A.A. 2019/2020

POLITECNICO
MILANO 1863

Computer Science and Engineering

Software Engineering 2

RASD

Requirement Analysis and Specification Document

version 1.1

Lorenzo Amaranto
Stefano De Cillis
Tayfun Gumus

1. Introduction

1.1 Purpose

1.1.1 General purpose

SafeStreets is a crowdsourced application where citizens can report traffic violations to authorities along with pictures and relative informations about the infractions. After their validation, if it's requested, the system provides the possibility to analyze the collected data, based on the role of the final user.

In addiction, SafeStreets can cross its informations with the ones provided by the municipality and suggest reasonable interventions for unsafe areas. Then it can also supply public authorities, such as the local police, with data in order to generate traffic tickets from them. Finally it elaborates statistics on different layers showing the effectiveness of SafeStreets and its impact on the community.

1.1.2 Goals

Here it follows a list of the identified goals of the S2B:

- **G1**: Allow citizens to report traffic violation with relative informations.
- **G2**: Allow both end users to mine data with different levels of data visibility.
- **G3**: Provide municipality with suggestions for unsafe areas.
- **G4**: Supply public entities with intelligence to generate traffic ticket.
- **G5**: Provide municipality with TI-Statistics in order to evaluate the effectiveness of the service.

1.2 Scope

The idea of SafeStreets is to improve the condition of the general mobility in the city exploiting a virtuous relationship of collaboration between citizens reporting violations and public authorities. The S2B has been thought to guarantee the privacy of citizens who report violations and to help municipality to check the city through a distributed network of reports.

Citizens will have the possibility to report traffic infractions providing pictures and informations about the violation and its type. In addiction, an optional field will be the license plate input in order to help the system recognize it. The system will evaluate either to accept or to refuse reports, that must be compliant to minimum acceptance requirements. Then,Safestreets will validate reports. People can also question the system in order to retrieve various kind of informations, like trends, based on the type of violation and other selectable parameters. Instead more informations can be mined by public authorities that will be also able to visualize the rankings about vehicles which committed more infractions and citizens who reported more violations.

SafeStreets shares a private interface to public authorities with which it can retrieve data and cross them with its own informations. The purpose it is to suggest reasonable solutions to each identified problem, such as adding barriers between streets and bike lanes or adding stakes on the sidewalk due to an aggressive parking, suggesting more checks in areas where there are double parking or car parked in reserved places for people with disabilities.

All the data collected by the system will be processed and made available for public authorities. In this way, the Local Police will use refined data in order to generate traffic tickets. Public entities can evaluate the influence of the system on the community with the statistics SafeStreets builds, retrieving indexes and trends, by using the feedbacks received by authorities.

1.3 Definitions, Acronyms and Abbreviations

1.3.1 Definitions

- **User**: generic end user who interacts with the application.
- **Citizen (or Reporter)**: users who reports the infraction.
- **Public Authority (or Entity)**: public institutions,such as Municipality, City Hall, Local Police.
- **Infraction**: a violation or infringement of a parking concerned law. They include:
    1. Double Parking (a).
    2. Aggressive parking on the sidewalk (b).
    3. Parking on the bike lane (c).
    4. Parking on reserved area (d).
    5. General Parking Violation (e).
- **Possible Solution**: answer that the system gives back to improve the condition of an area classified as unsafe. They include:
    1. Increase checking (a,e)
    2. Stakes on the sidewalk (b).
    3. Add barriers between streets and bike line (c).
    4. S.F. park solutions designed for people with disabilities (see Reference Documents) (d).
- **Unsafe area**: cut of the city with at least 100 accidents and 250 validated report per year.
- **System,Platform**: SafeStreets.
- **Statistics**: elaborated report informations.
    1. For citizens, they include:
        A. Sort of the streets according to the number of violations associated (abs value).
        B. Sort of the streets according to the number of associated violations/per type of violation.
        C. Trend of the reported violations during the day.
    2. For Public Authorities, they include:
        A. All the Statistics available for citizens.
        B. Vehicles that have more than thirty validated reports.
        C. Ranking of citizens that reported violations.

- **TI-Statistics:** information extracted analyzing feedback. They include:
    1. Ratio between number of reports that comes to the police over the number of reports that arrive to Safestreet (if it's bigger than 1, Safestreets is a better interlocutors with respect to the police).
    2. Ratio between the number of issued tickets over the number of validated report that Safestreets has registered(it's a measure of the effectiveness of the police action).
    3. Monthly based trend of the number of issued tickets.
- **Feedback**: daily record sent by the municipality concerning the total amount of traffic tickets issued.
- **Report**: a formal account of events given by a witness that any citizens can provide to the system with a list of pictures and a filled form with informations about the infraction.
- **Accepted Report**: state of a report that is compliant with the acceptance parameters.
- **Acceptance parameters**:
    1. Data and time must not be in the future.
    2. The licence plate must be inserted with the right number of characters and in the right position.
    3. There is no duplicate of the image in the database.
    4. In the database there is not a report inserted with FC, license plate, data and location all of them equal to the candidate report.
- **Acceptance procedure**: steps the S2B will perform to establish if the report fits the acceptance parameters. It is a NSC to discard any doubt about the usability of that specific report in the validation procedure.
- **Valid Report**: entity created based on accepted reports that have accomplished the validation process.
- **Validation Process**: two reports of the same violations are needed in order that violation valid. If the second report does not come within 30 days, the report is not valid.
- **Mobile Friendly**: a smart interface that makes the access to the services provided by the system easier.
- **Limited/Italian local reality**: any municipality in Italy with at least 100.000 residents.
- **Reasonable threshold:** two reports of the same violation have been accepted.
- **Intranet**: private network of computers.
- **Adult**: an eighteen years old individual or older.

1.3.2 Acronyms

- **S2B** = Software To Be.
- **API** = Application Programming Interface.

- **UI** = User Interface.
- **DD** = Design Document.
- **GDPR** = General Data Protection Regulation.
- **PA** = Public authority.
- **GMC =** General mobility conditions.
- **MITM** = Man In The Middle.
- **FC** = Fiscal Code.
- **NSC** = Necessary and Sufficient Condition.

1.3.3 Abbreviations

- **Gn** = nth goal
- **Dn** = nth domain assumption
- **Rn** = nth requirement

1.4 Reference Documents

- Specification document: “SafeStreets Mandatory Project Assignment”
- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.
- UML diagrams: https://www.uml-diagrams.org/
- Alloy doc: http://alloy.lcs.mit.edu/alloy/documentation/quickguide/seq.html
- GDPR: https://eur-lex.europa.eu/legal-content/IT/TXT/HTML/?uri=CELEX:32016R0679#d1e1807-1-1.
- Description of SF park system : http://sfpark.org/how-it-works/the-meters/.

1.5 Document Structure

The R.A.S.D. is composed by five sections, each one presented below:

- **Chapter 1**: This section is aimed to give a general introduction of the S2B, presenting the identified goals the system has to reach and a deeper description of the analysis of the world and the shared phenomena.

- **Chapter 2**: This section of the RASD should describe the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3 of the RASD, and makes them easier to understand.

- **Chapter 3**: This section of the SRS should contain all of the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements.

Throughout this section, every stated requirement should be externally perceivable by users, operators, or other external systems. These requirements should include at a minimum a description of every input (stimulus) into the system, every output (response) from the system, and all functions performed by the system in response to an input or in support of an output.

- **Chapter 4**: This section includes the alloy model and the discussion of its purpose. Also, some worlds generated running it are shown.

- **Chapter 5**: In this section are provided informations about the total amount of hours each member of the group spent working at the document.

1.6 Revision history

Version	Date	Description
1.0	07-11-2019	First RASD release
1.1	04-12-2019	Made clear the assumption about external interfaces in *Specific Requirements*

2. Overall description

2.1 Product perspective

In order to provide an idea about the relationships between different entities in the project, a class diagram is provided. It is the conceptual model for the application domain in which the software to be will be developed.

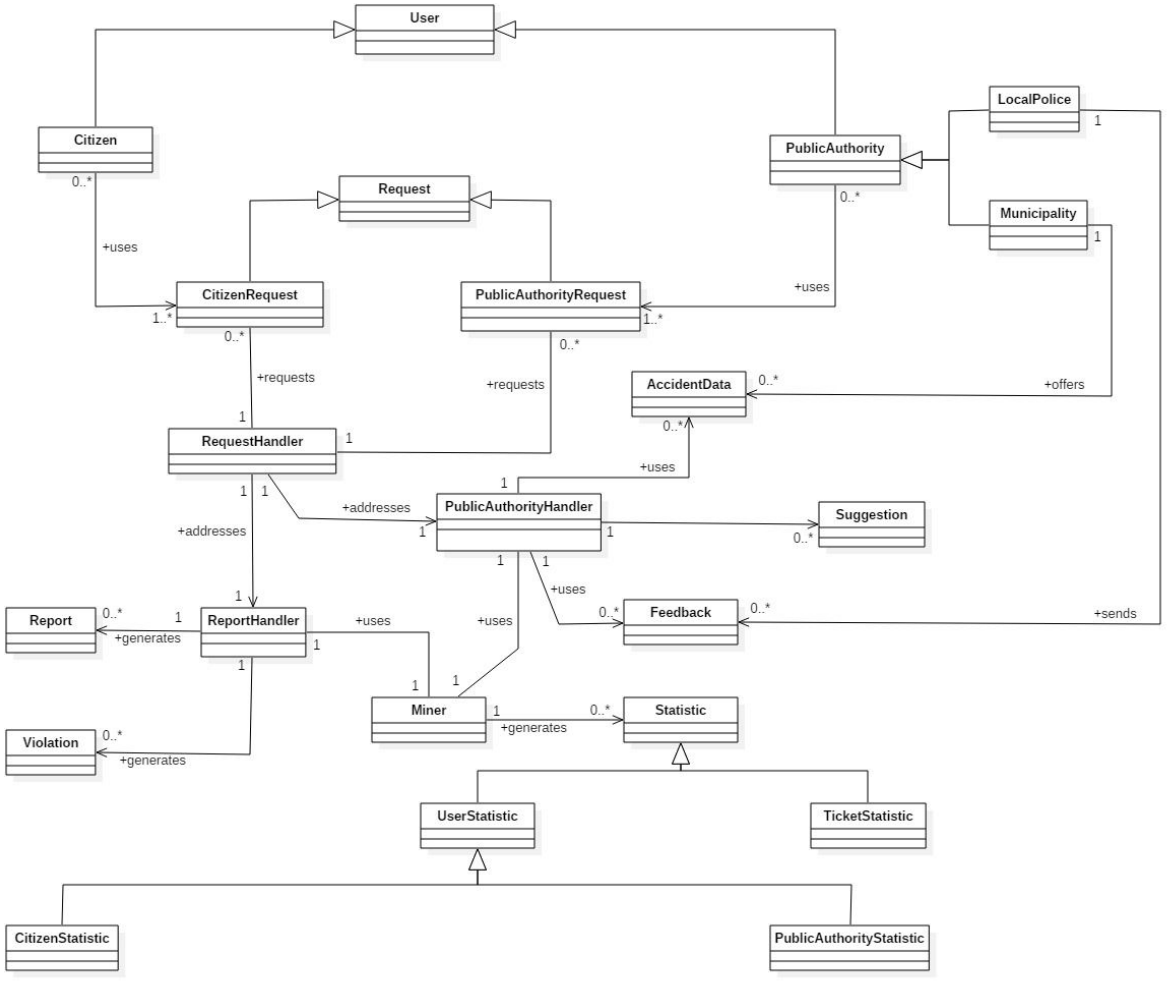


Fig. 1 S2B Structure

In order to better understand the structure, let's take in account the two sides of the diagram in Fig.1.

As mentioned above in this document, we have two different actors who interact in different way with the system: *Citizen* and *Public Authority*.

