Politecnico di Milano

AA 2019/2020

POLITECNICO
MILANO 1863

Safe Streets

RASD- Requirement Analysis and Specification Document

Version 1.1 - 15/12/2019

Authors:

Priyanka Rajendran

Sharapriyanka Vakati

Professor:

Matteo Rossi

Table of Contents

1. Introduction

1.1 Purpose

1.1.1 General Purpose

Safe Streets will act as a medium between the users and the authorities where the user notifies the authorities about the violations they come across. It is very difficult for the authorities to monitor the city for violations numerous parameters contribute to this fact. SS will help the authorities to have a better control over the city with the help of the citizens. SS will be a crowd sourced application which will use the data from multiple users and will be sent to multiple users. It will concentrate mainly on the parking violations but also collects data regarding traffic violations.

SS will receive all the information from the user regarding the violation. The SS will have to process the data received from the user to derive useful information for the authorities. The application should store all the data that are collected and processed using a suitable meta-data. The authorities should be able to receive the data regarding the violation through Safe Streets with the help of which they can take necessary action. Safe Streets will analyze its data and also provide some useful results. Both the user and the authorities will be able to access the Safe Streets data however the level of visibility will depend on the role assigned to the customers.

Safe Streets will also allow the user and the authorities to access the data related to the accidents from municipality database by using required services. Safe Streets will compare the data retrieved from the municipality service with its own data (i.e., Violation data) and identifies the areas that are potentially unsafe. After identifying the unsafe areas, it uses the analyzed data to provide useful solutions to the authorities. Such that by implementing these suggestions authorities can possibly reduce the number of violations thus transforming the unsafe areas to Safe Streets.

1.1.2 Goals

Goals are prescriptive assertions formulated in terms of world phenomena. Following are the world phenomena's that needs to be satisfied by the application.

G1: User should be able to report the parking or traffic violation to the authorities.

G2: Authorities should be able to access the data regarding the violation.

G3: User and Authorities should be able to access data from Safe Streets with limited access to the user.

G4: User and authorities should be able to access data regarding the accidents from municipality services.

G5: Safe Streets should identify unsafe areas and provide solution.

1.2 Scope

Safe Streets is aimed in reducing the traffic violations especially parking violations. The users registered with this application will be able to report violation by filling out some details regarding the violation. The application after receiving the information validates, process and stores the data. These information will be sent to the authorities who are registered with Safe Streets. Both the users and the authorities can access the data stored in Safe streets. The authorities can access all the data like the areas where the number of violations is high and the vehicles committing a greater number of violations whereas the users will have only limited access. It also helps its customers to access data regarding the accidents by providing useful services.

Safe Streets will use the data obtained from the municipality database with that of the violation data to identify the unsafe areas. It will also provide some suggestions for reducing the violations which when implemented by the authorities will reduce the number of violations. It will create a vigilance among the citizens about traffic and parking violations around the city and their consequences. It will help the authorities to get know about almost all the violations occurring in the city with the help of the public people who acts as the information source by providing information about the violations that they come across in their day-to-day life. This will also help the user to know about the incidents in the city by accessing the services provided

by the municipality to retrieve the accidents in the selected area and also the violations that are identified around the city. SS will help the users to stay alert in the areas that are marked unsafe. On the whole, Safe Streets acts as an intermediator between the user and authorities by facilitating some useful services.

Some of the machine phenomena that needs to be considered are as follows:

- The system will not highlight an area where the violations are more if no user reports the incidents in that area.
- The authority will not be able to find the vehicle violating the rules if the user did not take a picture covering the license plate or if the image quality is too low or shaky.
- The device that the user using should have a GPS with high accuracy.

1.3 Definitions, Acronyms, and Abbreviations.

1.3.1 Definitions

User: The customer of the application who provides information about the traffic violations, retrieve information from Safe Streets about the accidents occurring in the unsafe areas.

Authorities: Traffic officials who has the power or right to give orders, make decisions, and enforce obedience.

Violation: A violation is any act that fails to abide by the existing law.

Meta-data: Data that provides information about other data.

Algorithm: A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

Unsafe areas: The areas where large number of accidents occur and the area where the number of violations reported is high.

1.3.2 Acronyms

API: Application Programming Interface.

GPS: Global Positioning System.

UI: User Interface.

SS: Safe Streets.

ID: Identification number.

GDPR: General Data Protection Regulation.

1.3.3 Abbreviations

no. –number

1.4 Revision History

Version 1.0: First release.

Version 1.1: The following changes are made in the document.

- In Section 1.6, the chapter descriptions are detailed.
- Some attributes and functions are modified to make it consistent with the requirements and other attributes in the class diagram.
- In state diagram 4, the events flow through the SS application previously it was illustrated that the authorities and user can access the accident data directly which misses out SS application that acts as an intermediator.
- The State diagrams order has been changed to make the flow consistent throughout the document.
- The context of the scenarios has been modified to provide a clear understanding about the significance of SS.
- Use case diagrams and sequence diagrams are updated by removing or adding things.
- Requirement
- Some details have been modified/added throughout the document.

1.5 Reference documents

Specification document: “Safe Streets Mandatory Project Assignment A.Y. 2019-2020”.

1. IEEE standard 830-1998 IEEE Recommended Practice for Software Requirements Specification.
2. UML diagrams:
   https://www.uml-diagrams.org/
3. Alloy document:
   http://alloy.lcs.mit.edu/alloy/documentation/quickguide/seq.html/
4. Traffic rules:
   http://www.poliziamunicipale-online.it/?l=eng#/Legislation

1.6 Document Structure

This RASD (Requirements Analysis and Specifications Document) is composed of five chapters as outlined below:

Chapter 1: This section provides an introduction of the SS where the purpose and some of the requirements are stated plainly without much detail. It also lists the goals that needs to be achieved in order to satisfy all the needs. The scope section provides the aim of the project and also the machine phenomena that needs to be considered. It also has sections for providing definitions, abbreviations and acronyms about the terms used throughout the document. There is also a section to keep track of the revision history and details about the sites used as references.

Chapter 2: This section gives a better understanding of the requirements in detail and an overall description about the project. A class diagram which describes the static structure of the system is also illustrated to provide a clear understanding. Some state diagrams are also provided to define the events that takes place in the system along with their corresponding states.

Chapter 3: This section gives the detailed explanation about different scenarios of the application. Many sequence diagram has been illustrated to provide the flow between the actors and the system. All type of interfaces that needs to be implemented for this application has also been described. Explanation regarding the performance and design constraints are also detailed.

Chapter 4: This section provides the analysis of the model using Alloy Analyzer 4.0. It contains a documentation about the comments, signature definitions, facts, assertions and predicates for some of the formal verifications of the system.

Chapter 5: This section shows the effort spent by each member working for this project.

2. The Overall Description

2.1 Product Perspective

Safe Streets will be a crowdsourced and cloud-based application. The applications needs a GPS in order to send the location details to the authorities. SS uses Google map services to find the area where the violation took place. The system needs access to the device's camera to take the picture of the violation and this picture should be given as an input to a digital image processing algorithm which will find the license plate of the vehicle and extract the license number of the vehicle. The camera should be accessed through the application and the user will not be able to upload an image from their gallery. This will prevent the user from uploading an edited image. The system allows both the user and authorities to access data from SS if they are registered with the system. Role visibility will be set to prevent user from accessing all the data. The system needs to communicate with the municipality services to get the details regarding the accidents occurring in the city. The application needs to implement the required APIs provided by the municipality services. The system then runs a function to compare the data obtained from the municipality database and its own data to attain some conclusions. The system will conclude an area as an unsafe area if the frequency of accidents is higher and the number of violations reported in that area is comparatively higher than the other areas.

