Contents

Introduction

1.1 Purpose

The purpose of this Requirement Analysis and Specification Document (RASD) is to provide a detailed description about the software SafeStreets. In particular, this document is focused on important aspects that are useful during the design of the software architecture like: scope, functional and non-functional requirements, use cases and scenarios, constraints and assumption, UML diagrams, limitation and interfaces with other software. Overall the document is a useful guide for the developers that will have to follow and implement all the necessary requirements; nevertheless it is also a document that can be given to potential customers to get them an idea of what the software will be like.

The software features have been identified in the following list of goals.

1.1.1 Goals

- [G1]: Allow the user to send a violation report consisting in a picture and some metadata;

- [G2]: Allow the user to watch the history of his reports and their status;

- [G3]: Allow the user to watch on map areas and streets with an high frequency of violations;

  - [G3.1]: The user gets the latest violations near its current location;

  - [G3.2]: The user gets the latest violations near the location that he specifies;

- [G4]: Allow the local system administrator of the police station to create accounts for the police technicians;

- [G5]: Allow the authorities to visualize, schedule and change the status of reports submitted by the users;

- [G6]: Allow the authorities to mine data to make analysis and retrieve statistics;

- [G7]: If the municipality offers a service that provides data about car accidents then the system must be able to cross this information with its owns data in order to identify possible unsafe areas;

  - [G.7.1]: In this case the system must be able to suggest possible interventions;

  - [G.7.2]: The users must be able watch this unsafe areas;

  - [G.7.3]: The authorities must be able to consult the crossed data.

1.2 Scope

1.2.1 Description of the given problem

SafeStreets is a service that aims to improve the safety of the streets via the help of the users. They can notify violations or any illegal behaviour related to street parking to authorities. In particular users can iteract with the service via an application that can be used to send the violation reports; the latters mainly consist in a picture, taken by the user, of the vehicle responsible for the violation. Moreover users can send, along with the picture, location, date and additional information related to the infringement. The system also provides a Web interface that can be used by authorities in order to check the violations recieved. It is important to note that the picture sent by the user must contain the car plate in order to let police officers know which is the real vehicle that committed the violation. The application also offers the possibility to see areas/streets with the highest violations rates thanks to the data collected over time. As an advanced functionality the application can interact with services offered by the municipality; in particular if a service offers data related to car accidents, SafeStreets can cross this information with its owns, in order to get a better idea of the potentially unsafe areas and therefore suggest some possible interventions. Finally the application will have to be scalable and easy to use in

order to provide a fast and efficient utilization for users that see a violation and want to report it immediately.

1.2.2 Current system

SafeStreets is a new service that it is entering the market. There aren't any legacy systems that need to be integrated into the application, with the exception of the third party services that offers some functionality required by the application (like Maps and plate recognition). Those services will be better explained in the following paragraphs.

1.3 Definitions, acronyms and abbreviations

1.3.1 Definitions

- *Most Dangerous Streets*: Streets with the highest frequency of violation reports.

1.3.2 Acronyms

- EU: European Union;

- CET: Central European Timezone;

- API: Application Programming Interface;

- HTTP: Hyper Text Transfer Protocol;

- GPS: Global Positioning System;

- GDPR: General Data Protection Regulation.

1.3.3 Abbreviations

- [Gn]: n-th goal;

- [Dn]: n-th domain assumption;

- [Rn]: n-th functional requirements;

- MDS: Most Dangerous Street;

- LSA: Local System Administrator;

- PT: Police Technician.

1.4 Revision History

- Version 1.0:

  - First release.

1.5 References

- SafeStreets assignment document;

- Previous years RASDs of the Software engineering 2 project;

- IEEE Std 830--1998 IEEE Recommended Practice for Software Requirements Specifications;

- Slides from the course "Software engineering 2".

1.6 Document Structure

1. In the first part an overall vision of the system is given. We provide a rather summarized description of the purpose of the application and its scope as wall as a list of fundamental goals that the application must accomplish;

2. In the second part are described the general architecture of the software and its principal functions. Moreover in this section are also defined the classes of utilizators;

3. The third part is the core of the document. It defines: the requirements and the domain assumptions for each goal, a set of scenarios that describes real uses cases of the application as well as different types of UML diagrams related to different aspects of the service like use cases diagrams and sequence diagrams. It is also proposed a list of assumptions and design constraint and an analysis of the non-functional attributes of the system;

4. The fourth part propose the Alloy formalization of the system composed by its fundamental functionalities;

5. The last part states the hours of work division and the tools used to create all the part of this RASD document.

Overall Description

2.1 Product perspective

SafeStreets is a software that needs to be fast and reliable in order to allow the user to send a report immediately after he detects a violation. The goal is to design an architecture that can offer a good level of performance with respect to scalability and portability. The system is devided into three separate applications: a mobile front-end application for the users, a Web application for the authorities and a back-end system that manages all the operations. More in detail, the user application is a light-weight crossplatform mobile client; with this approach the client side is relieved from the computation and the result is a fast application for the users. With this mobile application the users can perform the main functions related to him, such as: registration, login, send reports, consult the history of his reports and watch a map with the MDS. The application is connected to the backend service which is the part of the architecture that handles all the main operations regarding the elaboration of data. In fact the backend is the core of the architetcure: it manages the incoming requests and handles the interaction with the third party system for the plate recognition. This is fundamental because the authorities need to receive the plate number in order to immediately identify the right veichle. This operation is done by sending the picture taken by the user to an external service that finds the plate number in the image and sends it back to the system as a string. Moreover the backend provides an interface that interacts with the cloud-based database in which the data will be saved. Finally it provides the functions to compute (on request) the MDS and other useful metrics. As an optional feature the backend can be configured to interact with the municipality service that offers data about car accidents. In this case, the backend compares the information sent with the data stored in the database. Therefore tries to identify the potentially unsafe areas and then suggests possible interventions. The Web application takes advantage

of the interactions with the remote database offered by the backend to provide a
simple interface for the authorities in order to let them consult the stored data:
they can query all reports, filter them by some specific criterias and analyze the
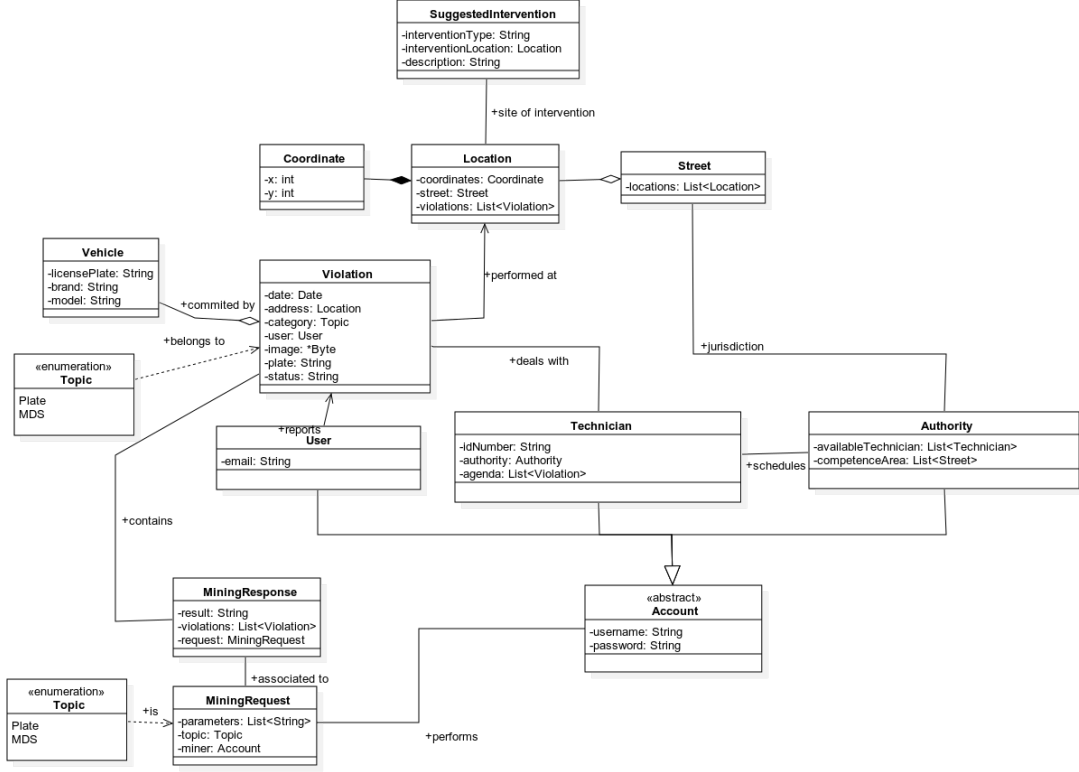data to get the statistics that they desire.