From the perspective of SafeStreets, citizens are viewed as source of informations that will be collected and mined. Even if citizens can retrieve some knowledge, the system aims to provide intelligence to PAs crossing its own informations with theirs. In this way PAs gets support with which they can improve the GMC.

It’s also interesting to notice how the dataflow is harvested and reorganized in a way that they won’t be used just for consulting actions but to provide an active support to PAs, such as generating useful data for local police in order to produce traffic tickets, too.

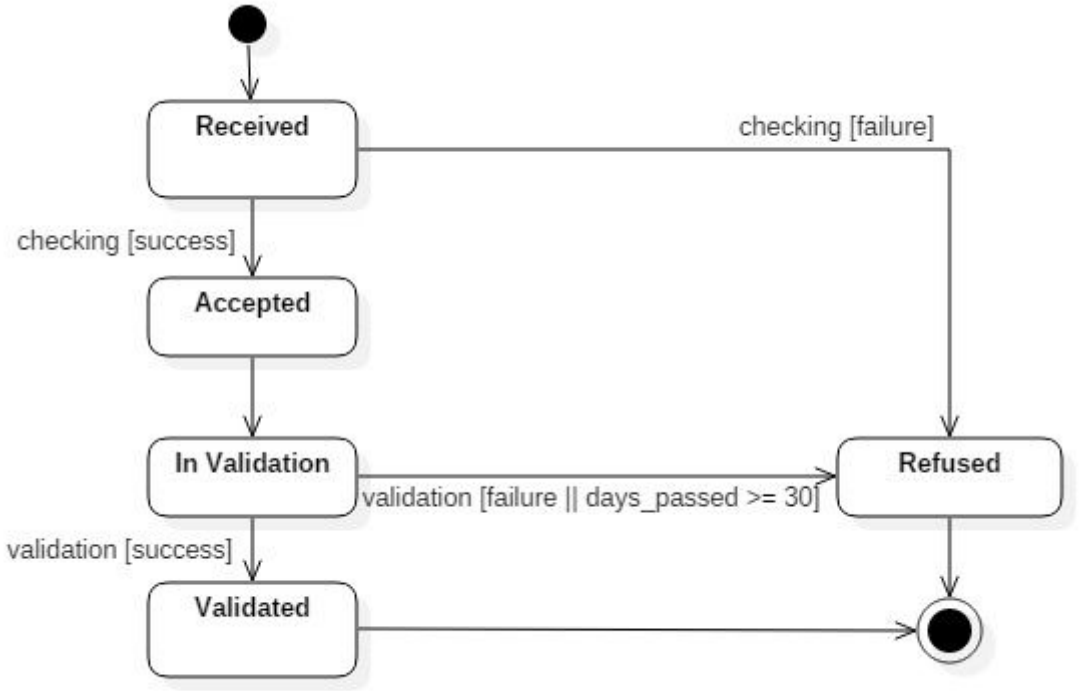


Fig. 2 Report Status

Fig.2 is used to highlight the procedure for processing a report uploaded by a citizen, by the analysis of the state of the report itself in the system. It is important to notice the fact that in order to be accepted, the report has to be examined and it has to pass a sequence of checks, this makes it meaningless the domain assumption of the user always providing correct inputs, and also improves the robustness of the S2B too.
After being accepted, it goes in a validation procedure that lasts at most thirty days in which another report of the same violation must occurs, otherwise it is refused. Finally the report is in the state in which it is considered valid, and thus usable for mining and TI-statistics. This way a violation has been properly confirmed. All of this is because, as the information that we produce from reports affect the community, it is needed a reasonable threshold to be set in order to build a reliable base for future investigation.

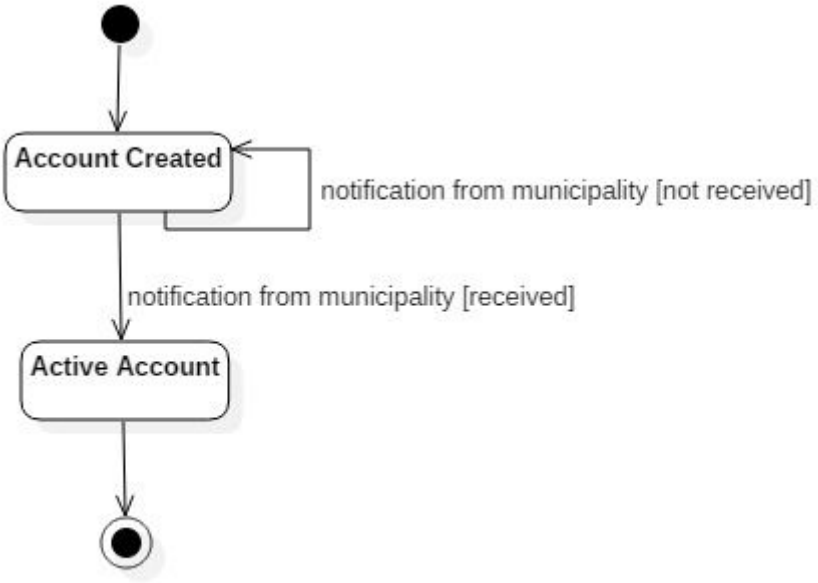


Fig. 3 Citizen Account Creation Status

The state diagram shown in Figure 3 helps understanding the peculiar conditions of the citizen's account. In particular, an account for each adult resident in the city is supposed to be created by the system. Whenever he/she decides to activate it, the municipality gives him/her a newly generated password, and notify the request to the S2B, that proceeds to activate the account and make it available for the citizen. Clearly the notification must include the transmission of the newly generated password too. This diagram also helps to visualize that the citizen account, once activated, lingers in that situation, without the possibility to be disabled.

2.2 Product Functions

The three main functionalities of the S2B are strictly correlated due to an exchange of information they perform. That is why they have been not considered separately. Still, from the product functions section, the division is needed in order to understand correctly what each part of the system should do.
Now, taking into account the identified goals, it is possible to go through the functionalities of the whole S2B.

2.2.1 Basic Service

1. It must allow access to authorities;
2. It must allow citizens to activate their accounts;
3. It must be able to handle the creation of accounts for citizens;
4. It must allow citizens to upload of violations and relative data;
5. It must allow citizens to upload the license plate of the vehicle committing the violation;
6. It must notify the results of the acceptance procedure;

7. It must allow the mining of the data to citizens;
8. It must allow the mining of the data to public authorities;

2.2.1 Advanced Function 1

1. It must be able to cross the collected data with the one the municipality has about accidents that occurs in its territory;
2. It must be able to produce solution to improve the situation of areas classified as not safe;

2.2.2 Advanced Function 2

1. It must grant access to authorities to a traffic tickets generation support service;
2. It produces periodically updated TI-statistics;

2.3 User Characteristics

For this project, it's foreseen to have two different types of users:

- Citizens:
    - People who can report violations on general mobility in the city or mine informations that have been received by the system, according to the privacy policies already mentioned.
- Public Authorities:
    - Municipality:
        - It's a public entity that can retrieve data from SafeStreets in order to analyze them and improve the environment in accordance with the suggestions the system is providing.
    - Police Station:
        - It's again a public entity who needs to mine data over all the validated ones with which supports their workflow in order to generate traffic tickets.

2.4 Assumptions, dependencies, and constraints

2.4.1 Assumptions

- [D1]: The fiscal code is unique;
- [D2]: The municipality gives the password to citizens who wish to activate their account;

- [D3]: The municipality offers a service that allows users to retrieve the information about the accidents that occur on the territory of the municipality;
- [D4]: The local police provides feedbacks;

2.4.2 Dependencies

All the functionalities, SafeStreets has, are closely related to each other as behind them there is the common activity of providing reports performed by citizens.
In addiction,given that we are dealing with limited realities, the system will be dependent on the municipality for what regards the availability of an updated list of street names. It is important to highlight this because the list will perform a crucial role in the validation process of a report.

2.4.3 Constraints

As the S2B is thought to be placed in an italian local reality, it must stick to the european Regulation on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, also known as GDPR.
Moreover, given the nature of the system, such as a mobile friendly browser based online platform, services provided by the S2B will be accessible only from users supporting a 2G/3G/4G or any active Internet connection, such as WiFi, on their device. SafeStreets will be available by the PAs within intranet networks in order to guarantee the security of the system itself.

3. Specific Requirements

According to the fact that we are strictly working with the municipality, we skipped the registration process on purpose. Given the GDPR regulamentation, the PA provides us the list with fiscal code and password of each adult. The password will be then requested by the user to the municipality.

A special remark has to be done for what regards the end of the *Problem* section in the assignment document. About the consideration made by the customer on the external interfaces design, the assumption made for the entire project is to build those required interfaces in order to support external activities, such as the retrieving of accidents by the citizens and the issuing traffic tickets by the police, whose implementation does not fall under the purview of the design of SafeStreets.

3.1 External Interface Requirements

3.1.1 User Interfaces

Different user interfaces are developed based on the type of end user. Each of them will be both accessible with a browser. A mobile-friendly interface is provided in order to facilitate the use with any device.This webpage is shown to the citizens before they can use the platform.

Fig. 4 Homepage SafeStreets

Based on the role of the user, SafeStreets displays different interfaces. The user will have access to some statistics, according to the project assignment and privacy policy, and all the PAs will have to own disposal a full dashboard (Fig. 4) in order to take actions and get full support from the system.

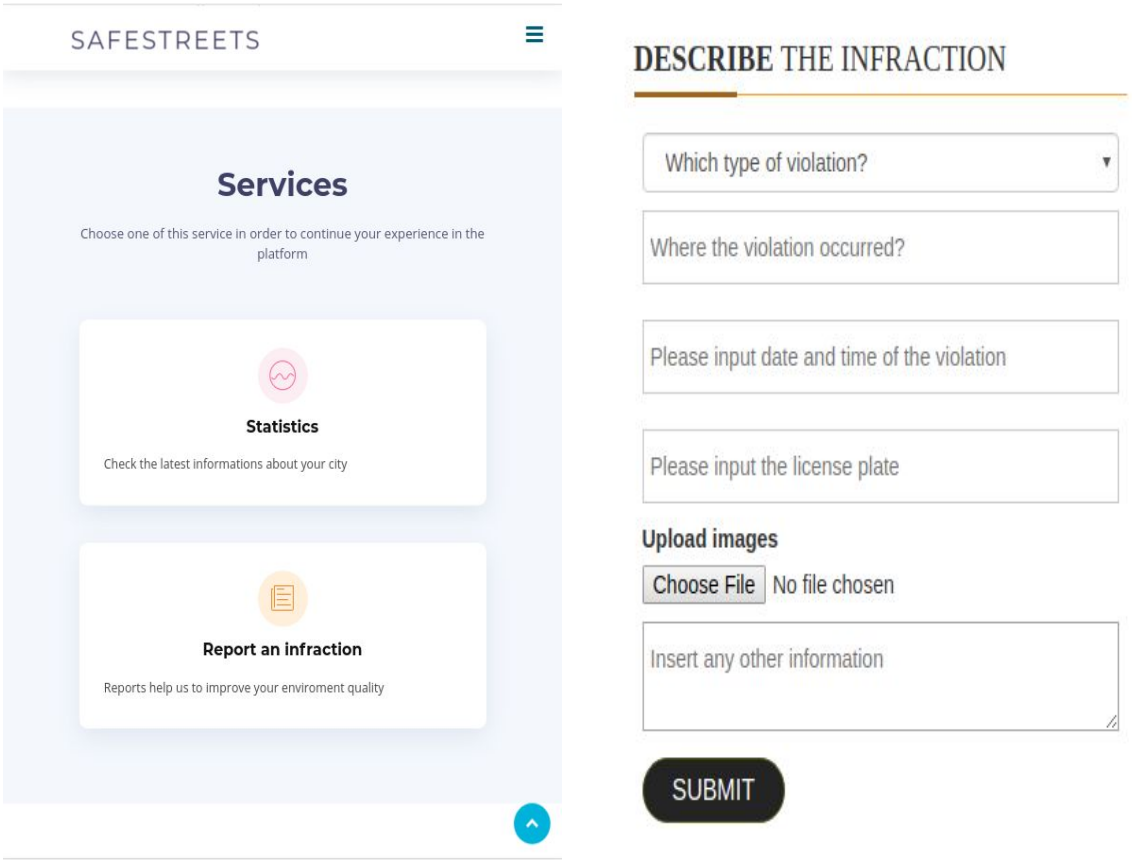


Fig. 5 Citizen Interface

As we can see from the image above (Fig.5), citizens have the possibilities to report an infraction, providing all the related informations. A minimalistic form, with fields to describe the infraction, is thought in order to make easier for a citizen its use.
All the data collected by the system will be then reorganized and refined in order to provide statistics and charts accessible by each user through the “statistics” section.

