Politecnico di Milano

AA 2019/2020

Software Engineering 2: SafeStreet Requirements Analysis and Specification Document

Authors:
Devi Radhakrishnan,
Gayathri Prakash Menakath

Professor:
Matteo Rossi

Politecnico di Milano November 10, 2019
Version 1.0

Table of contents

1. Introduction

1.1 Purpose

1.1.1 General Purpose

SafeStreets is a crowd-sourced application that enables users to send a notification to a third party like authorities such as police, when they come across a traffic violation, in particular, a parking violation such as vehicles parked in the middle of bike lanes or in places reserved for people with disabilities, double parking, etc. The application allows users to send some metadata such as pictures of the violation, type of violation, date, location, time, etc to support their notification of the violation.

The core service, Notify_violation provided by SafeStreets is the processing and storing of the information provided by the users and the mining of this stored information by both end-users and authorities, by highlighting the street names or areas with the highest frequency of violations or vehicles that commit most violations. Once the application receives the information provided by the user, it processes the data in various steps, in particular when it receives a picture, it runs an algorithm to read the license plate and it stores this information with the type of violation, which is input by the user, and the name of the street where the violation occurred.

The application also offers the possibility for the user to receive the statistics of the traffic violations of the place the user is currently located at, using the geographical positioning.

Another feature provided by SafeStreets is Prevent_violations where SafeStreets suggests possible methods to identify potentially unsafe areas and to provide possible interventions to prevent violations. The municipality provides a service that allows users

to retrieve information about the accidents that occur on the territory of the municipality. SafeStreets makes use of this information to cross-check with its own data to suggest methods of prevention of violation.

1.1.2 Goals

G1: The application must allow users to report a violation by uploading pictures and metadata of the violation.
G2: The application must notify the third party like authorities about the violation reported.
G3: An algorithm is run on the picture provided by the user to detect the license plate. The license plate number is stored along with other data such as type of violation, date, location is stored in the database.
G4: The application must provide the risk rates of location as per user's request on enabling the PreventViolations feature.
G5: The application must provide third parties with the details of vehicles that commit the most violations.
G6: The application must provide the users and third parties with the details of cities/areas with the highest rate of violation.

It is worth noting that the use of the 'must' verb points out that each goal represents a mandatory objective of the system.

1.2 Scope

Safe Street being a crowd-sourced application helps to provide users with the possibility to notify authorities when traffic or parking violations occur. The application offers public or government authorities to collect the necessary data of the violators and the areas

prone to violations. This will, in turn, help them to enforce more rules to prevent accidents that happen due to traffic rules violation.

The application is expected to have two types of customers, first being the users who wish to report a traffic violation or retrieve statistics on violation reports in a particular area and third parties like authorities who can analyse and take necessary action regarding the reported violations. The S2B will allow the user to insert details of the violations like date, time, location, and the provision to send pictures of the vehicles. The user can also mention the type of violation like traffic rule violation, parking violation, etc. The user can send photos of the vehicles violating the rules. SafeStreets' NotifyViolation service will run an algorithm that will read the license plate from the received images. This will help to keep track of the vehicles that commit the most number of violations. The user can also be asked to enter the vehicle number manually so that it can be cross verified with the data received as a result of running the algorithm. The location can be determined either by requesting the user input or by retrieving the geographical position of the violation. It must be assumed that the user's device is capable of retrieving geographical data (GPS), sending images (camera) and that the images sent are of decent quality so as for the algorithm to deduce the license plate from it. The location can be verified by cross-checking the user input with that of the geographical data from GPS and also from the co-ordinates retrieved while reading the images to retrieve license plates.

Once a violation is reported by the user NotifyViolation will send a notification about the violation to the authorities who are the third parties. While reporting a violation third parties like the police, municipalities or any other government organization will be able to access the reporting user's identity and contact information so as to verify that the user is not an imposter. This will help to prevent fake notification alerts. The third parties will also be able to contact the users if needed.

In addition to notifying violations, both end-users and the authorities can mine through the information about the occurrences of the accidents in the municipality. This will help the authorities to strategize their operations to prevent further accidents. On login, a normal user will be able to see the statistics of areas with the most number of violations of all the types. The user will be able to know about the areas that have the highest accident/violation rates. The application allows third parties like authorities to view the violations in a particular area and also the information of the vehicles that make the most number of violations.