Figure 2.1: Class diagram

2.2 Product Functions

2.2.1 Send a report

This is one of the most important function of the service. After the user has
successfully registered/logged in to the application, he can choose to send a
report of a violation to the officers. This is performed by taking a picture of
the car that committed the violation. After the device took the picture the
application asks the user to choose from a list of common violations the category
that best describes the event (i.e. No parking zone, Double parking). When this
information is provided the application gathers some other data in order to have
a correct and useful report. In particular it gets:

- The data related to the user who wish to send the report (User ID);

- The local time and date gathered from the phone OS;

- The information about the street where the user took the picture (Obtained by the GPS coordinates of the user).

The user can also optionally add the car plate numbers and the name of the street in case, for example, the application misrecognized the street from the GPS coordinates. When all the necessary data are correctly gathered, the report is sent to the backend system that will handle the request. The user can then consult the history section of the application to monitor the status of its report.

2.2.2 Receiving a report - Plate recognition

The specification document states that the license plate needs to be extracted from the image in order to save it as a correct report. In order to extract the plate number from an image an external service is used. This choice was made because, for this kind of application, an established and robust, high accuracy recognition service will certainly work better than a recognition system that has to be built from scratch only for this kind of purpose. Therefore the backend system runs a task that listens for new reports and as soon as it receives a new one it immediately sends the image contained in the message to the plate recognition service. After the elaboration, the service returns the response of the recognition algorithm that should contain the text transcription of the plate. Note that if for some reason (i.e. bad picture) the plate recognition system can't recognize the plate, the backend system will instead use the plate number inserted by the user if provided; if not provided the violation won't be associated to any plate number. After this recognition step the backend system will interact with the database to create a record that contains the licence plate (if recognized/provided) and all the other information sent in the report. It is important to highlight the fact that the photo won't be discarded after the recognition as they can still be used for legal purposes.

2.2.3 Information mining

The system offers to both users and authorities the possibility to analyze the registered data although the level of visibility differs with respect to the utilizator. Users can only see the streets with the highest frequency of violations via the map present in the mobile application. Authorities instead have a wider access

to data, that is to say they can access to all registered reports. Via the Web application they can also perform queries on the database (only selection queries) to mine the desired data and compute statistics and metrics for a deeper analysis. The role based approach has been designed in order to guarantee the privacy of car owners with respect to the users of the application. Therefore they will only be allowed to consult for the MDS.

2.2.4 Scheduling technicians on violations

The system offers to LSAs the possibility to manage the violation reports by scheduling them to one or more of their technicians. Scheduled technicians will have the possibility to change the violation report status (from scheduled) to solved once they patrolled the street and confirmed/rejected the violation.

2.2.5 Crossing data with the municipality

The backend system offers the possibility to interact with an external service, offered by the municipality, that provides data regarding car accidents happened in the municipality area. SafeStreets can define the potential unsafe areas of the municipality by merging car accidents data and viaolation reports stored by the application. The system can then suggest to the local system administrator possible interventions in order to improve the safety of this areas. It is important to note that the data offered by this service need to be structured in a standard way recognizable by the backend system, otherwise the information crossing will not provide any results.

2.3 User Characteristics

2.3.1 Actors

- *Visitor*: a person without a SafeStreet account. Visitors can only have access to the homepage and the registration form of the mobile application;

- *User/ Mobile user*: a person correctly registered to the SafeStreet account service. Users/Mobile users can perform any of the actions made available by the SafeStreet mobile application;

- *Recognized authority*: a recognized authority (Police station) which submitted to SafeStreet and can interact with it through its web application interface;

- *Local system administrator/Police corporal*: a person which belongs to a recognized authority, in charge of dealing police technician accounts and scheduling patrols;

- *Police technician*: a policeman encharged of dealing with the violations reports . He/She patrols the unsafe areas and gives fines to the reported cars in case of violations;

- *Third party recognition service*: an image recognition service which allows SafeStreet to extract car plates from the violation report's images.

2.4 Assumptions and dependencies

As far as the specification document is concerned, it is necessary to specify some details and to state clearly a few ambiguous points. In order to better clarify those situations, the following assumptions are introduced.

2.4.1 Text Assumptions

- **Violation report**

  - The information sent by user in the violation report includes: the license plate image and an optional textual transcription, its position coordinates (extracted automatically from the smartphone GPS) and the violation metadata;

  - The suitable metadata descripted in the specification document is intended as a choice of the law infringment and a textual description of the event;

  - Given the coordinates of the violation, the system is able to retrieve the address from which the report was sent;

  - Every time a report is received, the system will ask the image recognition service for the license plate textual transcription;

– A license plate textual transcription is considered correct if it fits into one of the common standards of the EU states. In case of wrong textual transcription of the license plate from the image recognition service:

  ∗ The textual transcription provided by the user is taken into account. After a previous check, the provided license plate will be considered as correct;

  ∗ If the user has not sent a textual transcription, the violation is not associated to any license plate number;

- **Mining information**

  – End users are allowed to mine information about the violation reports that they sent. Also, they can have access only to the (map visualization or list) MDS;

  – Authorities can mine information concerning all the stored reports in the SafeStreet system, such as MDS or the list of cars with an high number of violation reports;

- **Suggested intervention**

  – Only authorities are allowed to have access to the suggested intervention on unsafe areas identified by the system.

2.4.2 Domain assumptions

- [D1]: A picture taken with a smartphone is performed with a quality sufficient for the image recognition service to transcribe it;

- [D2]: The reported picture contains only the license plate of the car that committed the violation and not others;

- [D3]: The GPS information collected from the smartphone of a user has an accuracy of less than 5 meters;

- [D4]: The timestamp collected from the smartphone is synchronized with the CET standard;

- [D5]: The users can only report violations occurred in Europe;

- [D6]: Reports are only sent through a secure connection channel;

- [D7]: The users reports all the violation that they detect;

- [D8]: The users send report containing correct information about the detected violation;

- [D9]: The municipality system stores correctly all the information concerning car accidents;

- [D10]: The municipality system allows SafeStreet to retrieve information about the car accident among a certain area;

- [D11]: The municipality system sends data accordingly to a common standard which is comprehensible by the SafeStreet system;

- [D12]: The image recognition service is able to detect and transcribe license plates;

- [D13]: The police corporates and technicians can manage only the violation reports that belong to their jurisdiction (competence area);

- [D14]: The police corporates can manage only the police technicians of their jurisdiction;

- [D15]: The police technicians can only take care of the violations which are assigned to them by the police corporate.