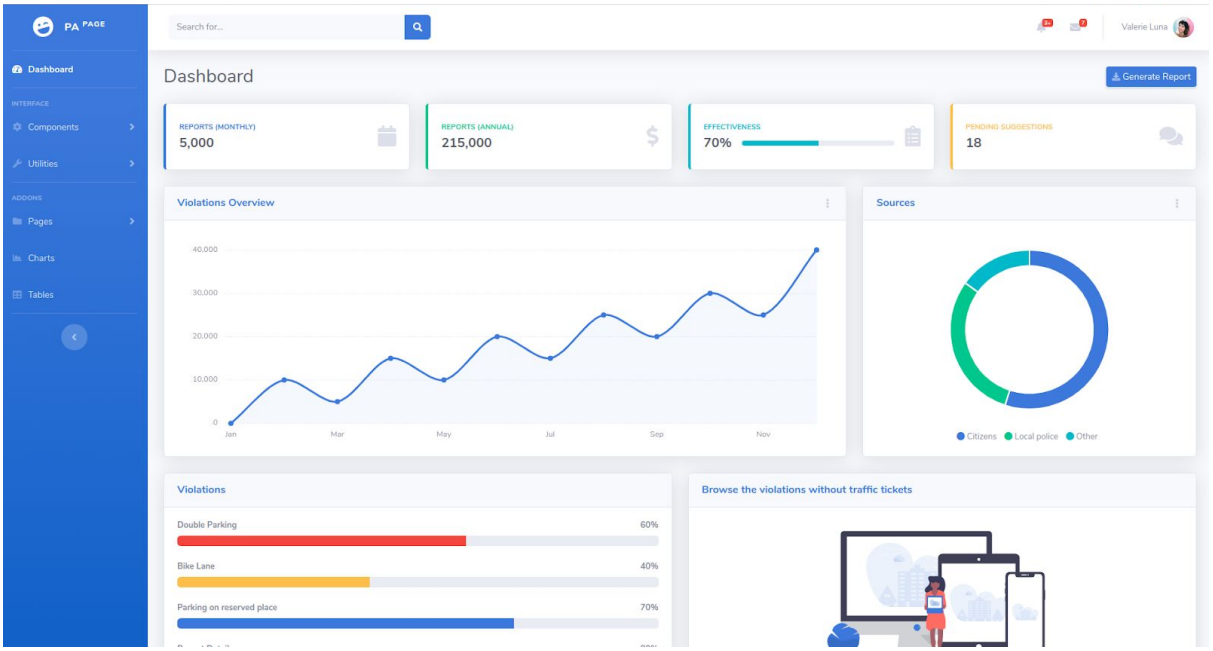


Fig. 6 PA Dashboard

PAs are already logged in, in a way that they can use SafeStreets's services within the PA's network (such as an Intranet or vpn). This interface provides charts and numbers to control trends and to get a reliable model of the city.
Components support the PA in its actions such as making decisions and generating traffic tickets.

3.1.2 Hardware Requirements

There are no relevant hardware interfaces to be used or developed.

3.1.3 Software Interfaces

In order to cross informations with the municipality, a software interface needs to be designed and developed. It will be used to provide an access point to PA with which they can provide all the data that will be then handled in order to suggest reasonable interventions to improve the environment.

3.1.4 Communication Interfaces

The communication interfaces, that will be used by the citizen, are the ones in order to connect the device (laptop or computer) to internet, such as 2G/3G/4G or WIFI interface.
A particular interface is needed by the PA in order to use the intranet.

3.2 Functional Requirements

In this section, it is shown that the requirements ensure the satisfaction of the goals in the context of the domain assumptions.
Throughout the whole RASD, it has been used the terms Municipality and Local Police as a specification of PA (see section “Definitions”). However, a better specification must take place, as in some situation these two entities are represented by humans, while some other times they’re represented by undergoing automated processes, that, given their nature will be better described in the DD. We will use the requirements to discriminate these situations.
Requirements R[4], R[9], R[10], R[12], R[13], R[14], R[15], R[16], R[18] consider physical people representing authorities exploiting SafeStreets services.
Requirements R[1], R[2], R[11], R[17] do not consider physical people but automated undergoing processes.

SafeStreets

[G1] it allows citizens to report traffic violation with relative informations.

- [D1]: The fiscal code is unique.
- [D2]: The municipality gives the password to citizens who wish to activate their account.
- [R1] it allows municipality to notify when a citizen becomes adult (account creation).
- [R2] it allows municipality to notify when an adult wants to activate his account.
- [R3] it supports the login of citizens.
- [R4] it supports the access of the platform by the municipality.
- [R5] it provides a form to be filled with pictures of the violation and its data.
- [R6] it provides a special field to be filled with the license plate of the vehicle committing the infraction.
- [R7] it notifies the results of the acceptance procedure to the citizen who reported a violation.

[G2] it allows both end users to mine informations.

- [R8] it provides charts and statistics for citizens.
- [R9] it provides charts and statistics for PA.

[G3] it provides suggestion for unsafe areas.

- [D3]: The municipality offers a service that allows users to retrieve the information about the accidents that occur on the territory of the municipality.
- [R10] it provides a search interface for PA in order to find unsafe areas.

- [R11] it allows municipality to upload data about the accidents that occurred in its territory.
- [R12] it allows PA to get a possible solution for an unsafe area.
- [R13] it supports city safeness by notifying PA about a new unsafe area.

[G4] it supplies local police with intelligence to generate traffic ticket.
- [R14] it supports the access of the platform by the local police.
- [R15] it provides to local police the records of infractions occurred on the territory of municipality.
- [R16] it allows local police to show an infraction in detail with all available data.

[G5] it provides TI-statistics in order to evaluate the effectiveness of the service.
- [D4]: The local police provides feedbacks.
- [R17] it allows local police to send feedbacks.
- [R18] it allows municipality to obtain the self-evaluated informations.

3.2.1 Scenarios

3.2.1.1 Scenario 1

Lorenzo is now an adult. He has been told by his parents that a new service, called SafeStreets, for the participation to the improvement of the GMC of Milan has been developed.
In order to contribute to the cause, Lorenzo decides to go to the municipality office to ask for his password. Then, once home, Lorenzo managed to active his account accessing to the service with his FC and the password the municipality office provided him.

3.2.1.2 Scenario 2

Tayfun cares very much about his city. While he is going home from university, he sees a car that is obstructing the bike lane with an invalid parking. Since he activated his account on SafeStreets, he can now report this violation. Right away, Tayfun takes his smartphone, access the platform and uploads a picture of the infraction filling the displayed form with license plate, type of violation, data and location. The infraction is now reported in the system and it will be eventually taken in account by the municipality.
Tayfun will know if his report has been accepted thanks to a notification sent by SafeStreets.

3.2.1.3 Scenario 3

Stefano is a very smart student. A research project about the safeness of his city has been assigned by the university he’s attending. So he’d like to gather some information about the

number of violations that occurred last year across the city. With his SafeStreets activated account, this possibility became real. He proceeds to do the login and then selects the option that allow him to visualize statistics, including the one he is interested in.

3.2.1.4 Scenario 4

At the end of the year, the financial situation of the city of Trani is way better than it was expected. Hence the city council, during the final meeting of the year, decided to invest part of the income in order to improve the general mobility in the city.
The municipality can only decide several areas of the city on which focusing. The decision is made based on the data mined in SafeStreets's platform which reports some city center zones in which school buses are always stuck during lunchtime because of the aggressive parking when parents collected their children from school.

3.2.1.5 Scenario 5

The municipality of Turin offers his citizens the possibility to request informations about the accidents that occur on its territory. Given the huge demand , the responsible office forwards a request to SafeStreets asking for the identification of unsafe area, according to the concerns of the citizens in the first place. Crossing its information with the ones it has received from the municipality about car accidents occurred in the territory of the municipality itself, Safestreets identify that an aggressive parking activity on the bike lane in the street Alfonso Varano. Thus, it suggest that adding a barrier between the bike lane and the part of the road for motorized vehicles to prevent unsafe parking.

3.2.1.6 Scenario 6

During the month of December the cold has seriously compromised the activity of control usually carried out by the police and the number of emitted traffic tickets has significantly dropped. To repair to the situation, the local police decides to avail of the services SafeStreets offers in the issuing of traffic tickets. So, it demands a list of validated reports.

3.2.1.7 Scenario 7

Mr. Alongi is the mayor of Palermo. Several months after the city council has decided to adopt Safestreets for supporting the local police in generating tickets, he wants to understand quantitatively how this collaboration is going and what results have been achieved, if any have. Also, he's interested in visualizing a measure of the effectiveness of the police action. So he decides to interrogate the municipality in order to retrieve these informations. The municipality forwards the request to SafeStreets which gives back the needed evaluations.

3.2.2 Use cases diagrams

Use cases diagrams are provided in order to represent user's interaction with the system, showing the relationship between the users and the different use cases in which the users are involved.

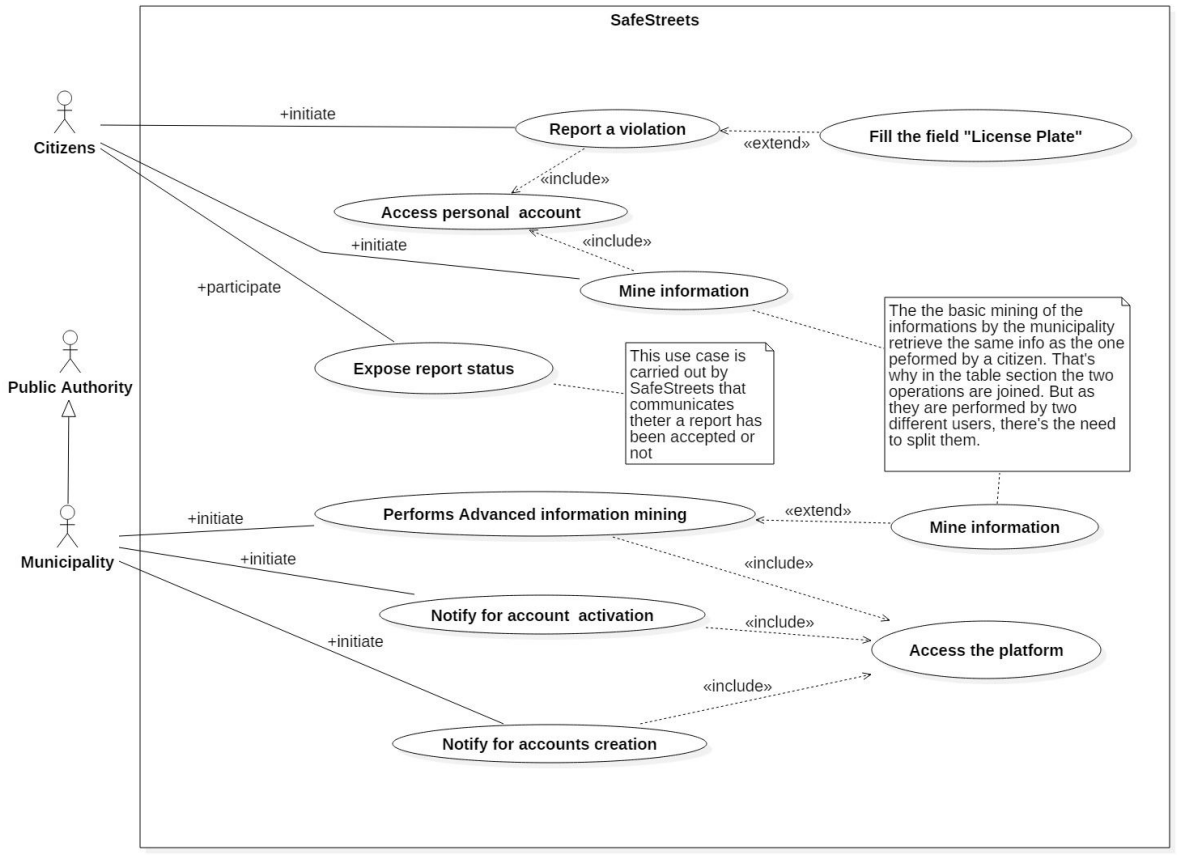


Fig. 7 Basic Service Use Case

Event	Access the platform (U1.1)
Actor	Municipality
Entry condition	Municipality needs to use SafeStreets services.