If the municipality or other organizations offer services that will have information about the accidents or violations that occur in their territory then PreventViolations service of the SafeStreets makes it possible for it to acquire this data for further operations. PreventViolations cross-checks this data with that of the already existing data that it has from the user inputs. This will help to potentially identify areas with risk of accidents and can suggest possible interventions like signboards, barriers between a bike lane and the part of the road for motorized vehicles, etc. Here the municipality is the third party and it is assumed that municipalities will have such a service for PreventViolations to acquire data. The user will be able to know the safety level of his/her location by enabling the location service of the device. Upon retrieving the geographical data of the user's current location PreventViolation can notify the user about violations happened in that area.

1.3 Definitions, Acronyms and Abbreviations

1.3.1 Definitions

- **User:** The normal customer of the application that exploits the basic features of the application to notify the authorities about violations. The user can also enable PreventViolations to get the risk rate of a particular location.

- **Third-party:** The customer of the application that gets notified when a normal user reports a violation. Third party gets the details of the vehicles that commit most number of violations.
- **Customer:** General customer of SafeStreet, can be both user or third party.
- **Traffic violation:** When a person parks the car on the wrong side of the road or whenever a person violates the traffic rules.

1.3.2 Acronyms

- **GPS:** Global Positioning System
- **API:** Application Programming Interface
- **S2B:** Software To Be
- **GDPR:** General Data Protection Regulation
- **EU:** European Union
- **EEA:** European Economic Area

1.4 Revision history

- Version 1.0:
    - First Release

1.5 Reference documents

- Specification Document: “Mandatory project Assignment AY 2019 - 2020”
- IEEE Std 830-‑1998 IEEE Recommended Practice For Software Requirement Specifications
- UML Diagrams: https://www.uml--diagrams.org/
- Alloy Doc: http://alloy.lcs.mit.edu/alloy/documentation/quickguide/seq.html

1.6 Document structure

The RASD document is composed of five chapters as mentioned below.

Chapter 1 is the introduction. It describes the purpose of the application informally and also mentions the goal of the application. It also describes the scope of the application in detail. All the features of the application and their purpose will be described in this section.

Chapter 2 offers an overall description of the application. This section has an elaborate explanation of the actors of the application. The class diagram provided here aids to the general understanding of the structure of the project. The state diagrams serve to understand the events flow at different stages of the application. All the features of the application are more explained more precisely.

Chapter 3 represents the body of the document. It contains the interface requirements like user interface, hardware interfaces and the software interface. It then lists a few real word scenarios and the expected behavior of the application. Use case diagrams and sequence diagrams are provided to explain the functioning of the application. The non-functional requirements are described through performance requirements, design constraints and software system attributes.

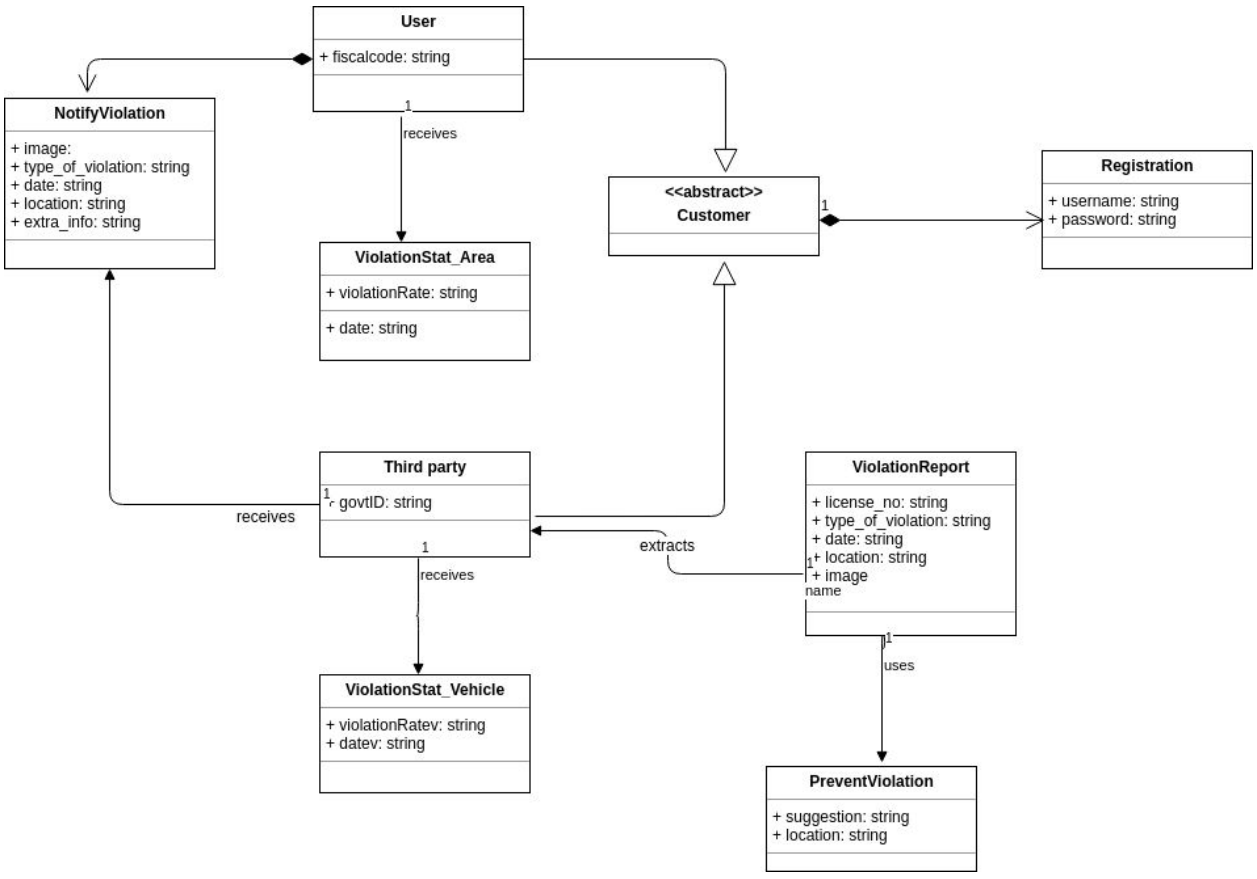
Chapter 4 contains the alloy model of some critical aspects of the system. The particular documentation is also provided.

