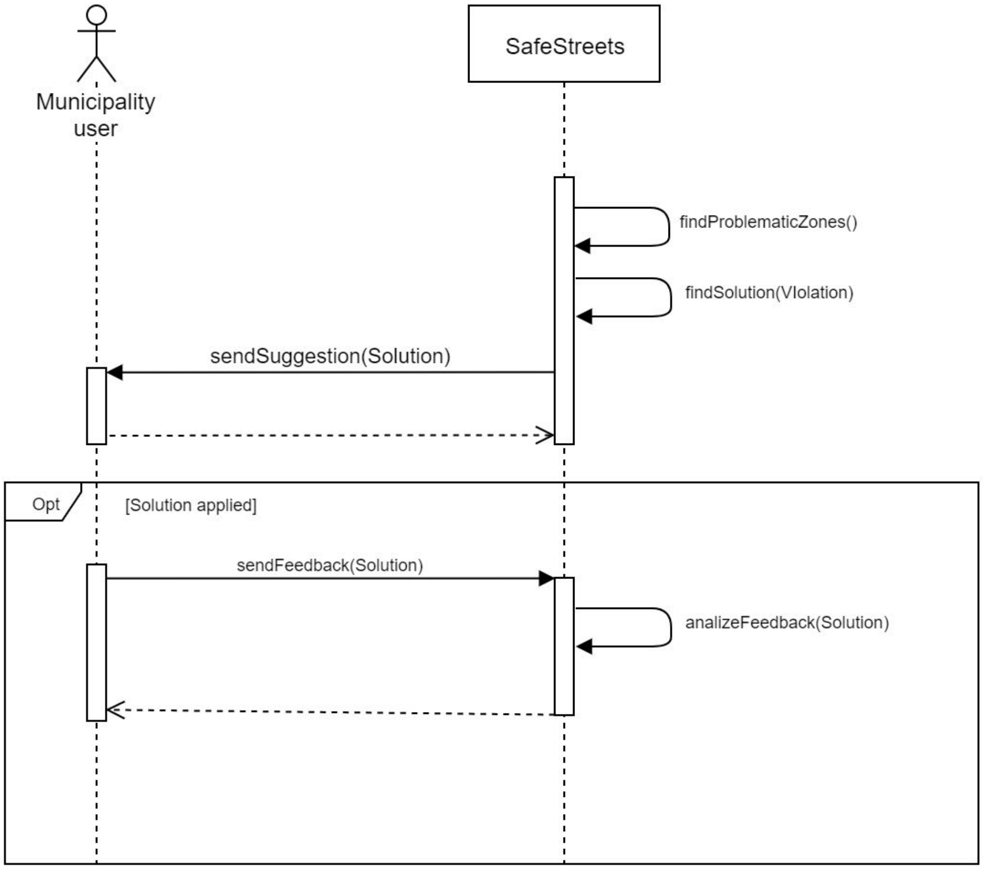


Figure 16 -Sequence diagram: send new segnalation

3.2.3 Requirements

G1: Users and municipality must be able to visualize on a map the zones where violations occur more frequently.

- **R1**: The system must allow the user to sign-up.
- **R2**: The system must allow the user to sign-in.
- **R3**: The system must store the segnalations sent by users in order to collect data of parking violations and where they occur.
- **R4:** The system must use only the segnalations whose state is “verified” to update its maps and statistics presented to users if the municipality which has control over that specific zone, on the other hand it will use the non verified segnalations if the municipality is not registered.

- **R5**: The system must allow the user to choose which statistics or maps visualize between those available.
- **R6**: SafeStreets should allow municipality users to use a tool that allow them to send data to the system.
- **R7**: When a municipality user visualize a specific segnalation the system must allow him to modify, using a tool, the current state of the report.
- **D1**: The user's smartphone GPS will provide a position with an error within 20 meters.
- **D3**: The user's Internet Connection works properly when he interacts with SafeStreets.
- **D4**: The data provided by Municipality are correct.
- **D6**: SafeStreets has access to the GPS and camera of the device that is running the app.
- **D7**: The municipality stores in a DB all the parking violations verified by authorities and the solutions adopted to face them.
- **D9**: SafeStreets receives instructions from each municipality on how to divide the zones.

