

Politecnico di Milano
AA 2019/2020



POLITECNICO
MILANO 1863

Safe Streets

RASD- Requirement Analysis and Specification Document

Version 1.0 - 10/11/2019

Authors:

Priyanka Rajendran
Sharapriyanka Vakati

Professor:

Matteo Rossi

Table of Contents

1. Introduction	4
1.1 Purpose	4
1.1.1 General Purpose	4
1.1.2 Goals	4
1.2 Scope	5
1.3 Definitions, Acronyms, and Abbreviations	5
1.3.1 Definitions	4
1.3.2 Acronyms	6
1.3.3 Abbreviations	6
1.4 Revision history	6
1.5 Reference Documents	6
1.6 Document Structure	6
2. Overall Description	7
2.1 Product Perspective	7
2.2 Product Functions	11
2.3 User Characteristics	11
2.4 Constraints	12
2.4 Assumptions and Dependencies	13
3. Specific Requirements	13
3.1 External interfaces requirements	13
3.1.1 User Interfaces	13
3.1.2 Hardware Interfaces	13
3.1.3 Software Interfaces	13
3.2 Functional requirements	13
3.2.1 User	13
3.2.2 Authorities	18
3.2.3 Requirements	18
3.3 Performance Requirements	18
3.4 Design Constraints	20
3.4.1 Standards Compliance	20
3.4.2 Hardware Limitation	21
3.4.3 Any Other Constraint	21
3.5 Software System Attributes	21
3.5.1 Reliability	21
3.5.2 Availability	21
3.5.3 Security	21

Requirement Analysis and Specification Document

3.5.4 Maintainability	21
3.5.5 Compatibility	21
4. Formal Analysis Using Alloy	22
5. Effort Spent	23

1. Introduction

1.1 Purpose

1.1.1 General Purpose

Safe Streets is a crowd sourced application which works in collaboration with the public and the authorities. It mainly concentrates on the parking violations and also on traffic violations. It acts as a medium between the users and the authorities where the user notifies the authorities about the violations they come across. Safe Streets receives all the data regarding the violation such as the type of violation, date and time, location and a picture describing the violation.

The system after receiving the image of the violation from the user it then runs an algorithm to get the license plate of the vehicle which will be very important for identifying the vehicle breaching the traffic rules or parking rules. It will also find the area where the violation takes place using the location provided by the user. The application will store these data using a suitable meta-data. The authorities can then receive the data regarding the violation and takes the necessary action. Safe Streets analyzes its data and will also provide some useful results. Such as the areas where the rate of violation is high and highlighting those areas to intimate the danger to the users as well as the authorities. Authorities can also access the data regarding the vehicles committing more violation, number of violations in each area, violations reported by each user and so on.

Both the user as well as the authorities can have access to the municipality services through Safe Streets. These services provide all the data regarding the accidents occurring in an area. Safe Streets will compare the data retrieved from the municipality service with its own data and identifies the areas that are potentially unsafe. After identifying the unsafe areas, it uses all the data it has to provide some solution to the authorities. Such that by implementing these suggestions can possibly reduce the number of violations thus transforming the unsafe areas to Safe Streets.

1.1.2 Goals

The goals of the project include:

G1: The application must act as a medium where the user can report the traffic violations especially parking violations to authorities.

G2: The system should allow the user to send the details of the violation such as type of violation, location, date and time, picture of the violation.

G3: The system should be able to process and then store these data using suitable meta-data.

G4: The system should send the data regarding the violation to the authorities.

G5: The user and the authorities should be able to retrieve data about the accidents occurring in the specified area using municipality services.

G6: The system should allow both the user and the authorities to access its data where the user is granted access to limited data.

G7: The application must be able to identify the unsafe areas by comparing the data from municipality services with its own data.

G8: The application should suggest some solutions to the authorities to reduce the accidents and violations in the unsafe areas identified.

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 send the details of the violation by filling out some details to proceed with reporting the violation. The application after receiving the information validates, process and store 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.

Safe Streets will be able to access the data from the municipality services. Using these data the safe streets will 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 violation. 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 source in providing these information 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. This will make the users stay alert in the areas that are marked unsafe. On the whole Safe Streets acts as an intermediary 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.

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.

