SAMUELE MOSCATELLI, ANDREA POZZOLI

SAFESTREETS - RASD

Version 1

10th November 2019

SAFESTREETS

Software Engineering 2 Project

Andrea Pozzoli and Samuele Moscatelli

SafeStreets RASD

Software Engineering 2 project

Politecnico di Milano

TITLEBACK

This document was written with LaTeX

CONTACTS

☞ pozzoliandrea97@gmail.com

☞ sem.mosca@gmail.com

CONTENTS

1 INTRODUCTION

1.1 PURPOSE

1.1.1 General purpose

This RASD (Requirement Analysis and Specification Document) document aims to give a complete and rigorous description of the system "SafeStreets". In particular, this will be done through a deep analysis of the customer's needs, so identifying the main goals of the project, describing the functional and non-functional requirements and then proposing a solution able to fulfil all of them. This document is addressed to clients, but not only, in fact it is meant to be used as a contractual basis to which all the members of the team designed to create "SafeStreets" have to refer.

1.1.2 Goals

The Goals listed below are the result of a detailed analysis process related to the purpose of the system and the phenomena that may characterize it. In particular, following the model proposed by M. Jackson and P. Zane, a distinction has been made between the phenomena that characterize the world, those that characterize the machine and finally also those shared between both the two parts. The World and Machine analysis will be shown in the next section.

G1 A person (end user) who sees a traffic violation should be able to notify authorities of the violation in every moment and situation (such as with lack of internet connection), participating to street regulation.

G2 A person (end user) should be sure that reporting a violation does not put him under any kind of risk of retaliation, so no one can know the identity of who has reported the violation.

G3 An authority should be able to know the recent traffic violations.

G4 A user should be able to know the statistics regarding traffic violations.

G5 Municipality should be able to collaborate with SafeStreets in order to provide information about accidents occurred in its territory.

G6 Municipality should be able to know the most unsafe areas of its territory.

G7 Municipality should be able to know possible interventions it can do in order to improve the unsafe areas of its territory.

G8 An authority should be able to know when violations occur around him.

G9 A user should be recognized and distinguished based on his role (end user, authority or municipality).

G10 An authority should be able to inform other authorities about its intentions to go to verify a traffic violation.

G11 A person (end user) should be able to know his contribution in traffic regulation.

1.2 SCOPE

1.2.1 Description of the problem

SafeStreets is a crowd-source application that aims to give users the opportunity and responsibility of contributing to road regulation, with more attention to parking violations. In particular, the system offers functionalities which allow users reporting infringements, describing when, where and how they have been perpetrated. In order to certify the reporting, the user has also to attach a picture of the violation, making sure to include the license plate of the vehicle, so that the system can have a guarantee of the truthfulness of the information and at the same time identify the transgressor. So, for example, if a citizen, while walking through via Golgi on 30th October 2019, sees a car with license plate XXX parked in the middle of the bike lane, he can open SafeStreets, take a photo of the vehicle location, insert the type of violation and a description and then send the report to the system. Once received the information, the application reads the license plate from the picture and store it together with the other data provided by the user and the position, date and time. So, in the example previously shown, SafeStreets would memorize a parking violation of the type "car on bike lane" in via Golgi, on 30th October 2019, with license plate XXX, and a brief description of the situation. All the traffic violations sent by an end user are not lost, in fact every end user can see on the application his past contribution to the traffic regulation. An end user cannot see the traffic violations sent by the other users.

Citizens are not the only type of users of the application, also authorities, in fact, can use it in different ways. In first place, they can consult SafeStreets in order to retrieve information about the situation of streets which are under their jurisdiction. Moreover, an authority can warn other authorities that he is going to verify if a reported traffic violation is really an infringement, so that two authorities don't go to verify the same violation.

Another type of user is municipality user, who can collaborate with SafeStreets with the aim of making roads under his jurisdiction safer through prevention. In particular, municipality provides information about accidents that has occurred and occur on its territory to the system. The system then can merge this data with those coming from violations and in this way it can identify the most dangerous areas, and at the same time suggest the best interventions that can be applied to make them safer.

In addition, when the system receive a traffic violation from an end user, it is elaborated and combined with the other data already stored, creating statistics. The possible statistics are: which streets are characterised by the highest number of infringements, in which moment of the day there are more violations, which type of violation is most perpetrated. The statistics regarding traffic violations can be accessed by all the three types of user. Instead data regarding a specific traffic violation is visible only by authorities.

1.2.2 "The World and The Machine" analysis

Phenomena	**Shared**	**Who controls**
User sees a traffic violation	NO	World
User wants to notify authorities	NO	World
User launches the application and logs in with his own credentials	YES	World
SafeStreets software is loaded checking the role of the user (end user or authority or municipality)	NO	Machine
User inserts the picture, the type and the description of the violation	YES	World
SafeStreets application detects date, time and position from the device	NO	Machine
SafeStreets asks to the user to insert again some wrong data	YES	Machine
User sends the violation report	YES	World
SafeStreets receives a violation	NO	Machine
SafeStreets runs the algorithm to read the license plate	NO	Machine
The algorithm can't read the license plate so SafeStreets asks to the user to insert another picture or to insert the license plate manually	YES	Machine
User inserts the license plate manually	YES	World
SafeStreets stores the retrieved information	NO	Machine
SafeStreets calculates the statistics	NO	Machine
User or authority or municipality mines the information	YES	World
SafeStreets receives data about accidents from municipality	YES	World
SafeStreets identifies unsafe areas	NO	Machine
SafeStreets suggests interventions	YES	Machine
SafeStreets goes out of service	YES	Machine
Safestreets notifies authorities near the violation which has occurred	YES	Machine
An authority goes in the place of the violation	NO	World
An authority notifies other authorities that he is going to check the violation	YES	World
An authority receives the notification that another authority is going to verify a violation	YES	Machine
An end user wants to see his past contributions to SafeStreets	NO	World

1.3 DEFINITIONS, ACRONYMS, ABBREVIATIONS

1.3.1 Definitions

- **End user:** The end user is a person that sees a traffic violation and wants to notify the authorities about it by using SafeStreets application. He can't see the violations sent by other users. He can see the statistics. End user is one of the three user types of the system.

- **Authority:** Authority is the second type of user and usually he is a police man. He does not send violation data to the system. He can see the violations sent by the end users. He can access to the statistics.

- **Municipality:** Municipality represents mayor and municipal employees of a city which decide to collaborate with SafeStreets and it is the third type of user. Municipality sends data about accidents to SafeStreets in order to allow it to cross them with violations data and find out the unsafe areas on its territory. He also can see the statistics.

- **System:** The system is a synonymous of SafeStreets. The system receives data form the end users, elaborates and stores data, shows data to the authorities, calculates statistics and unsafe areas, suggests interventions.

- **Traffic violation:** Data sent by a end user is called traffic violation or only violation. A traffic violation is composed by a license plate (taken from a picture or a text inserted by the user), date, time, GPS position, the type of violation and a description of it. Example of violations can be vehicles parked on the stripes or in places reserved to people with disabilities, double parking, parking in no parking places.

- **Statistics:** The statistics are some information calculated by SafeStreets in order to highlight the streets with a high number of violations, the days and times at which there are more violations, the most common types of violations.

- **Unsafe Area:** Municipality sends to SafeStreets all the accidents occurred in a city. The streets and the areas in which there is a high number of accidents are called unsafe areas. An unsafe area can be only in one municipality jurisdiction and two different municipality jurisdiction cannot have the same unsafe area.

- **Intervention:** After having discovered some unsafe areas, SafeStreets suggests to municipality some actions (interventions) to do in order to make these areas safer.

- **Accident:** an accident is data that municipality sends to SafeStreets in order to find the unsafe areas and suggest interventions. It describes a dangerous situation that occurred in one of the streets or areas under the municipality jurisdiction.

1.3.2 Acronyms

- RASD – Requirement Analysis and Specification Document
- API - Application Programming Interface
- GPS - Global Positioning System
- RMI - Remote Method Invocation
- UML - Unified Modeling Language

1.3.3 Abbreviations

- Gn: n-goal.
- Dn: n-domain assumption.
- Rn: n-functional requirement.
- rn: n-row of the matrix.

1.4 REVISION HISTORY

This is the first version of the document. The date of release of this document is 10th November 2019.

1.5 REFERENCE DOCUMENTS

- SafeStreets Mandatory Project Assignment
- Software Engineering 2 course slides
- Past projects of the course

1.6 DOCUMENT STRUCTURE

The first chapter is a brief introduction to the document. It describes the purpose and the scope of SafeStreets software. In order to understand better the document, the first part also presents definitions, acronyms and abbreviations that will be used in the document.

The second chapter enters more in detail in the description of the project SafeStreets. In the first section the shared phenomena are presented and the domain model is explained through a class diagram and some state-chart diagrams. Then we define the most important requirements that the software has to respect. They are divided in three subsections, each subsection presents a functionality of the system. In the third section instead we specifies which are the users of the system and which are their needs and the functionalities that they can use. Finally, there is a section in which there are the assumed domains and other constraints.

The third chapter focuses the attention on specific requirements. First of all, the requirements related to external interfaces are described, dividing them in different categories. Then, all the requirements are mapped and explained through different UML diagrams 1. In fact, we defined the use case diagrams for each type of user, the use cases for each functionality and the sequence diagrams for explaining better the order of the operations and the relationships between users. After that, we presented some performance requirements in a textual way. Then, in the conclusion of the chapter, the design constraints and the software system attributes are defined and described.

In the following chapter, we adopted Alloy in order to explain with a formal notation the most critical part of the requirements.

The last two chapters present the effort spent by Andrea Pozzoli and Samuele Moscatelli in order to complete this document. The effort is measured by the hours of work. Finally there are the references consulted during the development of the project.

2 OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

The main idea is firstly to build a basic service which represents the core of the application and which consists in giving to citizens a simple and fast way to report traffic violations which occur around them. In this way they could be encouraged to participate in the regulation of roads without the necessity of directly contact authorities and without exposing themselves to risks related to retaliation. Subsequently, the intention is to enrich the offer of the application also including the authorities and municipalities among the types of users and guaranteeing them functionalities that can be of great help in their daily tasks.

The class diagram showed below aims to give a high-level description of the system. In particular, it is possible to notice that the application is addressed to three types of users, the end users which represent the category of the citizens, the authority users and the municipality users. The starting point, as previously said, is represented by the end user, who sees a violation and sends a violation data to SafeStreets, including the date, the time, the position and the licence plate of the vehicle subject of the infringement. As a consequence of this, an authority user can see the traffic violation data and he can go to check the violation.

Moreover, municipality can also collaborate with SafeStreets by providing accidents data regarding its territory, in this way allowing the system to individuate and show information about the most unsafe areas within the municipality jurisdiction and about the best type of intervention that can be performed.

Finally, all the types of users are allowed to access to statistics on the traffic violations, which is a feature that can be very useful in the specific case of authorities and municipality users.

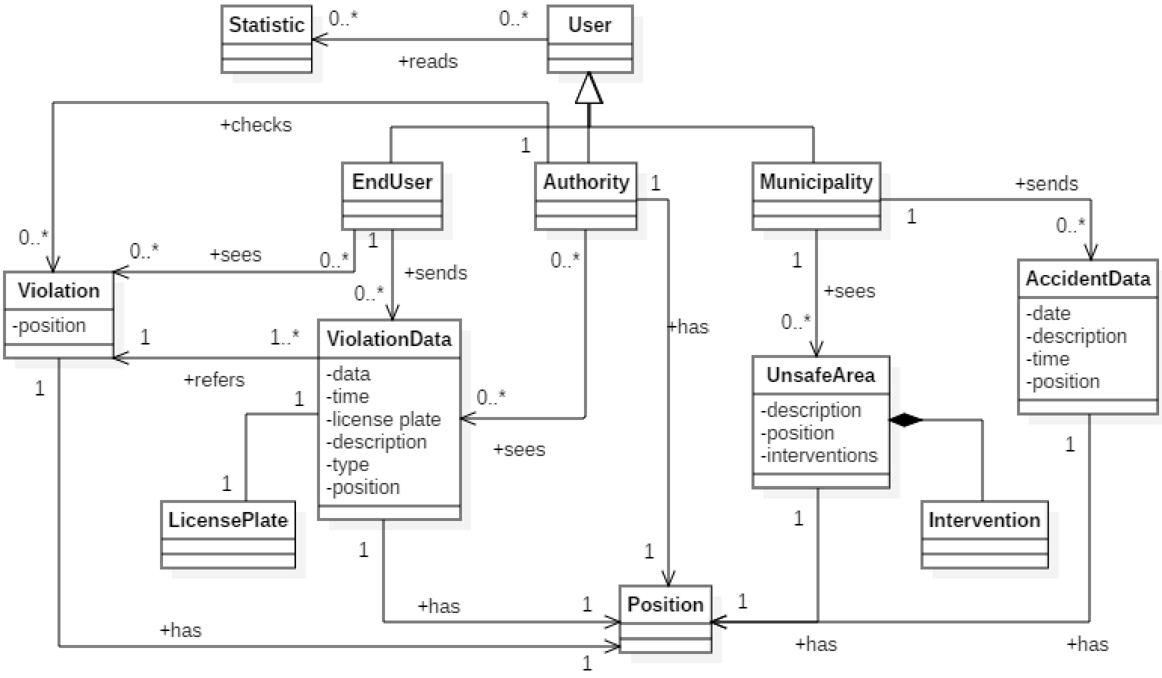


Figure 1: Class Diagram

Now, the attention is shifted on explaining with more precision the principal phenomena that characterize both the system and the outside world from the point of view of the application, and in order to do so 3 state-chart diagrams will be shown.

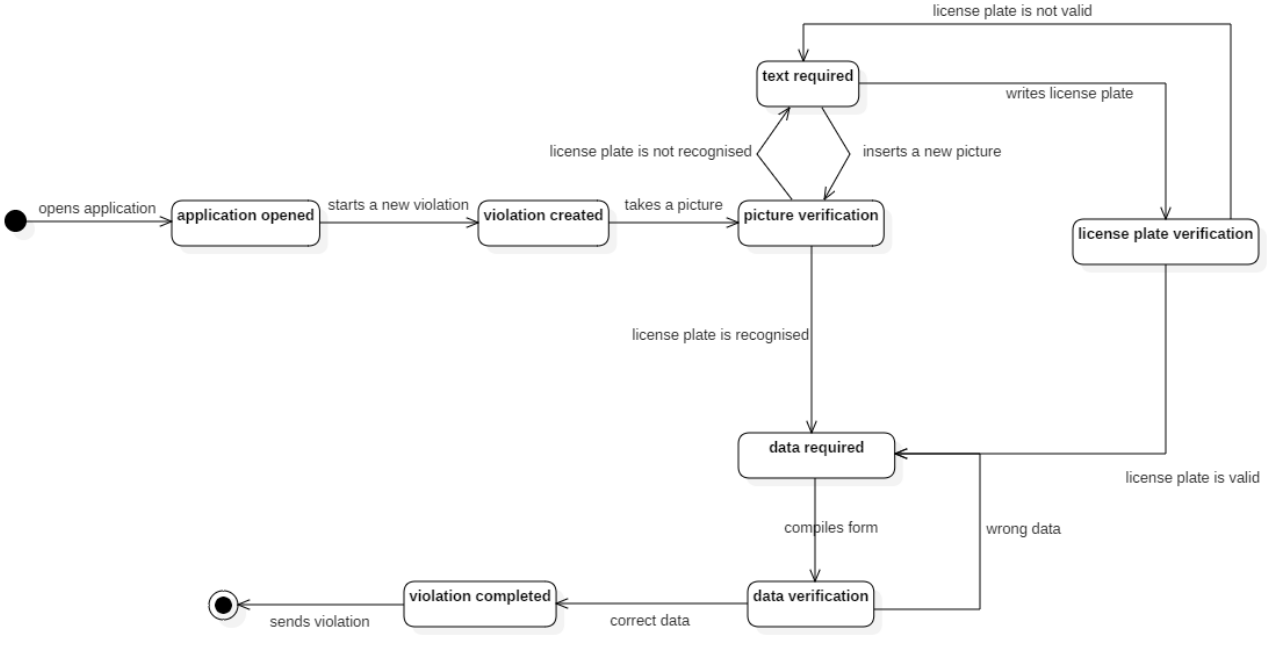


Figure 2: State-chart Diagram: Report traffic violation

In this first state-chart diagram, it is explained in further details the situation in which an end user reports a violation. It is taken into account also the case in which the system is not able to read the license plate through the picture inserted by the user, so he has to insert a new one or write it down manually.