Specific requirements

3.1 External interface requirements

3.1.1 User Interfaces

This section shows a mockup version of the main functionalities of the mobile application and the web application of the final version of SafeStreets.

- Mobile application:

Figure 3.1: LogIn interface

The Figure 3.1 represents the login form of the SafeStreets mobile application. The user can login through his username and password, or his Google
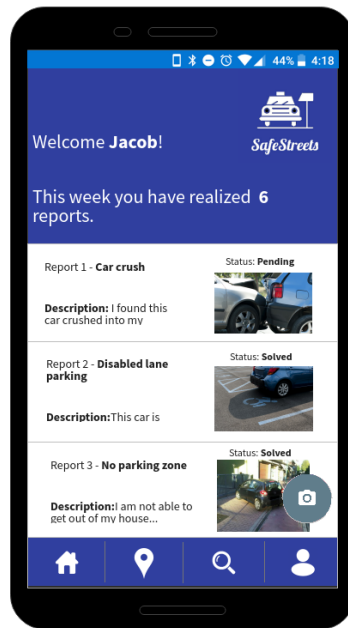
account.

Figure 3.2: Homepage

The figure 3.2 represents the homepage of the SafeStreets mobile application. The user is allowed to look at the list of the violation reports that he published, publish a new one through the floating button or visit other sections of the app.
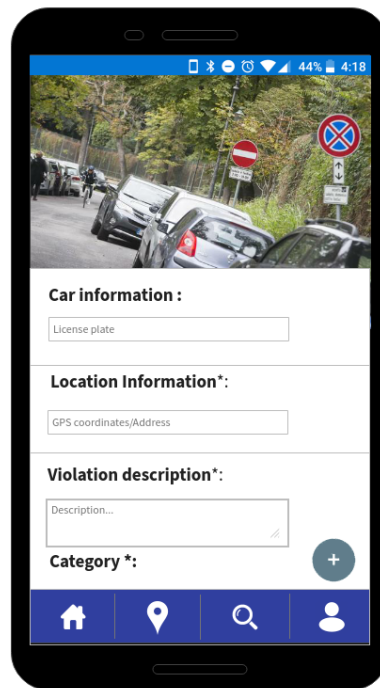
Figure 3.3: Violation report publishing page

The figure 3.3 represents the violations report publishing page of the SafeStreets mobile application. The user is allowed take a picture of the violation clicking on the camera button and fill in the form to realize the violation report.
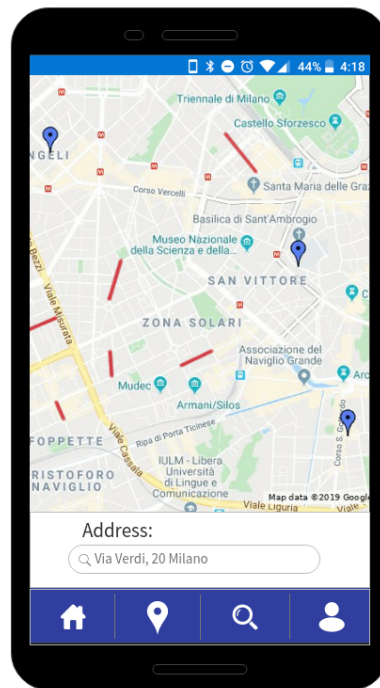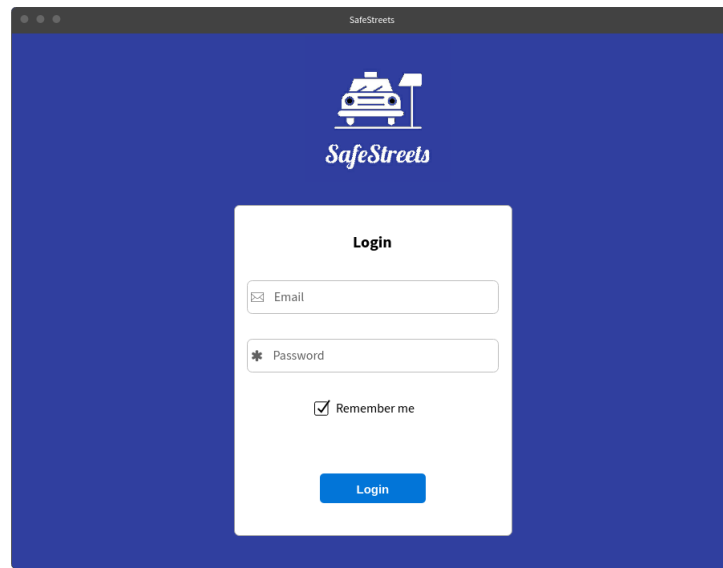
Figure 3.4: Map search page

The figure 3.4 represents the map page of the SafeStreets mobile application. The user is allowed to look at the MDS highlighted in red on the map. It is also possible to search for a specific address through the search bar.

- Web application;

Figure 3.5: LogIn interface
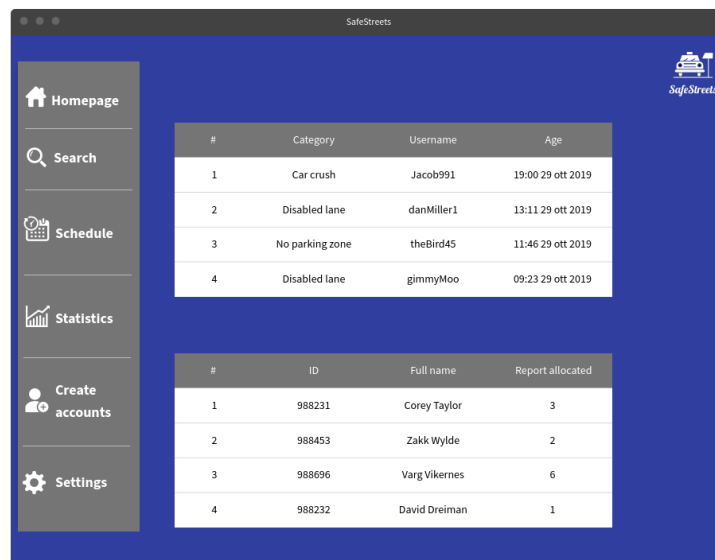
The Figure 3.5 represents the login form of the SafeStreets web application. The officer can login through his username and password.

Figure 3.6: Homepage

The figure 3.6 represents the homepage of the SafeStreets web application.