1.5 Reference documents

1. Specification document: “Safe Streets Mandatory Project Assignment A.Y. 2019-2020”.
2. IEEE standard 830-1998 IEEE Recommended Practice for Software Requirements Specification.
3. UML diagrams:
<https://www.uml-diagrams.org/>
4. Alloy document:
<http://alloy.lcs.mit.edu/alloy/documentation/quickguide/seq.html/>
5. 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 to the purpose where 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.

Chapter 2: This section gives a better understanding of the requirements and an overall description about the project. A class diagram which describes the static structure of the system is also illustrated to get a clear view. 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 of the scenarios. Many sequence diagram has been illustrated to provide the flow between the actors. All type of interfaces that needs to be implemented for this application has also been described. Explanation regarding to 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 the comments, signature definitions, facts, assertions and predicates for some of the aspects of the system in a documentation.

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 transfer 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. 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 editing the 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 APIs provided by the municipality services. The system then runs a function to compare the data obtained 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 than the other areas and also the number of violations reported in that area is more compared to other areas.

The class diagram illustrated below shows the model of the system that needs to be implemented. It gives an overview or a general structure of the classes used along with some objects and possible function. This will not provide details about all the classes that needs to be implemented. 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 they will be given a username and a password to access SS. The fiscal code and the authority will be verified to provide authenticity. The user sends details regarding the violation from their device. These data will be in Violation Class which can be sent to the authorities. The data in 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 is composed of MunicipalityServices and SafeStreets data.