Event flow	Local Police goes on the website inside its intranet.
Exit condition	Local Police successfully performed the access.
Exceptions	None

Event	Notify for accounts creation (U1.2)
Actor	Municipality
Entry condition	Since it is a daily process no particular entry condition are required.
Event flow	1. Municipality verifies which citizen became adult. 2. Municipality puts in a list each citizen who became adult during the day. 3. Municipality accesses the portal. 4. Municipality sends the list to Safestreet.
Exit condition	All new accounts have been created in SafeStreets system
Exceptions	A user already exists in the system.

Event	Notify for account activation (U1.3)
Actor	Municipality
Entry condition	A citizen reclaims its SafeStreets account
Event flow	1. The municipality generates a new password. 2. Municipality notifies the password to SafeStreets system and to the citizen.
Exit condition	The citizen account has been created.
Exceptions	1. The citizen account doesn't exist in the SafeStreets system.

Event	Access to SafeStreets account (U1.4)
Actor	Citizen
Entry condition	The citizen is on SafeStreets's Homepage website
Event Flow	1. He/She opens the login form. 2. He/she inserts the FC and the password he received from the municipality. 3. He/she does the login.
Exit condition	The user is logged in.
Exceptions	1. The user filled the form with wrong input data. Redirected to the login page. 2. There were communication errors between municipality and SafeStreets.

Event	Report a violation (U1.5)
Actor	Citizen
Entry condition	The citizen sees a traffic violation
Event flow	1. (S)he takes the smartphone and logs in the SafeStreets account. 2. (S)he opens the report form and fulfills all the fields that are required. 3. (S)he takes a picture of the infraction and then uploads it in the form. 4. (S)he manually fill a field with the license plate of the vehicle.[Optional] 5. (S)he submits the report.
Exit condition	The report is registered by the system.
Exceptions	As the acceptance procedure is carried out internally by the system, no exceptions can occur in this phase.

Event	Mine informations (U1.6)
Actor	User
Entry condition	The user needs to gather informations related to the number of

	violations across the city
Event flow	1. User enters in its SafeStreets account. 2. It goes in the statistics section. 3. It can mine data according to the default filters already implemented on the platform.
Exit condition	Data has been retrieved by the user.
Exceptions	None

Event	Perform advanced informations mining (U1.7)
Actor	Municipality
Entry condition	Municipality has accessed the platform.
Event flow	1. Municipality enters in the statistics section. 2. Municipality requests statistics that contains sensible informations.
Exit condition	Data are displayed to the municipality.
Exceptions	None

Event	Expose report status (U1.8)
Actor	SafeStreets, Citizen
Entry condition	The report has been evaluated.
Event flow	1. SafeStreets checks the report with all the acceptance parameters. 2. SafeStreets shows a notification pop-up. 3. Citizen clicks on the pop-up.
Exit condition	Citizen visualize whether the report has been accepted or not.
Exceptions	None

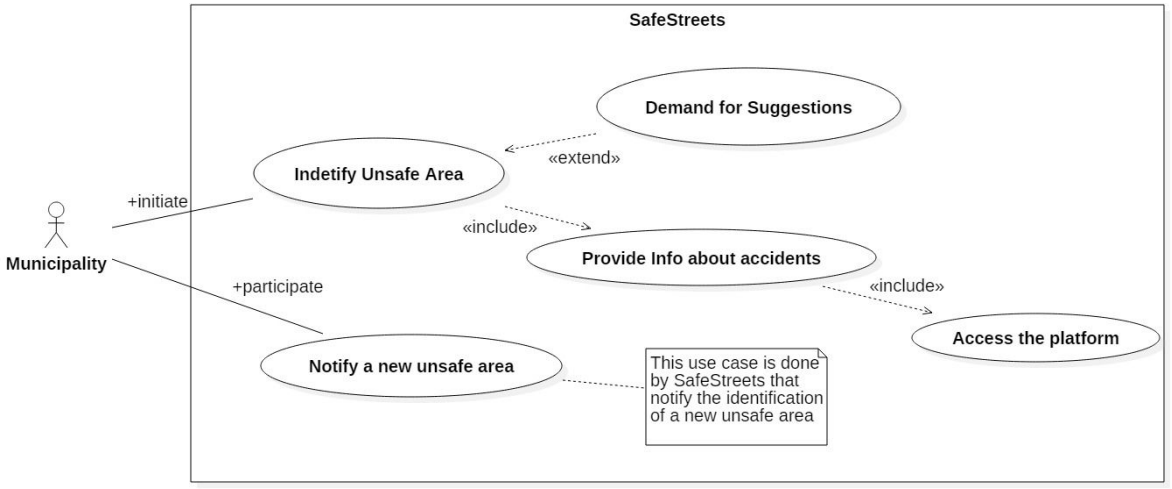


Fig.8 Advanced Function 1 Use Case

Event	Provide information about accidents (U2.1)
Actor	Municipality
Entry condition	No entry condition as it is a daily process.
Event flow	1. The municipality accesses its dashboard. 2. The municipality selects the data about the accidents it wants to upload. 3. The municipality uploads the data.
Exit condition	Data provided by the municipality are correctly stored.
Exceptions	None

Event	Identify unsafe area (U2.2)
Actor	Municipality, SafeStreets
Entry condition	Municipality is looking at his dashboard.
Event flow	1. Municipality searches for unsafe area of its territory 2. SafeStreets present a list of identified unsafe areas. 3. Safestreets present a list of unsafe areas.

Exit condition	Municipality visualize the results.
Exceptions	1. The system does not recognize the type of infraction. Hence it cannot provide a suggestion

Event	Notify new unsafe area (U2.3)
Actor	Municipality, SafeStreets
Entry condition	Municipality is looking at his dashboard.
Event flow	1. SafeStreets identifies a new unsafe area. 2. SafeStreets shows a notification pop-up. 3. Municipality clicks on “New unsafe area” pop-up.
Exit condition	Municipality visualize the results.
Exceptions	None

Event	Demand for suggestions (U2.4)
Actor	Municipality, SafeStreets
Entry condition	Municipality has received the list of unsafe area
Event flow	1. Municipality scrolls through a list of unsafe areas. 2. Municipality select a particular unsafe areas. 3. Municipality asks for suggestions. 4. Safestreets present a list of possible solution per area.
Exit condition	Municipality visualize the results.
Exceptions	2. The system does not recognize the type of infraction. Hence it cannot provide a suggestion

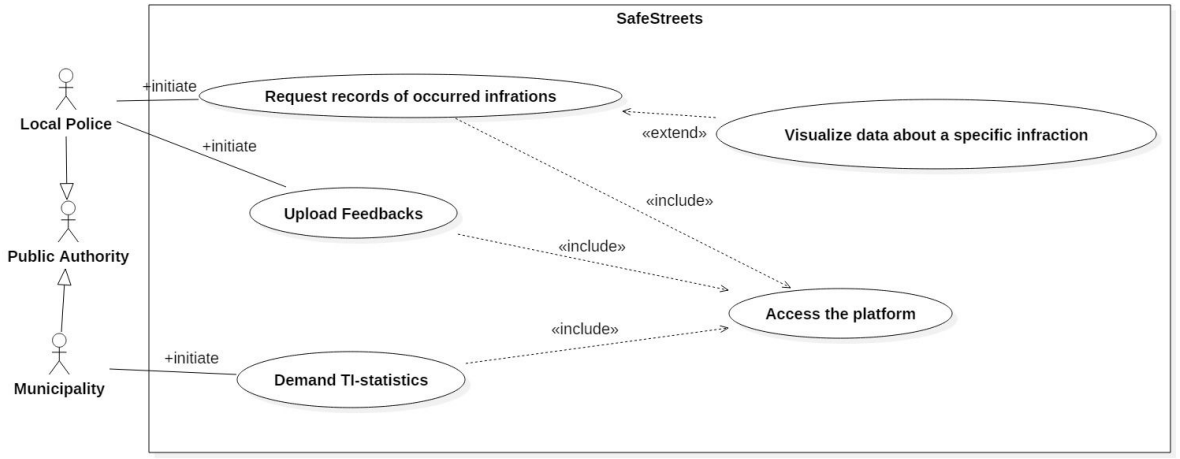


Fig. 9 Advanced Function 2 Use Case

Event	Access the platform (U3.1)
Actor	Local Police
Entry condition	Local Police needs to use SafeStreets services.
Event flow	1. Local police goes on the website inside its intranet
Exit condition	Local Police successfully accessed the platform
Exceptions	None

Event	Upload feedback (U3.2)
Actor	Local Police
Entry condition	Since it is a daily process no particular entry condition is needed.
Event flow	1. Local police access SafeStreets platform 2. Local police select a list of feedbacks it wants to provide to SafeStreets 3. Local police uploads the feedbacks.
Exit condition	Feedbacks are correctly stored by Safestreets.
Exceptions	None

Event	Demand TI-statistics (U3.3)
Actor	Municipality
Entry condition	An analysis about SafeStreets services needs to be done.
Event flow	1. Municipality accesses SafeStreets portal. 2. It gets the statistic about the effectiveness of this services.
Exit condition	An index about the effectiveness of SafeStreets is provided by SafeStreets itself.
Exceptions	None

Event	Request records of occurred infrations (U3.4)
Actor	Police office
Entry condition	Police office wants to look all the infractions it misses during the week.
Event flow	1. It accesses SafeStreets portal. 2. It goes in an apposite service SafeStreets offers. 3. It selects a certain period in time. 4. Local Police asks for all the violations without traffic ticket issued in that period.
Exit condition	A refined list of all the infractions without traffic ticket is provided by the system.
Exceptions	None

Event	Visualize data about a specific infraction (U3.5)
Actor	Police office
Entry condition	Local police obtained lists of occurred infractions.
Event flow	1. It selects a specific infraction.
Exit condition	All the specific data about the infraction are displayed back.
Exceptions	None

3.2.3 Traceability matrix

Row ID	Use Case ID	Req	Goal
r1	U1.1	R[4]	G1
r2	U1.2	R[1]	G1
r3	U1.3	R[4]	G1
r4	U1.4	R[3]	G1
r5	U1.5	R[5]\R[6]	G1
r6	U1.6	R[8]	G2
r7	U1.7	R[9]	G2
r8	U1.8	R[7]	G1
r9	U2.1	R[11]	G3
r10	U2.2	R[10]	G3
r11	U2.3	R[13]	G3
r12	U2.4	R[12]	G3
r13	U3.1	R[14]	G4
r14	U3.2	R[17]	G5
r15	U3.3	R[18]	G5
r16	U3.4	R[15]	G4
r17	U3.5	R[16]	G4

3.2.4 Activity diagram

The whole procedure of a report being made by a citizen and accepted by SafeStreets is modeled with an activity diagram.