Chapter 5 shows the effort each member has put in.

2. Overall Description

2.1 Product Perspective

The idea is to build PreventViolation along with NotifyViolation as it is an additional service that can integrate the features of NotifyViolation. It will provide additional information by retrieving data from third parties like municipalities. To retrieve the location of the users or the area where the violation happened the application exploits the device's GPS and the camera for sending good quality images.



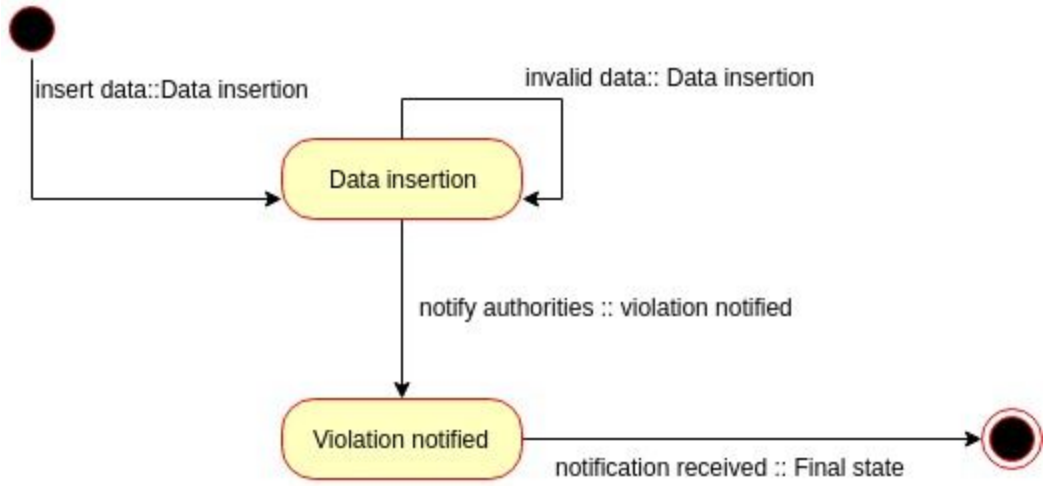


Figure 2: State Diagram: 1 Reporting a violation

In the first state diagram the user inserts the data necessary to report a violation. In case of invalid data user will be asked to insert the correct data. If correct then third party authorities will be notified and it will be acknowledged.