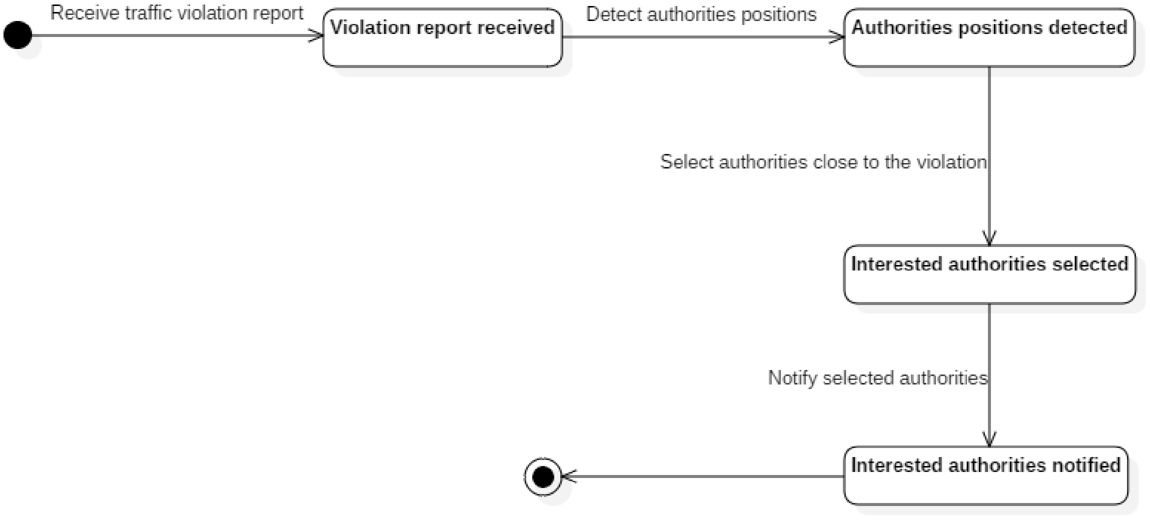

Receive traffic violation report
Violation report received
Detect authorities positions
Authorities positions detected
Select authorities close to the violation
Interested authorities selected
Notify selected authorities
Interested authorities notified

Figure 3: State-chart Diagram: Notify authorities

In this second state-chart diagram the attention is concentrated on the situation in which a violation report is received by the system. The system, as a consequence, detects the position of all the authorities and selects those who are close to the place in which the violation occurred, so that if someone of them wants, he can go to check the infringement in person.

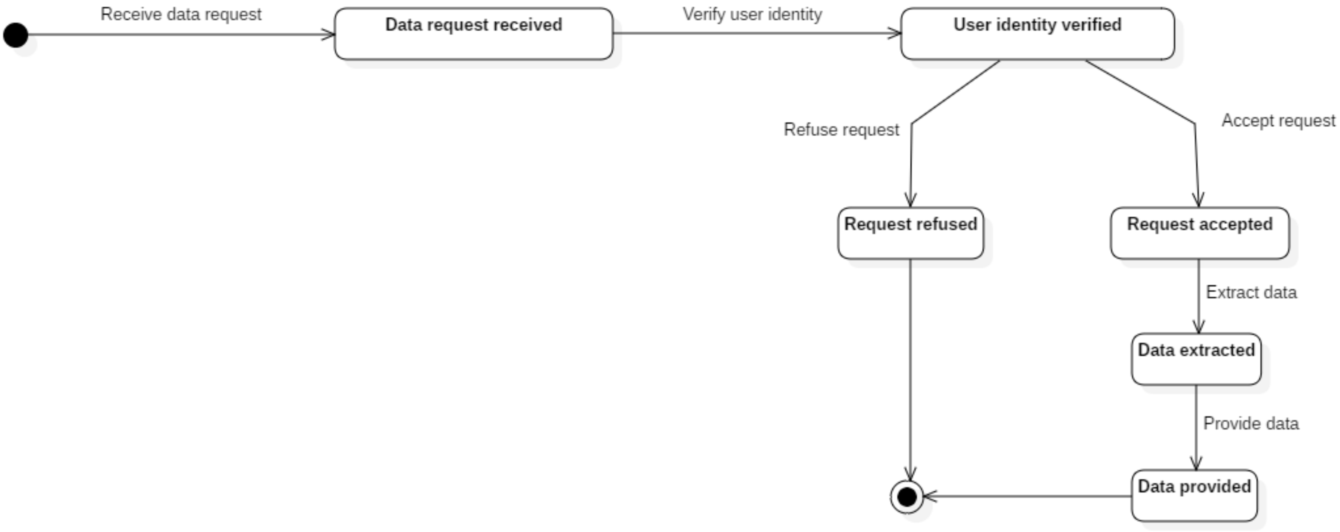


Figure 4: State-chart Diagram: Data requested

In this third state-chart diagram, the situation, in which the system receives a request from a user for one type of information, is shown (it can be a request for the traffic violations statistics or for the unsafe areas and suggestions). Safestreets firstly verifies the identity of the user who made the request and then verifies that, with his role, he can access to that kind of data.

2.2 PRODUCT FUNCTIONS

2.2.1 Traffic violations reporting

This function represents the core of the application, because, in fact, without it all the other functionalities offered by SafeStreets would be useless. The application, in particular, must allow any end user to easily report a violation in the instant in which he sees it. Going in further details, SafeStreets application firstly has to allow citizens to take a picture of the violation, remembering them to include in the photo also the license plate of the transgressor. Then it has to let user select a traffic violation type and also insert a brief description of the situation, so that they can express all the inconveniences originated by the abuse. While providing the information, the application software must check for the correctness of them. In particular, it must make sure of the legibility of the license plate from the picture, and, if it is not so, SafeStreets must warn the user, making him choose between taking another photo or writing the license plate number himself. Once provided all the information, the citizen can send the violation data, and in this instant the application detects from the user device the date, the time and also the position in which the violation has been sent. In case that it is not able to detect one or some of this data SafeStreets has to notify again the user so that he can fix the problem (for example enabling geolocation). Once received the complete information, SafeStreets dispatching software searches for the nearest authority users to the place of the violation and notifies them of the occurrence of it, showing all the details. After receiving it they can decide to go there to solve the problem, so the software system allows them to warn their colleagues who have been notified of the same violation, so that not too many authorities are going to be busy to solve the same problem. Authority users can also request the list of all the reported violations, and read the details of each one. In every moment, an end user can see his activity on the application and so all the past traffic violations he sent.

2.2.2 Statistics elaboration

Another fundamental service offered by SafeStreets is the possibility of mining traffic violations information from the data sent by the end users. In more details, each time the system receives a violation, this is analyzed, and some information are extracted, elaborated and used to create statistics reports. In particular, SafeStreets calculates the places with the highest number of infringements, the most common type of infringements and the period of time in which they are more frequent. This can be very useful for all the types of users, in fact end users can be aware of the most dangerous areas and try to avoid them, while authorities and municipality can exploit this data to improve road regulations by enforcing controls where violations are more frequent. For this reasons, SafeStreets must keep statistics always updated and provide them to each user who request them.

2.2.3 Unsafe area detection and intervention suggestions

In parallel with the elaboration of the statistics, SafeStreets must offer a function, this time dedicated only to municipality users, that allows them to know dangerous areas in their territories. A municipality user inserts data regarding accidents occurred in their region, such that the system can cross them with information related to traffic violations received by end users and provide a report on the safeness condition of the territory under the jurisdiction of that municipality. At the same time, SafeStreets has to analyse the elaborated data itself and by evaluating which type of events are more common in a specific area, it has to suggest the best interventions that can be done to make that area safer. In further details, the system first checks the type of the violations and accidents that frequently occur in a definite place and then, for those which happen really often, it also takes under consideration the description of them. In this way, if the problem mostly affects a particular category of citizens, SafeStreets can provide a more precise recommendation. As an example, if there is a specific bike lane on which drivers often park, the cyclists that uses to go biking there can report it each time this happens describing the situation. As a consequence, SafeStreets notices that and reports it to the municipality, with the suggestion of adding a barrier between the bike lane and the part of the road dedicated to motorized vehicles.

2.3 USER CHARACTERISTICS

2.3.1 End User

The end user is a person that can notify the authorities about a traffic violation. He must have a device with an internet connection, GPS and a camera. The end user must install on his device SafeStreets application and register in it. He must keep GPS and internet connection active when he uses the application. Every time the end user sees a traffic violation he can starts a new traffic violation report. The end user can see the statistics regarding traffic violations. He cannot see the traffic violations sent by other end users. He cannot see the unsafe areas and the suggested interventions. The end user is not notified when there is a new traffic violation near him. The end user can see his past contributions to the application. The end user cannot see the past contributions of other end users.

2.3.2 Authority

The authority is a person that can see the traffic violations sent by end users. He must have a device with internet connection. The authority must install SafeStreets software on his device and register in it. During the registration, it is necessary to verify that he is really an authority. He cannot create a new traffic violation report. The authority can see the statistics regarding traffic violations. He cannot see the unsafe areas and the suggested interventions. Authority must always keep GPS and internet connection active, also when he is not using SafeStreets software. He can be notified if a new traffic violation sent by an end user is near him. Authority can notify other authorities that he is going to verify if a traffic violation is true or not. If the authority checks (verifies) a violation, other authorities must no longer be able to see it as a notification. The authority user cannot see who create a traffic violation, and he cannot see the past contribution of end users.

2.3.3 Municipality user

The municipality user represents a municipality that collaborate with SafeStreets. He must send to SafeStreets data regarding accidents that occur in his jurisdiction. Municipality can see the unsafe areas calculated by SafeStreets in his jurisdiction. He can also see the possible interventions suggested by SafeStreets in his jurisdiction. Municipality must have a device with internet connection. He must install SafeStreets software on his device and register in it. During the registration, he must provide a governmental number that proves he is really a municipality. He cannot create a new traffic violation report. He cannot see traffic violations sent by end user. Municipality is not notified when there is a new traffic violation near him. He can see the statistics regarding traffic violations. The municipality user cannot see who create a traffic violation, and he cannot see the past contribution of end users. A municipality user cannot see the unsafe area of other municipality users.

2.4 ASSUMPTIONS, DEPENDENCIES AND CONSTRAINTS

D1 A person (end user) knows the traffic rules.

D2 A person (end user) knows that he can notify the authority if there is a traffic violation.

D3 A person (end user) has a device with a camera, internet connection and GPS sensor.

D4 A user (End user, Authority or municipality) knows SafeStreets and has a device on which there is SafeStreets software and internet connection.

D5 Users are fair with each other, so they do not lie when reporting a traffic violation.

D6 End users report a violation from the position where the violation occurred.

D7 An authority never communicates details of a traffic violation report to no one.

D8 An authority user is able to reach the position of a violations when notified.

D9 An authority has a device on which there is a GPS sensor.

D10 An interface to check through a governmental code if an authority or a municipality is really such is provided by the government.

D11 A municipality has data about accidents occurred in its jurisdiction stored and can provide them.

All traffic violations created by end users are sent to SafeStreets through internet connection. The position of a traffic violation is taken automatically from the GPS of the end user's smartphone. Date and time are taken automatically, too. The notifications received by the authorities are sent through internet connection. Authorities need GPS because they receive notifications only if they are near a new traffic violation. Authorities and Municipality must demonstrate their role during the registration trough a governmental code which is then verified through the government interface, such that it is possible to avoid that common citizens register themselves as authority or municipality. Accidents data, unsafe areas and suggested interventions are sent through internet connection. End users must have a device with a camera because they have to take a picture of the license plate of the vehicle that commits the traffic violation. A traffic violation report is shown to authorities and municipalities in an anonymous way. When an end user inserts data regarding a traffic violation he has to choose a type of traffic violation among different predefined types and then he can give a description of the situation, so he has to know the street rules. Each municipality has its own territory, and it can see the unsafe areas only of its territory. Two territories cannot have the same unsafe area.

3 SPECIFIC REQUIREMENTS

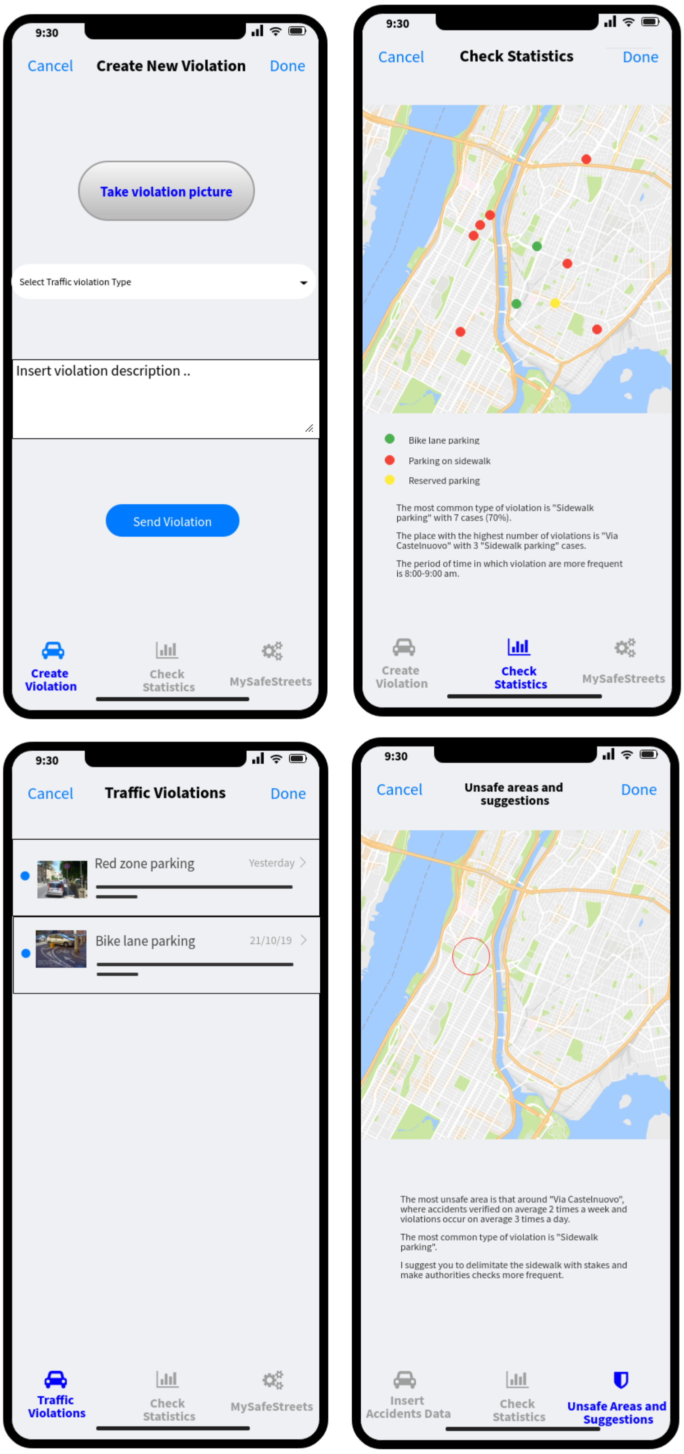
3.1 EXTERNAL INTERFACE REQUIREMENTS

3.1.1 User Interfaces

SafeStreets has three different user interfaces, one for each type of user:

- End user: application with a way to insert a new traffic violation and see past contributions.
- Authority: application with a way to see all the traffic violations, a way to notify other authorities, a way to be notified.
- Municipality: software with a way to see unsafe area and suggested interventions, a way to send accidents.
- All users: software with a way to see statistics.