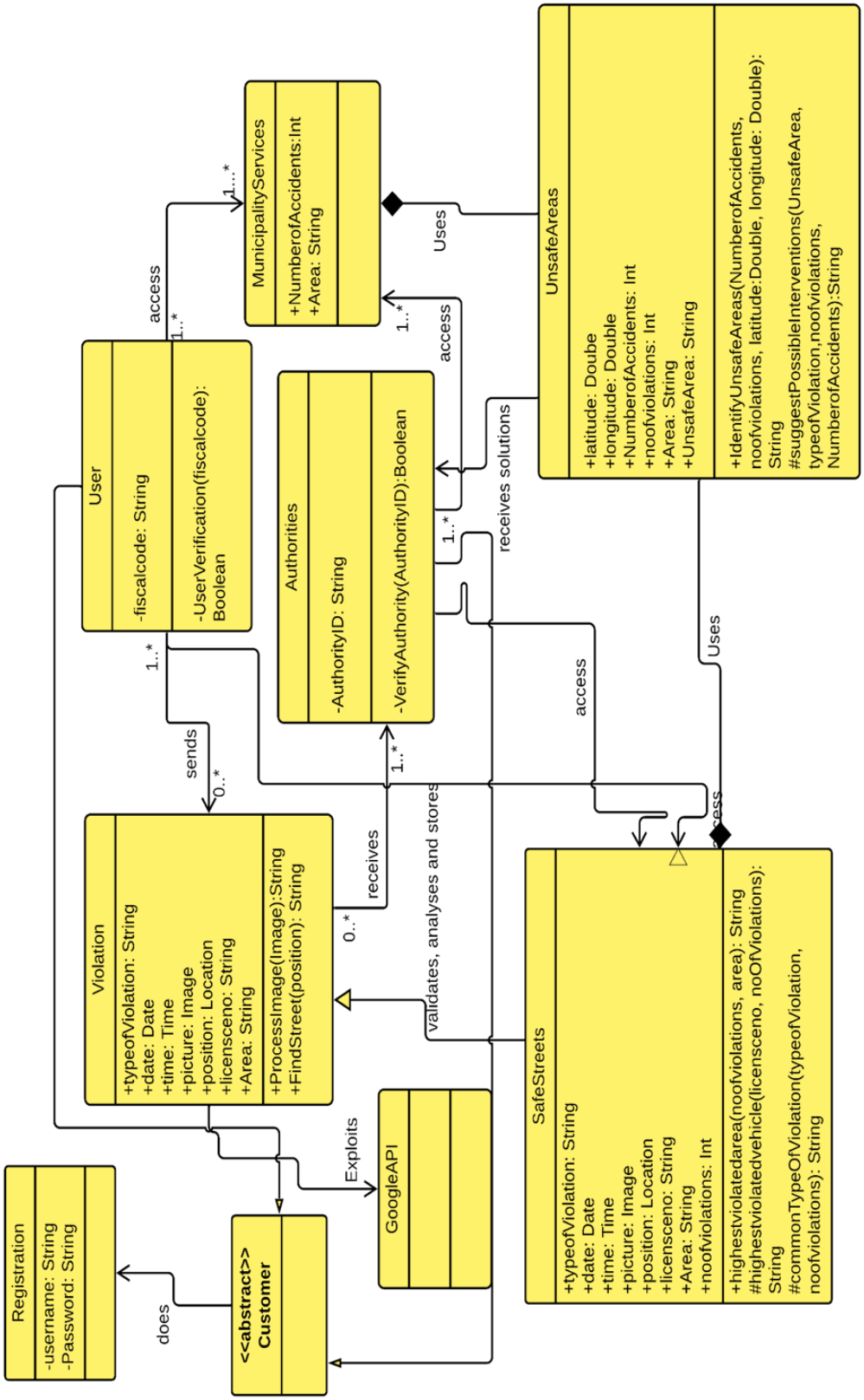


Figure 1: Class Diagram

The class diagram illustrated above shows the structure of the system that needs to be implemented. It gives an overview or a general structure of the classes along with some associated objects and possible functions that can be implemented. This will not provide details about all the classes that needs to be implemented but can give a high level perspective about the system. The user and the authorities are connected using Safe Streets. The user is identified through their fiscal code which will provide their identity. Authorities will be identified by using their authority ID. Both the users and the authorities need to register with the system after which they will be provided with a username and a password to access SS. The fiscal code and the authority ID will be verified to ensure authenticity. The user sends details regarding the violations by accessing SS from their device. These data will be handled by the Violation Class which can be accessed by the authorities. The data about the violation will be processed and stored in the associated SafeStreets class. User and authorities can retrieve data from SafeStreets according to their roles. MunicipalityServices provides data about accidents to both the users and the authorities.

UnsafeAreas class is composed of Municipality Services and Safe Streets data.

The dynamic behavior of the application is illustrated using some of the state diagrams. This will help us identify the important objects to be analyzed, states and the events. Figure 2- State diagram 1 describes the flow from the user sending the data to the authorities receiving it.

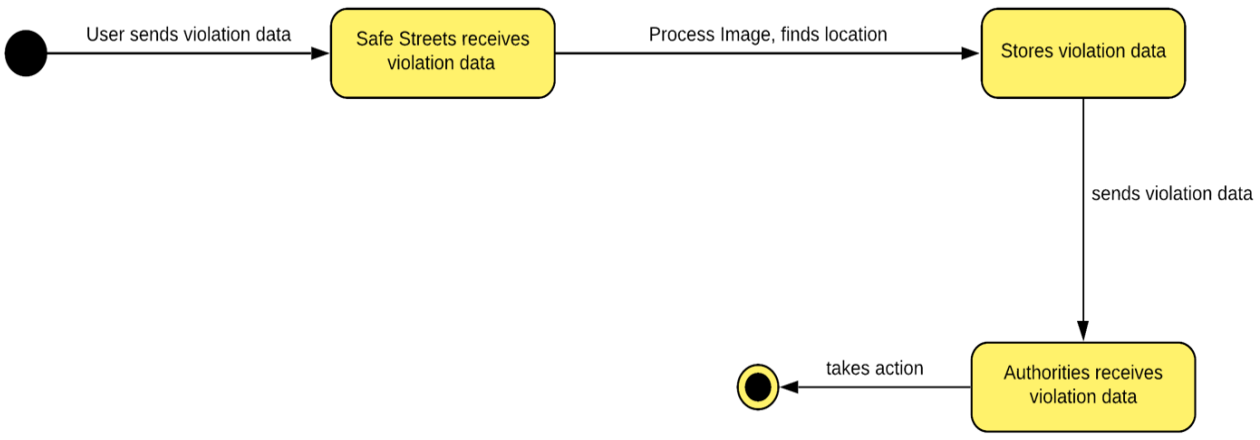


Figure 2- State diagram 1: User reports violation data and authorities takes action.

In the diagram below, Figure 3- State diagram: 2, the diagram represents the actions involved in accessing the data from SS. Access to the data will be provided according to the roles.

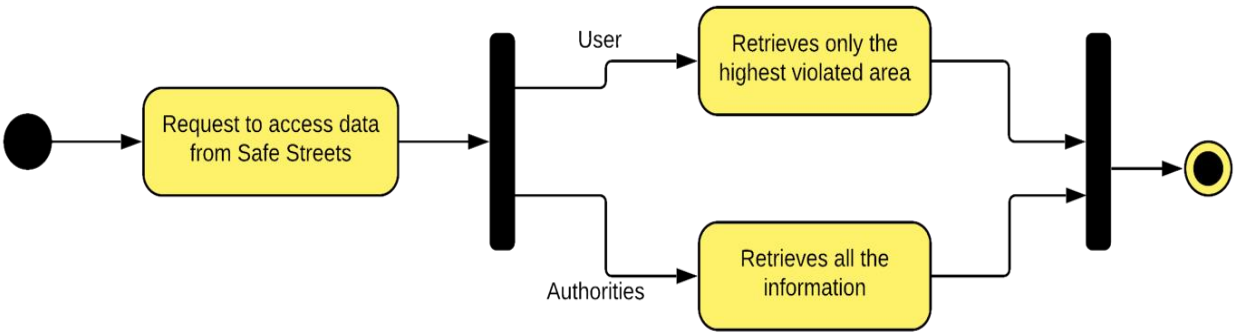


Figure 3- State diagram 2: Accessing data from Safe Streets

In the below diagram, Figure 4- State diagram 3, the events involved in accessing the data about accidents from the municipality services is described.

Figure 4- State diagram 3: Accessing municipality data.

In the below diagram, Figure 5- State diagram: 4, the events involved in identifying the unsafe streets by receiving the data from municipality services and SS and also sending the authorities the solutions by analyzing the data.