G2: Users and municipality must be able to visualize statistics regarding violations.

- **R1**: The system must allow the user to sign-up.
- **R2**: The system must allow the user to sign-in.
- **R3**: The system must store the segnalations sent by users in order to collect data of parking violations and where they occur.
- **R4:** The system must use only the segnalations whose state is "verified" to update its maps and statistics presented to users if the municipality which has control over that specific zone, on the other hand it will use the non verified segnalations if the municipality is not registered.
- **R5**: The system must allow the user to choose which statistics or maps visualize between those available.
- **R6**: SafeStreets should allow municipality users to use a tool that allow them to send data to the system.
- **R7**: When a municipality user visualize a specific segnalation the system must allow him to modify, using a tool, the current state of the report.
- **D1**: The user's smartphone GPS will provide a position with an error within 20 meters.
- **D3**: The user's Internet Connection works properly when he interacts with SafeStreets.
- **D4**: The data provided by Municipality are correct.
- **D6**: SafeStreets has access to the GPS and camera of the device that is running the app.
- **D7**: The municipality stores in a DB all the parking violations verified by authorities and the solutions adopted to face them.
- **D9**: SafeStreets receives instructions from each municipality on how to divide the zones.

G3: The municipality must be able to confirm reports or mark them as fake

- **R1**: The system must allow the user to sign-up.
- **R2**: The system must allow the user to sign-in.

- **R3**: The system must store the segnalations sent by users in order to collect data of parking violations and where they occur.
- **R7**: When a municipality user visualize a specific segnalation the system must allow him to modify, using a tool, the current state of the report.
- **R8**: The system must provide the requested data if and only if the user has the permission to visualize them.
- **R9**: The system must allow municipality users to visualize the current state of a specific segnalation.
- **D3:** The user's Internet Connection works properly when he interacts with SafeStreets.

G4: Municipality must receive solutions that suggest how to solve the problems of violations in problematic zones.

- **R1**: The system must allow the user to sign-up.
- **R2**: The system must allow the user to sign-in.
- **R6**: SafeStreets should allow municipality users to use a tool that allow them to send data to the system.
- **R8**: The system must provide the requested data if and only if the user has the permission to visualize them.
- **R10**: SafeStreets must use the data sent by municipalities to update the solution table.
- **R11**: Given a problem, SafeStreets must be able to find the most suitable solution using the solution table in order to send it to the municipality.
- **R12**: SafeStreets must send suggestions to the municipality users who have the problematic zone under their control.
- **R13**: The system should recognize zones where the occurrence of parking violations can be reduced and find a solution to propose.
- **R14**: The system must update its solution table, evaluating if the occurrence of violations has decreased, whenever the municipality reports that a solution has been adopted.
- **D4**: The data provided by Municipality are correct.
- **D5**: SafeStreets has a mapping that associates every possible type of violation with a possible solution to fight the specific problem.
- **D7:** The municipality stores in a DB all the parking violations verified by authorities and the solutions adopted to face them.
- **D9**: SafeStreets receives instructions from each municipality on how to divide the zones.

G5: Citizens must be able to report parking violations.

- **R1**: The system must allow the user to sign-up.
- **R2**: The system must allow the user to sign-in.
- **R3:** The system must store the segnalations sent by users in order to collect data of parking violations and where they occur.
- **R15**: The system must allow the user to attach to the segnalation a photo just taken through the app.
- **R16**: The system must allow the user to specify the type of parking violation that just occurred.