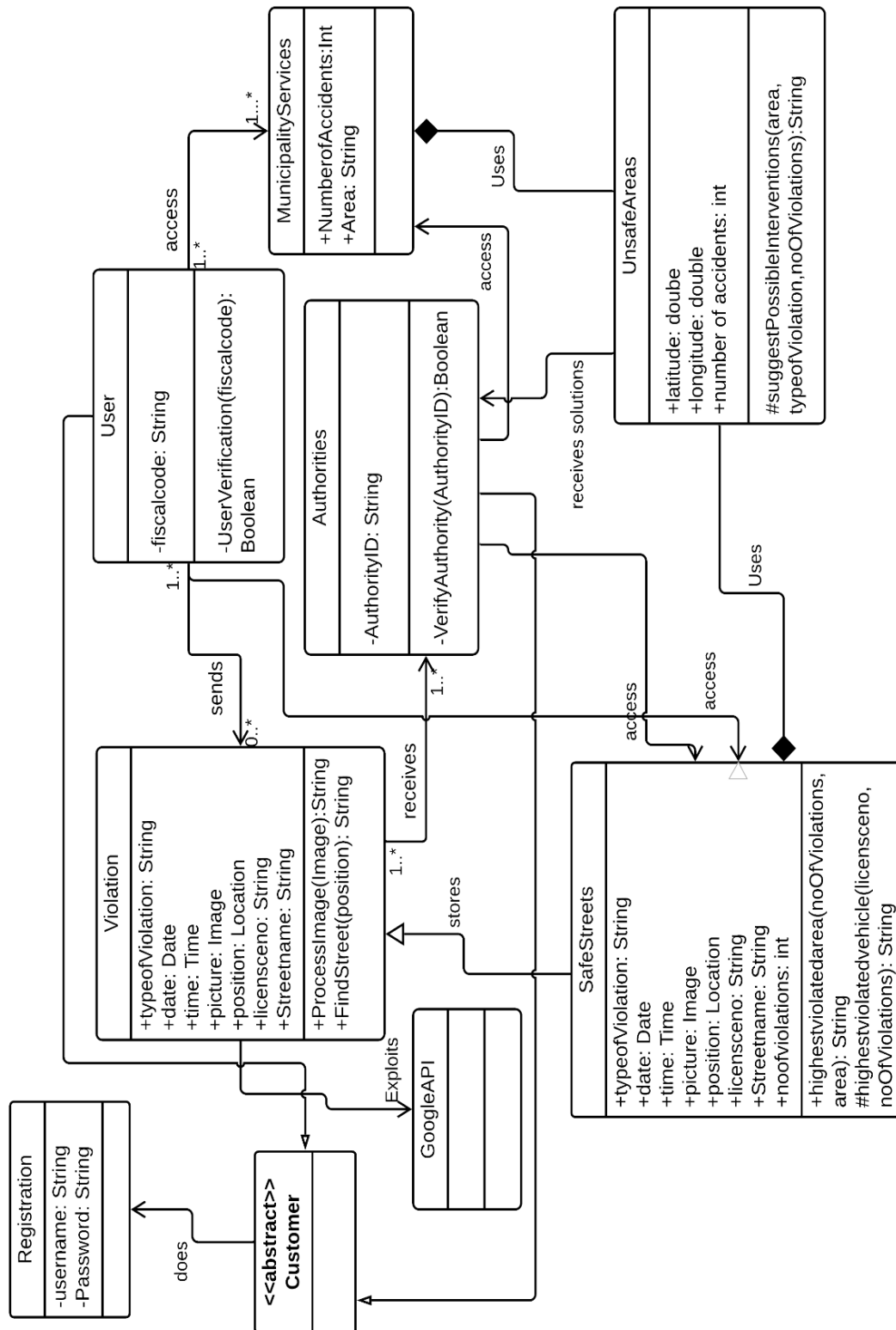


Figure 1: Class Diagram

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.

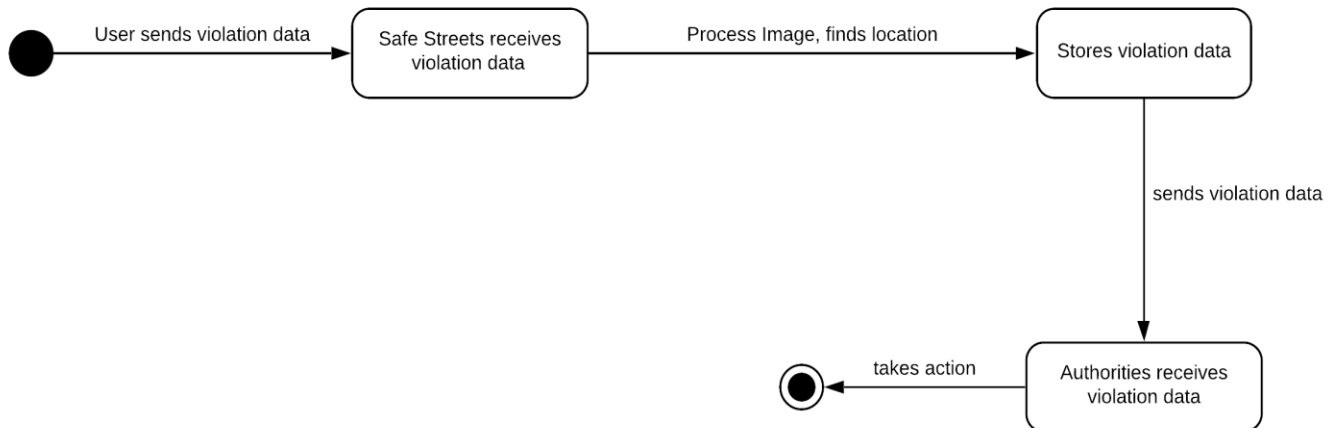


Figure 2- State diagram 1: User reports violation data and authorities takes action.

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.

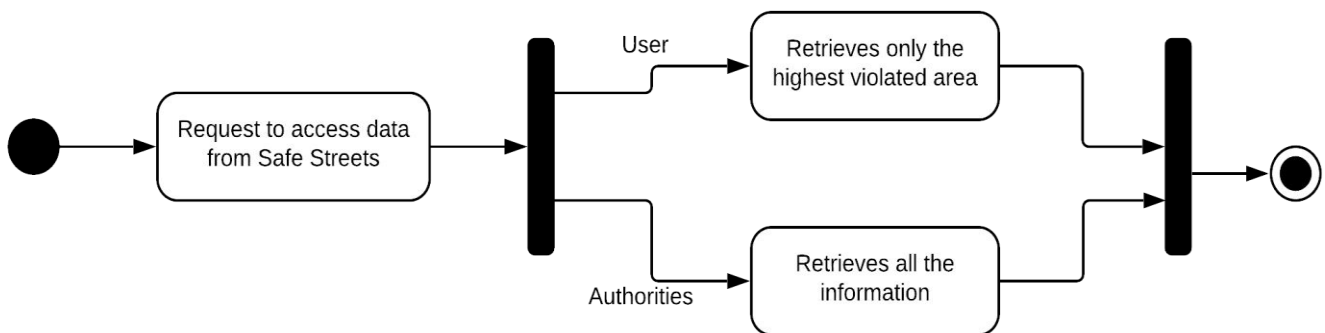


Figure 3- State diagram 2: Accessing data from Safe Streets

In the below diagram, Figure 4- State diagram: 3, the events involved in identifying the unsafe streets by receiving the data from municipality services and SS and also sending the authorities the solutions by analyzing the data.