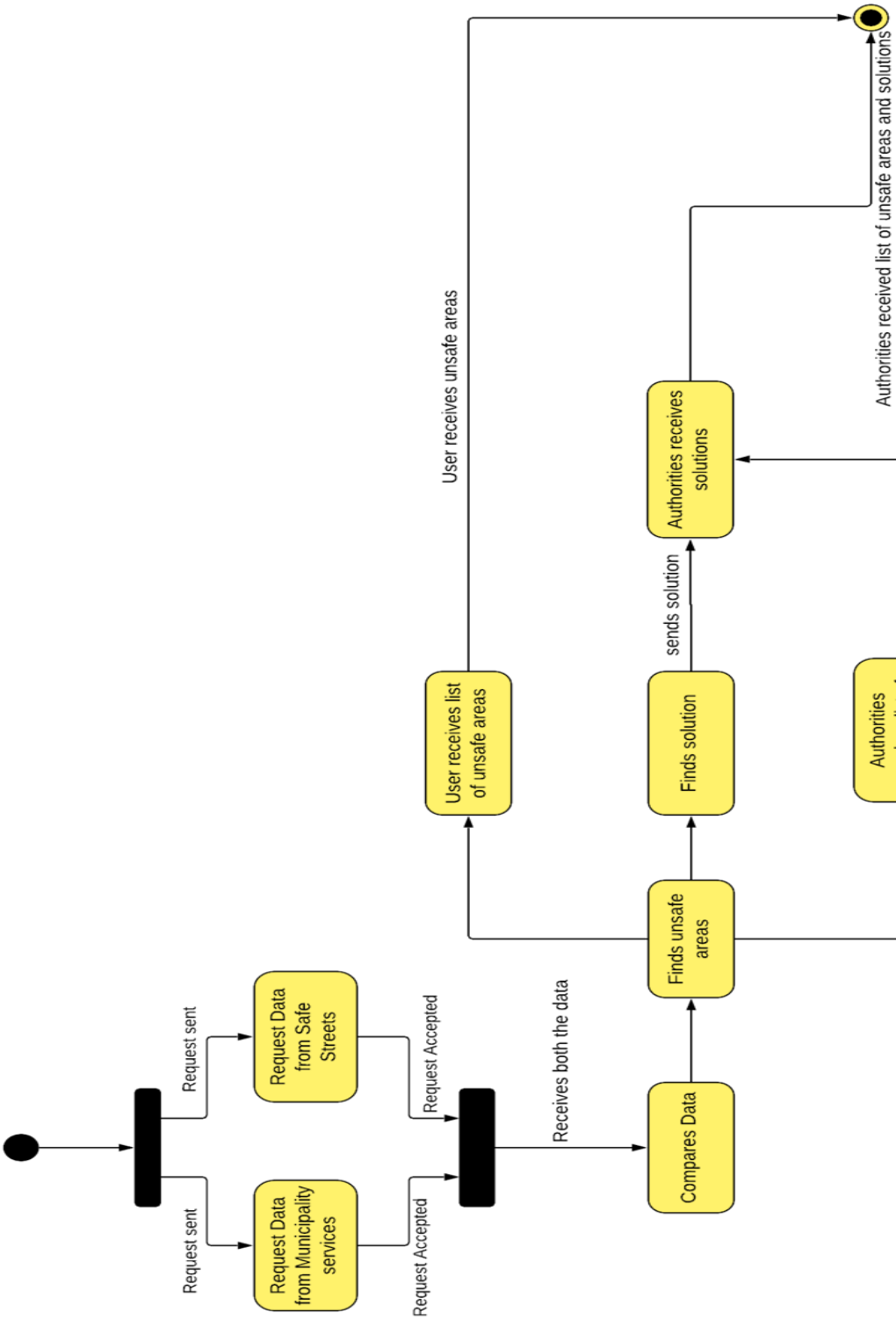


Figure 5- State diagram 4: Finding unsafe areas.

2.2 Product Functions

The important product functions involved in the system are described briefly in the sections below.

2.2.1 Reporting Violation and accessing data from Safe Streets

This is the most important function and is the main objective of SS. The system will allow the user to send the details of violation after once their registered details with SS are verified. Each user will be identified uniquely using their fiscal code. So it is mandatory to enter the fiscal code while registering with SS. This is important to avoid anonymous reporting of violations. Also this will create a vigilance among the users not to report scenarios which is not at all a violation. On the other hand authorities will be asked to register with their authority ID which is also unique and SS confirms whether the ID belongs to the particular authority or not. The user whenever sees a vehicle violating parking rules or traffic rules have to just capture the image of the vehicle with the license plate of the vehicle visible. The camera to capture violation can be accessed only from the application. User will not be able to upload an image from the gallery which will prevent the user from uploading an edited image.

Some of the possible type of parking/traffic violations that can be reported are as follows:

1. Parking vehicles on the edge of the roadway.
2. Parking vehicles opposite to the prescribed sign.
3. Double parking.
4. Vehicles that are parked with the payment certificate for parking (ticket) not visible.
5. Vehicles parked in the end of the driveway.
6. Vehicles parked in the place for taxis, buses, ambulances, police vehicles, fire brigadiers, garbage vehicles, areas allocated for loading/unloading goods.
7. Parking in the area reserved for people with disabilities.
8. Vehicles parked in the area indicated as No Parking or in limited parking after prohibited time.
9. Vehicles parked within 5 meters to the intersection of the roadway cross.
10. Vehicles parked on the sidewalks, in the area for pedestrian and bicycle lane.

11. Leaving the vehicle parked in the parking area to the parking meter payment by exposing ricarcabile vehicle is not running.
12. Breach of road signs art.
13. Vehicles exceeding the prescribed speed limit.

When the user submits the type of violation and the image, other details like location and date and time will also be sent to the SS. The SS then process the image to identify the license plate and uses the location data to identify the area where the violation took place. These data will also be sent to the authorities who can access it by logging into SS. SS then stores all the processed data reported by the users in its database.

SS also allows the users to view the list of highlighted areas. SS highlights an area if the no. of violations in that particular area is more than 10 in a week. Authorities can access all the data stored in the SS such as the violations reported by all the users, the highlighted areas, highlighted vehicles, and the most common type of violations. Some of the data will be the output of SS after analyzing the raw data from the user.

The user and the authorities are allowed to access the data from SS regarding the accidents occurring in a particular area. The Safe Streets application retrieves these information from the municipality database by implementing an API for the municipality services. It then provides the fetched results to the user and the authorities.

2.2.2 Classifying Unsafe areas

Safe Streets identify the unsafe areas in the city by comparing the data retrieved from municipality database and the data from its own database. The system will conclude an area as an unsafe area if the frequency of accidents in a particular area is higher than other areas and the no. of violations reported in that particular area is higher. The threshold for an area to be considered safe keeps changing according to the new data.

For example if an area has a frequency of 10 accidents per week and some 4 to 5 violations being reported by the users in which 7 cases there are no injuries, in two or three incidents the victims has a light injury, in two or three incidents the victims has serious injuries and in two cases it caused the death of a victim. Then it will be labelled as an unsafe area. All other areas data will be compared with the area labelled unsafe. If it is likely to have at least half of the parameters of

the unsafe area then it will also be marked as an unsafe area. Some more conclusions like the road in which most the violations or accidents occur, in which time the frequency is higher will also be analyzed by SS and it will provide a solution in order to transform an unsafe area to Safe Streets which is the ultimate goal of SS application. These solutions will be sent to the authorities who are responsible to implement those solutions provided. The list of unsafe areas will be sent to both the user and authorities.

2.3 User Characteristics

The actors of the application are as follows:

1. **User:** A person who is a common public registered with Safe Streets.
   The role of the user in each of the product functions will give a clear differentiation based on the function.
   - **Reporting Violation and accessing data from Safe Streets:** The user will report the incidents that are violations. They will also be able to some data from SS and some through SS.
   - **Classifying unsafe areas:** The user will be able to access the list of unsafe areas.
2. **Authorities:** A person who is a traffic official and will be responsible to enforce laws and take actions.
   The role of the authorities in each of the product functions will give a clear differentiation based on the function.
   - **Reporting Violation and accessing data from Safe Streets:** The authorities will be able to retrieve the data sent by the user, can take actions then close the violation. They will be able to access data from SS and some of the processed and analyzed data from and through SS.
   - **Classifying unsafe areas:** The authorities will be able to access the list of unsafe areas along with corresponding solutions to transform unsafe areas to SS.

2.4 Constraints

The application is constrained by the system interface to the GPS system within the mobile phone compatible with running this application. Since there are multiple system, different

Operating systems and multiple GPS manufacturers, the interface will most likely not be the same for every one of them. The Internet connection is also a constraint for the application. Since the application fetches data to and from the cloud database over the Internet, it is crucial that there is an Internet connection for the application to function. So, both the user and the authorities should have a device with a 2G/3G/4G or at least a WiFi connection.

2.5 Assumption and Dependencies

Some of the domain assumptions include:

D1: Every user of the system has a fiscal code that is unique.

D2: Every authority has a unique authority ID.

D3: User has a mobile phone equipped with a GPS, camera and an internet connection.

D4: Safe Streets can access the location and date and time from the user's device automatically.

D5: Authorities should have a computer with an internet connection.

D6: The devices on which the services are exploited can provide real time information.

D7: Unsafe areas are characterized with the highest number of violations and accidents.

The application will always be used on a device that have enough performance. The device should have enough hardware resources available to run seamlessly. Another assumption is that the GPS components in all phones work in the same way. If the phones have different interfaces to the GPS, the application need to be specifically adjusted to each interface and that would mean the integration with the GPS would have different requirements than what is stated in this specification. The user’s device will be a smartphone with a camera. It is also assumed that the user knows the traffic and parking rules well. SS will be developed in compliance with GDPR regulation.

3. Specific Requirements

3.1 External Interfaces

3.1.1 User Interfaces

The user interface for the software should be compatible to any browser such as Google Chrome, Internet Explorer, Mozilla or Safari by which user or the authorities can access to the system.

Detailed mockups describing all the components in the UI will be presented in the Design Document.

3.1.2 Hardware Interfaces

The system has no hardware interface. The physical GPS is managed by the GPS application in the mobile phone and the hardware connection to the database server is managed by the underlying operating system on the mobile phone and the web server.

3.1.3 Software Interfaces

The system doesn't provide any API to external applications to use its data. The application communicates with the GPS application in order to get geographical information about where the user is located and with the database in order to get the information from Safe Streets.