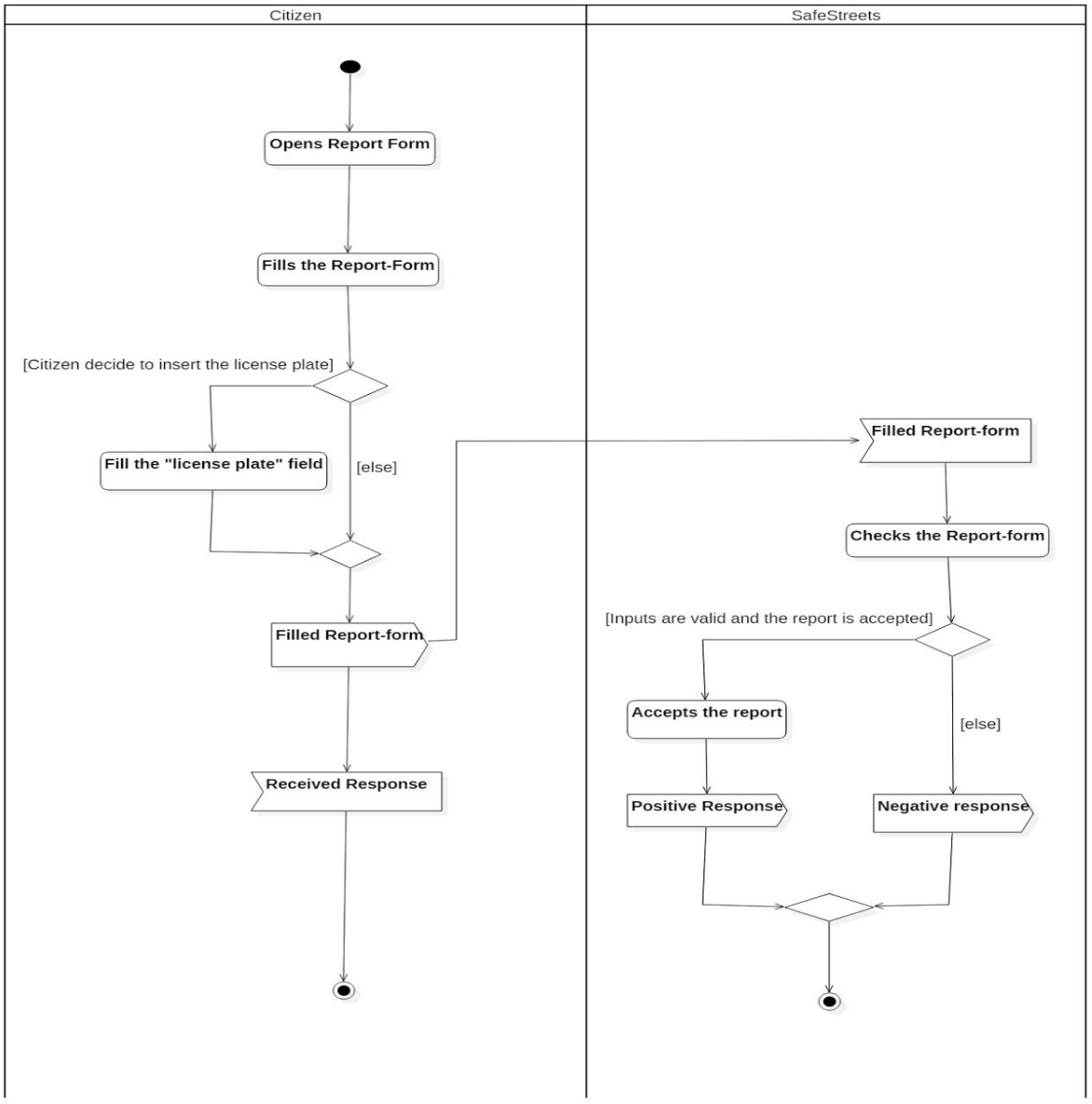


Fig. 10 Citizen Reporting Violations Activity Diagram

3.2.5 Sequence diagrams

Important interactions between the different actors are here represented in the sequence diagrams below.

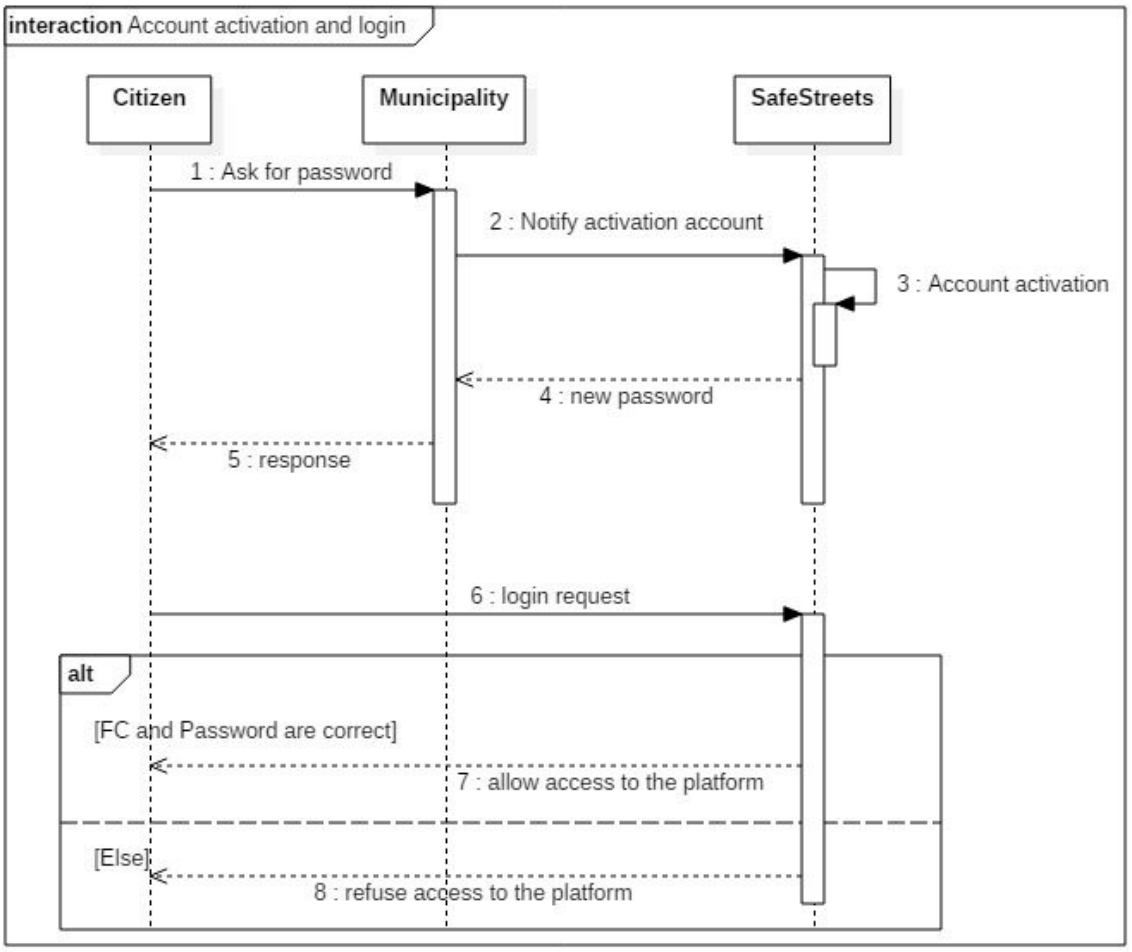


Fig. 11 Account Activation and Login Sequence Diagram

In Fig.11 the whole interaction, that involves a Citizen, the Municipality and SafeStreets, and aimed to the activation of an account and the consequent login is shown.

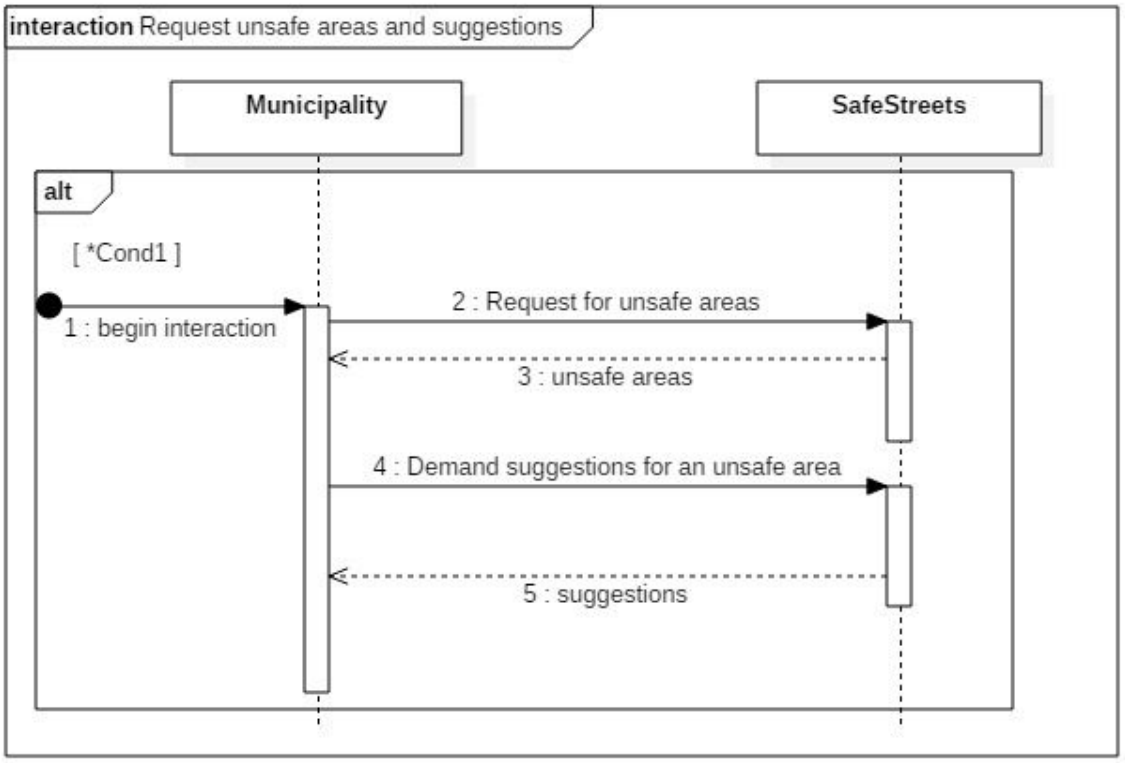


Fig. 12 Unsafe Areas and Suggestions Request Sequence Diagram

*Cond1 → it is referred to the fact that the municipality exposes the service that citizens can exploit in order to retrieve accidents that occur in its territory. If these data are exposed and the service is enabled, then the condition is true and the crossing of the data is possible, otherwise this cannot happen.

3.3 Performance Requirements

No particular performance requirements are needed for this application.

3.4 Design Constraints

In the citizen UI the report upload needs to provide a minimalist and cleaned form in order to minimize the complexity of the process and help the user to fast describe the infraction.

3.4.1 Standards compliance

The S2B leds to manage a significant amount of data that we can classify in two sets: personal user data and private data provided by the municipality. They must be compliant to GDPR specifications.
HTTPS protocol is used as standard in order to guarantee a secure connection between the system and the clients.

3.5 Software System Attributes

3.5.1 Reliability

Keeping in mind the deep connection that occurs between the functionalities the system offers, the system must stick to an high level of reliability. Redundant storage disks are employed to avoid system failures in case of data losses.

3.5.2 Availability

Given the nature of the S2B, as such of a platform of public utility that strongly interacts with both citizens and authorities, the application must guarantee a full time service availability.