The local system administrator is able to look at the list of recent violations, the list of the registered police technicians or to visit other pages by the side menu.
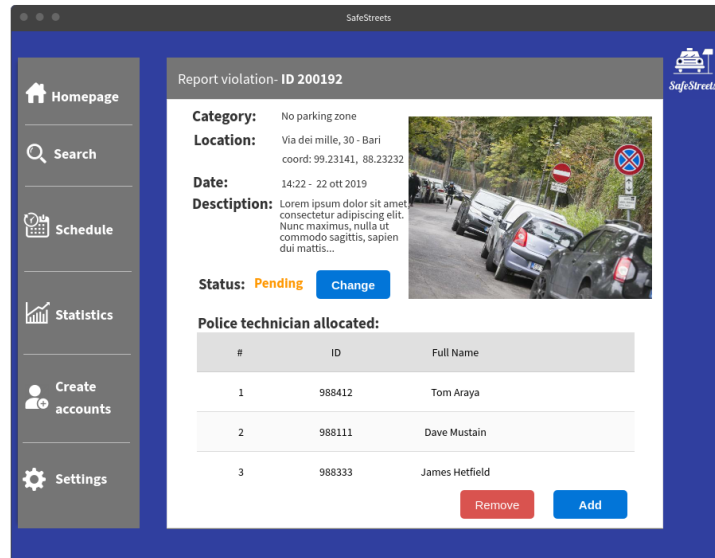
Figure 3.7: Specific violation report page

The figure 3.7 represents the specific violations report page of the SafeStreets web application. Both the LSA and the PT are able to look at the full violation report and they can change the report status from Pending to Closed (or viceversa). However, the LSA only can allocate/deallocate police technicians from this specific report.
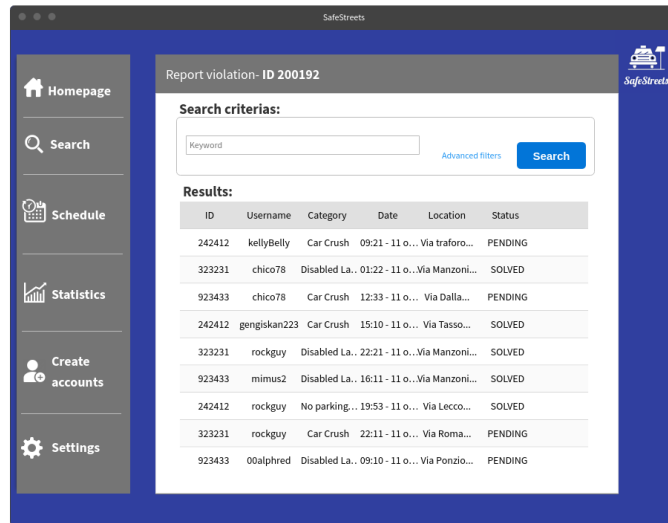
Figure 3.8: Report mining page

The figure 3.8 represents the report search page of the SafeStreets web application. The local system administrator is able to search for a set of violation applying a keyword filtering and some advanced filters, such as date range or location.

3.1.2 Hardware Interfaces

SafeStreets do not require any specific hardware device, thus there are no hardware interfaces to other existing systems.

3.1.3 Software Interfaces

SafeStreets is a standalone service and does not share any API to external systems. However, the system requires the interaction with the Image recognition service and with the municipality's legacy system to perform its tasks. Also, a map service is required in order to show the MDS positions on the mobile application.

The software requires a standard format in order to retrieve car accident. data from the municipality's system. Such standard will be specified more in details in the DD.

3.1.4 Communication Interfaces

External services are not allowed to interact actively with SafeStreets, therefore the system does not offer any communication interface to external systems.

3.2 Functional Requirements

3.2.1 Scenarios

Scenario 1

John is a very sportsman person who likes to ride his bicycle a lot. Therefore, every Sunday afternoon he decides to go for a ride along the bicycle lane near his house. After some time he sees a car parked on the bike lane that limits the passage of bicycles. Thus, John decides to open up the SafeStreets application on his smartphone and takes a picture of the car. After that he selects the "No parking zone" as a violation category and adds the car plate number to the report in order to make sure that it will get correctly processed by the officers.

Scenario 2

Albert's son, Freddy, has recently been operated to his leg. Thus, the municipality released him a pass which allows Albert to park in places reserved for people with disabilities. During the week, Albert has to take Freddy to the doctor for some checks. Sometimes, he manages to find a free parking spot in front of the hospital reserved for people with disabilities, but other times the place happens to be occupied by another car without a disabilities' pass. When this happens he needs to find another parking spot more distant from the hospital and therefore he his forced to make his son walk to the healthcare center. This can be dangerous as he has just been operated. After some weeks in this condition, Albert decides to use the SafeStreets service in order to prove if the service can be really useful: he sends a report every time he finds himself in this kind of situation. After a certain period, he notices that the parking spots are either free or regularly occupied. In order to prove the functioning of the app, Albert checks his report history and finds out that all of his requests have been solved by the police.

Scenario 3

Every morning Kevin wakes up to go to work. Around 7 o'clock he leaves home to get to the parking spot where he parked his car the day before. Once he arrives at his car location, he notices that the car parked behind his' has crushed into his rear bumper leaving a noticeable dent. He does not have time to call the police because he is in a hurry for an important meeting at work, thus he decides to immediately send a crush report by taking a picture of the unpleasant event. After that, he rapidly gets on his car in order to arrive in time at work. Once he finishes his shift, he opens up the SafeStreets app to check the status of his report. Fortunately, he finds out that his request has been taken into account by a local police officer.

Scenario 4

Marco has been invited to Luigi's graduation party organized to celebrate his master degree in Computer Science and Engineering obtained at Politecnico di Milano. Marco lives in a small city with very limited public transportation solutions. In order to get to Milan, he decides to go with his car to the event. Once he is almost close to the designated place, he starts to looking for a parking spot. Soon later, he remembers that Luigi told him to download the SafeStreets app in order to check where the most unsafe streets in the area could be. He stops the car and opens the mobile application. Then, he navigates to the map section and finds on the map the location where the party is held. The application highlights the most dangerous streets near this location so that Marco can avoid parking his car there.

Scenario 5

Leon is a newly hired police technician at the Berlin police department. For his first days of work the corporal assigned him some routine work, mainly focused on patrols and giving fines. In order to check where he needs to patrol, he logs into the SafeStreets Web application from his computer with his credentials. Once logged in, He consults the scheduled violation reports submitted by the police corporate, in order to find out their locations.

Scenario 6

Daniele is the local system administrator of the Genova police department and he is responsible of dispatching patrols. He decides that he wants to schedule the officers in a more intelligent way. In order to do that, he consults the SafeStreets report database by logging in the web application. He would like to send the more experienced officers to areas in which there is a high number of violations categorized as "crash violations" whereas he would assign regular officers to other streets with an high frequency of violation of any category. In order to find out those streets, he filters the reported violations by the web application;immediately after, the interface displays the desired results. Now he is able to schedule the patrols as he wants. Daniele would also like to communicate which are the 10 cars that committed the highest number of violations in the city to all the officers. To do that he filters once again by the SafeStreets web application searching for the 10 car plates with the highest rate of violations.

Scenario 7

Robert is a civil councillor encharged of the street's safety of the city. He would like to find out which are the possible interventions that can be performed on the city streets in order to provide a safer circulation. Robert has limited budget and he wants to perform only the paramount interventions: he asks to a friend that works in the police department of the city to retrieve some suggestions using the newly purchased SafeStreets system. Robert knows that the SafeStreets system crosses the data offered by the municipality about car accidents with its own information about violations happened in the city. The system finds out that there is a relation between car accidents and unsafe parking in zones near biking lanes.Thus,in order to limit the number of incidents/violations, the system suggests to build barriers that divide the bike lane from the street. Robert likes the idea and start to work on a plan that he will propose to his manager.

3.2.2 Use cases

Name	Sign Up
Actors	Visitor
Entry Conditions	The visitor has opened the application on his device
Event flow	1. The Visitor chooses the Sign Up option; 2. The Visitor fills the mandatory fields regarding his personal data (i.e. email, password); 3. The Visitor fills the optional fields (i.e. Name, Surname, age) 4. The Visitor confirms the registration; 5. The system saves the information and creates the User account.
Exit conditions	The User is registered.
Exceptions	1. The user was already registered. In this case the application warns the user that there is another account associated to the specified email; 2. The user does not fills all the mandatory fields. In that case the application warns the user by highlighting the empty fields.