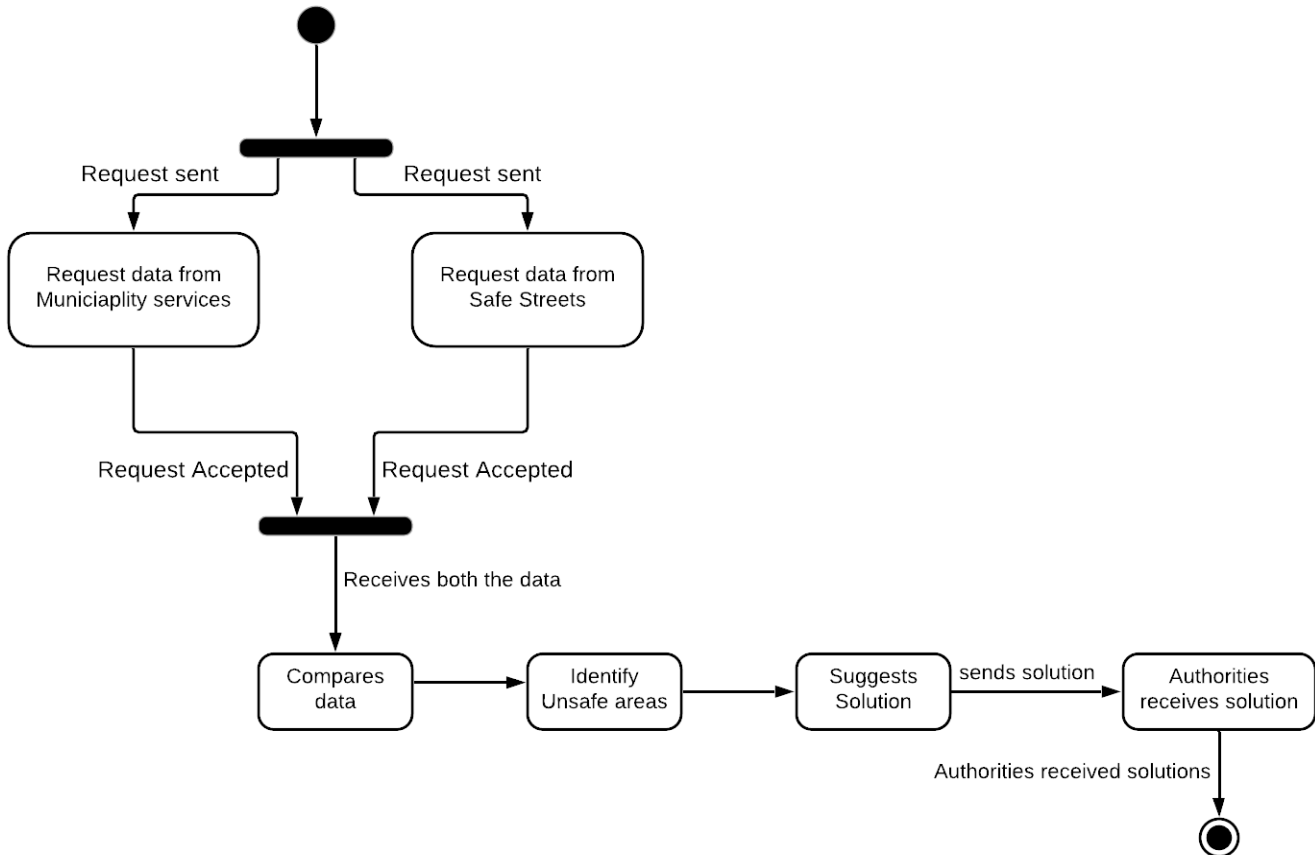


Figure 4- State diagram 3: Finding unsafe areas.

In the below diagram, Figure 5- State diagram 4, the events involved in accessing the data about accidents from the municipality services is described.

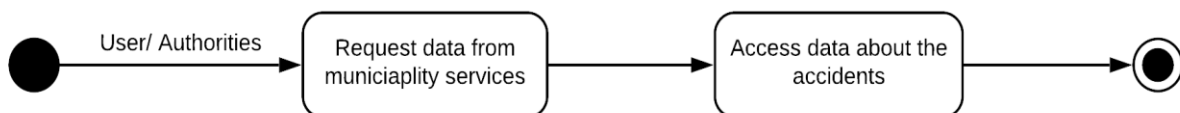


Figure 5- State diagram 4: Accessing municipality data.