3.2 Functional Requirements

3.2.1 User

Scenarios

Scenario 1:

Marco, a student of polimi who is working as a delivery boy in just eats company as a part time job. He uses his bicycle for delivering the food to the customers. One day when he was attending a food delivery near to crossroads junction found a parking violation. The car was parked near to the junction of the roads. This will lead to accidents since the driver emerging from or turning into the road may not get a clear view of what is ahead of the road. He opens his Safe Streets application and clicked a photo of that car's license board and uploads it. Thanks to the

Safe Streets application it helps the authorities to take an immediate action as soon as possible such that preventing accidents in that junction.

Scenario 2:

One fine weekend two friends, Joel and Sam went out to have fun. They went to a bar in Piazzale Loreto. They both enjoyed a lot in the bar drinking and dancing. To reach their home they have to take a bus from piazzale Loreto. While walking towards the bus stop they found a vehicle in the end of the street that is parked in the no parking area. Sam took a picture of that car covering license number board and uploaded it to Safe Streets application and entered all the detail regarding the violation. This notification is received by the authorities and they took immediate action by removing the vehicle from the No Parking area and the owner of the ticket is accused.

Use Cases

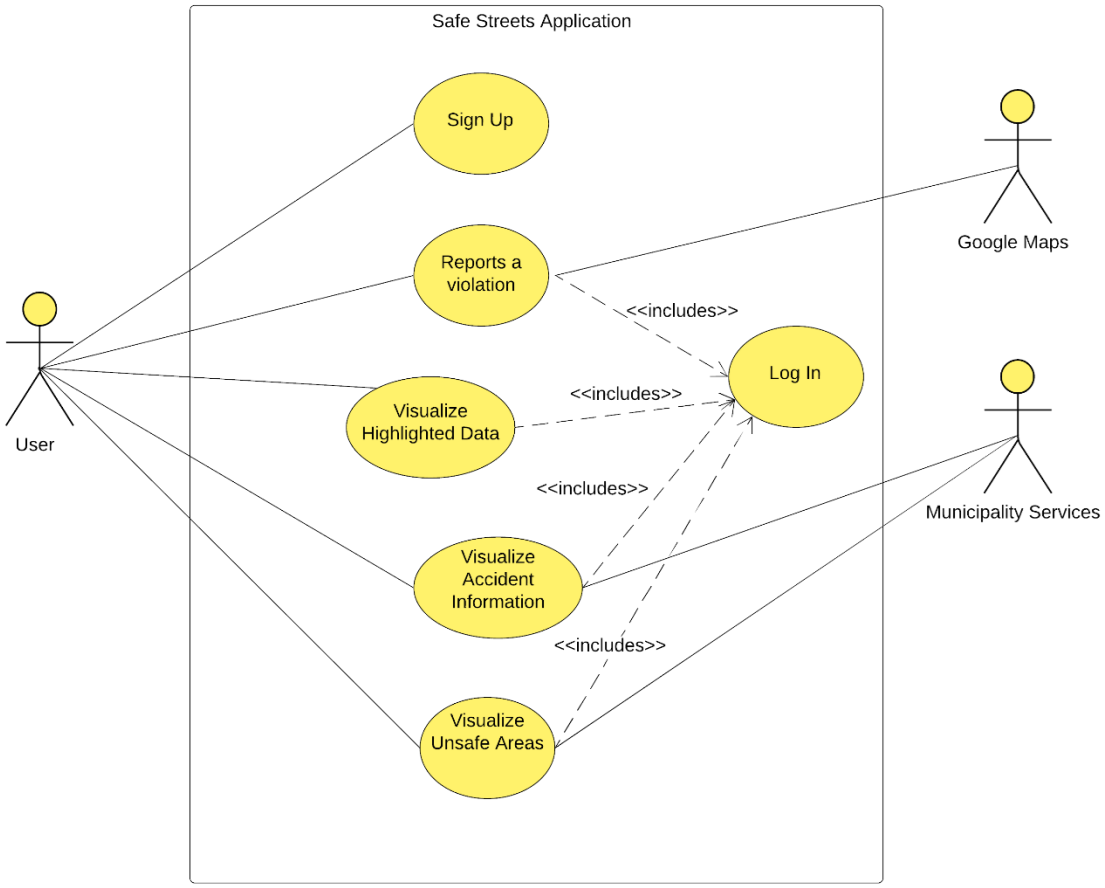


Figure 6: Use case diagram_User

Name	**Sign Up**
Actor	User
Entry Conditions	The user must open this application on his/her device and the user should be new to the application.
Events Flow	1. The user needs to choose the "Sign Up" option. 2. The user must fill all the mandatory fields (fiscal code). 3. The user can fill the data in the optional fields as well. 4. The system saves the data.
Exit Conditions	The user should have entered the correct data and SS must store it preserving integrity.
Exceptions	1. The user already registered in this case Safe Streets should display a warning message and ask him/her to login. 2. The username is already taken so it warns to select any other username to continue. 3. The password may not satisfy all the mandatory conditions so it will ask the user to create a password which satisfies everything.

Name	**Login**
Actor	User
Entry Conditions	1. The user must open SS on his/her device. 2. The user should have already signed up.
Event Flows	1. The user chooses the login option. 2. The user must enter username and password correctly. 3. User clicks on Log In button.
Exit Conditions	The user must login if the details entered by him/her are correct and should be directed to the home page.
Exceptions	1. The user enters the wrong username or password. In this case the system must warn the user to enter the correct details and try logging in again.

Name	**Report a Violation**
Actor	User
Entry Conditions	The user must have already done the login activity.
Event Flows	1. User will be directed to the Report Violation page by default. If the user is on some

	other page, then he/she can click on the Report Violation from the menu. 2. The user clicks the picture of the violation and uploads. 3. The user will be directed to the next page where a drop-down box is displayed to choose the type of violation. There are some mandatory fields. A list of possible violations that can be reported and an option to access camera clicking on which the camera will be opened through which they can capture the picture of the vehicle violating the parking/traffic rules can be uploaded. 4. An optional field describing the violation can be filled by the user.
Exit Conditions	1. The user data along with the location, date and time will be sent to the system.
Exceptions	If the user missed out filling any of the mandatory field user will be notified with a warning message.

Name	**Visualize Highlighted Area**
Actor	User

Entry Conditions	The user must have already done the login activity.
Event Flows	1. The user clicks on the Highlighted Area option from the menu. 2. A page with a list of highlighted areas where the violation is high along with the name of the area and no. of violations is displayed.
Exit Conditions	The user is provided with the requested data.
Exceptions	\

Name	**Visualize Accident Information**
Actor	User
Entry conditions	The user must be logged in using his credentials on a device.
Events Flows	1. Click on the Accident Information button. 2. The user will be directed to the page where they have to enter an area and search.

	3. The accident information associated with that area will be displayed.
Exit conditions	The user is provided with the requested data.
Exceptions	\
Name	**Visualize Unsafe Areas**
Actor	User
Entry conditions	The user must be logged in using his credentials on a device.
Events Flows	1. Click on the Visualize unsafe areas. 2. The user will be directed to the page which displays the list of unsafe areas.
Exit conditions	The user is provided with the requested data.
Exceptions	\

Sequence Diagram:

The sequence diagram shows the actions involved in reporting a violation. It involves user and safe streets. It also uses google maps for getting the location details.