3.5.3 Security

With regards to GDPR specifications, collected data cannot be stolen or used for different purposes for which they have been collected. More in detail, considering handled informations,the system must not provide personal informations about reporters.
This consideration is strengthened by the reflection on physical security issues citizens may be exposed to, due to the violations reporting/committing activity. High level security protocols must be put in place to prevent access to sensible information from malicious entities. As mentioned before, the connections to the system stick to HTTPS protocol in order to mitigate attacks e.g. MITM attack. The PAs will be considered by the S2B as trusted elements in the analysis of the security process.
All the reports are subjected to the acceptance and later to the validation procedure. This way the chain of custody of the information exchanged with users is put in place.

3.5.4 Maintainability

An high level of modularity is required in order to ease the process of maintenance of the S2B. Furthermore, as the analysis tools for statistical applications are continuously increasing, modularity will help also the process of update of existing functions will be subjected to e.g. machine learning and data mining algorithms.

3.5.5 Portability

No particular portability constraints are at stake, given the browser based access to the platform.

4. Formal analysis using alloy

4.1 Alloy code

In order to formally describe the domain with its properties, the Alloy modeling has been carried out. This is also useful to see if the stated properties will be satisfied by the system and the constraints will never be violated.
The assertions have been used to test several properties of the model. If an assertion shows counterexamples, an instance that breaks a property exists and modifications on the model are needed. Even if they are already fulfilled by facts, assertions have been left in order to keep track of the development of the alloy model.
Finally the assertions can be used for future testing of the model.

```
//--------Signatures--------
abstract sig AccountStatus{}
one sig Active extends AccountStatus{}
one sig Created extends AccountStatus{}
//All the status for reports
abstract sig ReportStatus {}
one sig Accepted extends ReportStatus{}
one sig Refused extends ReportStatus{}
one sig Validated extends ReportStatus{}
//Actors
abstract sig User {}
abstract sig PA extends User {}
//Used in order to collect all the Users for each kind of statistic
one sig EndUser {composedBy: some User}
one sig Municipality extends PA {}
one sig PoliceOffice extends PA {}
```

```
sig Citizen extends User {

	fiscalCode: one FiscalCode,
	status: one AccountStatus

}

//Fiscal code

sig FiscalCode{}

sig Location {}

sig Date {}

sig LicensePlate {}

abstract sig TicketStatus {}

one sig IssuedTicket extends TicketStatus {}

one sig NotIssuedTicket extends TicketStatus{}


//One report is associated to one citizen through its fiscal code

sig Report {

	status: one ReportStatus,
	fiscalCode: one FiscalCode,
	licensePlate: one LicensePlate,
	date : one Date,
	location: one Location,
	type: one TypeInfraction

}

sig Violation {

	licensePlate: one LicensePlate,
	reports: set Report,
	date : one Date,
	location: one Location,
	ticketStatus: one TicketStatus,
	type: one TypeInfraction

}

sig Feedback {

	violation: one Violation,
	madeBy: PoliceOffice

}

//Statistics - Two types of statistics are defined in order to highlight the different level of visibility based on the role

abstract sig Statistic {}
```

```
one sig StatBase extends Statistic  {
	accessibility: EndUser
}
one sig  StatAdvanced extends Statistic {
	accessibility: some PA
}
//Statistic that refers to advanced functionality 2
one sig  TIStatistics extends Statistic {
	accessibility: some PA
}
sig Suggestion {}
sig TypeInfraction {
	suggestion:Suggestion
}
sig Incident {
	location: one Location,
	date : one Date
}
```

```
sig UnsafeArea {

	location : one Location,
	violations: set Violation,
	incidents: set Incident

}

//--------Facts--------

//A report associated to a violation is validated

fact AViolationContainsOnlyValidatedReport {

	all r: Report | r.status = Validated <=> r in Violation.reports

}

//Fiscal code is unique
fact UniqueFiscalCode{

	no c:Citizen |  c.fiscalCode in (Citizen-c).fiscalCode

}




fact AViolationExists{

	//All the reports associated in a violation have the license plate equal to the one in the violation
	all v: Violation | all r: v.reports | v.licensePlate = r.licensePlate

	//A report is used only for a violation
	no r: Report, v: Violation, s: Violation | v != s and r in v.reports and r in s.reports

	//Violation is created when we have at least 2 reports
	all v: Violation | #v.reports > 1

}

fact singleEvent{

	//Data, position and license plate define an event. All reports of the same event are treated in the same way
	no r, r' : Report | r.date = r'.date and r.location = r'.location and r.licensePlate = r'.licensePlate and r.status != r'.status and r=r'

	//No more than one report of the same citizen about the same event
	no r,r' : Report | r.date = r'.date and r.location = r'.location and r.licensePlate = r'.licensePlate and r.fiscalCode = r'.fiscalCode and r != r'

	//Date and Location are the same in the reports as in the Violation they are in
	all v: Violation, r: v.reports | r.date = v.date and r.location = v.location

}

fact noMoreThanOneReportInValidation{

	//There can be at most one report in validation about an event. With at least two reports accepted, a violation can be generated and the reports become validated
	no r : Report, r':(Report-r) | r.date = r'.date and  r.location = r'.location and r.licensePlate = r'.licensePlate and r.fiscalCode != r'.fiscalCode  and r.status=Accepted and r'.status=Accepted

}
```

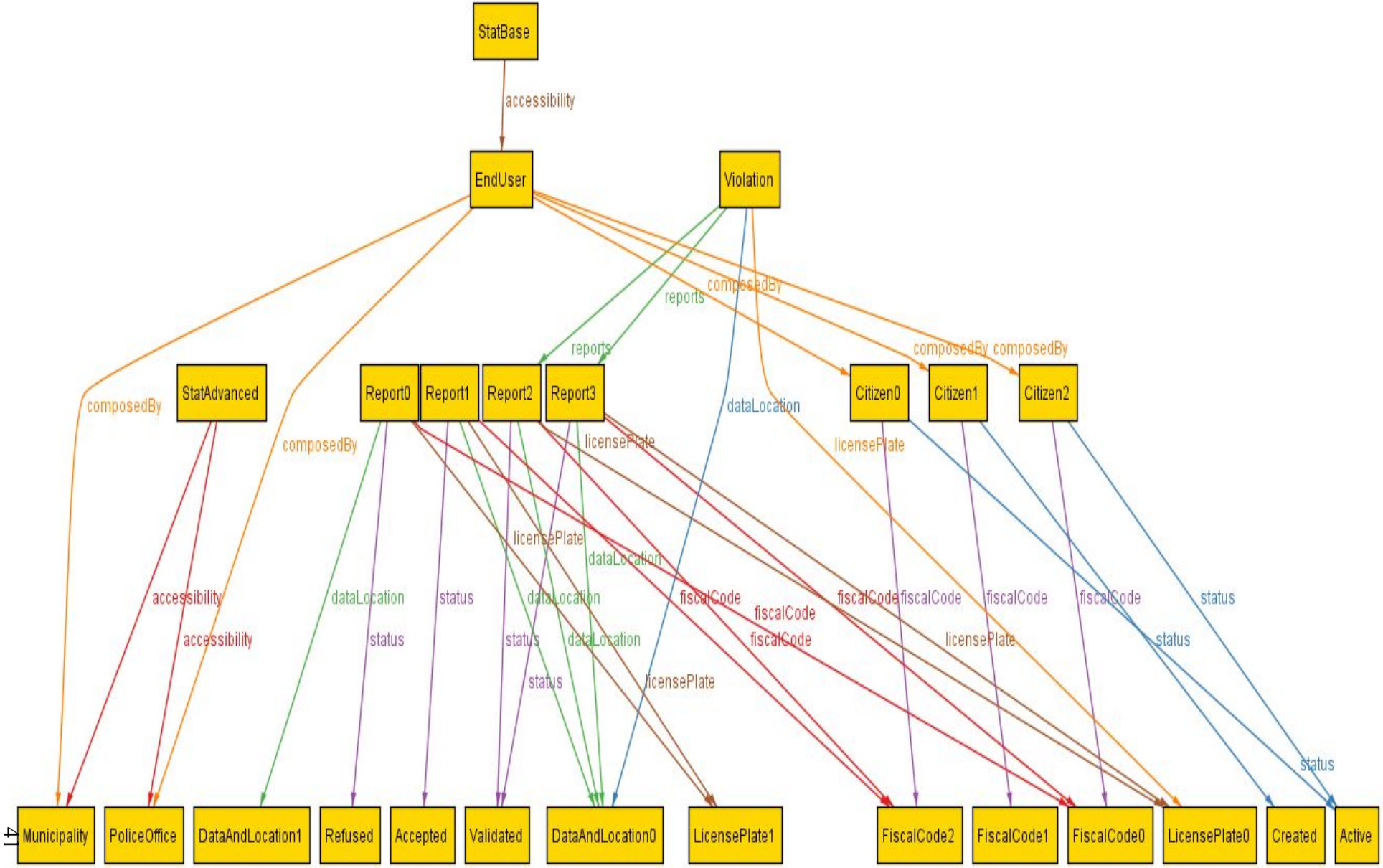
```
//Each of these needs a parent. They cannot float in the model
fact noFloatingEntities{

	all d : Date | d in Report.date
	all loc : Location | loc in Report.location
	all fc : FiscalCode | fc in Citizen.fiscalCode
	all lp : LicensePlate | lp in Report.licensePlate
	all tp:TypeInfraction| tp in Violation.type
	all sg:Suggestion| sg in TypeInfraction.suggestion

}

fact citizenMustBeActiveInOrderToReport {

	no c: Citizen | c.status = Created and c.fiscalCode in Report.fiscalCode

}

fact feedbackIssued {

	//No more than one feedback for violation
	no  f,f'' : Feedback | f != f'' and f.violation = f''.violation

	//No feedback with more than one violation
	no v,v': Violation, f : Feedback | v != v' and f.violation = v and f.violation = v'

	//No issued status without feedback
	no v : Violation | (v.ticketStatus = NotIssuedTicket and v in Feedback.violation) or ( v.ticketStatus = IssuedTicket and v not in Feedback.violation)

}

//Define which kind of user can access to each type of statistic
fact defineAccessibilityContrainst {

	all u : User | u in EndUser.composedBy
	all u : PA | u in StatAdvanced.accessibility
	all u : PA | u in TIStatistics.accessibility

}


fact cannotExistTwoViolationsThatAreTheSame{

	no v1,v2 : Violation | v1 != v2 and v1.date = v2.date and v1.location = v2.location and v1.licensePlate = v2.licensePlate

}

fact typeViolationIsTheSameOfItsReports{

	all v: Violation, r: v.reports | r.type = v.type

}

//For each type of infraction there is only a suggestion
fact differentInfractionsHaveDifferentSuggestions {

	no t,s:TypeInfraction | t!=s and s.suggestion = t.suggestion

}

//In order to reduce the graphic complexity, we reduced the threshold for establishing if an area is unsafe
fact NoDuplicateIncidents{

	no i1,i2: Incident | i1.location = i2.location and i1.date=i2.date and i1 != i2
	all un:UnsafeArea | #un.incidents > 1
	all ua: UnsafeArea, disj i1,i2 : ua.incidents | i1.location =i2.location and i1.location = ua.location

}
```