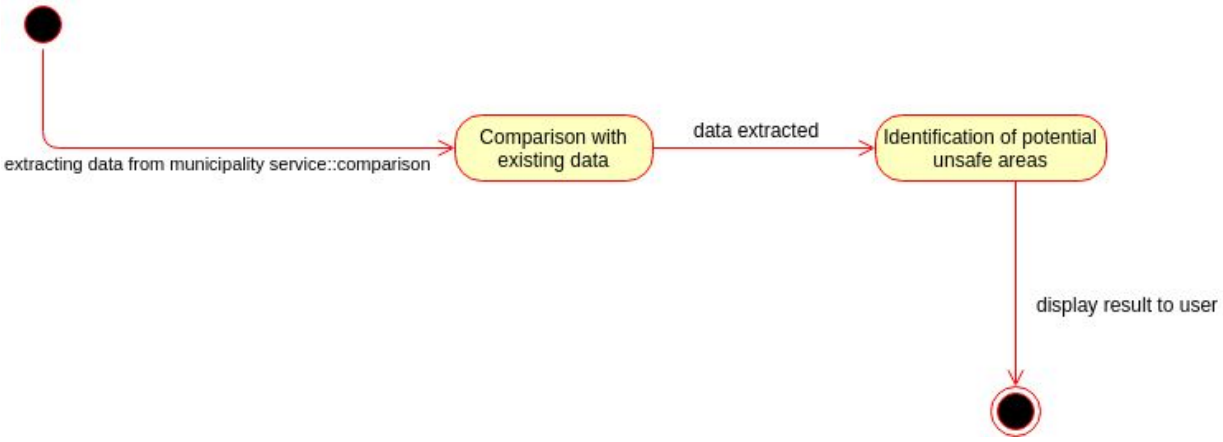


Figure 3: State Diagram: 2 Displaying high risk rate areas

The application extracts data from the services offered by third parties and will compare it with its own data. As a result, the application will display the names of the cities with high risk rate.

Figure 4: State Diagram: 3 Violation rate of the specific area

When a user inputs a particular location the application will display the risk rate at that particular location. This is done by extracting data from third parties and comparing it with that of the application.

2.2 Product Functions

In the following section, the most important product functions are reported.

2.2.1 Notifying Violations management

This is the core function of the application. The application will allow the user to sign up entering a username, password, and a few other necessary information. Some data like the identification document, contact information will be necessary. In order to secure the privacy of the individuals, personal data will be shared with the third party only if it is a recognized authorization or a government undertaking. A user can check for areas with the highest rates of violation or accident rates. The application will allow the user to report violations along with the necessary details like the location, type of violation, date and images of the vehicle. This will notify the third parties who are concerned authorities. From the uploaded images the application will read the license plate. To a third party authority, the application provides the details (vehicle registration number, color, model, etc) of the vehicles that commit the most number of violations. The third-party like the municipality can verify the credibility of the users who report a

violation. The application allows third parties like authorities to view the violations in a particular area and also the information of the vehicles that make the most number of violations.

2.2.1 Preventing Violations management

This product function must be guaranteed by the application if and only if there is a third party offering the necessary service. In this case, the third party will be the municipality, local police or any other government organization who offers their data about the traffic violations or accidents. The application will retrieve data from the third party service and will compare this data with that of its own. The application will already have its own share of data collected from the user inputs. On cross-checking the available data, it will be easier to identify potential accident-prone zones. The application will then notify the third parties about the possible interventions. This may include any kind of suggestions that will help prevent any more accidents or other violations in that area. The user will be able to check the safety level of his/her location by enabling the location service of the device. PreventViolation will retrieve the geographical data of the user's current location can notify the user about violations happened in that area.

2.3 User Characteristics

The actors of the application are the following:

1. **User:** a person who registers to SafeStreets. The user's role will be better explained in the following distinctions depending on the service.
    - **NotifyViolations:** a user is a person who reports the violations by giving necessary details. The user can see all his/her past reporting and also the areas with high-risk rates.

    - **PreventViolations:** A user will be able to see the violation/accident rates at a particular location.
2. **Third-party:** it is necessary to distinguish what the third party is depending on the service.
    - **NotifyViolations:** the third party is any government authority who should be notified of traffic violations or accidents. (local police etc). They will be able to get the details of vehicles that commit most violations. Verify the details of the users.
    - **PreventViolations:** the party will be an organization or any authority which will provide statistical information on the occurrences of violations or accidents.

2.4 Assumptions, dependencies and constraints

- **D1:** The location of the area is provided with an error of at most 5 meters.
- **D2:** Each fiscal code number is unique.
- **D3:** The government ID provided at the time of organization (third party) registration is unique.
- **D4:** The images uploaded by the user are of good quality.
- **D5:** The third parties are acknowledged organizations.
- **D6:** The internet connection works properly without failure.

Any device that can connect to the internet through a browser is enough for both the user and the third party to access the application. (registering, reporting violations, checking risk zones, etc). The customer's device should have a 2G/3G/4G internet connection. In order to retrieve the geographical data, the device should have GPS. The S2B is supposed to be distributed in Italy due to which we are considering fiscal code as an identity document.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The following mockups give an approximate idea of how the user interface of the application should look like. It differentiates the interactions of both user and third party with the SafeStreet application

Figure: 5 Mockup: Registration for users

Figure: 6 Mockup: Registration for organization

The registration for both users and third parties are different. In the case of third party registration, the respective identification number of organizations is required. A user who is an individual will have to provide the fiscal code.