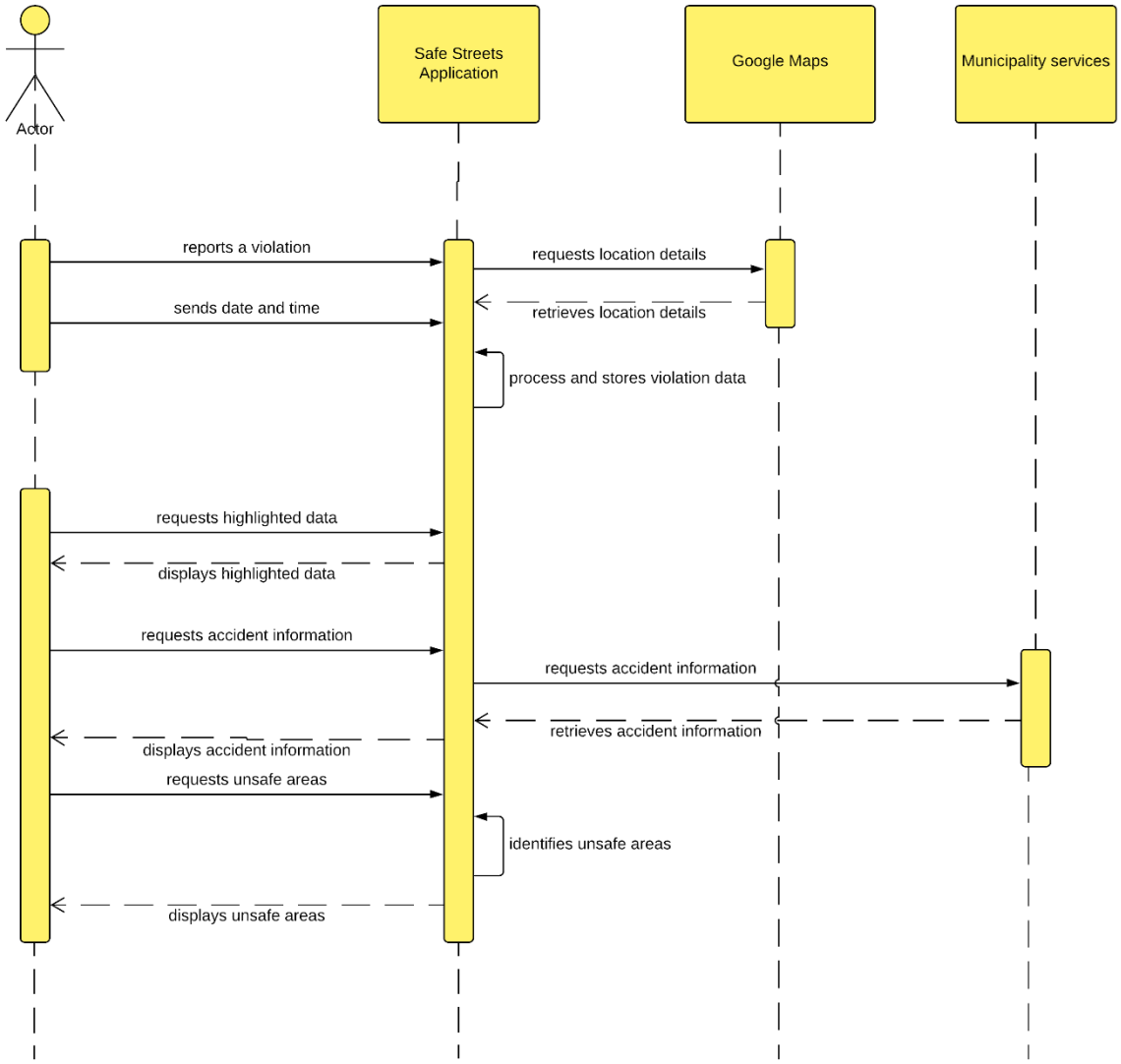


Figure 7: Sequence diagram- User

3.2.2 Authorities

Scenarios

Scenario 3:

Neeta, who is working as a traffic violation monitor in the traffic control office opens Safe Streets app to see the violations submitted by the user. It is a parking violation in Viale Romagna, the license number is 160117 MI and the date and time is also displayed. The user mentioned in the description that the vehicle is parked in a paid parking area without displaying the receipt paid

in the front side of the vehicle. Neeta then checks the area from where the violation is identified and through the traffic camera installed in that street, she confirmed the violation and necessary actions were taken. She then closes the violation.

Scenario 4:

Luca, who is working as a traffic violation monitor in the traffic control office received a violation notification in which it is reported that a vehicle is parked in a footpath. This is creating a mess and Luca immediately took necessary action to remove the vehicle from the footpath thus helping the pedestrians. He also fined the vehicle owner. Thanks to Safe Streets the authorities cleared the congestion and helped the public.

USE CASES

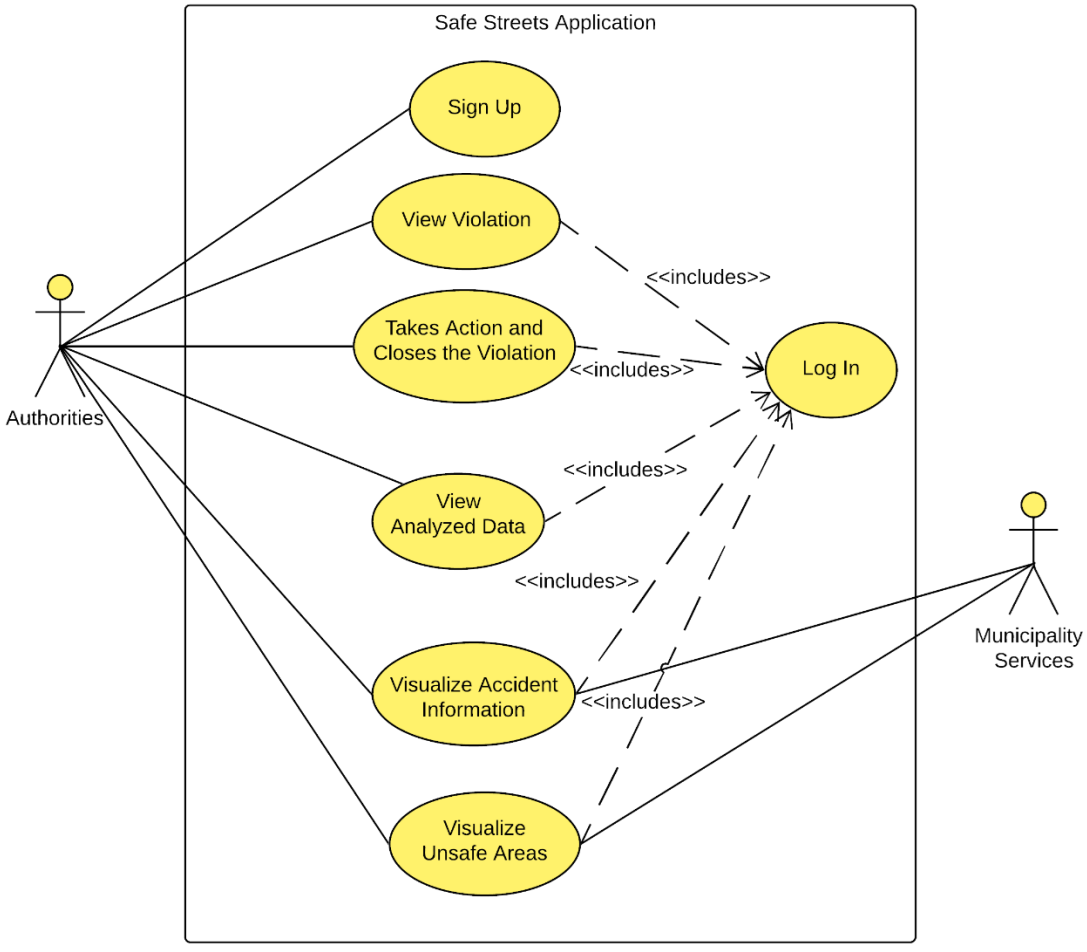


Figure 8: Use case diagram- Authorities

The Login and sign up use cases are identical to ones described for the user so they are not repeated here.

Name	**View Violation**
Actor	Authority
Entry Conditions	1. The Authority must login into the application. 2. User have reported some violation.
Event Flows	1. In the home page, click on View Violation button. 2. The authority will get the information about the violation.
Exit Conditions	1. The authority will check whether it is a violation or not. 2. The authority will take necessary action. 3. The authority will close the issue once it has been solved.
Exceptions	\

Name	**View Analyzed data**
Actor	Authority
Entry Conditions	1. The Authority must login into the application.
Event Flows	1. In the home page, click on Analyzed data button. 2. Authorities will be directed to the page where all the details regarding the area with highest

	violations, the vehicle committing more violation and the most common type of violation in the city.
Exit Conditions	1. The Authority can see the analyzed data to arrive at some useful conclusions
Exceptions	\

Name	**Visualize Accident Info**
Actor	Authorities
Entry conditions	The authorities must be logged in using his credentials on a device.
Events Flows	1. Click on the Accident Information button. 2. The user will be directed to the page where an area needs to be entered and searched. 3. The number of accidents in that area is displayed.
Exit conditions	The authorities will be provided with the requested data.
Exceptions	\

Name	**Visualize Unsafe Area**
Actor	Authority

Entry Conditions	1. The Authority must login into the application.
Event Flows	1. In the home page, click on Unsafe Area button. 2. The authorities will be directed to the page where the list of unsafe areas identified, number of accidents in that area, frequency of accidents and all the data related to the unsafe areas. 3. A solution for converting it into a Safe Street is also provided.
Exit Conditions	The Authority can see the solution provided by the Safe Streets to convert unsafe areas to Safe Streets
Exceptions	\

Sequence Diagram:

The sequence diagram shows the interaction between the authorities, Safe streets and municipality services.