- **R17**: The system must complete the segnalation with the correct date, time, the licence plate, the name of the street and the username of the author of the segnalation.
- **D1**: The user's smartphone GPS will provide a position with an error within 20 meters.
- **D2:** The picture is taken with a camera that has at least 5 Mpx to allow the OCR algorithm to retrieve the licence plate.
- **D3:** The user's Internet Connection works properly when he interacts with SafeStreets.
- **D6:** SafeStreets has access to the GPS and camera of the device that is running the app.
- **D9**: SafeStreets receives instructions from each municipality on how to divide the zones.

G6: Municipality must be able to have access to all the anonymous segnalations sent to SafeStreets.

- **R1**: The system must allow the user to sign-up.
- **R2**: The system must allow the user to sign-in.
- **R3:** The system must store the segnalations sent by users in order to collect data of parking violations and where they occur.
- **R8:** The system must provide the requested data if and only if the user has the permission to visualize them.
- **R18**: The system, when it shows to municipality users the list of all the segnalation of the zones under their control, must take care of removing the email of the author from the reports.
- **R19**: When a municipality user chooses to visualize the list of all the violations, SafeStreets must provide him only those which occurred in the zones under the control of the specific user.
- **R20**: SafeStreets must send an email to the correct municipality to notify the violation each time it stores a segnalations sent from a zone under the control of that specific municipality.
- **D3**: The user's Internet Connection works properly when he interacts with SafeStreets.
- **D8**: SafeStreets knows an email address for each not registered municipality to which the system refers when it notifies a segnalation.

G7: Each user must be able to visualize the list of segnalations sent by him.

- **R1**: The system must allow the user to sign-up.
- **R2**: The system must allow the user to sign-in.
- **R3:** The system must store the segnalations sent by users in order to collect data of parking violations and where they occur.
- **R8**: The system must provide the requested data if and only if the user has the permission to visualize them.
- **R21**: The system must present in the homepage of each user the list of segnalations whose author matches the email of the user who logged in.
- **D3:** The user's Internet Connection works properly when he interacts with SafeStreets.

Traceability matrix

The following table shows for each requirement the relations that it has with respect to use cases. Here are listed only the use cases that are strictly linked with the corresponding requirement in order to avoid a huge number of repetitions.

R1	● Sign In
R2	● Visualize maps ● Visualize statistics ● Send segnalations ● Visualize Segnalations send by him ● Mark a segnalation as fake or verified ● Send to SafeStreets data ● Visualize all the segnalations ● Receive suggested solutions
R3	● Visualize statistics ● Visualize maps ● visualize segnalations send by him ● Visualize all the segnalations ● Mark a segnalation as fake or verified
R4	● Visualize statistics ● visualize maps
R5	● Visualize maps ● Visualize statistics
R6	● Visualize maps ● Visualize statistics ● Send data to SafeStreets ● Receive suggested solutions
R7	● Visualize maps ● Visualize statistics ● Mark a segnalation as fake or verified
R8	● Visualize Segnalations send by him ● Mark a segnalation as fake or verified ● Visualize all the segnalations ● Receive suggested solutions
R9	● Mark a segnalation as fake or verified
R10	● Receive suggested solutions
R11	● Receive suggested solutions

R12	● Receive suggested solutions
R13	● Receive suggested solutions
R14	● Receive suggested solutions
R15	● Send segnalations
R16	● Send segnalations ● Receive suggested solutions
R17	● Send segnalations ● Mark a segnalation as fake or verified ● Visualize all the segnalations ● Visualize maps ● Visualize statistics ● Visualize Segnalations send by him
R18	● Visualize all the segnalations
R19	● Visualize all the segnalations
R20	● Visualize all the segnalations
R21	● Visualize Segnalations send by him

3.3 Performance requirements

SafeStreets will reasonably have a large user base and a large amount of information within its database, so it must be able to manage everything offering a pleasant user experience for anyone who interacts with the system without any particular slowdown.