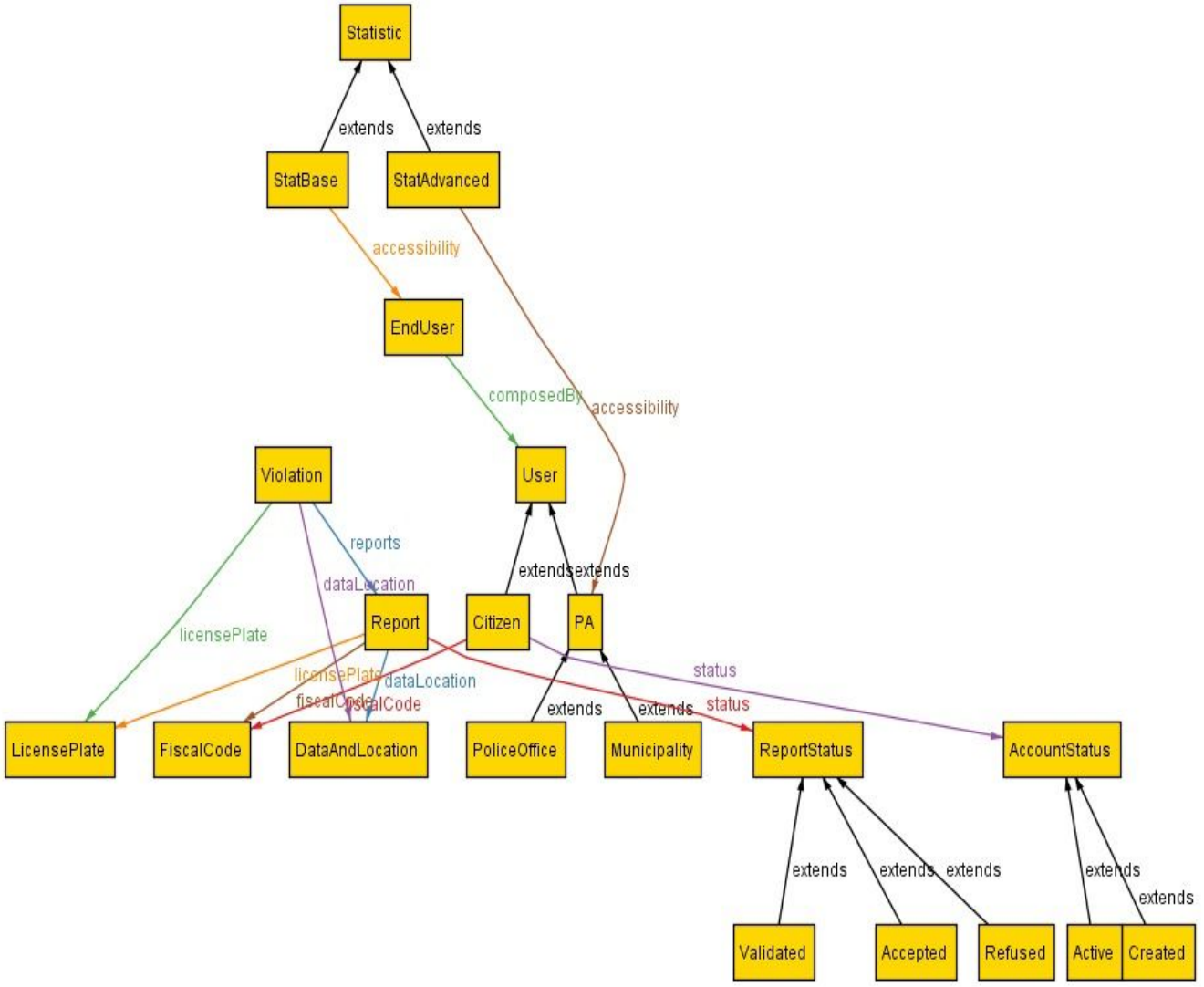
```
//Unsafe area has the same location as for the violations in it
fact NoLocationInUnsafeAreaNotInViolation{

	// To be an unsafe area, it must have 3 or more violations.
	all ua : UnsafeArea| #ua.violations>1

	// Given two different unsafe areas, they do not have the same location. For all the unsafe areas.
	no ua, ua' : UnsafeArea | ua != ua' and ua.location = ua'.location

	// For all the violations associated with an unsafe area, the location is the same
	all ua: UnsafeArea, disj v,v' : ua.violations | v.location =v'.location and v.location = ua.location

}


//--------Assertions--------

//There is no report that is not validated but there is already a violation
assert noSameReportForTheSameEvent {

	no disj r1,r2 : Report | r1.location = r2.location and r1.date = r2.date and r1.licensePlate = r2.licensePlate  and r1.fiscalCode = r2.fiscalCode

}


assert noInvalidStateCondition {

	all r: Report | r.status = Validated iff r in Violation.reports
	all c: Citizen | all r: Report | r.fiscalCode = c.fiscalCode implies  c.status = Active
	all f: Feedback | all v: Violation | (f.violation = v iff v.ticketStatus = IssuedTicket) and (v.ticketStatus = NotIssuedTicket iff v not in Feedback.violation )


}

assert violationConsistency {

	no v1,v2 : Violation | v1 != v2 and v1.date = v2.date and v1.location = v2.location and v1.licensePlate = v2.licensePlate
	all v: Violation, r: v.reports | r.type = v.type

}

assert noDifferentReportLocationInViolation {

	all v: Violation | all r : v.reports | r.location = v.location and v.date = r.date

}

assert extractUnsafeAreas {

	all ua: UnsafeArea |  all v: Violation | all r: Report | (ua.location = v.location and v.location =  r.location) iff( v in ua.violations and r in v.reports)
	all r:Report|r.status=Validated or r.status = Accepted or r.status = Refused

}

check noSameReportForTheSameEvent for 4
check noDifferentReportLocationInViolation for 5
check extractUnsafeAreas for 3
check noInvalidStateCondition for 3
check violationConsistency for 5
```

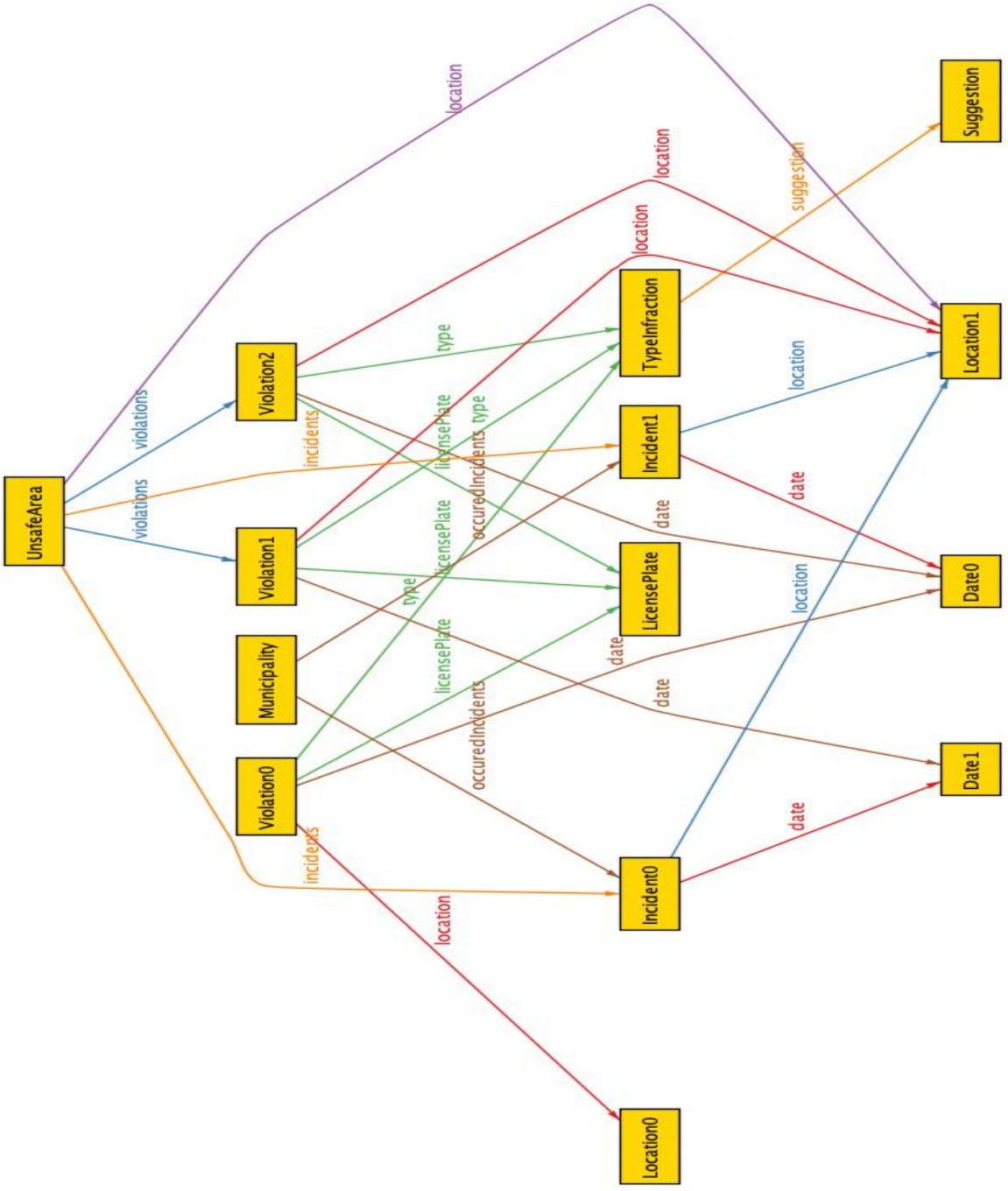
```
//Generate basic service world

pred basicService {

	#Report = 4
	#Violation = 1
	#Citizen = 3

}

//Generate Advanced Function2 world

pred advancedFunction2{

	#Report = 5
	#Violation = 2
	#Feedback = 1
	#Citizen = 3

}

//Generate Advanced Function1// world

pred advancedFunction1{

	#Violation =3
	#UnsafeArea = 1
	#Date=2
	#Incident = 2
	#Location = 2

}

run   basicService for 6
run  advancedFunction1 for 8
run advancedFunction2 for 6
```