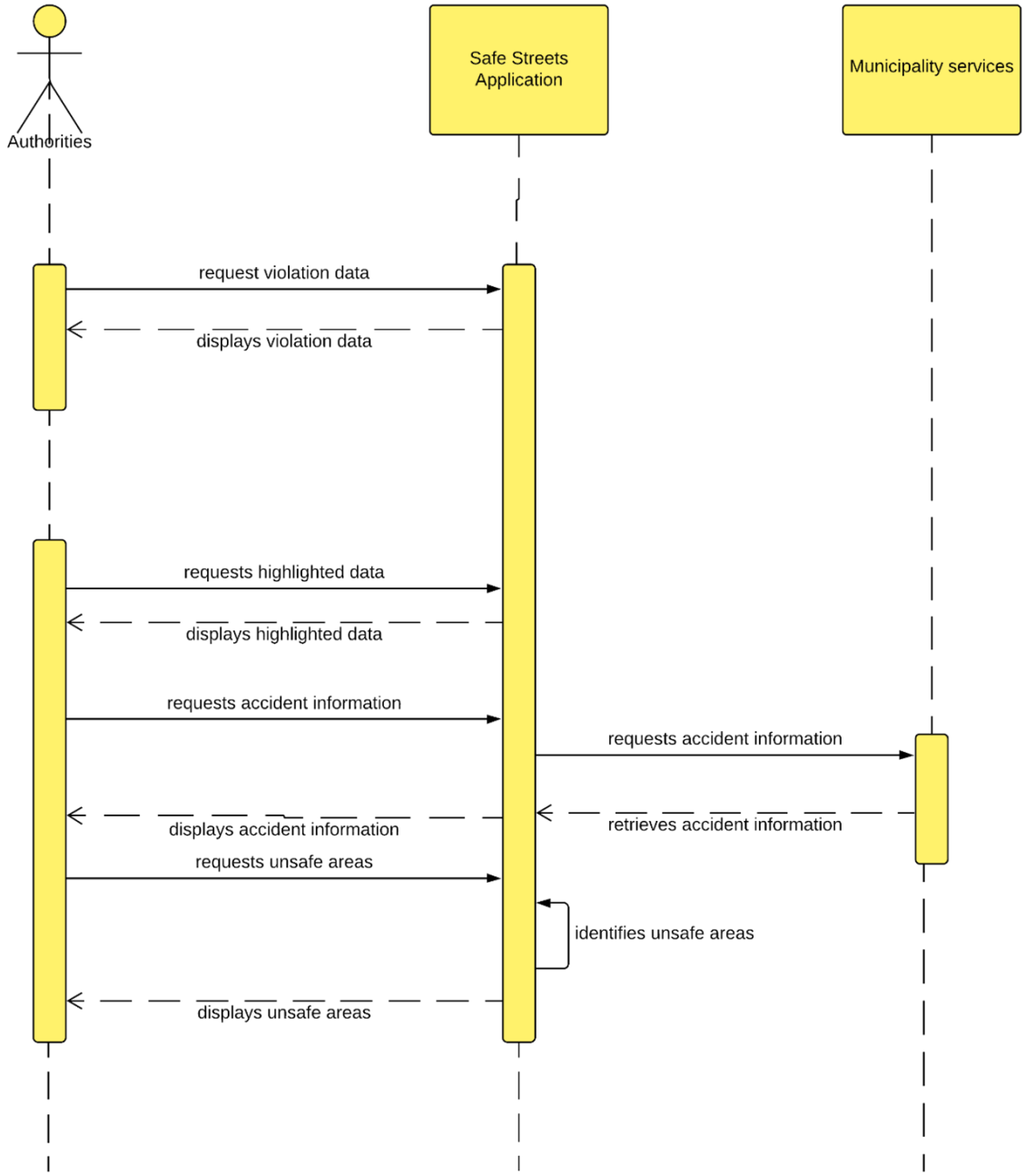


Figure 9: Sequence diagram- Authorities

3.2.3 Requirements

G1: User should be able to report the parking or traffic violation to the authorities.

D1: Every user of the system has a fiscal code that is unique.

D3: User has a mobile phone equipped with a GPS, camera and an internet connection.
D4: Safe Streets can access the location and date and time from the user's device automatically.

R1: User should be able to upload the image with vehicles violating the rules. The system should not allow users to upload images from Gallery to prevent data manipulation rather it should access the device's camera through the application.

R2: The system should allow the user to select the type of violation and provide a description about the violation.

R3: The system should be able to access the location, date and time from the user's mobile automatically.

G2: Authorities should be able to access the data regarding the violation.

D2: Every authority has a unique authority ID.

D5: Authorities should have a computer with an internet connection.

D6: The devices on which the services are exploited can provide real time information.

R4: Authorities should be able to access all the data regarding the violation like type of violation, location and date and time.

R5: Authorities should be able to access the processed data such as the license no. from the image submitted by the user and the area where the violation occurred.

R6: After taking actions, authorities should close the issue. Such that investing time on the same violation that has been reported and solved already is reduced.

G3: User and Authorities should be able to access data from Safe Streets with limited access to the user.

D6: The devices on which the services are exploited can provide real time information.

R7: The application should allow both the users and the authorities to retrieve information regarding the area with highest violation recorded.

R8: The application should allow only the authorities to retrieve data regarding the vehicle that is committing most violation and the most common type of violation.

G4: User and authorities should be able to access data regarding the accidents from municipality services.

D6: The devices on which the services are exploited can provide real time information.

R9: The user and the authorities should be able to access the data regarding the accidents from municipality services through Safe Streets.

R10: The municipality services should provide the data regarding the no. of accidents in each area.

G5: Safe Streets should identify unsafe areas and provide solution.

D7: Unsafe areas are characterized with the highest number of violations and accidents.

R11: Safe Streets should be able to compare the data from the municipality services with its own data.

R12: An area will be characterized as an unsafe area if the frequency of accidents and the no. of violations reported in that particular area is higher than other corresponding areas.

R13: The threshold for an area to be considered safe keeps changing according to the new data.

R14: After each update of the data, it will be compared with the area labelled unsafe. If it is likely to have at least half of the parameters of the unsafe area then it will also be marked as an unsafe area.

R15: Information regarding the road in which most the violations or accidents occurred, the time in which the frequency is higher should also be provided by SS.

R16: Safe Streets should suggest solutions based on the analyzed data.

R17: The solutions formulated will be sent to the authorities who will be considered responsible for implementing the solutions provided.

R18: Safe Streets should also provide expected results when the solutions are implemented.

3.2.3.1 Traceability Matrix:

In the below Requirement Traceability Matrix or RTM all the requirements are associated with its goals as well as the use cases specifying the scenarios. It is important to emphasize that the use cases that are illustrated are associated with the requirements. However some of the use cases that are indirectly related to each requirement are not listed to have a clear view. For instance, the R1 which says that the Safe Streets should act as an intermediary between the user and the authorities is applicable to almost all the cases. So it will not be indicated for all the goals. It will be mentioned under goals that has high dependency on the requirement. Hence we focus on the most specific things related to each requirement.

Raw ID	Goal ID	Requirement ID	Use Case ID
r1	G1	R1	Report a violation
r2	G1	R2	Report a violation
r3	G1	R3	Report a violation
r4	G1	R4	View violation
r5	G2	R5	View violation
r6	G2	R6	View violation
r7	G3	R7	View Highlighted area
r8	G3	R8	View Analyzed Data
r9	G4	R9	Visualize Accident Info
r10	G4	R10	Visualize Accident Info
r11	G5	R11	Visualize unsafe areas
r12	G5	R12	Visualize unsafe areas
r13	G5	R13	Visualize unsafe areas
r14	G5	R14	Visualize unsafe areas
r15	G5	R15	Visualize unsafe areas
r16	G5	R16	View solutions
r17	G5	R17	View solutions
r18	G5	R18	View solutions

3.3 Performance Requirements

The system should be able to serve a great number of users and authorities simultaneously. It should guarantee reactive and correct responses.

3.4 Design Constraints

3.4.1 Standard compliance

The system adopts precise units of measure:

Date (mm/dd/yyyy)

Time (hr:min:sec)

With regards to the privacy of data, since the application processes sensitive ones, the entire project is subject to the General Data Protection Regulation [GDPR], a regulation in EU law on data protection and privacy for all individuals within European Union [EU] and the European Economic Area [EEA].

3.4.2 Hardware Limitations

The user's and the authorities device should be equipped with a 2G/3G/4G or a WiFi connection. The application should have been allocated at least a limited amount of space to run seamlessly.

3.4.3 Any Other Constraint

The system must respect privacy policies, in particular the privacy of the users so that there will be unique numbers of users using this application. The authorities should be given access only to read the data provided by the users but are not supposed to do any modifications at any cost.

3.5 Software System Attributes

3.5.1 Reliability

The system must be able to run continuously without any interruptions. In order to do that, it must be ensured that the system is fault tolerant. For example, the central server, which contains the data, should be duplicated, just like the running processes, which provide the services. Some techniques, like the FloodSet algorithm, can be adopted to ensure the required reliability.

3.5.2 Availability

Since this system is not a Safety Critical system so an availability of 99.9 is enough for this type of application.

3.5.3 Security

The data provided by the user contains sensitive information, so the aspect is primary importance. The central database on which the data reside must be protect by all the necessary measures to avoid any external and internal attack and also to handle malfunctions of the hardware. For the purpose of sending the data, encryption technique must be used in order to guarantee for privacy and consistency.

3.5.4 Maintainability

The development of the application must be done so that in the future it will be easy to fix and modify it, according to the circumstances, and also in order to let cost of these operations be cheap. Appropriate design patterns will be used, as it will be explained in further document.