Here are shown four mockups in order to give an approximate idea of how SafeStreets user interfaces should appear. The first mockup (top-left) shows how an end user can insert data regarding a traffic violation, specifying license plate, type of infringement and a brief description. The second mockup (top-right) represents the page of the statistics that is visible to all the types of user. In the third picture (bottom-left) an authority user can see some recent traffic violations posted by the end users. The last mockup shows the page of unsafe areas and suggested interventions that a municipality can see in order to improve his territory.



The most common type of violation is "Sidewalk parking" with 7 cases (70%).

The place with the highest number of violations is "Via Castelnuovo" with 3 "Sidewalk parking" cases.

The period of time in which violation are more frequent is 8:00-9:00 am.

The most unsafe area is that around "Via Castelnuovo", where accidents verified on average 2 times a week and violations occur on average 3 times a day.

The most common type of violation is "Sidewalk parking".

I suggest you to delimitate the sidewalk with stakes and make authorities checks more frequent.

Figure 5: Mockup examples of SafeStreets user interfaces 2

3.1.2 Hardware Interfaces

For end users: device with internet connection for sending traffic violations, GPS for catching the position while he sends a traffic violation and camera for taking the picture of the license plate.

For authorities: device with internet connection for sending and receiving notifications and for downloading traffic violations sent by end users, and GPS in order to be notified for a near violation.

For Municipality: device with internet connection for downloading data regarding unsafe areas and suggested interventions and for sending accidents data.

3.1.3 Software Interfaces

The system uses a map in order to show traffic violations statistics to the users, and also to show unsafe areas to municipality users. Moreover the application uses a software that allows end users to take pictures.

3.1.4 Communication Interfaces

The system uses internet and connectivity protocols and interfaces.

3.2 FUNCTIONAL REQUIREMENTS

This section firstly shows the use case diagrams associated to each of the three types of user of SafeStreets, then, in the remaining part, functional requirements are analyzed in three subsection, each one of them referring to one of the product functions individuated in the homonymous paragraph above in the chapter 2. Moreover, for the most critical situations also the sequence diagram is showed. Finally, functional requirements elicitated from the analysis are presented.

Use case diagram 1: End User

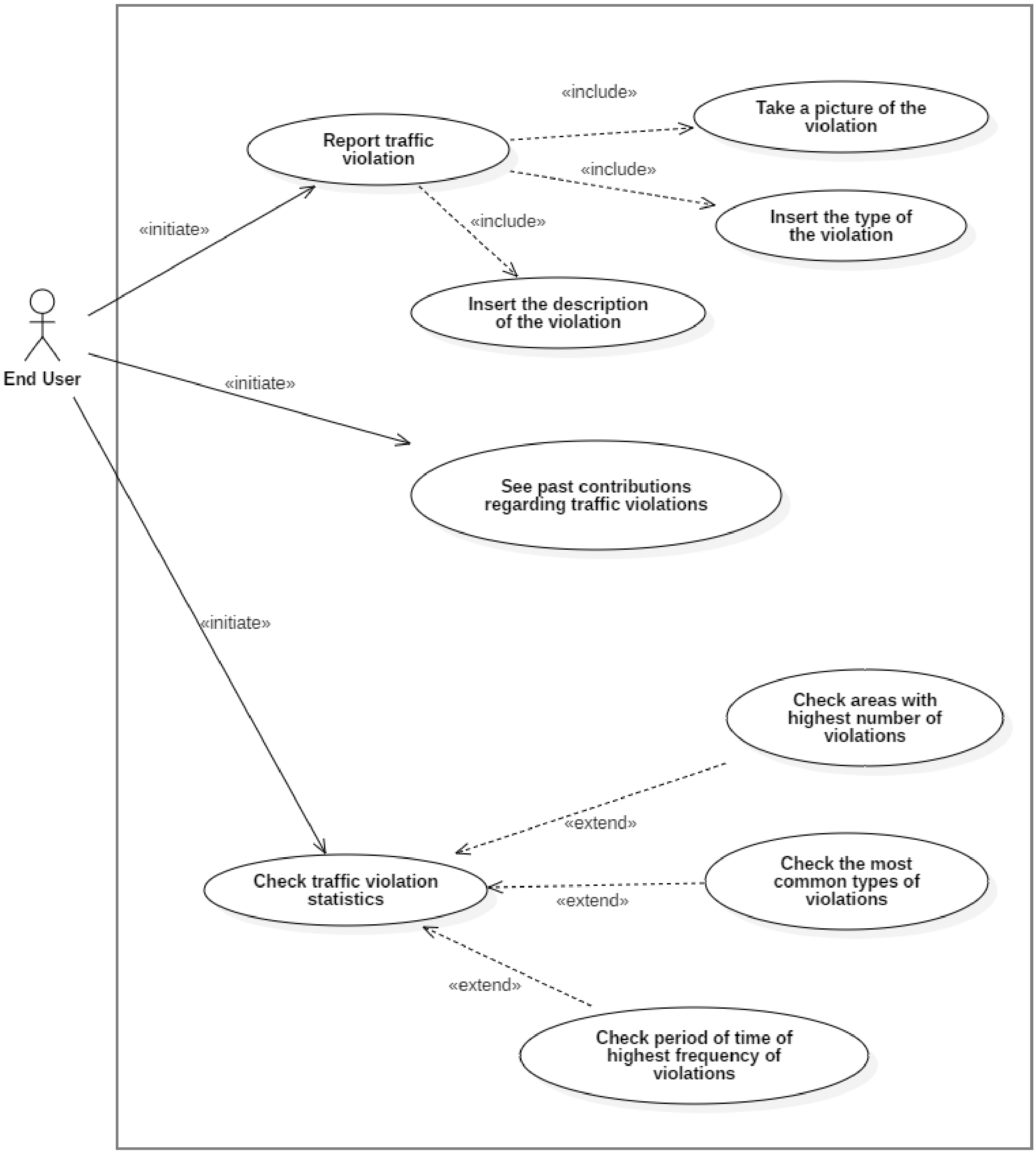

«include»
Take a picture of the violation
Report traffic violation
«include»
Insert the type of the violation
«initiate»
«include»
Insert the description of the violation
End User
«initiate»
See past contributions regarding traffic violations
«initiate»
Check areas with highest number of violations
«extend»
Check the most common types of violations
Check traffic violation statistics
«extend»
«extend»
Check period of time of highest frequency of violations

Figure 6: Use Case Diagram: End User

Use Case Diagram 2: Authority

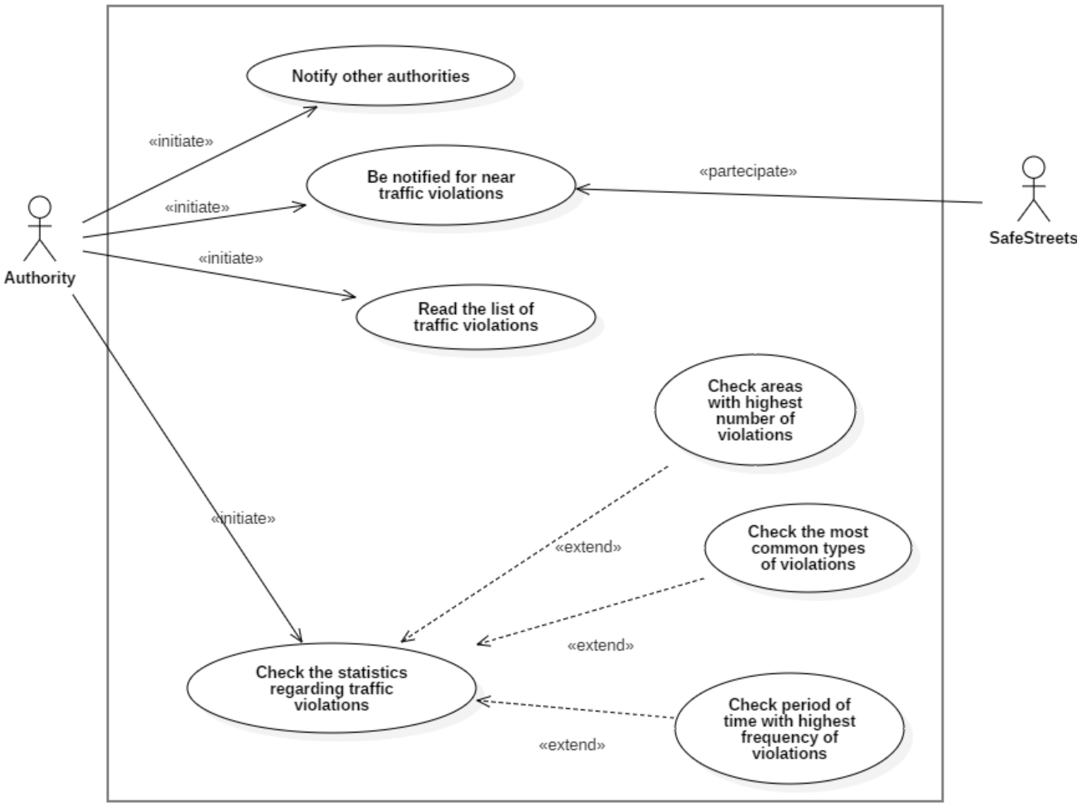

Notify other authorities
«initiate»
Be notified for near traffic violations
«partecipate»
«initiate»
Authority
SafeStreets
«initiate»
Read the list of traffic violations
Check areas with highest number of violations
«initiate»
«extend»
Check the most common types of violations
«extend»
Check the statistics regarding traffic violations
Check period of time with highest frequency of violations
«extend»

Figure 7: Use case diagram: Authority User

Use Case Diagram 3: Municipality

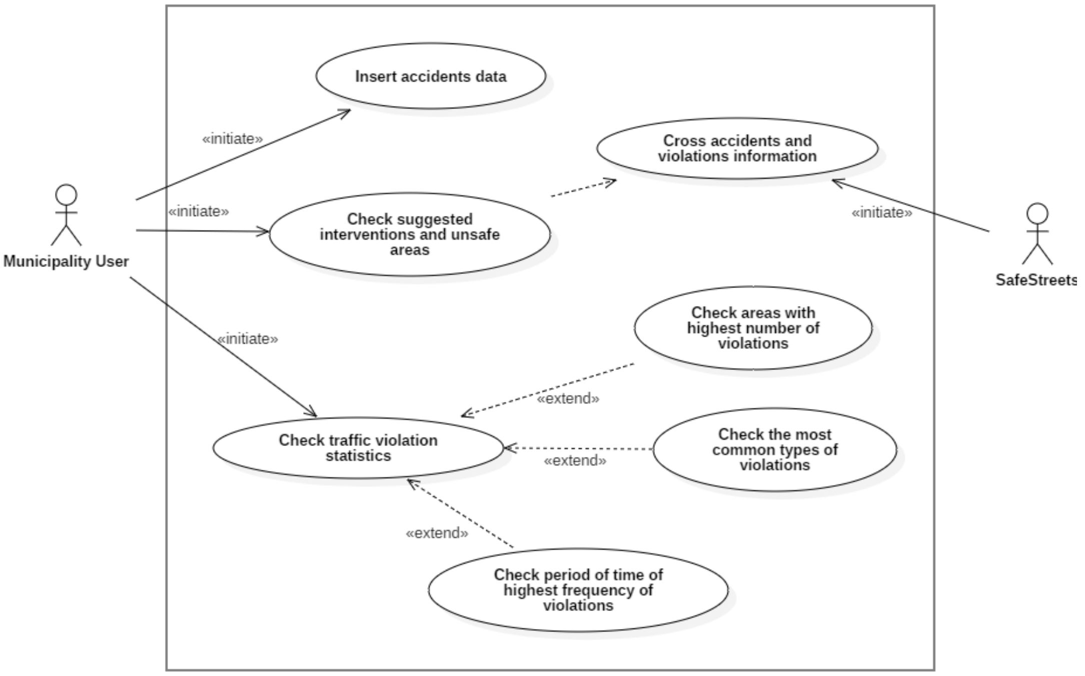

Insert accidents data
«initiate»
Cross accidents and violations information
«initiate»
Check suggested interventions and unsafe areas
«initiate»
Municipality User
SafeStreets
Check areas with highest number of violations
«initiate»
«extend»
Check traffic violation statistics
«extend»
Check the most common types of violations
«extend»
Check period of time of highest frequency of violations

Figure 8: Use case diagram: Municipality User

3.2.1 Traffic violations reporting

Scenarios

Scenario 1

Mr. Fanelli has just had a bad injury while playing a rugby match with his team. Since he broke his left elbow and his right knee, he can't use crutches, so he is obliged to use a wheelchair for a couple of months. During this period, he really likes being outdoors, but one day, he is forced to stop his morning tour because of a car parked immediately after a tight curve in the middle of the sidewalk. For this reason, in order to be able to resume his ride, he is obliged to pass in the middle of the road reserved to motorized vehicles, exposing himself to the danger of being hit by a car. At that moment, he remembers of having SafeStreets application installed on his smartphone, so he extracts the device, opens the application, logs in and starts creating the violation report. So, he takes a picture of the car with its license plate and sends it together with the type and the description of the violation to SafeStreets, which detects date, time and position of the infringement and stores it.

Scenario 2

Luca is a very civilized businessman who is really annoyed by who does not respect the rules, so after learning about the existence of SafeStreets he installed it on his device and each time he is in the city he always pay attention that traffic violations do not occur around him. One day, being late for work he was rushing to the office and while crossing the street he noticed that a scooter was parked just on the pedestrian crossing. Even if he was really in a hurry he decided to report the violation to SafeStreets, so he took a picture of the vehicle and after having filled out the other fields he tried to send the report, but just after having clicked the "Send" button the application warned him that the photo was not clear enough to allow the reading of the license plate. Luca had no time to take another picture because he was really late. However, fortunately, SafeStreets, allows to choose between taking the picture again or write the license plate manually, so the businessman chose the second option and managed to write down the information in the proper field and send the violation report to the system without loosing too much time.