3.4 Design constraints

3.4.1 Hardware limitations

SafeStreets doesn’t have hardware limitations difficult to satisfy, in fact the only ones required guarantee the correct functioning of the system and the services offered by it. Below there is a brief outline that lists the minimum requirements that the device used by the user must satisfy:

- Internet connection
- GPS (if the user wants to send reports)
- Camera (if the user wants to send reports)

3.4.2 Any other constraint

As previously mentioned SafeStreets wants to protect the privacy of its users and for this reason, although it allows each of them to view the reports they have sent in the past, when it offers to the municipality the reports saved in its database does not allow to trace the author of the report .

3.5 Software system attributes

3.5.1 Reliability

In order to guarantee and improve reliability of the software some tools of Statistical Modeling and Estimation of Reliability Functions are used; in addition, downtimes, and how they recur at certain time intervals, are continuously monitored. In this way it is possible to forecast the reliability through testing phase; when the results obtained are not good, a special team is responsible for reinforcing the system in order to make it more robust and more fault tolerant.

3.5.2 Availability

SafeStreets wants to achieve the highest level of availability through the implementation of individual components characterized by reliability and a complex system capable of responding quickly to errors. In any case, the goal is for the system to be available at least 99.9% of the time

3.5.3 Security

Security is a fundamental aspect of every software and the same can be said for SafeStreets that wants its users' identity and privacy to be safe as well as the confidential data that the municipality sends to the system. This is why the information in the database are encrypted.

3.5.4 Maintainability

Nowadays maintainability is a prerequisite if you want to offer a software product that can always be modern and up to date. In fact, through this attribute it can evolve continuously limiting the efforts required and the associated expenses. All this is ensured by the use of design patterns and the respect of the design principles.

3.5.5 Portability

SafeStreets can work in different environments even if not all the functionalities are available on all the platforms because of some hardware limitations. However, it is possible to access the services offered by SafeSteets using a special application or the website through devices that have installed the main OS of smartphones and computers.

4 FORMAL ANALYSIS USING ALLOY

4.1 Alloy Model

In this section Alloy is exploited to model some important aspects of the system. Static constraints are modelled such as:

- Two different users have different email.
- The solution a Municipality receives concerns a zone which is in the territory of that Municipality
- All the segnalations that triggered the process that brings SafeStreets to propose a solution to the municipality occurred in the same zone.
- All the segnalations that triggered the process that brings SafeStreets to propose a solution to the municipality concern the same type of violation.
- Two different Municipality cannot share the same zone.
- A Municipality which is not registered to SafeStreets cannot receives solutions.
- A Normal User can see all the segnalations he sent.

Other more trivial, yet necessary, constraints are included. All the constraints are verified by visualizing some worlds that Alloy generates and some assertions are checked.
It should be remarked that while Location's coordinates are real numbers, we used integer numbers since Alloy supports only them. Solutions expressing floating-point numbers using integers do exists, but this is not necessary for our purpose.

```
/// signatures ///

abstract sig Bool {}
one sig True extends Bool {}
one sig False extends Bool {}

sig Email { // NormalUser's email
} {one user: NormalUser | user.email = this }

sig TypeOfViolation { // Segnalation's typeOfViolation
}

sig Location { // a location is identified via latitude and longitude
      latitude:         one Int,
      longitude: one Int
}
```