3.5.5 Compatibility

The application offers multiple services and can be used by a variety of people (it's a quite heterogeneous application) that is why it must be compatible to as many devices and technologies as possible, in order to meet the constraints contained in "Hardware and Software interfaces" section.

4. Formal Analysis Using Alloy

In this section some of the critical aspects of the system is checked using the alloy analyzer 4.0. Some of the static constraints are focused in specific.

- No two customers should have the same username or that of the same fiscal code or authority id.
- Each violation should be assigned to only one authority.
- Highlighted areas contain the most number of violations.
- Unsafe areas are identified using the number of accidents and the number of violations in each area.

It's important to note that some numerical values that were too big for having a correct analysis in Alloy have been reduced.

//every customer can have only one account registered.

```
abstract sig Customer{
registration: one Registration,
}
// defining some of the signatures.
sig Fiscalcode, AuthorityID, Username, Password {}
sig Picture, Date, Time {}
sig MunicipalityService{}
// data associated with violation
sig Violation{
        tv: set String,
        d: one Date,
        p: one Picture,
        t: one Time,
        loc: one Position,
        Area: set String,
        lno: set String
}
//for each registration there is one username and password
sig Registration{
        username: one Username,
        password: one Password
}
//parameters that are associated with user
sig User extends Customer{
        fiscalcode: one Fiscalcode,
        uv: set Violation
}
//associating user with one fiscal code
fact{
all fc: Fiscalcode| some u: User|fc in u.fiscalcode
}
```

```
//checking whether the fiscal code is unique
fact UniqueFiscalCode
{
no disj u1,u2: User| u1.fiscalcode=u2.fiscalcode
}
//All Usernames have to be associated to a Registration
fact UsernameRegistrationConnection{
all u: Username | some r: Registration | u in r.username
}
 //All Passwords have to be associated to a Registration
fact PasswordRegistrationConnection{
all p: Password | some r: Registration | p in r.password
}
fact RegistrationCustomerConnection {
all r: Registration | some c: Customer | r in c.registration
}
//Every Customer has a unique username
fact NoSameUsername {
no disj c1,c2: Customer | c1.registration.username = c2.registration.username
}
//parameter associated with authorities
sig Authorities extends Customer{
authorityID: one AuthorityID,
receivedViolation: set Violation
}
//associating authorities with one authority ID
fact{
all aid: AuthorityID| some au: Authorities|aid in au.authorityID
}
//checking whether the authority ID is unique
fact UniqueAuthorityID
```

```
{
        no disj au1,au2: Authorities| au1.authorityID=au2.authorityID
}
sig SafeStreet{
        ss: lone Int,
        Area: set String
}
sig Position{
        latitude: one Int,
        longitude: one Int
}
{
//{latitude>=-90 and latitude<=90 and longitude>=-180 and longitude<=180}
// values are scaled for simplicity
latitude>=-2 and latitude<=2 and longitude>=-4 and longitude<=4
}
assert AuthDiffViolation{
        no disj a1, a2: Authorities| some v: Violation| v in a1.receivedViolation
and v in a2.receivedViolation
}
check AuthDiffViolation for 4
sig HighlightArea extends Violation{
        ha: String
}
{
ha=Area iff #Violation>2
}

sig MunicipalityData {
        NoOfaccidents: set Int
}
```

```
sig UnsafeArea extends HighlightArea{
        harea: set HighlightArea,
        md: lone MunicipalityData,
}
{
        all ua: String | ua= Area iff md.NoOfaccidents>5
}
pred SS{
 #Violation=5
}
run SS for 5
```