Scenario 3

Claudio is a young adult and he has a one year old daughter, Sofia. One Saturday afternoon, Claudio was at home with Sofia, and because of it was a beautiful and sunny day, he decided to go out with her to take a walk. So he took the stroller and went out, but suddenly he had to stop because a car was parked on the sidewalk. Claudio is an user of SafeStreets, so he took his smartphone and sent a traffic violation. After that, he continued his walk with Sofia and they stayed out for about one hour. While they were coming back, a car was parked again in the same place, but Claudio wasn't sure that it was the same car as before. So he opened SafeStreets, he clicked on the section of the past contributions and looked at the last traffic violation sent. With his surprise he discovered that the car on the sidewalk wasn't the same, so, again, he took a photo of the license plate, inserted type and description of the violation and sent it.

Scenario 4

Mr. Donini is a police officer who works at the police station of Lecco. Every Wednesday he has the duty of fulfilling a patrol ride with the aims of discovering traffic violations in Lecco. Because of Lecco is a quite big city, it is a bit difficult to be efficient in finding traffic violations, and so he wastes a lot of his time. One day, after a football match with his colleagues, Mr. Del Forte said him that he had installed a fantastic application called SafeStreets that helps authorities in discovering traffic violations. Therefore, Mr. Donini downloaded SafeStreets app and the following Wednesday he tested it. With his surprise, that day he found more traffic violations than the times before. He followed his usual patrol ride, but each time, he received a notification by SafeStreets regarding a traffic violation, he changed course and he went to the position indicated by the application. In fact, SafeStreets must allow an authority to be notified when he is near to a new traffic violation published by the end users. The notification includes the type of traffic violation, a description of it and its position, so Mr. Donini could decide if it was useful changing course or not. If Mr. Donini decided to go to the traffic violation reported by SafeStreets, he was able to communicate it to his colleagues by sending a notification through the app. In this way, he avoided that two police officers went to the same position for a single traffic violation. Moreover, he could see if there were traffic violations, for which he was not notified (maybe because he was a bit far from them) and so he could decide to go there to verify them. In fact, if an authority opens SafeStreets application, there is a list of traffic violations published by the end users again with type of traffic violation, date, time, position and description.

Use case 1: Report traffic violations

Name	Report traffic violation
Actor	End User
Entry condition	The user sees a traffic violation and wants to report it to authorities.
Event flow	1. In the homepage, the End User clicks on the "Create Violation" button entering in the violation creation page. 2. The Application requests to the End User to take a picture of the violation (remembering him to include the license plate of the car). 3. The End User takes a picture of the violation and inserts it in the application. 4. The Application requests to the End User to select the type of the violation. 5. The End User selects the type of the violation. 6. The Application requests to the End User to insert a description of the violation. 7. The End User inserts the description of the violation. 8. The End User clicks on the "Send violation" button. 9. The Application detect date, time and position from the device and then send the violation report to the System.
Exit condition	The violation report is created and sent to SafeStreets.
Exception	The Application can't read the license plate from the provided picture, so it notifies the End User asking him to take another picture or to insert the license plate manually. The Application can't detect the position from the device, so it notifies the End User asking him to activate the geolocation.

Sequence diagram 1: Report traffic violations

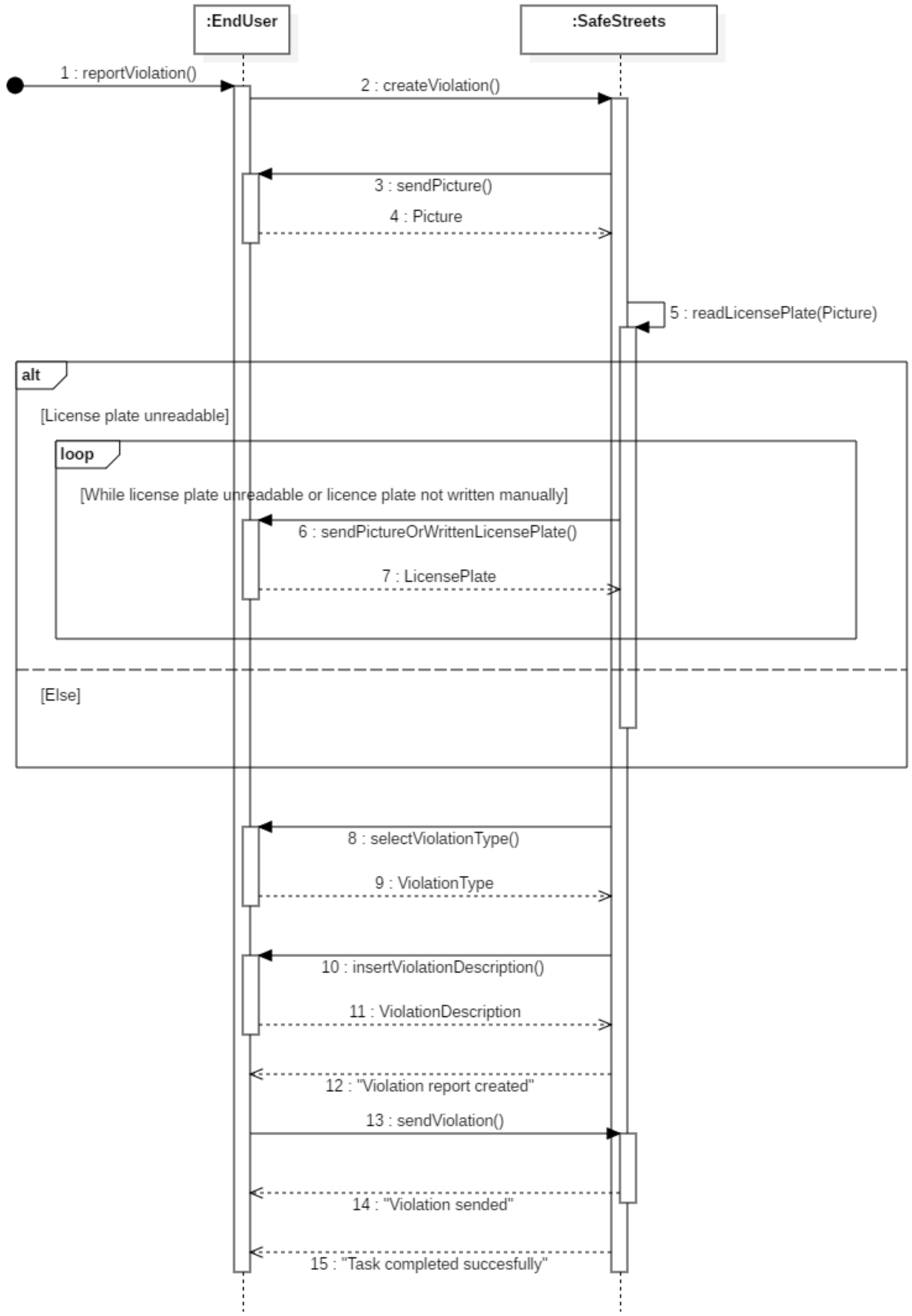

:EndUser
:SafeStreets
1 : reportViolation()
2 : createViolation()
3 : sendPicture()
4 : Picture
5 : readLicensePlate(Picture)
alt
[License plate unreadable]
loop
[While license plate unreadable or licence plate not written manually]
6 : sendPictureOrWrittenLicensePlate()
7 : LicensePlate
[Else]
8 : selectViolationType()
9 : ViolationType
10 : insertViolationDescription()
11 : ViolationDescription
12 : "Violation report created"
13 : sendViolation()
14 : "Violation sended"
15 : "Task completed succesfully"

Figure 9: Sequence Diagram: Report traffic violation

In this first sequence diagram, we want to show the flow of actions that are necessary to an end user to create and send a new traffic violation. A critical aspect of this sequence of tasks are the steps linked to the insertion of the license plate, so we put a bit more attention on this passage.

Use case 2: See past contributions

Name	See past contributions
Actor	End User
Entry condition	An end user wants to see one or some contributions he sent to SafeStreets.
Event flow	1. In the homepage of end user, the end user clicks on "past contributions" button. 2. The application enters in past contributions page. 3. The end user sees all the traffic violations he sent. 4. The end user clicks on a traffic violation. 5. The end user sees data regarding that traffic violation.
Exit condition	The end user checked the past contributions and closes the application.
Exception	The end user has never sent a traffic violation.

Sequence diagram 2: See past contributions

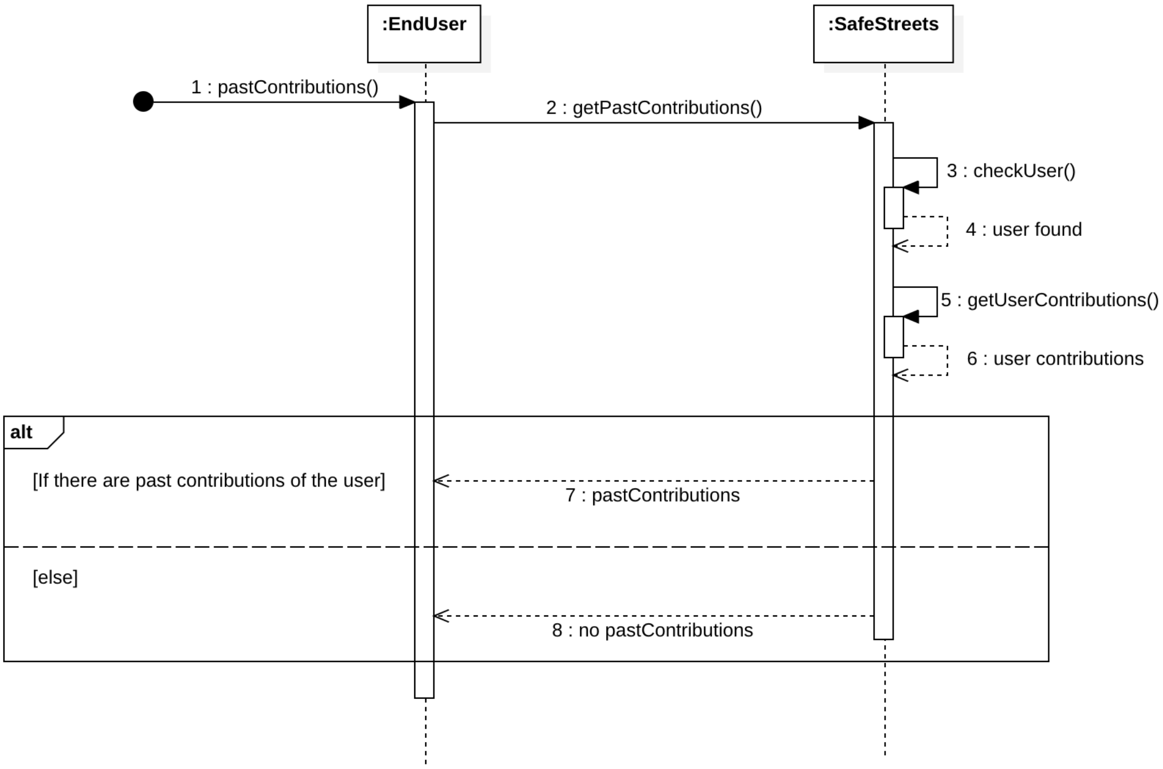


Figure 10: Sequence Diagram: See past contributions

This sequence diagram shows how an end user can see his past contributions sent to SafeStreets. We focus a bit more attention on the possibility that an end user doesn't have past activity. So in the diagram both the cases (the presence or not of past contributions) are shown.

Use case 3: Be notified for near traffic violations

Name	Be notified for near traffic violations
Actor	Authority
Entry condition	1. A traffic violation has been reported. 2. An Authority is near the traffic violation.
Event flow	1. SafeStreets takes the position of all the authorities. 2. The System checks which authorities are near the new traffic violation. 3. SafeStreets notify the Authorities of the occurrence of a traffic violation near them. 4. The Authority open the Application and goes in the "Traffic Violations" section. 5. The Authority checks the violation data reported (photo,type and description).
Exit condition	The Authority knows that a traffic violation occurred near him and he checks the violation report notified to him.
Exception	1. The Authority is not logged in the application. 2. The System is not able to detect the position of the Authority.

Sequence diagram 3: Be notified for near traffic violations

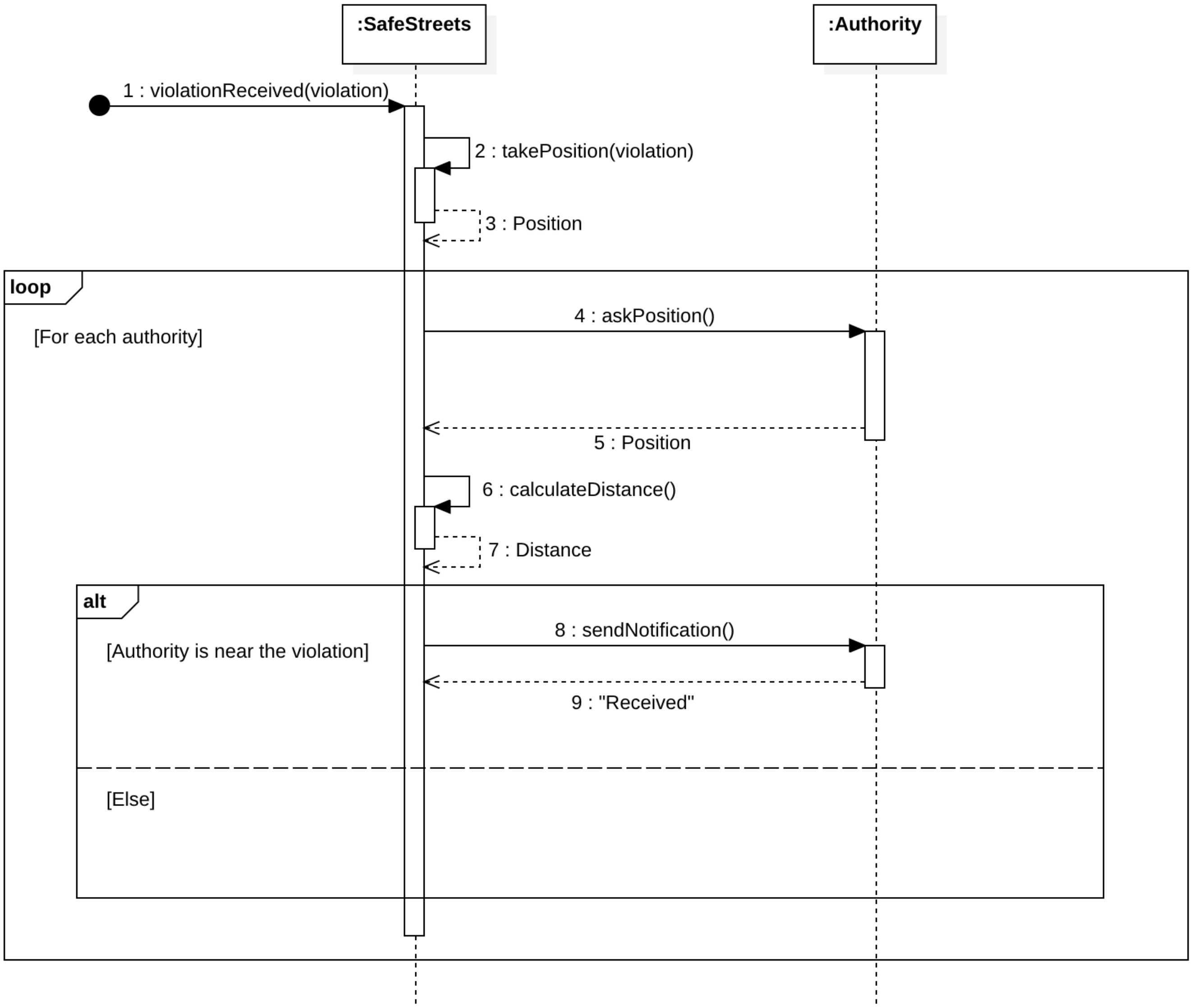


Figure 11: Sequence Diagram: Be notified for near traffic violations

Here is shown as SafeStreets decide which authorities notify when a new traffic violation is reported. For each authority, the system asks the position and confront it with the position of the traffic violation. If an authority is near the violation, the system notifies him.