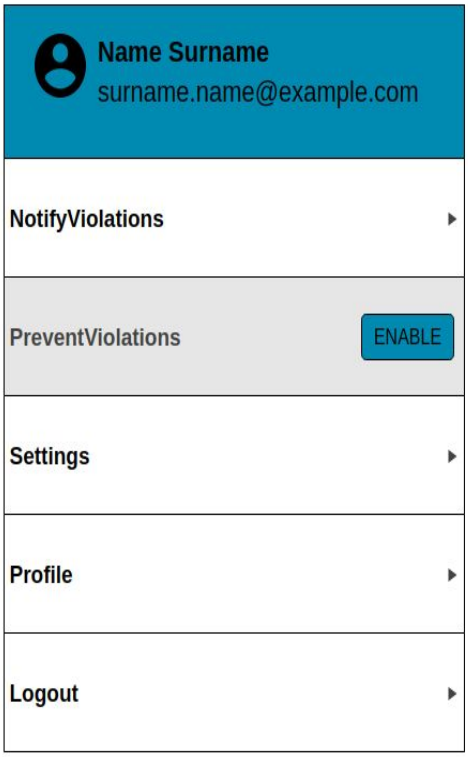


Figure: 7 Mockup: Main menu

If the user has not enabled the PreventViolatons feature then that feature alone will be disabled and the rest of the features will be available.

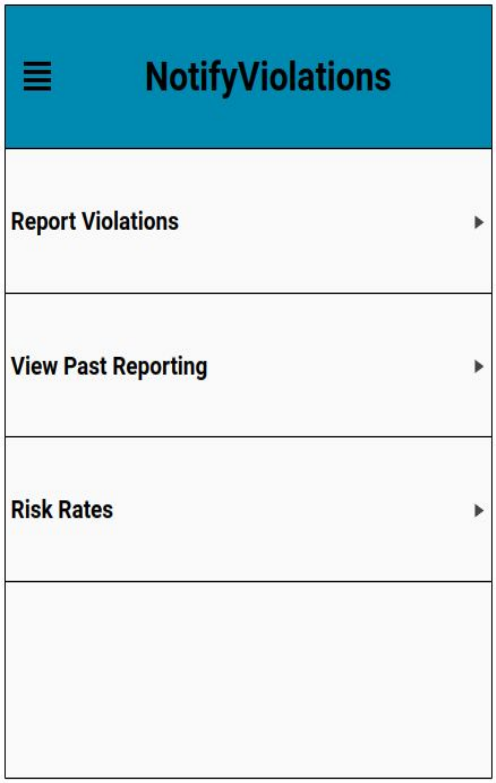


Figure: 8 Mockup: The NotifyViolation feature

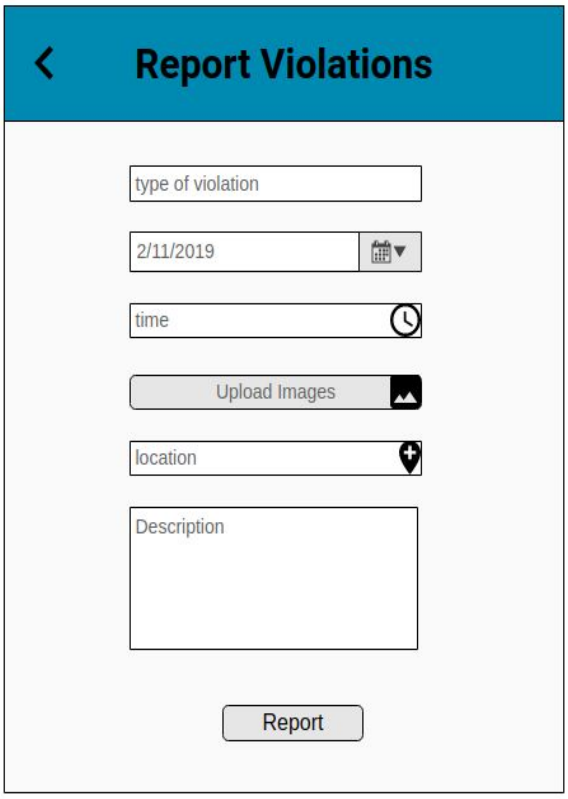


Figure: 9 Mockup: Reporting a violation

The user can notify the third party authorities by reporting the violations. The third-party will be notified as soon as the violation has been reported. The user can also view the past reportings and also the cities with the highest rate of violations.