Name	Log in
Actors	User
Entry Conditions	1. The User has opened the application on his device; 2. The User has already done the Sign Up activity.
Event flow	1. The User chooses the Log-In option; 2. The User fills the email and password fields required for the authentication; 3. The User presses the login button.
Exit conditions	The User is logged and the application homepage is displayed on the screen.
Exceptions	1. The User enters the wrong email; 2. The User enters the wrong password.

Name	Send report
Actors	User
Entry Conditions	1. The User has opened the application on his device; 2. The User has already done the Log in activity.
Event flow	1. The User presses the button to send a report; 2. The User takes a picture of the event; 3. The application displays a miniature of the picture and the location information retrieved from the phone's GPS; 4. The User chooses the category associated with the type of violation that he wants to report; 5. The User optionally adds the plate number of the car and a short description of the violation; 6. The User sends the report.
Exit conditions	1. The report has been correctly received by the back-end system; 2. The User can see the status of his report on the homepage of the application.
Exceptions	1. The User did not choose the category of violation. The application warns the user to select one type of violation; 2. The application misrecognized the location from the GPS. In this case the User can modify the location information.

Name	Watch the MDS
Actors	User
Entry Conditions	1. The User has opened the application on his device; 2. The User has already done the Log in activity.
Event flow	1. The User navigates to the map section; 2. The system displays the MDS of the surrounding area centred on the current position of the User; 3. If the User chooses a specific address then the MDS will be searched in the area around that address.
Exit conditions	The User watches the MDS around a specific area.
Exceptions	The Users inserted an non-existent address. The application warns the User that such address does not exists.

Name	Mine information
Actors	Local system administrator, Police technician
Entry Conditions	1. The Operator has the SafeStreets Web application opened; 2. The Operator have already logged in the Web application.
Event flow	1. The Operator selects the option to search information; 2. The Operator inserts his desired search criteria; 3. The system performs the query based on the inserted filters; 4. The system returns the results of the query.
Exit conditions	The Operator is provided with the desired results.
Exceptions	The Operator inserted a set of criteria which leads to no results. In that case the Web application warns the Operator with a message that says that no records were found that satisfy the requested query.

Name	Create Police Technician accounts
Actors	Local system administrator, Police technician
Entry Conditions	1. The LSA has the SafeStreets Web application opened; 2. The LSA have already logged in the Web application; 3. The LSA have the mandatory information needed to create a Police technician account.
Event flow	1. The LSA selects the "Create Accounts" section in the Web application; 2. The LSA insert the mandatory data related to the police officer account that he wants to create (i.e. email, password, badge number); 3. The LSA chooses the confirmation option; 4. The system saves the new account.
Exit conditions	The new Police technician account has correctly been created.
Exceptions	1. There exist another account with the specified information. The system warns the LSA that an account for that officer already exists; 2. The LSA does not fill all the required fields. The Web application warns the user to fill in all the required information.

Name	Assign reports to technician
Actors	Local system administrator, Police technician
Entry Conditions	1. The LSA has the SafeStreets Web application opened; 2. The LSA have already logged in the Web application; 3. The Police technicians have an account registered by the LSA.
Event flow	1. The LSA selects the "Schedule" section in the Web application; 2. The LSA chooses which violations need one or more Police technician assigned; 3. The LSA selects the officers that he wants to assign to the violation; 4. The LSA confirms the assignment; 5. The system register the schedule.
Exit conditions	The Police technicians are provided with the violations to which they are associated.
Exceptions	1. The LSA tries to assign officers to a violation with the status "Closed". In that case the application warns the LSA that the violation has already been resolved. 2. The LSA tries to associate more than 5 technicians to the same violation. The application warns the LSA that the maximum number of Officers per violation is 5.

Name	Consult assigned reports
Actors	Police technician
Entry Conditions	1. The PT has the SafeStreets Web application opened; 2. The PT have already logged in the Web application.
Event flow	1. The PT selects the "Schedule" section in the Web application; 2. The system retrieves the violation reports associated to the PT; 3. The Web application displays to the PT the list of violations to which he is assigned; 4. The PT can inspect the details of each violation; 5. The PT can modify the status of the violation. If he chooses to solve the violation the report will then be archived.
Exit conditions	The Police technician is provided with the information regarding the reports to which he is assigned.
Exceptions	There are no violations assigned to the PT. The Web application then warns the user that there is nothing on his schedule.

Name	Ask for suggestions
Actors	Local System Administrator, Police technician
Entry Conditions	1. The Operator has the SafeStreets Web application opened; 2. The Operator have already logged in the Web application; 3. The municipality offers a service to retrieve data related to car accidents.
Event flow	1. The Operator selects the "Intervention suggestions" section in the Web application; 2. The system asks to the municipality service the data about car accidents happened in the city; 3. The system crosses the data received with the violations report submitted in the city; 4. The system finds the area where there is a strong correlation between car accidents and violations; 5. The system, based on the type of violations, tries to suggest some possible interventions to do on those areas; 6. The system displays a list of interventions for each unsafe area.
Exit conditions	The Operator is provided with a list of possible suggestions.
Exceptions	1. The system cannot find a correlation between the information, therefore it is not able to calculate suggestions. In that case the application warns the Operator that no possible interventions have been found; 2. The data sent by the municipality are few compared to the violations (or viceversa) so the system cannot make accurate suggestion. In that case the application warns the Operator that not enough data were provided in order to compute possible interventions.

3.3 Requirements

- **[G1]: Allow the user to send a violation report consisting in a picture and some metadata;**

    - [R1]: The mobile application must be able to gather all the required data such as: location, date and time from the user's device;

    - [R2]: The mobile application has to allow the user to choose between different categories of violations;

    - [R3]: The mobile application has to allow the user to optionally add information about the event and the car plate number;

    - [R4]: The mobile application has to allow the user to retake the picture if desired;

    - [R5]: The mobile application must be able to interact with the back-end system to send/retrieve information;

    - [R6]: The backend system must be able to send the violation image to the plate recognition service in order to retrieve the car plate number from the image;

    - [R7]: Once the car plate number has been retrieved the system must save the report information in the database;

    - [D1]: A picture taken with a smartphone is performed with a quality sufficient for the image recognition service to transcribe it;

    - [D2]: The reported picture contains only the license plate of the car that committed the violation and not others;

    - [D3]: The GPS information collected from the smartphone of a user has an accuracy of less than 5 meters;

    - [D4]: The timestamp collected from the smartphone is synchronized with the CET standard;

    - [D5]: The users can only report violations occurred in Europe;

    - [D6]: Reports are only sent through a secure connection channel;

    - [D7]: The users reports all the violation that they detect;

    - [D8]: The users send report containing correct information about the detected violation;

- **[G2]: Allow the user to watch the history of his reports and their status;**

  - [R5]: The mobile application must be able to interact with the backend system to send/retrieve information;

  - [R8]: The backend system must retrieve the latest reports submitted by the user from the database. Once the data is gathered, the backend system must send it to the mobile application that will display them;

  - [R9]: The user must be able to consult the status associated with each report;

  - [D6]: Reports are only sent through a secure connection channel;

- **[G3]: Allow the user to watch on map areas and streets with an high frequency of violations;**

  - [R5]: The mobile application must be able to interact with the backend system to send/retrieve information;

  - [R10]: The application has to allow the user to insert an address near which to search the MDS. If no address is provided the application will instead search the MDS near the current position of the user (collected from the GPS);

  - [R11]: The backend system must calculate the MDS close to the given location;

  - [R12]: The mobile application must enlighten on the map the MDS;

- **[G4]: Allow the local system administrator of the police station to create accounts for the police technicians;**

  - [R13]: The Web application must be able to interact with the backend system to send/retrieve information;

  - [R14]: The Web application must provide a special interface for the LSA in order to perform special task reserved for his role;

  - [R15]: The Web application must allow the LSA to securely create account reserved for PT;

  - [R16]: The Web application must allow the LSA to associate the PT's badge number with his related account;

- [R17]: The system automatically generates a temporary password associated to the new account. Upon the first login, the application asks the PT to change the password;

- [R18]: The system correctly registers the new accounts and allows access to the Web application to the PT registered;

- [D14]: The police corporates can manage only the police technicians of their jurisdiction.