Use case 4: Notify other authorities

Name	Notify other authorities
Actor	Authority
Entry condition	1. The Authority wants to go to check a traffic violation in person. 2. The Authority wants to warn other authorities that he is going to check that violation.
Event flow	1. The Authority opens SafeStreets application. 2. The Authority logs in. 3. The Authority goes in the "Traffic Violations" section. 4. The Authority finds a traffic violation which he is interested in checking directly. 5. The Authority clicks on the chosen traffic violation. 6. The Authority reads the details of the traffic violation (picture, type and description). 7. The Authority decides to go to verify if the traffic violation is true. 8. The Authority clicks on the "Warn" button which allows to warn other authorities that he is going to verify in person this traffic violation.
Exit condition	The Authorities know that a certain traffic violation is going to be checked by another Authority.
Exception	The Authority has no internet connection.

Sequence diagram 4: Notify other authorities

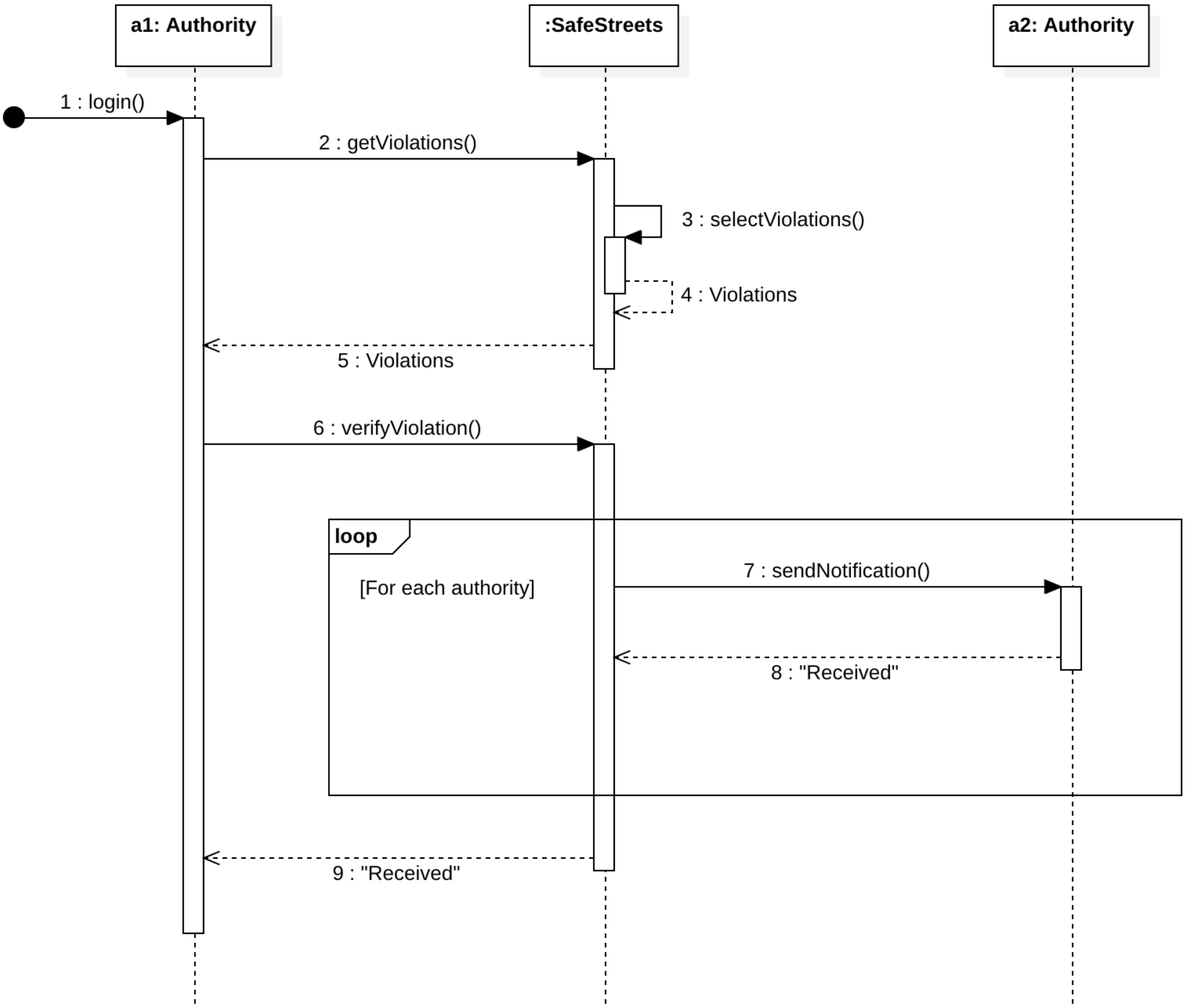


Figure 12: Sequence Diagram: Notify other authorities

In this sequence diagram, we explain the steps required for avoiding that two authorities go to verify the same traffic violation. The authority a1 sends to the system his willingness to go to verify a traffic violation. As a consequence the system notifies all the other authorities about a1's intention.

3.2.2 Statistics elaboration

Scenarios

Scenario 5

Andrea, a very sporty and competitive guy, each Sunday morning wakes up early so that he can go bike riding without any disturbance. During these morning training sessions, however, it's really common for him to brake sharply for car parked in the middle of the bike lane or to go in the middle of the road for motorized vehicles because of double parking. This week, however, his friend Samuele talked about SafeStreets and about the traffic violation statistics reporting service that this application offers. So, Andrea installed it on his device, opened the application, signed up and accessed to the statistics page. By looking to the data provided by the application he has found that there is a bike lane near Como lake in which traffic infringements are really rare. After trusting the application, he discovered he could not do without it anymore, because he was able, for the first time in a long time, to take a bike ride without annoying interruptions.

Scenario 6

Mr. Ryan is the new police commissioner of the police station of Milan. Among his duties, he has also to decide in which areas of Milan it is better to control if there are traffic violations and in particular parking violations. At the beginning, he used to maintain the same areas supervised by his predecessor, but now, with the birth of SafeStreets he understood that a more efficient work can be done. After having opened the application and logged in, Mr. Ryan accessed the traffic violation statistics area that SafeStreets provides and began to check the data deriving from the combination of all the traffic violation report done by end user. By crossing this new information with those of his predecessors he discovered that the areas kept most under control until now are almost no longer subject to this type of events, while many others considered safe are now much more popular for breaking traffic rules. So, thanks to SafeStreets application can grant to citizen a more targeted service, going to cover all the most critical places.

Scenario 7

Giovanni, an employee of the municipality of Como, has the task of understanding why every morning an endless queue is formed along "Via Castelnuovo". His municipality has recently joined SafeStreets project by installing SafeStreets software on its computers, so he opens the application, logs in with the municipality credentials and select the function that permits to see the traffic violation statistics of the territory. As soon as the system shows it to him, Giovanni sees that the highest number of traffic violations of type "No parking zone" occurs just in that street between 8:00 and 9:00 am. Once noticed this, he remembers that exactly in the position where infringements concentrates, there is the high school and understand that the principal cause of the problem is that parents, in order to bring their children to school, stop by the side of the road, hindering the passage of other cars. After becoming aware of this, he communicates the results obtained from his analysis to the mayor of his municipality, who can now decide the best solution for the problem.

Use case 5: Check traffic violation statistics

Name	Check traffic violation statistics
Actor	User
Entry condition	The User wants to know information about traffic violation statistics.
Event flow	1. In the homepage, the User clicks on the "Check Statistics" button entering in the statistics page. 2. The User select which type of statistic he wants to see. 3. The Application elaborates the request and extracts the selected type of statistics. 4. The Application shows the selected type of statistics to the User.
Exit condition	The User has checked the statistics and closes the statistics page.
Exception	There are too few traffic violation reports, so the System is not able to elaborate reliable statistics, so the Application notifies the User of it.

Sequence diagram 5: Check traffic violation statistics

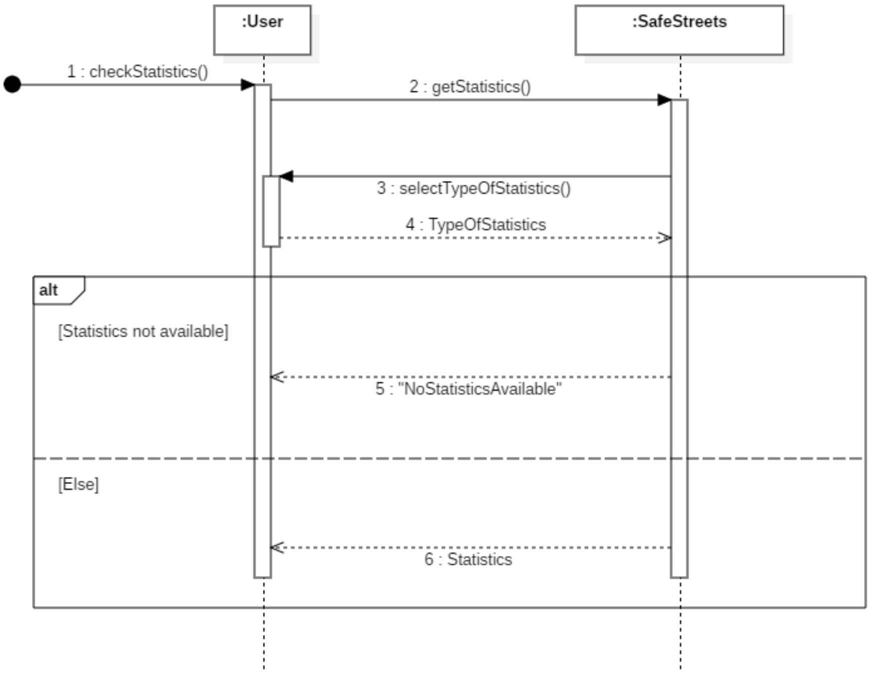


Figure 13: Sequence Diagram: Check traffic violation statistics

Each type of user can see the statistics calculated by SafeStreets about traffic violations. In this sequence diagram it is shown the needed actions to see the statistics. It is possible that there are no statistics due to a lack of data, so we present both the cases, the presence or not of statistics.

3.2.3 Unsafe area detection and intervention suggestions

Scenarios

Scenario 8

Roberto, wise mayor always careful of his citizens needs, noticed that in the last few months, traffic violations and accidents have become more frequent, growing together in their number. He immediately understands that there must be a correlation between the two, but at the same time he realises that he has only data regarding accidents. Doing some researches, he discovers SafeStreets and installs it on municipal computers. Opening the software, he registers as a municipality and through the use of the service offered by the system which allows to insert accidents data, he begins to insert these types of information. After inserting enough data, the system crosses them with information deriving from violations reported by end users, elaborates possible suggestions and sends them to Roberto. The mayor, watching at the interventions proposed, discovers that there are two unsafe areas, in fact, in a couple of streets drivers used to leave the car in double parking creating traffic jams that lead to an increase in rear-end collisions, and that to solve the problem it is enough to augment the controls on those areas.

Use case 6: Insert accidents data

Name	Insert accidents data
Actor	Municipality user
Entry condition	Municipality has data about accidents occurred in its territory and wants to insert them in the System so that SafeStreets can cross all his data.
Event flow	1. In the homepage, the Municipality User clicks on the "Insert Accidents Data" button entering in the accident creation page. 2. The System requests to the Municipality User to insert data about accidents. 3. The Municipality User groups all the accidents data in a single file. 4. The Municipality User inserts the created file into the System. 5. The Municipality User clicks on the “Send” button.
Exit condition	Data on accidents are sent to Safestreets and the Municipality User goes out of the "Insert Accidents Data" page.

Use case 7: Check suggested interventions and unsafe areas

Name	Check suggested interventions and unsafe areas
Actor	Municipality User
Entry condition	1. Municipality User wants to know the most unsafe areas and possible interventions he can apply to get it safer. 2. SafeStreets has crossed accidents information with those deriving from violation reporting. 3. SafeStreets elaborated possible interventions to make unsafe areas safer.
Event flow	1. In the homepage, the Municipality User clicks on the "Unsafe Areas and Suggestions" button entering in the interventions page. 2. The Application elaborates the request and extracts the most unsafe areas and the possible interventions. 3. SafeStreets supplication shows to the Municipality User the most unsafe areas and the possible interventions to get them safer.
Exit condition	Municipality User has checked the most unsafe areas and the possible interventions to get them safer and close the interventions page.
Exception	1. SafeStreets software is not able to read the provided accidents data and to extract information from them, so it notifies the Municipality User. 2. There are too few traffic violation reports or too few accidents data, so the System is not able to elaborate reliable suggestions, so the Application notifies the Municipality User of it.