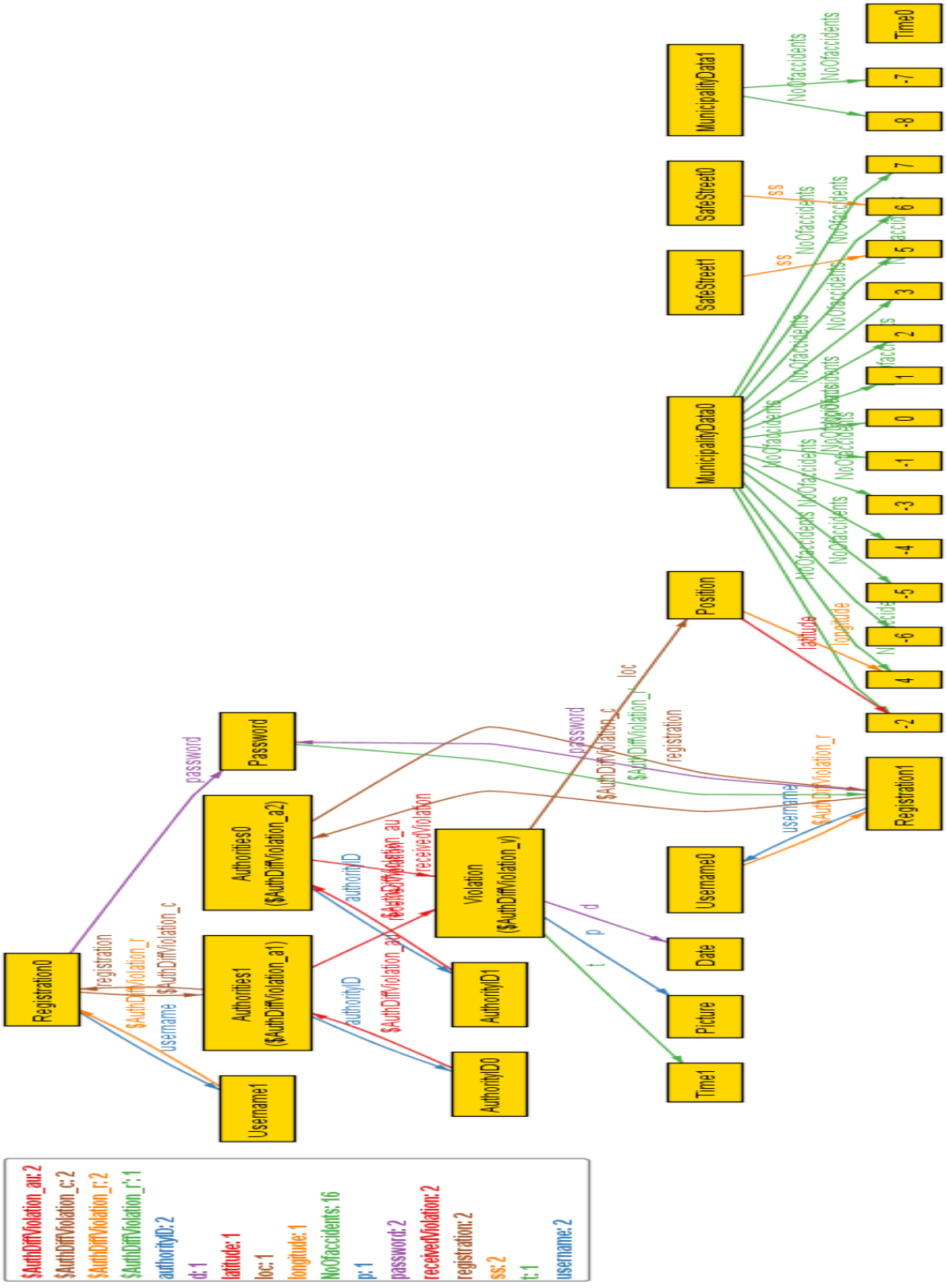


Figure 10: World 1: Analysis using Alloy

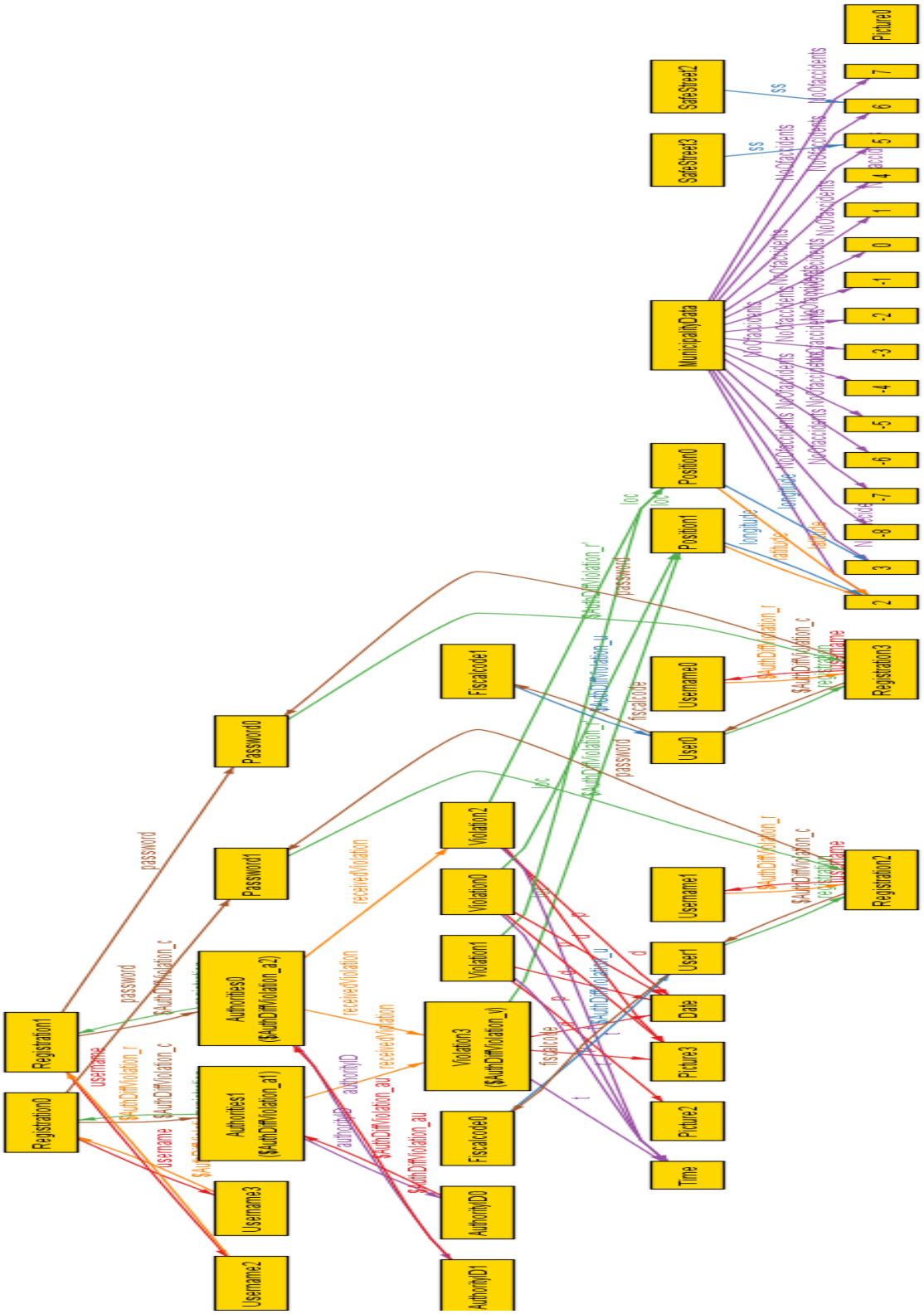
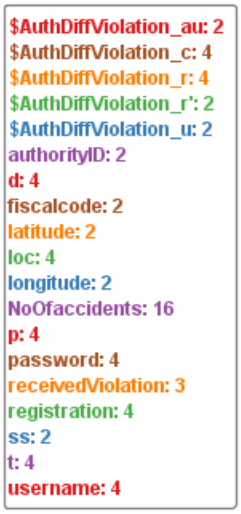
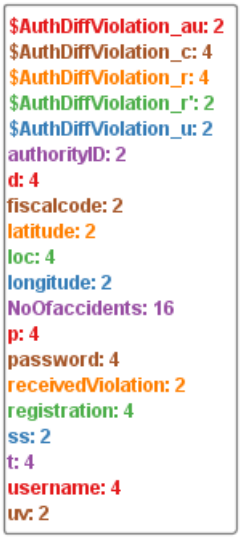

RegistrationO
Registration1
Authorities0
($AuthDMViolation_a2)
Authorities1
($AuthDMViolation_a1)
Username2
Username3
Password0
Password1
FiscalCode1
AuthorityID0
AuthorityID1
Violation0
Violation1
Violation2
Violation3
($AuthDMViolation_v)
FiscalCode0
User0
User1
Username0
Username1
Position0
Position1
Date
Time
Picture2
Picture3
MunicipalityData
SafeStreet2
SafeStreet3
Picture0
Registration2
Registration3
password
username
receivedViolation
authorityID
loc
fiscalCode
latitude
longitude
NoOfAccidents

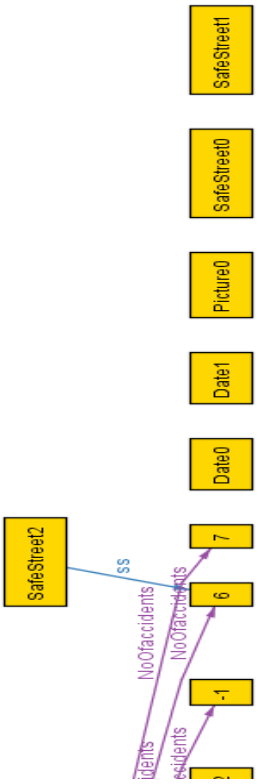
Figure 11: World 2: Analysis using Alloy



For World 2



For World 3



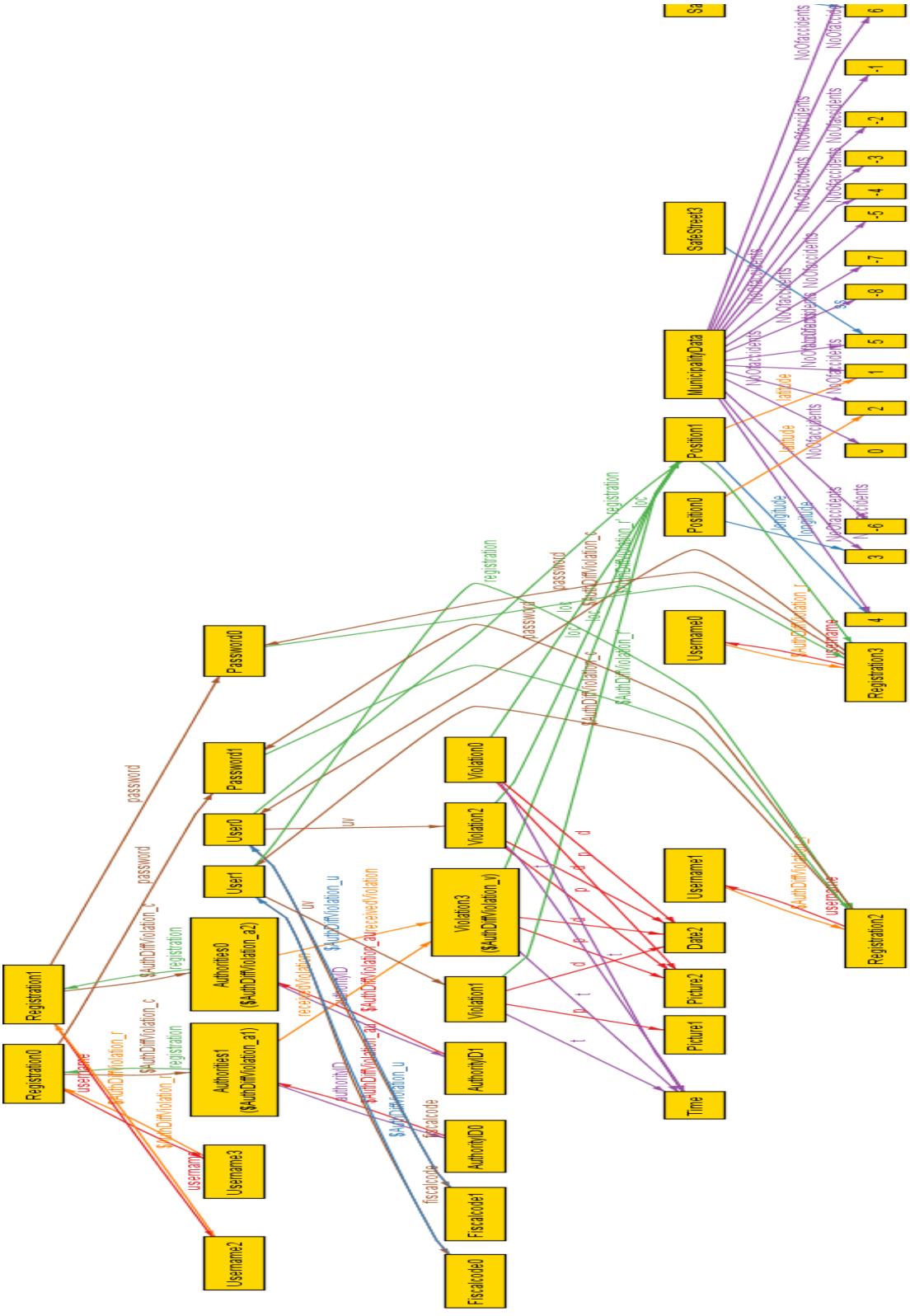


Figure 12: World 3: Analysis using Alloy

5. Effort Spent

ST1

Description of the task	**Hours**
Introduction	6
Product Perspective	8
Product Functions	13
Domain assumptions	3.5
External interface requirements	2
Functional requirements	8
Non-functional requirements	0.5
Formal analysis using Alloy	16

ST2:

Description of the task	**Hours**
Introduction	5
Product Perspective	3
Product Functions	4.5
Domain assumptions	5
External interface requirements	10
Functional requirements	6
Non-functional requirements	2.5
Formal analysis using Alloy	4