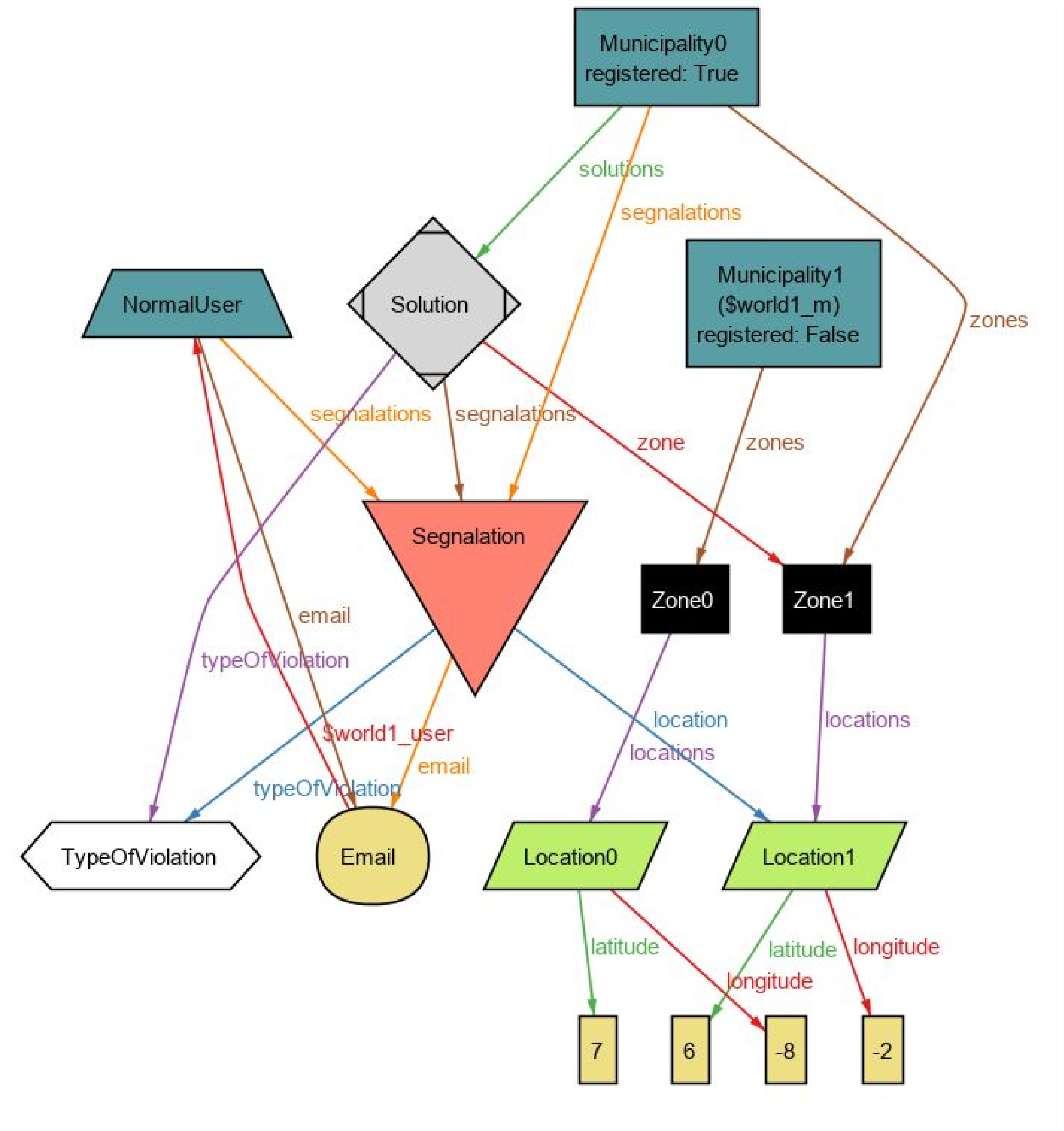
```
sig Zone { // a zone is identified via the set of its locations
      locations: some Location
}

sig Segnalation { // a location is composed of a location, its type and the normal user's email who sent it
      location:               one Location,
      typeOfViolation: one TypeOfViolation,
      email: one Email
}

sig Solution { // a solution includes a set of segnalations that occurred in a zone
      segnalations: some Segnalation,
      zone:       one Zone,
      typeOfViolation: one TypeOfViolation
}

abstract sig User { // user can be Municipality or Normal User
      segnalations: set Segnalation
}

sig Municipality extends User {
      zones:         some Zone      , // zones that are in the territory of the Municipality
      solutions: set Solution,
      registered: one Bool
}

sig NormalUser extends User {
      email:                  one Email
}

/// ~signatures ///


/// facts ///

// each municipality has all the segnalations sent in its zones
fact municipalityHasItsZonesSegnalations {
      all m: Municipality, s: Segnalation | one z: m.zones |
            s.location in z.locations implies s in m.segnalations
}

// each normal user has the segnalations he sent
fact normalUserHasSegnalationsHeSent {
      all u: NormalUser | all s: Segnalation |
            s.email = u.email implies s in u.segnalations
}

// a municipality receives solutions only if it is registered to safestreet
fact solutionToMunicipalityOnlyIfRegistered {
      all m: Municipality | m.registered = False implies m.solutions = none
}

// two different locations cannot share the same coordinates
fact locationsDiffersForCoordinates {
      no disj l1, l2: Location | l1.latitude = l2.latitude and l1.longitude
= l2.longitude
}

// each segnalation is associated to a single normal user
fact segnalationAssociatedToUniqueUser {
      all segnalation: Segnalation | one user: NormalUser | segnalation in