- **[G5]: Allow the authorities to visualize, schedule and change the status of reports submitted by the users;**

  - [R13]: The Web application must be able to interact with the backend system to send/retrieve information;

  - [R19]: The Web application allows the visualization of reports to both the LSA and the PTs via a dedicated section;

  - [R20]: The Web application allows the LSA to schedule reports to PT by associating their account to the report;

  - [R21]: The Web application allows PTs to visualize their scheduled reports;

  - [R22]: The Web application allows both the LSA and the PTs to change the status of reports (i.e. from "PENDING" to "SOLVED");

  - [D13]: The police corporates and technicians can manage only the violation reports that belong to their jurisdiction;

  - [D15]: The police technicians can only take care of the violations which are assigned to them by the police corporate;

- **[G6]: Allow the authorities to mine data to make analysis and retrieve statistics;**

  - [R13]: The Web application must be able to interact with the backend system to send/retrieve information;

  - [R23]: The Web application provides a section that can be used by both LSA and PTs to mine the data.

  - [R24]: The system must be able to collect the filter criteria inserted by the authorities and then compose a query that will be executed to retrieve the matching data;

- [R25]: The Web application provides a section that can be used by both LSA and PTs to visualize statistics/metrics about the data;

- [R26]: The system must offer some functionalities to calculate the most useful statistics/metrics related to the data;

- **[G7]: If the municipality offers a service that provides data about car accidents then the system must be able to cross this information with its owns data in order to identify possible unsafe areas;**

  - [R13]: The Web application must be able to interact with the backend system to send/retrieve information;

  - [R27]: The Web application provides a section from which the authorities can ask for suggestions;

  - [R28]: The backend system must be able to communicate and retrieve data from the municipality service at any given time;

  - [R29]: The system has to merge and find correlations between the violations and the car accidents;

  - [R30]: The system has to identify potentially unsafe areas and then estimate possible interventions based on the correlations that have been found;

  - [R31]: Once the computation is over the Web application must display the suggested interventions on the interface along with the location;

  - [D9]: The municipality system stores correctly all the information concerning car accidents;

  - [D10]: The municipality system allows SafeStreet to retrieve information about the car accident among a certain area;

  - [D11]: The municipality system sends data accordingly to a common standard which is comprehensible by the SafeStreet system.

3.4 Performance requirements

In order to promptly tackle violations that are reported, SafeStreets has to be a low latency service: police technicians have to intervene quickly, in order to fine

the transgressors before they leave the scene of the violation. Also, the system has to guarantee an efficient service to an high number of users and police technician connected simultaneously.

3.5 Design constraints

3.5.1 Regulatory policies

The user information is stored accordingly to the GDPR policies in order to guarantee the privacy of individuals.

- No data is shared with third parties for commercial purposes;

- All reported images and violation data are stored safely through encryption methods. The third party system used for image recognition can not store any information/image which identifies;

- Any additional information, such as GPS position or Camera access, is promptly asked to the user before performing the specific operation, accordingly to the Android/IOS standards.

3.5.2 Hardware limitations

- Mobile applications:

    - Any kind of modern smartphone;

    - Internet connectivity;

    - Camera;

    - GPS.

- Web application:

    - Browser (any HTTP client);

    - Internet connectivity.

3.5.3 Interface to other applications

- A certified external image recognition service deals with the car plate recognition through report's images;

- Report's information are stored in a cloud database which guarantees data encryption and information retrieving through authentication;

- The municipality information center is periodically asked to provide data about the car accidents that happen among the municipality's territory.

3.6 Software system attributes

3.6.1 Reliability

The system needs to always be online in order to correctly store and manage violation reports and to be robust in case of failures. Moreover a cloud based database can be used in order to guarantee data redundancy and fast access.

3.6.2 Availability

As previously stated, the system must always guarantee an operating service of violation reports. These requirements can be guaranteed through a virtual replication of the backend system: a possible solution could be implemented through Docker containers managed by a container orchestrion such as Kubernetes or DockerCompose. The container architecture automatically restarts containers in case of failures (fault tolerant) guaranteeing the system functioning in every situation.

3.6.3 Security

As the data transferred and stored in the SafeStreets system deals with private citizen's data, it must be guaranteed an high level of security. In order to meet the GDPR conditions, all data has to be correctly encrypted through several security methods before being sent and stored.

3.6.4 Maintainability

The software maintainability is one of the most critical aspect in the development of this software. In order to allow cheap and easy fix, SafeStreets is developed by using several design patterns that make the software flexible and scalable. Such patterns can be recognized both in the previous UML class diagram and in further documents.

3.6.5 Portability

The SafeStreets mobile application is intended to work on every mobile device, thus the technologies used for its development have to be cross-platform (such as Flutter or any other language which can generate applications for IOS and Android systems).

Formal analysis using Alloy

4.1 Overview

In this section an analysis of some critical aspects of the system is provided exploiting Alloy. The focus is on some static constraints, in particular:

- Uniqueness of usernames and every other univoque parameter of the system, such as plates, violationIds and so on;

- Users can only mine information concerning MDS and not Plates. Also they cannot explicitly query reports that has been performed by other users.

- LSAs can deal only with violation occured in their competence area and thus scheduling such violations to the technician that they manage;

- SuggestedIntervention can be generated only if at least one violation occured in the same location of the suggestion.

Other constraints are specified for keeping a steady structure of the alloy model, in order to avoid dangling entities and preserving foreign constraints.
Four different worlds have been generated in order to verify the correctness of the constraints.