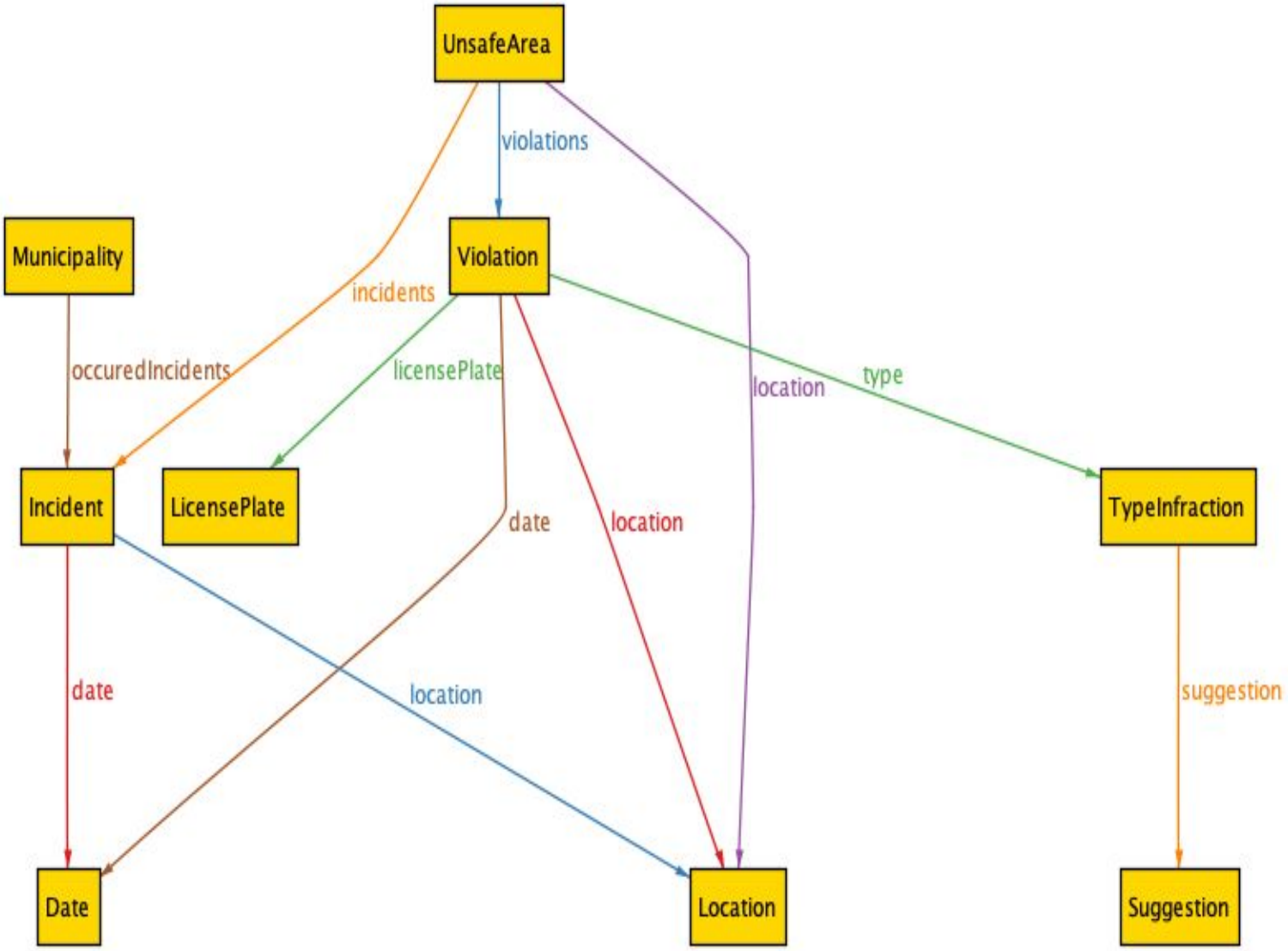
4.2 Alloy instances

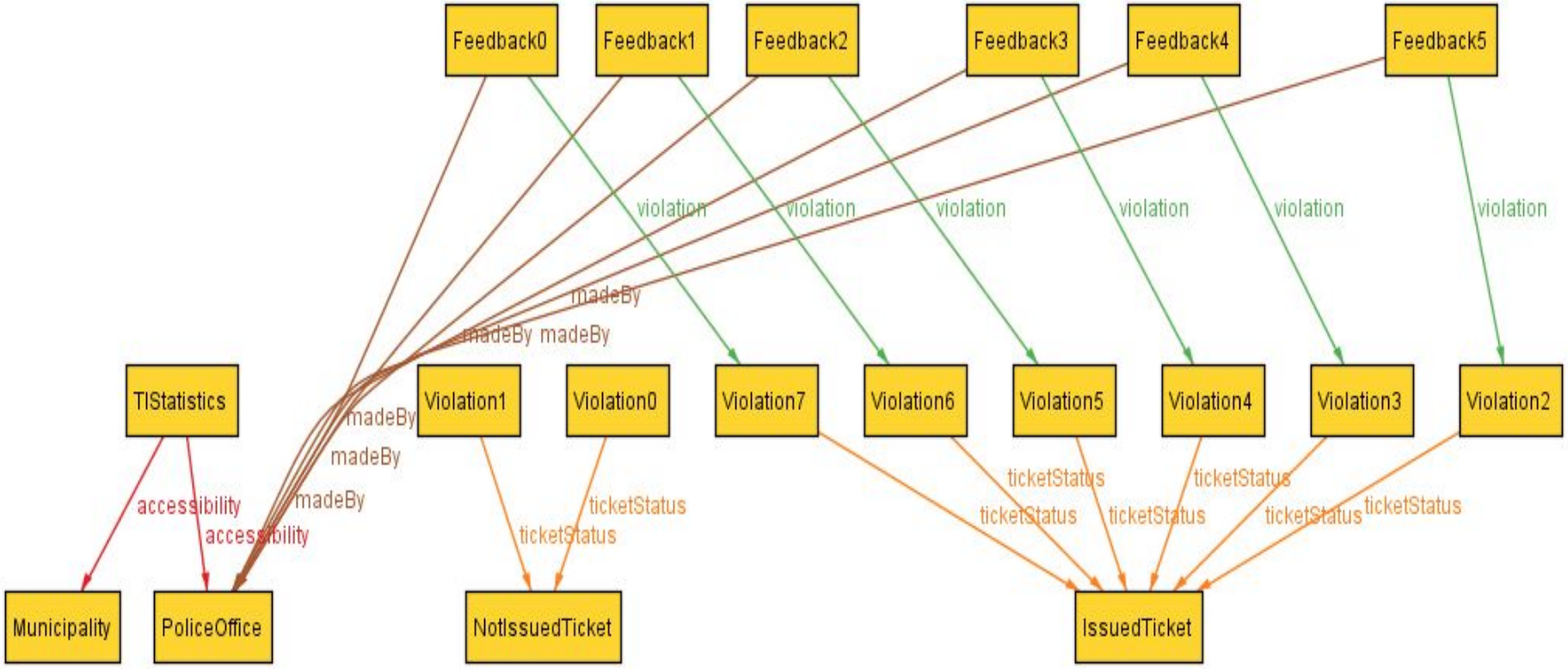
Here some instances of the world, along with the metamodels, generated by the three different functionalities of SafeStreets are presented; the first two models refer to BasicService, the second ones to Advanced Function 1 and the third ones to Advanced Function 2, respectively. With instances, we check the consistency of the model.
It has been chosen not to present an instance of the whole models as it would have been too complex, penalizing the clearness and comprehensibility of the world itself.

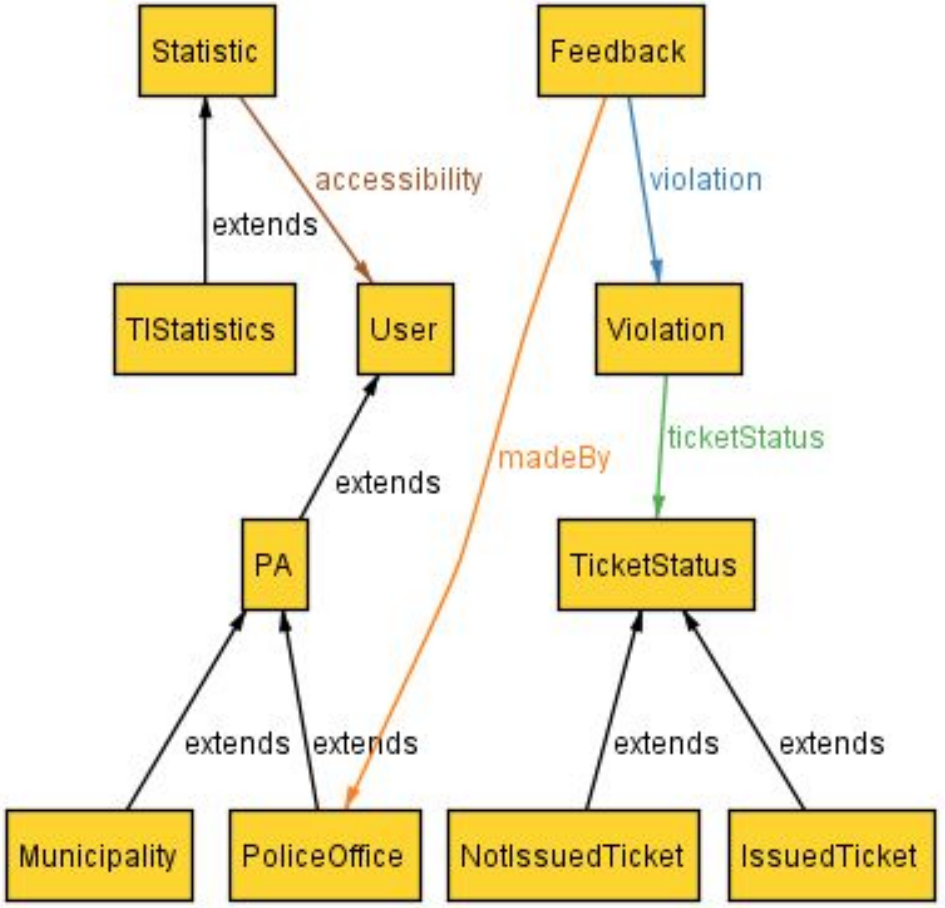

StatBase
accessibility
EndUser
Violation
composedBy
reports
composedBy
reports
composedBy
composedBy
StatAdvanced
Report0
Report1
Report2
Report3
Citizen0
Citizen1
Citizen2
composedBy
composedBy
dataLocation
licensePlate
licensePlate
licensePlate
dataLocation
accessibility
accessibility
dataLocation
status
dataLocation
fiscalCode
fiscalCode
fiscalCode
fiscalCode
fiscalCode
fiscalCode
status
status
status
dataLocation
fiscalCode
licensePlate
status
licensePlate
status
Municipality
PoliceOffice
DataAndLocation1
Refused
Accepted
Validated
DataAndLocation0
LicensePlate1
FiscalCode2
FiscalCode1
FiscalCode0
LicensePlate0
Created
Active


Statistic
extends
extends
StatBase
StatAdvanced
accessibility
EndUser
composedBy
accessibility
Violation
User
reports
dataLocation
extends
extends
Report
Citizen
PA
licensePlate
licensePlate
dataLocation
fiscalCode
fiscalCode
status
status
extends
extends
LicensePlate
FiscalCode
DataAndLocation
PoliceOffice
Municipality
ReportStatus
AccountStatus
extends
extends
extends
extends
extends
Validated
Accepted
Refused
Active
Created


UnsafeArea
violations
violations
location
Violation2
Violation1
Municipality
Violation0
incidents
incidents
location
type
licensePlate
occuredIncidents
type
licensePlate
occuredIncidents
licensePlate
occuredIncidents
TypeInfraction
Incident1
LicensePlate
Incident0
location
location
suggestion
location
date
date
date
location
date
date
Suggestion
Location1
Date0
Date1
Location0


UnsafeArea
violations
Municipality
Violation
incidents
occuredIncidents
licensePlate
location
type
Incident
LicensePlate
date
location
TypeInfraction
date
location
suggestion
Date
Location
Suggestion


Feedback0
Feedback1
Feedback2
Feedback3
Feedback4
Feedback5
violation
violation
violation
violation
violation
violation
madeBy
madeBy
madeBy
madeBy
madeBy
madeBy
TIStatistics
Violation1
Violation0
Violation7
Violation6
Violation5
Violation4
Violation3
Violation2
accessibility
accessibility
ticketStatus
ticketStatus
ticketStatus
ticketStatus
ticketStatus
ticketStatus
ticketStatus
ticketStatus
Municipality
PoliceOffice
NotIssuedTicket
IssuedTicket


Statistic
Feedback
accessibility
violation
extends
TIStatistics
User
Violation
ticketStatus
madeBy
extends
PA
TicketStatus
extends
extends
extends
extends
Municipality
PoliceOffice
NotIssuedTicket
IssuedTicket

5. Effort Spent

Here is reported the incremental effort spent by each member of the group on the project, measured in hours.

Date	Amaranto	De Cillis	Gumus	Subject Matter
21/10/2019	2:45	1:45	0	Goals & Req & Document writing
22/10/2019	2:45	2:45	2:30	Goals & Req & Document writing
23/10/2019	8:45	11	7:30	Goals & Req & Document writing
24/10/2019	13:15	15:30	11:30	Class & States Diagrams & Document writing
25/10/2019	14:15	16:45	11:30	Use Case Diagrams & Document writing
26/10/2019	14:15	18:30	11:30	Use Case Diagrams & Document writing
28/10/2019	19:15	23:30	15:30	Use Case Diagrams & Document writing
29/10/2019	22:45	25:30	17	Activity Diagram &

				Document writing
30/10/2019	27:15	31:50	21	Alloy model
31/10/2019	31:15	35:50	25:15	Alloy model
1/11/2019	33:15	35:50	30	Alloy model
2/11/2019	37:30	38	36:30	Alloy model
4/11/2019	45:30	46	43	Alloy model
5/11/2019	53	54	50:30	Alloy model & Document writing
6/11/2019	58	58	58	Document general review
7/11/2019	64	64	64	Last review and submit
Total Amount	64	64	64	/