2.2 Product Functions

The important product functions involved in the system are described briefly in the section 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 registering their details with SS. 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 it confirms whether the ID belongs to an 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. The camera 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 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 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, which user reported, 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.

2.2.2 Classifying Unsafe areas

The user and the authorities are allowed to access the data from SS regarding the accidents occurring in the particular area which is queried by the user. SS fetches these data from the municipality database by implementing an API for accessing the municipality services. SS also uses the data retrieved from municipality services to compare with its own data to derive conclusions about the unsafe areas.

The system will conclude an area as an unsafe area if the frequency of accidents is higher than the 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 is no injuries, in some 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 is responsible to implement those solutions provided.

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.
- **Classifying unsafe areas:** The user will be able to access data from municipality services through SS about the accidents in the particular area.

2. Authorities: A person who is a traffic official and will be responsible to enforce laws and take actions.

- **Reporting Violation and accessing data from Safe Streets:** The authorities will receive the data sent by the user and some data from SS which is processed for example license plate. They will be able to access almost all data from SS.
- **Classifying unsafe areas:** The authorities will be able to access data from municipality services through SS about the accidents in the particular area. They will also receive 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 or the one in laptop or desktop or tablet or any other devices 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 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 stores the information with suitable meta-data.

D5: The devices on which the services are exploited can provide real time information.

D6: Municipality services allows Safe Streets to exploit its data.

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 can access to the system. The user of the application should see the log-in page when he/she opens the application. If the user either it is a public or the authority who has not registered, he/she should be able to do Sign-in on the log-in page.

In the registration page, user will be asked to register with their name, surname, area and their fiscal code and the authority will be asked to register with their authority ID, name and surname.

The user chooses the button Report Violation in the home page and the user will be directed to the Violation page where a drop down box with 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. The user can go back to the home page where there is an option to view the highlighted area. By clicking on the Highlighted Area button user will be displayed with a list of highlighted

areas where the violation is high along with the name of the area and no. of violations. User can go back to the home page and select the button Accident Info upon clicking the button the user will be directed to the page which displays the number of accidents in each unsafe area.

The authorities will have View Violation, Analyzed data and unsafe areas buttons in the home page. The authorities when clicking on the View Violation will be able to see all the list of violation notifications that are reported by the user and upon clicking on a row they will be directed to the page where all the details of the selected notification will be displayed. On clicking the Analyzed data 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. By clicking on the Unsafe areas 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. A solution for converting it into a Safe Street is also provided.

Logout option will be provided at the top right corner upon clicking the button user or the authority will be logged out.

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 his part time. He uses his bicycle for delivering the food to the customers. One day when he was attending a delivery in Viale Tibaldi/Via Carlo Bazzi crossroads he met with an accident where a car from another side hit him and he was injured. Luckily, that day his friend also got order in the same place and same time they were doing it together. While this incident takes place his friend who is using Safe Streets application he just clicked a photo of that car license board and uploaded then the notification is received by the authorities and took an action. Here by Gowtham got some rewards for using this Safe Street application to complain about his issue happened in an unsafe area.

Scenario 2:

As it was a cozy winter season, one day two friends named Joel and Sam went out to enjoy their weekend. They went to a bar in Piazzale Loreto. They both enjoyed a lot in the bar drinking whisky of two bottles each. As they came out of the bar they started walking near piazzale Loreto to take a bus to reach their home. 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

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	<ol style="list-style-type: none"> 1. The user needs to choose the “Sign Up” option. 2. The user must fill all the mandatory fields. 3. The user can fill the data in the optional fields as well. 4. The system saves the data.
Exit Conditions	The user had entered the correct data and SS must store it preserving integrity.
Exceptions	<ol style="list-style-type: none"> 1. The user already registered in this case it should display 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.

Requirement Analysis and Specification Document

Name	Login
Actor	User
Entry Conditions	<ol style="list-style-type: none"> 1. The user must open SS on his/her device. 2. The user should have already signed up.
Event Flows	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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 Violation
Actor	User, Authority
Entry Conditions	The user must have already done the login activity.
Event Flows	<ol style="list-style-type: none"> 1. The user clicks on the Report Violation button. 2. The user will be directed to the Violation page where a drop down box with 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.