4.2 Alloy code

```
--Account Parameters
sig Username{}
sig Password{}
------------------

--Topic of the Mining request
abstract sig Topic{}
```

```
one sig PlateTopic extends Topic{}
one sig MDSTopic extends Topic{}
----------------------------

--Vehicle parameters
sig Plate{}
sig Model{}
sig Brand{}
-------------------

--Account is the generic signature for every registered entity.
--User, LSA and technicians are the specific istances for each
    ↪ specific person.
abstract sig Account{
    userName: one Username,
    password: one Password
}
sig User extends Account{}
sig LSA extends Account{
    availableTechnicians: set Technician,
    competenceArea: set Street
}

sig IdTechnician{}
--The agenda is the list of the scheduled violations allocated by
    ↪  the boss to the technician.
--Also, every technician has exactly one boss and he can receive
    ↪ scheduled violation only by him
sig Technician extends Account{
    idNumber: one IdTechnician,
    agenda:set ScheduledViolation,
    boss: one LSA
}

--Parameters are generic strings that the miner can specify
    ↪ inside his request
sig Parameter{}

sig MiningRequest{
    topic: one Topic,
    parameters: set Parameter,
    miner: one Account
}
```

```
--Result can be either positive or negative.
--Violations is the resulting set of the request.
sig Result{}
sig MiningResponse{
    violations:set Violation,
    request:one MiningRequest,
    result:one Result
}


sig Coordinates{
    x:one Int,
    y:one Int
}


--One location could be placed on the intersection of more than
    ↪ one street
sig Location{
    coordinates: one Coordinates,
    streets: set Street
}
--Viceversa one street can contain more than one location
sig Street{
    locations: set Location
}


sig Date{}
sig IdViolation{}
--Violation is a generic entity defined for the state design
    ↪ pattern.
abstract sig Violation{
    id: IdViolation,
    date: Date,
    address:Location,
    user: User,
    plate: Plate
}
--The scheduler can be a LSA which competence area covers the
    ↪ location of the violation.
sig ScheduledViolation extends Violation{
    scheduler: one LSA,
    allocatedTechnicians: set Technician
}
```

```
--By default , violations  are  instanced  as  PendingViolations  ,
    ↪ which  are  all  those  violation  which  has  yet  to  be  handled.
sig PendingViolation extends Violation{}
--Solved  violations  are  archived  violations  which  have  been
    ↪ already  dealt  by  technicians.
sig SolvedViolation extends Violation{}

sig Vehicle{
    plate:Plate ,
    model:Model ,
    brand:Brand
}


--SuggestedIntervention  can  be  defined  only  over  a  location  in
    ↪ which  at  least  one  violation  occured
sig InterventionCause{}
sig SuggestedIntervention{
    location:one Location ,
    reason: one InterventionCause ,
}


--There  could  not  be  a  pair  of  accounts  with  the  same  userName.
fact differentUsernames{
    no disj a1,a2:Account | a1.userName=a2.userName
}


--There  could  not  be  a  pair  of  violations  with  the  same  violation
    ↪  id.
fact differentViolationIds{
    no disj v1,v2:Violation | v1.id=v2.id
}


--There  could  not  be  a  pair  of  vehicles  with  the  same  plate.
fact differentPlates{
    no disj v1,v2:Vehicle | v1.plate=v2.plate
}


--There  could  not  be  a  pair  of  technician  with  the  same  id  number
    ↪ .
fact differentTechnicianIds{
    no disj t1,t2 : Technician | t1.idNumber=t2.idNumber
}
```

```
--users can not make mining requests on car plates as only
    ↪ authorities are allowed to perform such queries
fact noUserPlateMining{
    all r:MiningRequest | (some u:User | r.miner=u) implies not r
        ↪ .topic=PlateTopic


}


--Users can not request other users violation in an explicit
    ↪ query
fact noOtherUsersViolation{
    all mr:MiningResponse | all m:mr.violations | mr.request.
        ↪ miner in User implies m.user=mr.request.miner
}


--All the available technician of a LSA must have as boss
    ↪ parameter the LSA himself
fact noDifferentBosses{
    all l:LSA | some t: Technician| t in l.availableTechnicians
        ↪ iff t.boss=l
}


--Only the technicians' own boss can schedule them for a
    ↪ violation
fact ownBossScheduling{
    all sv:ScheduledViolation | some t:Technician | t in sv.
        ↪ allocatedTechnicians iff t.boss=sv.scheduler
}


--At most 5 different technicians can be allocated to the same
    ↪ scheduled violation
fact noMorethanFiveTechnPerViolation{
    all v:ScheduledViolation | #(v.allocatedTechnicians)<6
}


--This function returns a (set in case of an intersection) of
    ↪ streets in which a certain violation occured
fun getStreetsFromViolation[v:Violation] : set Street{
    v.address.streets
}
--LSAs can schedule only violations that occur in their
    ↪ competence areas
fact LSACompetenceArea{
```

```
        all sv:ScheduledViolation | one s:getStreetsFromViolation[sv]
          ↪  | s in sv.scheduler.competenceArea
}


--Every location must be at least in one street and viceversa,
    ↪ every street must contain at least one location in order to
    ↪  be valid
fact LocationInStreet{
    all l:Location | some s:Street | s in l.streets and l in s.
        ↪ locations
}


--Every mining request must be associated to a response
fact OneResponsePerRequest{
    all re:MiningRequest | one rs:MiningResponse | rs.request=re
}


--A suggested intervention can be generated only if at least one
    ↪ violation occured in the same place
fact AtLeastOneViolationSuggestedIntervention{
    all si:SuggestedIntervention| some vi:Violation | si.location
        ↪ =vi.address
}


--Every id must be associated to the relative entity
fact NoDanglingIds{
    (all id:IdTechnician | some te:Technician | te.idNumber=id)
    and
    (all iv:IdViolation | some v:Violation | v.id=iv)
}


--Every password must be associated to the relative account
fact NoDanglingPassw{
    all p:Password | some a:Account | a.password=p
}


--Every InterventionCause must be associated to the relative
    ↪ SuggestedIntervention
fact NoDanglingIntCauses{
    all ic:InterventionCause | some si:SuggestedIntervention | si
        ↪ .reason=ic
}
```

```
--Every plate,model and brand of a vehicle must be associated to
   ↪ the relative vehicle
fact NoDanglingVehicleParams{
    all p:Plate | some v:Vehicle | v.plate=p
    and
    all m:Model | some v:Vehicle | v.model=m
    and
    all b:Brand | some v:Vehicle | v.brand=b
}


--Every parameter must be associated to the relative mining
   ↪ request
fact NoDanglingParams{
    all p:Parameter | some re:MiningRequest | p in re.parameters
}


--Every result must be associated to the relative mining response
fact NoDanglingResults{
    all r:Result | some mr:MiningResponse | mr.result=r
}


--
   ↪ ----------------------------------------------------------------
   ↪
pred world1{
    #Street=2
    #Location=2
    #Coordinates=2
    #Violation=2
    #Vehicle=1
    #Date=2
    #User=1
    some disj s1,s2:Street | s1!=s2 and some l1:Location | l1 in
        ↪ s1.locations and not l1 in s2.locations
    and some disj v1,v2:Violation |  v1.address!=v2.address and v
        ↪ 1.date!=v2.date
    and some disj c1,c2:Coordinates | c1.x!=c2.x and c1.y!=c2.y
        ↪ and c1.y!=c2.x and c2.x!=c1.y
}
run world1 for 2 but 0 LSA, 0 Technician
--
   ↪ ----------------------------------------------------------------
   ↪
```

```
pred world2{
    #Technician=1
    #User=1
    #MiningRequest=2
    #MiningResponse=2
    #Result=2
    #Vehicle=0
    #Brand=0
    #Date=0
    #Model=0
    some disj m1,m2:MiningRequest | some disj mr1,mr2:
        ↪ MiningResponse| m1.topic=PlateTopic and m2.topic=
        ↪ MDSTopic and
     m1.miner!=m2.miner and #m1.parameters<3 and #m2.parameters<3
        ↪  and mr1.request=m1 and mr2.request=m2
     and some u:User| m2.miner=u

}
run world2
--
   ↪ ----------------------------------------------------------------
   ↪
pred world3{
    #ScheduledViolation=1
    #Technician=1
    #Coordinates=1
    #SolvedViolation=0
    #PendingViolation=0
    #Parameter=0
    #Vehicle=1
    #SuggestedIntervention=0
    }

run world3
--
   ↪ ---------------------------------------------------------------
   ↪
pred world4{
    #SuggestedIntervention=2
    #Violation=2
    some disj si1,si2:SuggestedIntervention | si1.location!=si2.
        ↪ location
}
```

```
run world4
```