Sequence diagram 6: Insert accidents data and check suggested interventions and unsafe areas

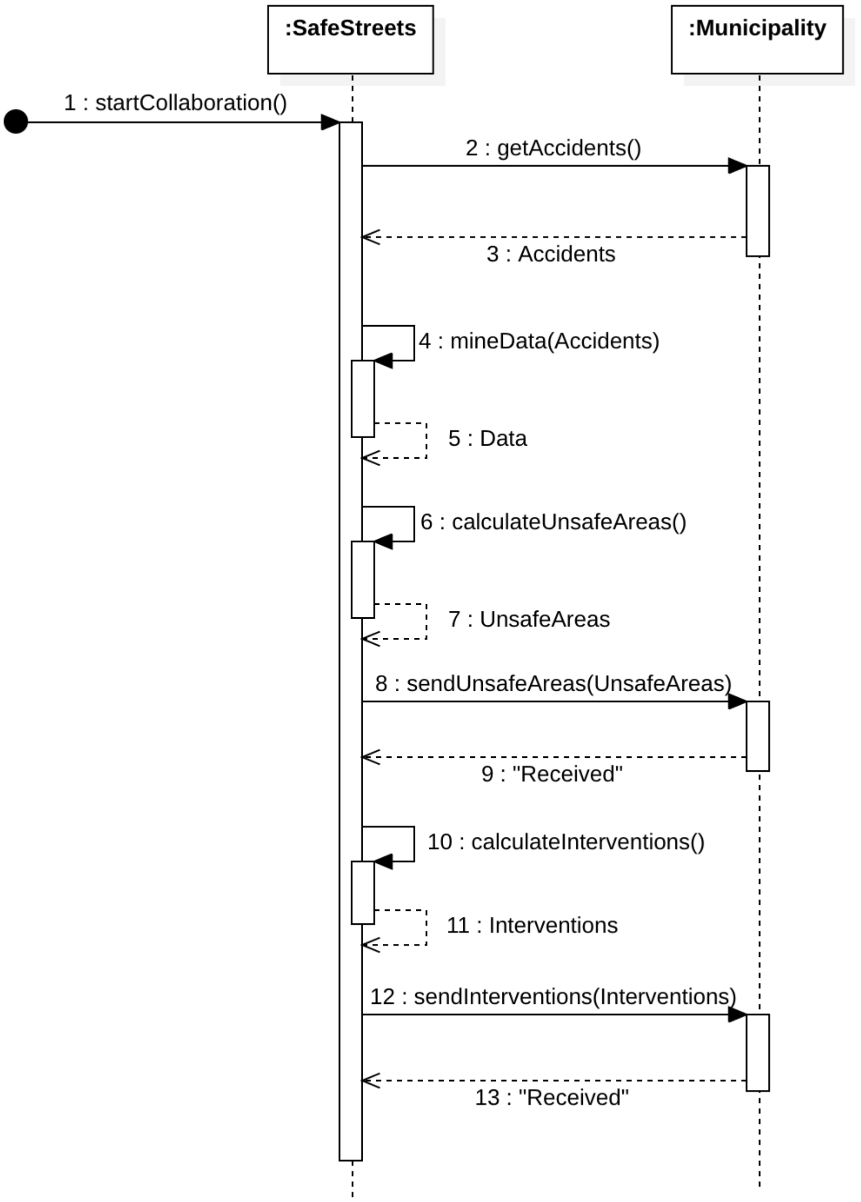


Figure 14: Sequence Diagram: Insert accidents data and check suggested interventions and unsafe areas

This last sequence diagram resumes the interaction between municipality user and the system. Municipality provides data about accidents to SafeStreets. The system calculates unsafe areas and suggested interventions and send both to municipality.

3.2.4 Requirements

The requirements elicitated from the previous analysis are now presented. In particular, in order to better show the relation between them, the goals and the domain assumptions exposed before, the goals are again listed, but this time indicating below them the requirements which aim to satisfy them and the domain assumptions that permit to those requirements to reach that specific goal.

G1 A person (end user) who sees a traffic violation should be able to notify authorities of the violation in every moment and situation (such as with lack of internet connection), participating to street regulation.

R1 When a user sees a traffic violation, SafeStreets application must allow him to take a picture of it, insert a description and immediately send the information to authorities.

R2 When an end user sends a violation report, SafeStreets application must detect automatically the date, the time and the position from the device. The position is taken from the GPS of the user's device.

R3 When detecting the date, the time and the position from the device, if it is not able to take one of these information, SafeStreets application must notify the user telling him which is the problem.

R4 When the picture of the violation inserted by the end user is not readable by the application, the system must ask him to insert it again or to write manually the license plate number.

R5 When a violation is sent, SafeStreets dispatching software must find the nearest authority users and notify them.

R6 When an end user wants to report a traffic violation and there is no internet connection, SafeStreets software must allow him to save it and send it when internet connection has been restored.

D1 A person (end user) knows the traffic rules.

D2 A person (end user) knows that he can notify the authority if there is a traffic violation.

D3 A person (end user) has a device with a camera, internet connection and GPS sensor.

D4 A user (End user, Authority or municipality) knows SafeStreets and has a device on which there is SafeStreets software and internet connection.

D5 Users are fair with each other, so they do not lie when reporting a traffic violation.

D6 Users report a violation from the position where the violation occurred.

G2 A person (end user) should be sure that reporting a violation does not put him under any kind of risk of retaliation, so no one can know the identity of who has reported the violation.

R7 When a violation is reported, SafeStreets must not show the identity of the end user that created it, so that to guarantee anonymity.

R8 When an end user or a municipality user logs in, SafeStreets must not allow him to see the traffic violations sent by the other end users.

D7 An authority or a municipality never communicate details of a traffic violation report to no one.

G3 An authority should be able to know the recent traffic violations.

R9 When an authority logs in from his device, SafeStreets must allow him to see information about the traffic violations sent by the end users.

R14 When a user registers to SafeStreets application, the system must ask him to select the role he held (end user, authority or municipality), such that it is possible to distinguish him and provide him the features associated with his role. In particular, if the user select "Authority" or "Municipality" the system must ask him also the governmental code and verify their identity through the provided interface.

R15 When a user logs in, SafeStreets must recognise him and his role (end user, authority or municipality), such that to provide him the right features.

D4 A user (End user, Authority or municipality) knows SafeStreets and has a device on which there is SafeStreets software and internet connection.

D5 Users are fair with each other, so they do not lie when reporting a traffic violation.

G4 A user should be able to know the statistics regarding traffic violations.

R10 When a user logs in from his device, SafeStreets must allow him to see statistics about the traffic violations.

R11 When a violation is reported, SafeStreets must update the traffic violation statistics and the unsafe areas information.

R16 When a user requests to see traffic violations statistics, but there are too few traffic violation reports, the System must notify him of that.

D4 A user (End user, Authority or municipality) knows SafeStreets and has a device on which there is SafeStreets software and internet connection.

D5 Users are fair with each other, so they do not lie when reporting a traffic violation.

D6 Users report a violation from the position where the violation occurred.

G5 Municipality should be able to collaborate with SafeStreets in order to provide information about accidents occurred in its territory.

R12 When a municipality user logs in, SafeStreets must allow him to provide information about accidents occurred in its territory.

R15 When a user logs in, SafeStreets must recognise him and his role (end user, authority or municipality), such that to provide him the right features.

D11 A municipality has data about accidents occurred in its jurisdiction stored and can provide them.

G6 Municipality should be able to know the most unsafe areas of its territory.

R13 When a municipality user logs in from his device, SafeStreets must allow him to see unsafe areas and possible interventions.

R15 When a user logs in, SafeStreets must recognise him and his role (end user, authority or municipality), such that to provide him the right features.

R11 When a violation is reported, SafeStreets must update the traffic violation statistics and the unsafe areas information.

R17 When a Municipality User requests to see the most unsafe areas and the possible interventions, but there are too few traffic violation data and too few accident data, the System must notify him.

R23 SafeStreets must not allow a municipality to see the unsafe areas information of other municipalities.

D4 A user (End user, Authority or municipality) knows SafeStreets and has a device on which there is SafeStreets software and internet connection.

D5 Users are fair with each other, so they do not lie when reporting a traffic violation.

D6 Users report a violation from the position where the violation occurred.

G7 Municipality should be able to know possible interventions it can do in order to improve the unsafe areas of its territory.

R13 When a municipality user logs in from his device, SafeStreets must allow him to see unsafe areas and possible interventions.

R11 When a violation is reported, SafeStreets must update the traffic violation statistics and the unsafe areas information.

R15 When a user logs in, SafeStreets must recognise him and his role (end user, authority or municipality), such that to provide him the right features.

D4 A user (End user, Authority or municipality) knows SafeStreets and has a device on which there is SafeStreets software and internet connection.

G8 An authority should be able to know when violations occur around him.

R9 When an authority logs in from his device, SafeStreets must allow him to see information about the traffic violations sent by the end users.

R18 When a violation is reported, SafeStreets must detect the position of all the authorities from their device in order to know who can be interested in knowing the occurrence of the violation.

R5 When a violation is sent, SafeStreets dispatching software must find the nearest authority users and notify them.

R15 When a user logs in, SafeStreets must recognise him and his role (end user, authority or municipality), such that to provide him the right features.

D4 A user (End user, Authority or municipality) knows SafeStreets and has a device on which there is SafeStreets software and internet connection.

D5 Users are fair with each other, so they do not lie when reporting a traffic violation.

D6 Users report a violation from the position where the violation occurred.

D8 An authority user is able to reach the position of a violations when notified.

D9 An authority has a device on which there is a GPS sensor.

G9 A user should be recognized and distinguished based on his role (end user, authority or municipality).

R14 When a user registers to SafeStreets application, the system must ask him to select the role he held (end user, authority or municipality), such that it is possible to distinguish him and provide him the features associated with his role. In particular, if the user select "Authority" or "Municipality" the system must ask him also the governmental code and verify their identity through the provided interface.

R15 When a user logs in, SafeStreets must recognise him and his role (end user, authority or municipality), such that to provide him the right features.

D10 An interface to check through a governmental code if an authority or a municipality is really such is provided by the government.

G10 An authority should be able to inform other authorities about its intentions to go to verify a traffic violation

R9 When an authority logs in from his device, SafeStreets must allow him to see information about the traffic violations sent by the end users.

R15 When a user logs in, SafeStreets must recognise him and his role (end user, authority or municipality), such that to provide him the right features.

R19 When an authority user is notified, SafeStreets software must allow him to warn other authorities that have received the same notification that he is going to check the violation so that not too many authorities deal with the same violation.

R20 When an authority checks a violation, the System must not allow other authorities to check it and authorities must no longer be able to see it as a notification (with "check a violation" it is meant that the authority expresses to the System the willingness of going verify it in person).

R21 When an authority is warned about a violation checked by another authority, the System must not allow him to check the violation again.

D4 A user (End user, Authority or municipality) knows SafeStreets and has a device on which there is SafeStreets software and internet connection.

D6 Users report a violation from the position where the violation occurred.

D8 An authority user is able to reach the position of a violations when notified.

D9 An authority has a device on which there is a GPS sensor.

G11 A person (end user) should be able to know (only) his contribution in traffic regulation

R7 When a violation is reported, SafeStreets must not show the identity of the end user that created it, so that to guarantee anonymity.

R8 When an end user or a municipality user logs in, SafeStreets must not allow him to see the traffic violations sent by the other end users.

R15 When a user logs in, SafeStreets must recognise him and his role (end user, authority or municipality), such that to provide him the right features.

R22 SafeStreets must store all the traffic violations sent by each end user, so that it can show to an end user (only) own contributions.

D3 A person (end user) has a device with a camera, internet connection and GPS sensor.

D4 A user (End user, Authority or municipality) knows SafeStreets and has a device on which there is SafeStreets software and internet connection.

3.2.5 Traceability matrix

In order to make the links between goals, domain assumptions, requirements and use cases clearer, a traceability matrix is now showed.

Row ID	Goal ID	Domain Assumptions ID	Requirements ID	UseCases ID
r1	G1	D1 ,D2, D3, D4, D5, D6	R1, R2, R3, R4, R5 R6	U1
r2	G2	D7	R7, R8	U1
r3	G3	D4, D5	R9, R14, R15	U3
r4	G4	D4, D5, D6	R10, R11, R16	U5
r5	G5	D11	R12, R15	U6
r6	G6	D4, D5, D6	R11, R13, R15, R17, R23	U7
r7	G7	D4	R11, R13, R15	U6
r8	G8	D4, D5, D6, D8, D9	R5, R9, R15, R18	U3
r9	G9	D10	R14, R15	
r10	G10	D4, D6, D8, D9	R9, R15, R19, R20, R21	U4
r11	G11	D3, D4	R7, R8, R15, R22	U2

3.3 PERFORMANCE REQUIREMENTS

The system has to be able to respond to a possibly great number of simultaneous requests and more generally to a great number of requests throughout the day. Moreover, it has also to deal with a great number of data, because of the large number of violation reports that every day can be sent and because of the task of elaborating statistics and unsafe areas information.

Considering only the city of Como for example, carrying out a brief research it is possible to understand that out of about 86,000 (eighty-six thousands) 3 inhabitants, between end users, authorities and municipality users, at least half of them have the means and the skills to be able to subscribe to the service. Moreover, by knowing that during a day, on average, in Como 1000 (one thousand) traffic violations occur, is possible that every day 1500 (one thousand and five undreds) violation reports are sent to SafeStreets (considering also those sent more than one time from different people) only from the city of Como. Looking at these data and considering that SafeStreets has to be proposed to the regional level, it is evident that the amount of information that the system has to deal with is really large.

3.4 DESIGN CONSTRAINTS

3.4.1 Standards compliance

Regarding the data that SafeStreets has to provide to the users (traffic violations statistics and unsafe areas), they are shown both with absolute values and with percentage values, so as to provide a vision that is as complete and precise as possible and at the same time readable for everyone without any specific knowledge or competence.

3.4.2 Hardware limitations

For what concerns hardware limitations, it is required that the devices used by the users have some specific functionalities. Going into more details, each type of user must have a device equipped with a network adaptor to be able to have an internet connection (wi-fi or cellular network). Then, end users and authorities are also required to have a GPS sensor installed on their devices so that SafeStreets can detect their position while reporting a violation (for the first category) and being notified (for the second one). It's not all about end user, because in fact they are needed to also have a camera to take the violation picture when reporting it.

3.4.3 Any other constraint

The system also has to give a minimum guarantee regarding the anonymity of users reporting violations. The reason is that end users who made the violation report can be exposed to the risk of retaliation by those who have been signalised to the authorities.

3.5 SOFTWARE SYSTEM ATTRIBUTES

3.5.1 Reliability

The system does not require strict reliability guarantees, but, despite this, a duplication of the central server could be considered, also using some techniques like the FloodSet algorithm, in this way providing a service more resistant to failures.

3.5.2 Availability

According to what said with reference to reliability, the system does not have strict constraints regarding availability, however, in order to provide a service that guarantees good quality and good user experience it is necessary that it has an availability that does not fall below 99%.

3.5.3 Security

User data are stored in a secure way. Anonymity is guaranteed. End users and municipalities users cannot see traffic violations published by other users. The license plate are not shared. Authority position is private, none can know it. Accidents from municipality are not visible from other users and other municipalities. An unsafe area is visible only by the municipality user that has that are under its jurisdiction. Authority and municipality must demonstrate their role.

3.5.4 Maintainability

We use design patterns for having a better maintainability. In the future the application will be easy and cheap to modify and fix.

3.5.5 Portability

The application runs on smartphone, tablet and computer, with different operative systems, different measures and different characteristics. So it has to be implemented in a way that respects the portability.