Requirement Analysis and Specification Document

	3. An optional field describing the violation can be filled by the user.
Exit Conditions	<ol style="list-style-type: none"> 1. The user data along with the location, date and time will be sent to the system. 2. Authorities will be notified about the violation reported along with all the data.
Exceptions	If the user missed out filling any of the mandatory field user will be notified with a warning message.

Name	Visualize Highlighted Data
Actor	User
Entry Conditions	The user must have already done the login activity.
Event Flows	<ol style="list-style-type: none"> 4. The user clicks on the Highlighted Area button. 5. A page will be displayed with a list of highlighted areas where the violation is high along with the name of the area and no. of violations.
Exit Conditions	The user is provided with the requested data.
Exceptions	\

Name	Accident Info
Actor	User
Entry conditions	The user must be logged in using his credentials on a device.
Events Flows	<ol style="list-style-type: none"> 1. Click on the Accident Info button.

	2. The user will be directed to the page which displays the number of accidents in each unsafe area.
Exit conditions	The user is provided with the requested data.
Exceptions	\

3.2.2 Authorities

Scenarios

Scenario 3: Neeta, who is working as a traffic violation monitor in the traffic control office receives a notification from the specific user namely John stating that there is a traffic violation in the Viale Romagna in which he mentioned that a vehicle whose license number is 160117 MI. He just mentioned in the violation is that this 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 has occurred and through the traffic camera installed in that streets she confirmed the violation and necessary actions were taken.

Scenario 4: Luca, who is working as a traffic violation monitor in the traffic control office received a notification from the user named Anna where she reported a vehicle which 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.

USE CASES

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	User, Authority
Entry Conditions	<ol style="list-style-type: none"> 1. The Authority must login into the application. 2. User have reported some violation.
Event Flows	<ol style="list-style-type: none"> 1. In the home page, click on View Violation button. 2. The authority will get the information about the violation.
Exit Conditions	<ol style="list-style-type: none"> 1. The authority will check whether it is a violation or not.

Requirement Analysis and Specification Document

	2. The authority will take necessary action.
Exceptions	\

Name	Analyzed data
Actor	Authority
Entry Conditions	1. The Authority must login into the application.
Event Flows	<ol style="list-style-type: none"> 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	Unsafe Area
Actor	Authority
Entry Conditions	1. The Authority must login into the application.
Event Flows	<ol style="list-style-type: none"> 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	\

3.2.3 Requirements

G1: The application must act as a medium where the user can report the traffic violations especially parking violations to authorities.

R1: The System must save the collected data of users registered to Safe Streets application in real time.

R2: The System has to allow the authority that wants to retrieve some data to know the user details.

R3: The system must not allow the users to modify the pictures taken using this application.

R4 When the request for the data is approved the system has to make previously saved data available for the authorities.

R5 The system has to save the data of unsafe areas which can be useful for users when they use that locations again by this the users and will stay safe in the unsafe areas.

D2 Each user Fiscale code must be unique.

D3 The devices inbuilt camera application must work properly.

D4 The device internet connection must work properly.

D5 The device must save automatically regarding unsafe area whenever there is a violation in that particular area.

3.3 Performance Requirements

The system has to able to serve a great number of users and authorities simultaneously. It has to 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 9.999 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.

//every customer can have only one account registered.

```
abstract sig Customer{
    registration: one Registration,
}
```

// defining all the signatures.

```
sig Fiscalcode, AuthorityID, Username, Password {}
sig Typeofviolation, Date, Time, Picture, Position, Licenseno, Streetname,
NoOfaccidents, NoOfviolations {}
sig HighlightVehicle, HighlightArea, MunicipalityService{}
//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,
    violations: 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
}
```

//parameter associated with authorities

```
sig Authorities extends Customer{
    authorityID: one AuthorityID
}
```

```
//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
}

// data associated with violation
sig Violation{
    tv: Typeofviolation,
    d: Date,
    t: Time,
    pic: Picture,
    loc: Position,
    lno: Licenseno,
    sn: Streetname
}
//check whether user sends all the data
assert Violation{

}
//highlighting areas
sig MunicipalityData extends NoOfaccidents{
    na: set NoOfaccidents
}
sig UnsafeArea extends HighlightArea{
    harea: HighlightArea,
    md: MunicipalityData,
    ua: UnsafeArea,
}
{
    all ua: UnsafeArea | >10
}
pred SS{

}
run SS for 3

```

5. Effort Spent