Figure: 10 Mockup: The Find Risk Rate Screen

If the user enables the PreventViolations notification, then the user will be able to find the risk rate at a particular location. The application will display a graphical representation of the violation in the previous months.

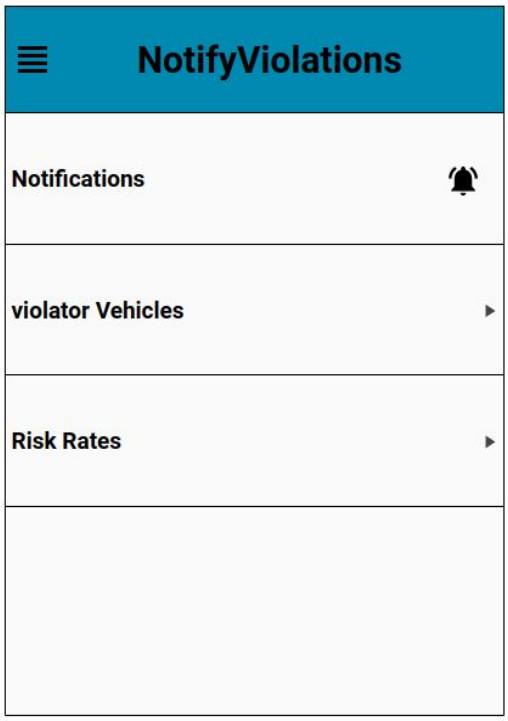


Figure: 11 Mockup: Third Parties screen

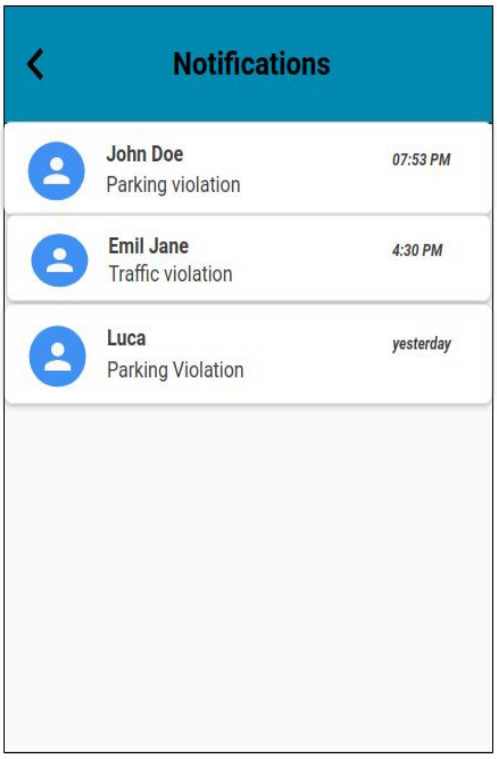


Figure: 12 Mockup: Notifications for third parties

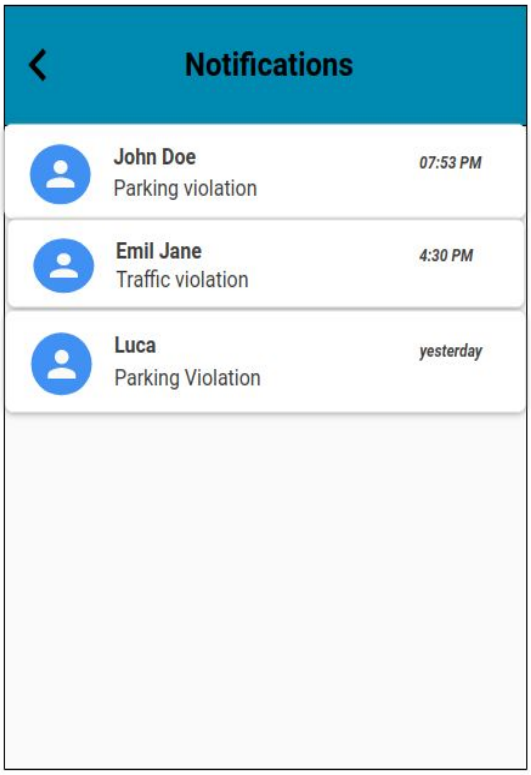


Figure: 13 Mockup: Notifications for third parties

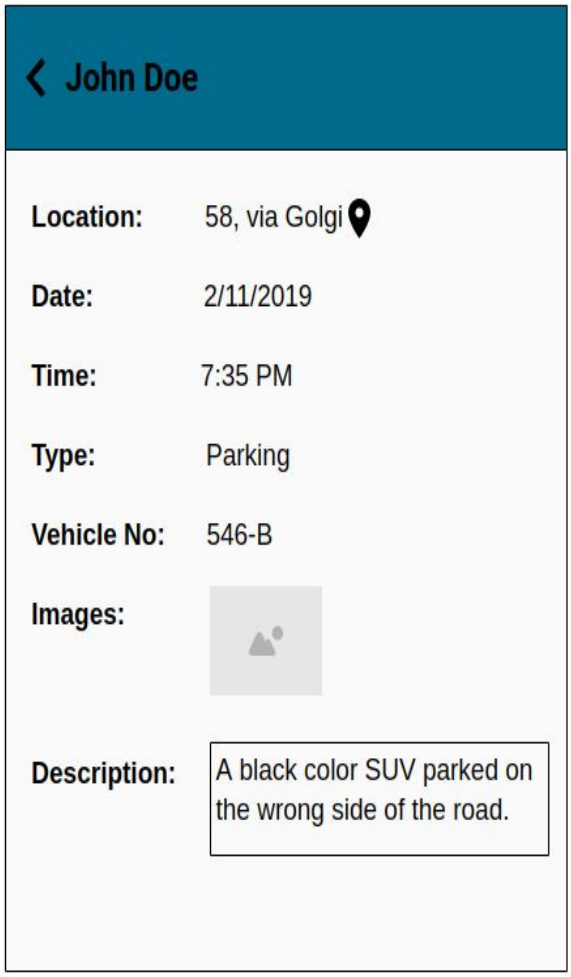


Figure: 14 Mockup: Third-party view for a reported violation

3.1.2 Hardware interfaces

The application does not have any hardware interfaces.

3.1.3 Software Interfaces

The application does not provide any APIs to any external application.

3.2 Functional Requirements

3.2.1 User

Scenarios

Scenario 1

Sarah, a young woman, is driving her SUV to purchase groceries. She decides to park the car a few meters from the grocery store. It is then she notices that another car has been parked inappropriately due to which she is not able to park her vehicle. She waits for a few seconds but does not see anyone coming. She then decides to use SafeStreet's NotifyViolation feature to report the parking violation. She clicks the photo of the car which has the license plate and uploads it along with the details of the area. SafeStreet notifies the concerned third parties and the police come in a few minutes to tow away the wrongly parked car.

Scenario 2

Jo is new to the city of Milan. Before going out on a drive he would like to know the accident/rules violation rates in the city. He decided to check it with the help of SafeStreet's PreventViolation app. The app has its data cross-checked with that of third parties to provide the risk rate of particular locations. Jo gives the location and checks the risk rate to avoid any mishaps.