4 FORMAL ANALYSIS USING ALLOY

4.1 SIGNATURES

The Alloy model starts by declaring all the signatures. The attention is focused on the three types of users. First of all, there is an abstract signature "User". Then, this signature is extended by three other signatures representing each user type (EndUser, Authority and Municipality). Other signatures are Violation, ViolationData, UnsafeArea, Intervention and Position. Violation represents an infringement that occurs in the real world, that is seen by the end user. ViolationData instead refers to a traffic violation report that is created and sent by the end user. An authority can see all the violations sent, he can be notified for a traffic violation that occurs in his assigned area, he can go to check a violation and he can warn other authorities about his intentions to verify that. In particular, it must be specified that, in order to calculate the authorities who are close to a reported violation, the area assigned to each authority (it is modelled as a set of positions) is considered. In fact, if the traffic violation occurs in a position that is inside the assigned area of an authority, that authority will be notified. A municipality user has a set of unsafe areas under his jurisdiction and each unsafe area is composed by a set of positions and can be linked to an intervention.

```
abstract sig User {}

sig EndUser extends User {
violationsSent: set ViolationData,
violationsSeen: set Violation
}

sig Authority extends User {
violations: set ViolationData,
notifications: set ViolationData,
checked: set ViolationData,
warned: set ViolationData,
assignedArea: set Position
}

sig Municipality extends User{
unsafeareas: set UnsafeArea
}

sig ViolationData {
violation: one Violation,
position: one Position
}

sig Violation {
position: Position
}

sig UnsafeArea{
area: set Position,
intervention: lone Intervention
}
```

```
sig Intervention{}

sig Position {}
```

4.2 FACTS

After the signatures, the facts are presented. Fifteen facts have been found and they are necessary in order to build a correct Alloy model.

```
--FACTS
--1. a checked violation cannot be in notifications
fact {
all a: Authority |
 no vd: ViolationData |
 vd in a.notifications and vd in a.checked
}

--2. a checked violation cannot be in checked by other authorities
fact {
all vd: ViolationData |
no disj a1, a2: Authority |
  vd in a1.checked and vd in a2.checked
}

--3. all ViolationData must be in all authority.violations
fact {
all a: Authority |
  ViolationData in a.violations
}

--4. all ViolationData in authority.checked cannot be also in authority.warned and
--viceversa
fact {
all a: Authority |
 no vd: ViolationData |
 vd in a.warned and vd in a.checked
}

--5. all ViolationData in authority.warned cannot be in authority.notifications
fact {
all a: Authority |
 no vd: ViolationData |
  vd in a.notifications and vd in a.warned
}

--6. all ViolationData must be in one user.violationsSent
fact {
all vd: ViolationData |
one eu: EndUser |
vd in eu.violationsSent
}

--7. two different end users cannot have the same violationdata in endUser.vionationssent
fact {
no disj eu1, eu2: EndUser |
(eu1.violationsSent & eu2.violationsSent) ≠none
}

--8. all the violationdata sent by the end user must be seen by it (and sent
--from the position the violation occurred)
fact {
all vd: ViolationData, eu: EndUser |
(vd in eu.violationsSent) iff (vd.violation in eu.violationsSeen)
}
```

```
--9. all violationdata must be in authority.notifications only if violationData.position
--in a.assignedarea
fact {
all a: Authority, vd: ViolationData |
(vd in a.notifications) implies (vd.position in a.assignedArea)
}

--10. all violationdata.position must be equal to violationdata.violation.position
fact {
all vd: ViolationData, v: Violation |
(vd.violation =v) iff (vd.position =v.position)
}

--11. two different municipalities cannot have the same unsafe area
fact{
no disj mun1, mun2: Municipality |
(mun1.unsafeareas & mun2.unsafeareas) ≠none
}

--12. unsafe area must be associated to a municipality
fact{
all ua: UnsafeArea |
some mun: Municipality |
ua in mun.unsafeareas
}

--13. all interventions must be in at least one unsafe area
fact{
all inte: Intervention |
some ua: UnsafeArea |
inte in ua.intervention
}

--14. two different unsafe areas cannot have the same area
fact{
no disj ua1, ua2: UnsafeArea |
(ua1.area & ua2.area) ≠none
}

--15. all unsafe areas must have at least one position
fact{
all ua: UnsafeArea |some pos: Position |
pos in ua.area
}
```

4.3 PREDICATES

In this section, the predicates are listed. The predicates are conditions that can happen in the world that has been described. Below each predicate there is the "run" instruction executed in order to verify that an instance is found and so to be sure that the model is consistent.

```
--PREDICATES
--1. an end user sees a new violation and sends it
pred seeViolation [v: Violation, vd: ViolationData, eu, eu’: EndUser] {
--preconditions
vd not in eu.violationsSent
v not in eu.violationsSeen
--postconditions
eu’.violationsSeen =eu.violationsSeen + v
eu’.violationsSent =eu.violationsSent + vd
}

--run seeViolation for 5 but exactly 3 EndUser, exactly 1 UnsafeArea, exactly 1 Authority,
     ↪ exactly 3 ViolationData

--2. an authority checks a violationdata
pred checkViolation [a, a’: Authority, vd: ViolationData] {
--preconditions
vd not in a.checked
vd in a.notifications
--postconditions
a’.checked =a.checked + vd
a’.notifications =a.notifications - vd
}

--run checkViolation for 3 but exactly 2 Authority, exactly 3 ViolationData, exactly 1
     ↪ EndUser

--3. authority notified
pred notifyAuthority [a, a’: Authority, vd: ViolationData] {
--preconditions
vd.position in a.assignedArea
--postconditions
a’.notifications =a.notifications + vd
}

--run notifyAuthority for 4 but exactly 2 EndUser, 3 Violation

--4. a new unsafearea is detected
pred newUnsafeArea [mun, mun’: Municipality, ua: UnsafeArea]{
--preconditions
ua not in mun.unsafeareas
--postconditions
mun’.unsafeareas =mun.unsafeareas + ua
}

--run newUnsafeArea for 4 but exactly 4 UnsafeArea
```

4.4 ASSERTIONS

Finally, the assertions are presented. Predicates shown above have been used to build assertions. Each assertion is checked, and the "check" instruction executed is written at the end of each assertion.

```
--ASSERTIONS
--1. if a end user sees and send a violation, the violation must be
--viewable by all the authorities
assert viewSentViolation {
all a: Authority, v: Violation, vd: ViolationData, eu, eu’: EndUser |
seeViolation [v, vd, eu, eu’] implies vd in a.violations
}
--check viewSentViolation for 3

--2. if an end user send a new violation data, another end user cannot see it
assert seeAnotherUserSentViolation {
all vd: ViolationData, v: Violation |
no eu1, eu1’, eu2: EndUser |
(seeViolation [v, vd, eu1, eu1’] and eu1 ≠eu2 and eu1’ ≠eu2 and
(eu1’.violationsSent & eu2.violationsSent ≠none))
}
--check seeAnotherUserSentViolation for 10

--3. if an authority checks a violation data another authority cannot check the
-- same violationdata
assert doubleCheck {
all vd: ViolationData |
no a1, a1’, a2: Authority |
 (a1 ≠a2 and a1’ ≠a2 and checkViolation [a1, a1’, vd] and
      (a1’.checked & a2.checked ≠none))
}
--check doubleCheck for 10

--4. if a violation data is notified to an authority it must be notified to all the
-- other authorities assigned to an area which comprehend the position of the
-- violation
assert allInterestedAuthoritiesNotified {
all a1, a1’, a2: Authority, vd: ViolationData |
 (notifyAuthority [a1, a1’, vd] and a1 ≠a2 and a1’ ≠a2 and
  vd.position in a2.assignedArea) implies (vd in a2.notifications)
}
--check allInterestedAuthoritiesNotified for 3

--5. if a new UnsafeArea is inserted, it is associated to only one municipality
assert onlyOneMunicipalityAssociated{
all mun, mun1, mun2: Municipality, ua: UnsafeArea |
(newUnsafeArea [mun, mun1, ua] and mun1≠mun2) implies
(ua in mun1.unsafeareas and ua not in mun.unsafeareas and ua not in mun2.unsafeareas)
}
--check onlyOneMunicipalityAssociated for 5
```

4.5 RESULTS

Here are shown the worlds generated by executing the previously presented predicates.

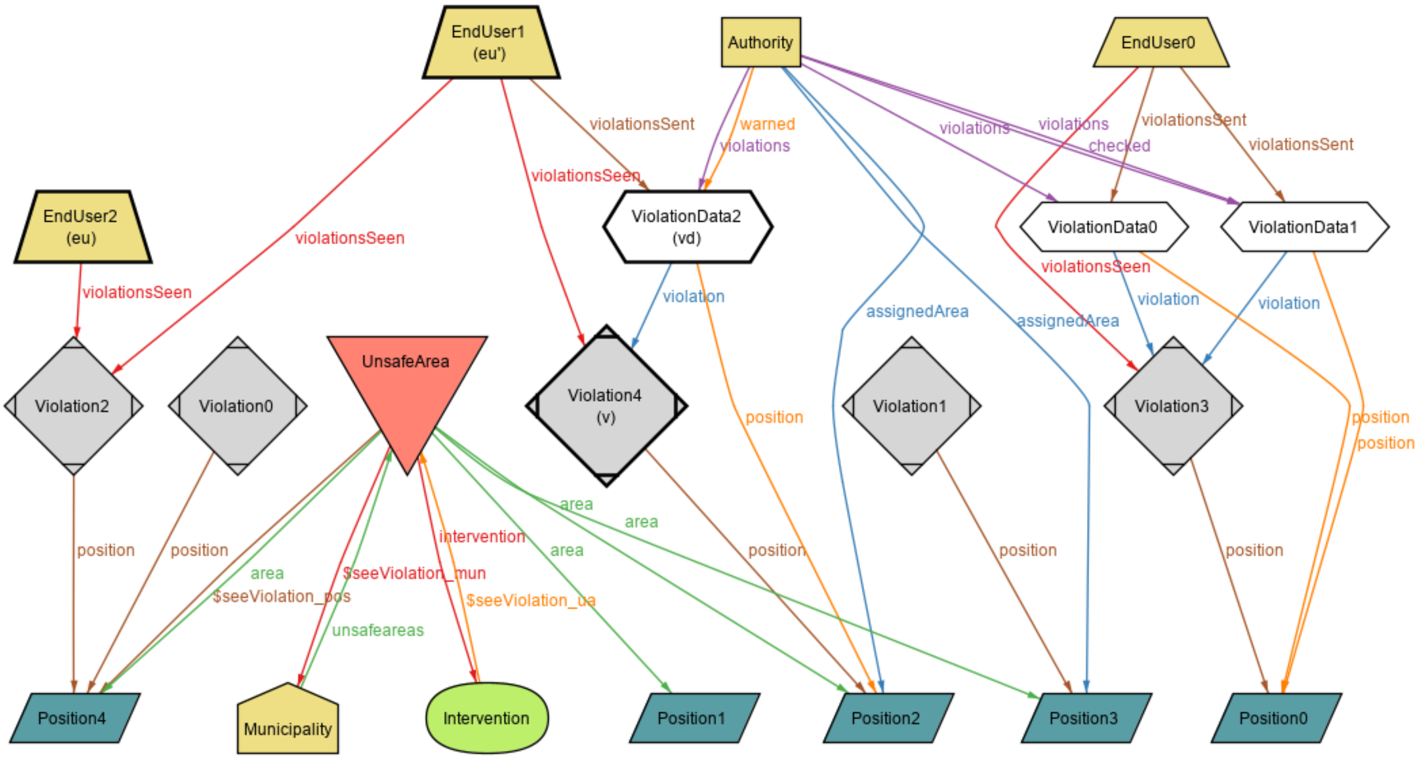


Figure 15: Predicate 1: seeViolation

The first world shown is obtained by executing the instruction "run seeViolation for 5 but exactly 3 EndUser, exactly 1 UnsafeArea, exactly 1 Authority,exactly 3 ViolationData". The aim is to formally model the situation in which an end user sees a violation and as a consequence of this sends the related report to authorities.

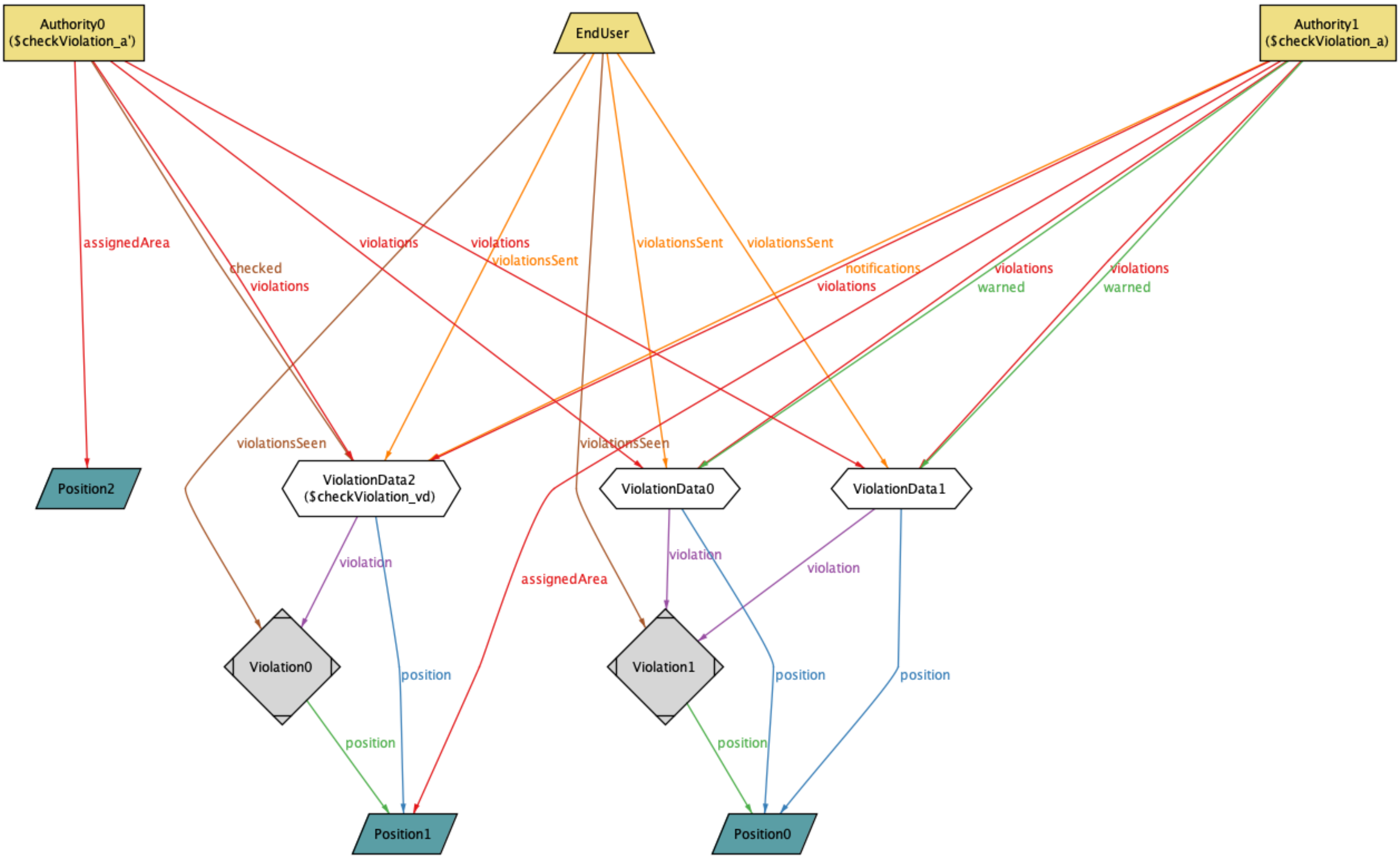


Figure 16: Predicate 2: checkViolation

The second world shown is obtained executing the instruction "run checkViolation for 3 but exactly 2 Authority, exactly 3 ViolationData, exactly 1 EndUser". The aim is to formally model the situation in which a violation that has been received by an authority as a notification (because it has occurred in his territory) can be checked by that authority.