user.segnalations
}

// each segnalation occur in a zone
fact segnalationOccurredInZone {
      all segnalation: Segnalation | one zone: Zone | segnalation.location
in zone.locations
}


// all segnalations that triggered a solution occurred in the same zone
fact segnalationsInSolutionSameZone {
      all solution:Solution | all segnalation: solution.segnalations
            | one zoneLocation: solution.zone.locations |
segnalation.location in zoneLocation
}

// all segnalations that triggered a solution concern the same type of
violation
```

```
fact segnalationsInSolutionConcernSameTypeOfViolation {
      all solution:Solution | all segnalation: solution.segnalations
            | one tov: TypeOfViolation | segnalation.typeOfViolation = tov
and solution.typeOfViolation = tov
}

// the solutions that municipality receives occur in a zone which is in the
municipality
fact solutionConcerningMunicipalityZones {
      all municipality: Municipality | all solution: municipality.solutions
            | solution.zone in municipality.zones
}

// each municipality has different zones
fact differentMunicipalityZones {
      no disj m1, m2: Municipality | m1.zones & m2.zones != none
}

// each location belongs to only one Zone
fact locationBelongsToOneZone {
      all location: Location | one  zone: Zone | location in zone.locations
}

// each zone belongs to a municipality
fact zoneBelongsToMunicipality {
      all zone: Zone | one municipality: Municipality | zone in
municipality.zones
}

// each segnalation belongs to 0 or 1 solution
fact segnalationBelongsToMax1Solution {
      all segnalation: Segnalation | lone solution: Solution | segnalation
in solution.segnalations
}

// each segnalation is associated to a NormalUser
fact segnalationAssociatedToNormalUser {
      all segnalation: Segnalation | one user: NormalUser | segnalation in
user.segnalations
}

/// ~facts ///
```