4.3 Worlds

In order to prove the correctness of the alloy model, some different real situation has been created by usign the world predicates. Here are the generated world replicas by predicates:
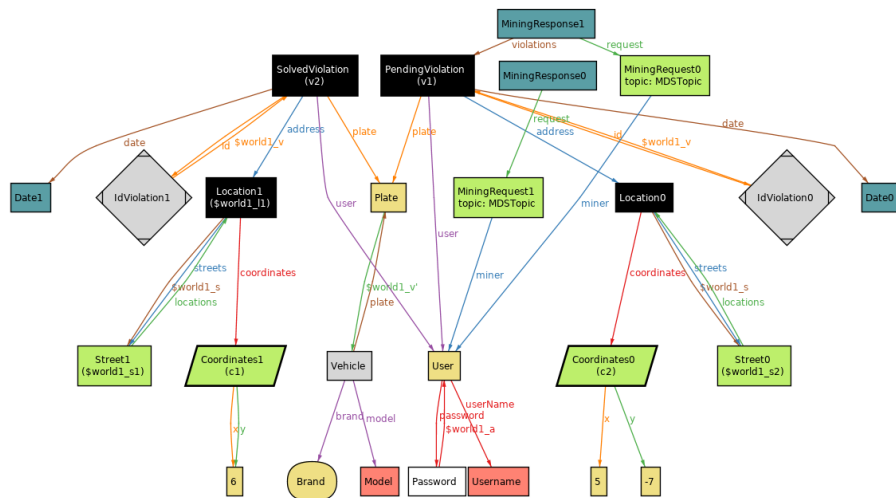

Figure 4.1: Alloy world 1

- World 1: The first world pictures the situation in which a user publishes two different violation reports in different locations on different dates. Here is possible to underline the hierarchical relations between coordinates, locations and streets and the way reports are associated to users.
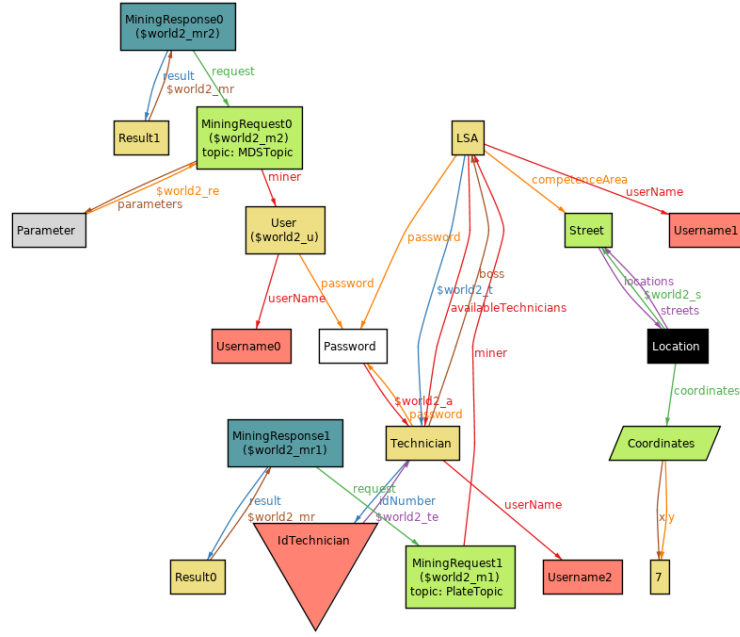
Figure 4.2: Alloy world 2

- World 2: The second world pictures the situation in which different accounts have performed mining request on different topics. Here it is underlined that user can perform only MDS requests while technicians and LSAs can also be looking for Plate topics and deeper queries. Also, the number of parameters of each mining request have been limited to 3 for view purposes.
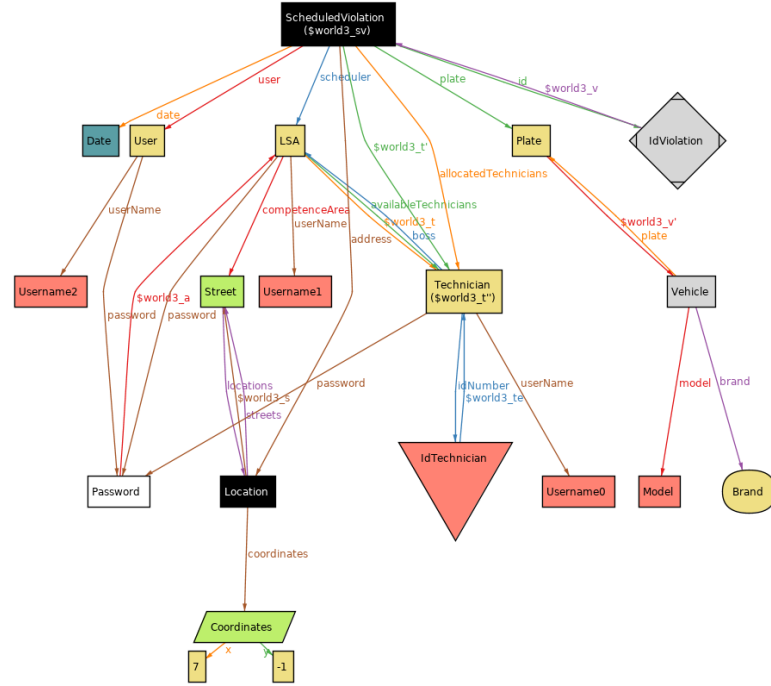
Figure 4.3: Alloy world 3

- World 3: The third world pictures the situation in which a violation has been scheduled to a technician. Here the relations between technicians, scheduled violations and LSAs are underlined.
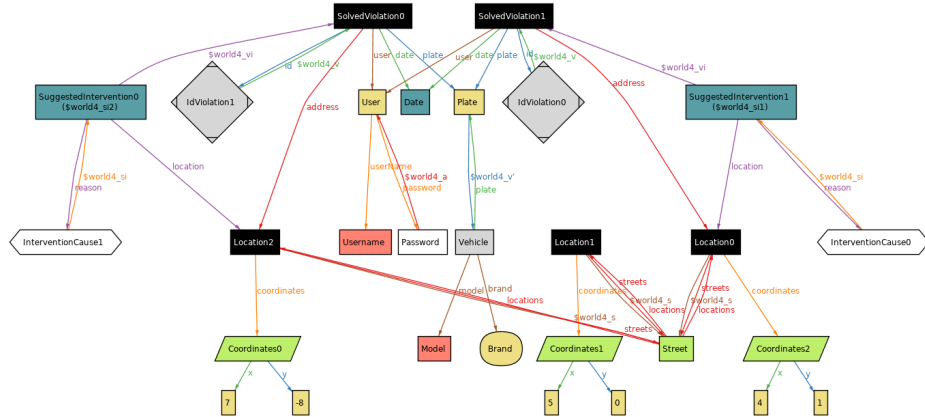
Figure 4.4: Alloy world 4

- World 4: The fourth world pictures the situation in which a suggested intervention has been generated for a specific location. Here it is visible to have a look at the relations between suggested interventions, locations and violations. More specificly, it is possible to evince the fact that a suggested

intervention can be performed on a certain location only if at least one violation occured at the same spot.

4.4   Alloy model performance

Figure 4.5: Alloy worlds performance

Effort Spent

References

- LateX Workshop extension for Visual Studio Code:
  https://github.com/James-Yu/LaTeX-Workshop/

- LateX compiler:
  https://www.latex-project.org/

- StarUML for UML diagrams:
  http://staruml.io/

- Moqups for creating Mockups:
  https://app.moqups.com/

- Alloy extension for Visual Studio Code:
  https://github.com/s-arash/org.alloytools.alloy/tree/ls

- Alloy compiler and visualizer:
  http://alloy.lcs.mit.edu/alloy/