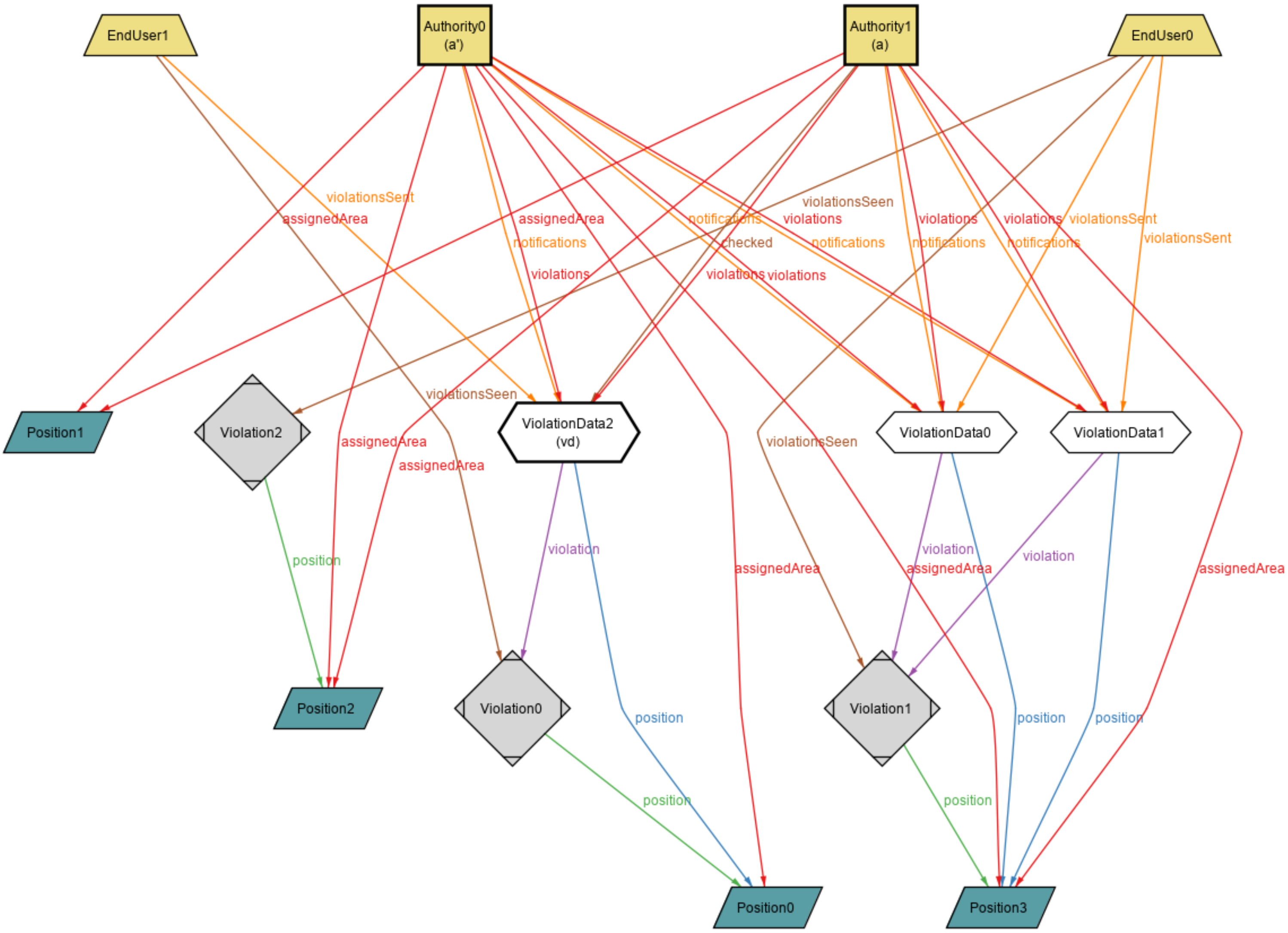


Figure 17: Predicate 3: notifyAuthority

The third world shown is obtained executing the instruction "run notifyAuthority for 4 but exactly 2 EndUser, 3 Violation". The aim is to formally model the situation in which a violation that has been occurred in a certain position and that has been reported by an end user, is notified to all the authorities to whom that position is assigned.

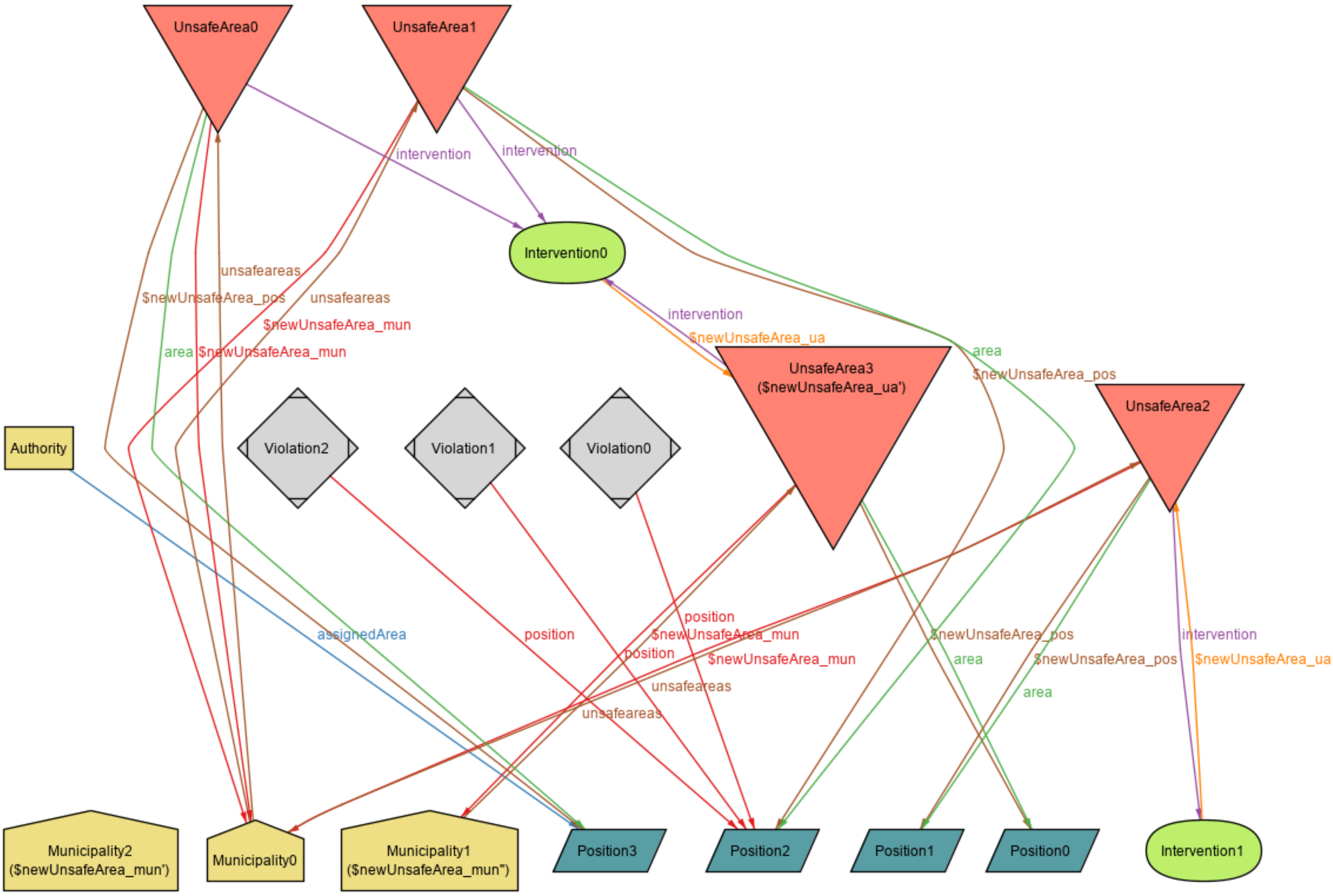
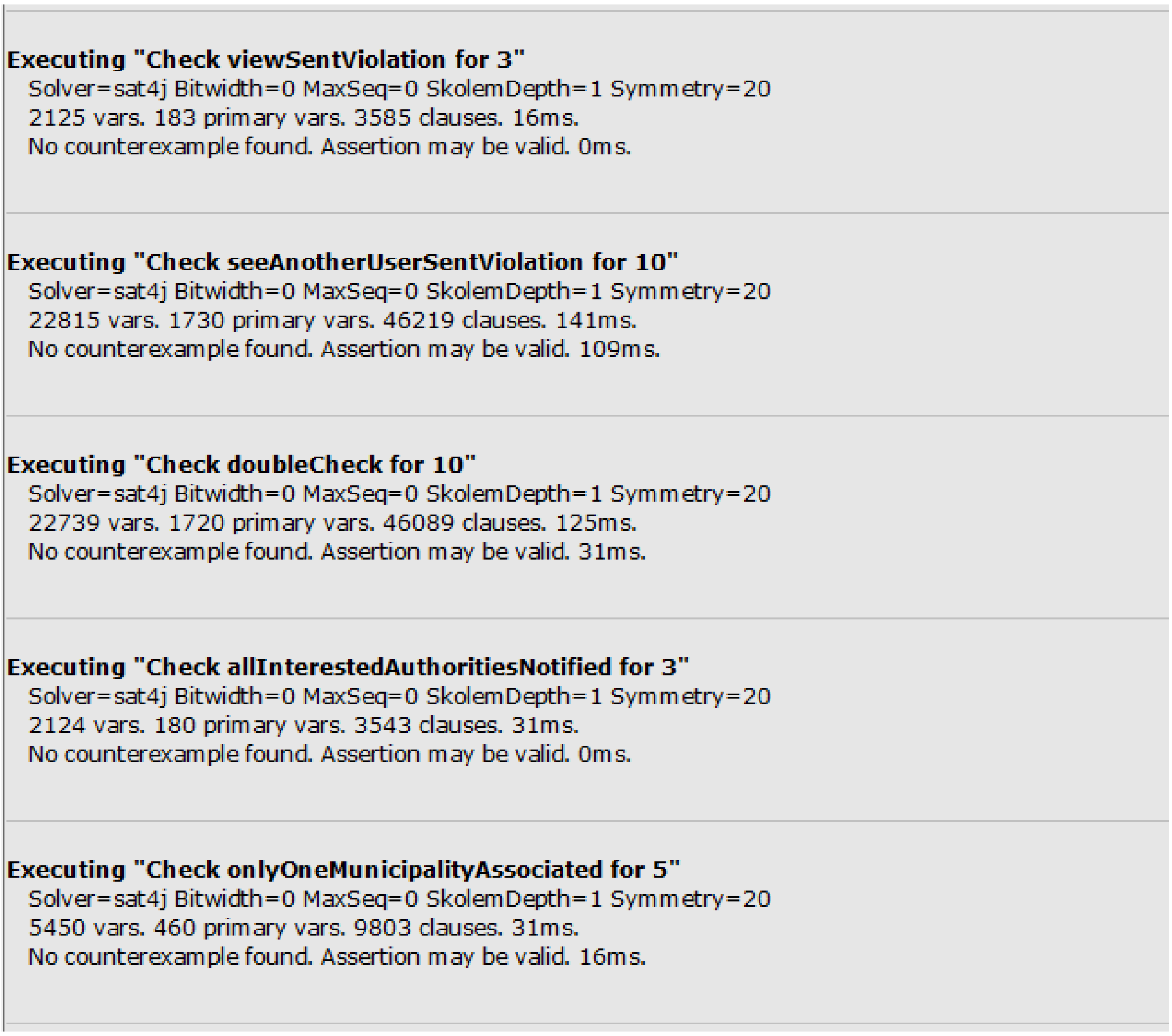


Figure 18: Predicate 4: newUnsafeArea

The third world shown is obtained executing the instruction "run newUnsafeArea for 4 but exactly 4 UnsafeArea". The aim is to formally model the situation in which a new unsafe area is calculated and generated by the system and then assigned to the municipality to which that territory belongs.



```
Executing "Check viewSentViolation for 3"
   Solver=sat4j Bitwidth=0 MaxSeq=0 SkolemDepth=1 Symmetry=20
   2125 vars. 183 primary vars. 3585 clauses. 16ms.
   No counterexample found. Assertion may be valid. 0ms.

Executing "Check seeAnotherUserSentViolation for 10"
   Solver=sat4j Bitwidth=0 MaxSeq=0 SkolemDepth=1 Symmetry=20
   22815 vars. 1730 primary vars. 46219 clauses. 141ms.
   No counterexample found. Assertion may be valid. 109ms.

Executing "Check doubleCheck for 10"
   Solver=sat4j Bitwidth=0 MaxSeq=0 SkolemDepth=1 Symmetry=20
   22739 vars. 1720 primary vars. 46089 clauses. 125ms.
   No counterexample found. Assertion may be valid. 31ms.

Executing "Check allInterestedAuthoritiesNotified for 3"
   Solver=sat4j Bitwidth=0 MaxSeq=0 SkolemDepth=1 Symmetry=20
   2124 vars. 180 primary vars. 3543 clauses. 31ms.
   No counterexample found. Assertion may be valid. 0ms.

Executing "Check onlyOneMunicipalityAssociated for 5"
   Solver=sat4j Bitwidth=0 MaxSeq=0 SkolemDepth=1 Symmetry=20
   5450 vars. 460 primary vars. 9803 clauses. 31ms.
   No counterexample found. Assertion may be valid. 16ms.
```

Figure 19: Check Assertions

This picture demonstrates the correctness of the assertions modelled in the alloy model by showing that checking them no counterexamples are found.

5 EFFORT SPENT

5.1 SAMUELE MOSCATELLI

- 10/10/2019: 1h
- 12/10/2019: 3h
- 18/10/2019: 2h 30min
- 19/10/2019: 7h
- 20/10/2019: 6h 30min
- 21/10/2019: 30min
- 22/10/2019: 2h
- 24/10/2019: 3h
- 25/10/2019: 3h 15min
- 26/10/2019: 10h
- 27/10/2019: 2h
- 28/10/2019: 30min
- 29/10/2019: 2h
- 30/10/2019: 2h
- 09/11/2019: 3h
- 10/11/2019: 1h 30min

Total: 49h 45min

5.2 ANDREA POZZOLI

- 11/10/2019: 2h
- 12/10/2019: 3h
- 18/10/2019: 2h 30min
- 19/10/2019: 5h
- 20/10/2019: 4h 30min
- 22/10/2019: 5h

- 30/10/2019: 3h
- 31/10/2019: 4h
- 01/11/2019: 4h
- 02/11/2019: 4h
- 05/11/2019: 2h
- 06/11/2019: 2h 30min
- 08/11/2019: 2h 30min
- 09/11/2019: 3h
- 10/11/2019: 2h 15min

Total: 49h 15min

6 REFERENCES

1. For UML diagrams we used StarUML software.
2. For the mockups we used mockflow.com.
3. The number of inhabitant of the city of Como is taken from https://it.wikipedia.org/wiki/Como.
4. For the coordination and organization of the work we used a repository of github.com.