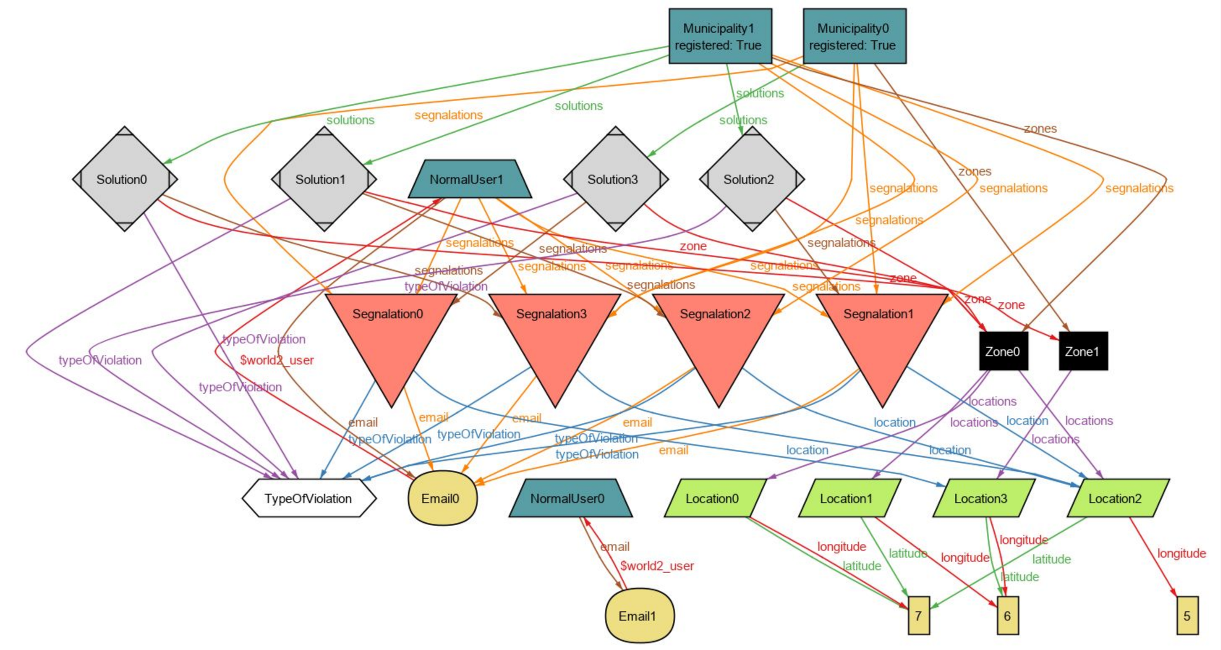
4.2 World generated

Here we present some worlds generated by Alloy which are compatible with the model.

```
pred world1 {
      (#Municipality > 1) and
      (some m: Municipality | m.registered = False) and
      (#Segnalation > 0)
}
run world1 for 3
```



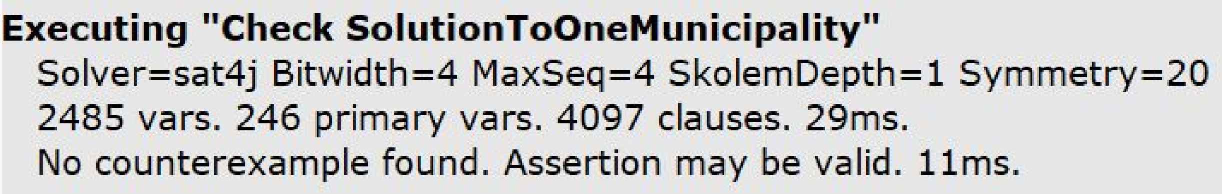
```
pred world2 {
      (#Municipality = 2) and
      (#NormalUser >=  2)  and
      (all m: Municipality | m.solutions != none)
}
run world2 for 4
```



4.3 Assertion check

We ensure that some facts follow from the above defined model.

```
assert SolutionToOneMunicipality { // a solution must be sent to only one
municipality
      no disj m1, m2: Municipality | some s: Solution | s in m1.solutions
and s in m2.solutions
}
check SolutionToOneMunicipality
```



```
Executing "Check SolutionToOneMunicipality"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   2485 vars. 246 primary vars. 4097 clauses. 29ms.
   No counterexample found. Assertion may be valid. 11ms.
```

5 EFFORT SPENT

De Vita Simone

15/10/2019: 3h
17/10/2019: 30min
18/10/2019: 45min
20/10/2019: 5h
23/10/2019: 1h
24/10/2019: 30 min
27/10/2019: 2h
05/11/2019: 2h
06/11/2019: 5h
07/11/2019: 3h
10/11/2019: 1h

Fabio Fontana

15/10/2019: 3h
17/10/2019: 45min
18/10/2019: 1h 30min
19/10/2019: 4h
21/10/2019: 45min
23/10/2019: 1h
26/10/2019: 4h
28/10/2019: 30min
29/10/2019: 2h
30/10/2019: 2h
01/11/2019: 2h
02/11/2019: 1h
03/11/2019: 2h
07/11/2019: 4h
08/11/2019: 1h
10/11/2019: 1h