Use Case Diagram

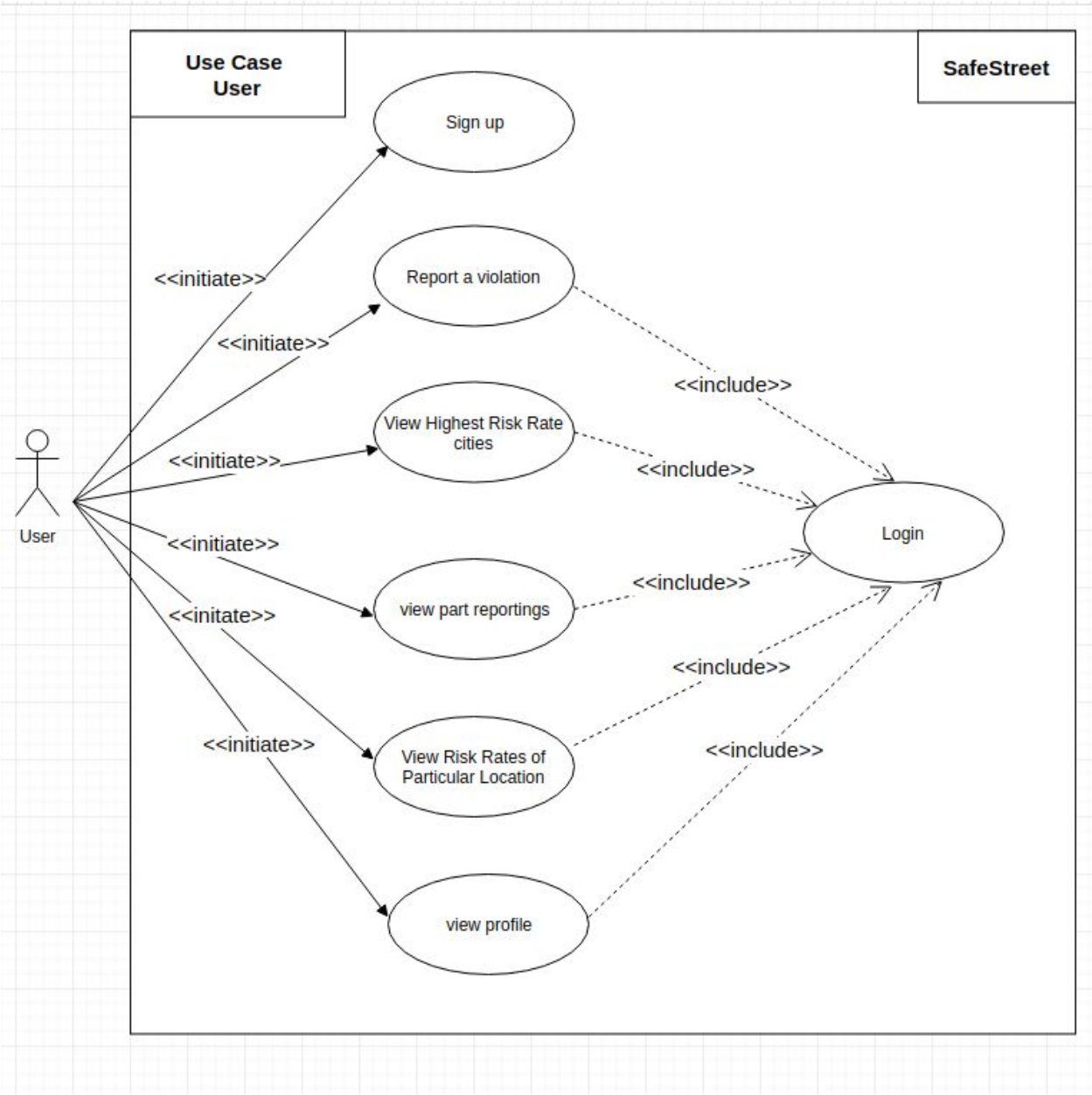


Figure: 14 Use Case Diagram: User

Use Cases

Name	Sign up
Actor	User
Entry Conditions	The user should open the application on the device.
Events Flow	1. The user chooses the sign-up option. 2. Fill the sign-up form with necessary data. 3. Confirm sign up. 4. The application saves the data.
Exit conditions	The user is registered and the data is stored in the application.
Exceptions	1. The user is already registered. In this case, application recognizes the email id and will suggest login. 2. When the user fills the form with invalid data, the application will indicate it be registering the user.

Name	Login
Actor	User
Entry Conditions	The user has already created an account.
Events Flow	1. The user chooses login option.

	2. Fills the email and password fields. 3. Confirm Login.
Exit conditions	The user is logged in moves to other features of the application.
Exceptions	1. User is not registered. 2. The user enters a wrong email id 3. User enter a wrong password In all the cases, the user will be notified about what went wrong.

Name	View Profile
Actor	User
Entry Conditions	The user is logged-in to the system
Events Flow	The User will be able to see the details on selecting the profile option from the main menu.
Exit conditions	Personal data will be displayed.
Exceptions	\

Name	Report Violation
Actor	User

Entry Conditions	The user has been successfully logged-in.
Entry Flow	1. The user fills up the form with mandatory data. 2. Uploads the image of the vehicle. 3. Adds any additional data. 4. Confirms the reporting. 5. Third party will be notified.
Exit Condition	The third party is correctly notified.
Exceptions	\

Name	View highest risk rate cities
Actor	User
Entry Conditions	The user has been successfully logged-in.
Events Flow	1. The user chooses the option to view the cities with the highest violation/risk rates. 2. The stats will be displayed
Exit conditions	The data is displayed
Exceptions	The comparison of data to calculate the risk factor has not taken place. In that case stats may not be accurate.

Name	View Past Reportings
Actor	User
Entry conditions	The user has been successfully logged-in.
Events Flow	1. The user chooses the option to view the past reporting. 2. Particular data will be displayed.
Exit conditions	The past reportings made by the user will be displayed.
Exceptions	\

Name	View Risk Rates of Particular location
Actor	User
Entry conditions	1. The user has been successfully logged-in. 2. The user has enabled the 'PreventViolations' feature.
Events Flow	1. The user chooses the option to view the risk rate of a particular location. 2. The user gives a particular

	location. 3. The risk rate of that particular area will be displayed.
Exit Conditions	The risk rate of a particular area will be shown.
Exceptions	The comparison of data to calculate the risk factor has not taken place. In that case, stats may not be accurate.

Sequence Diagrams

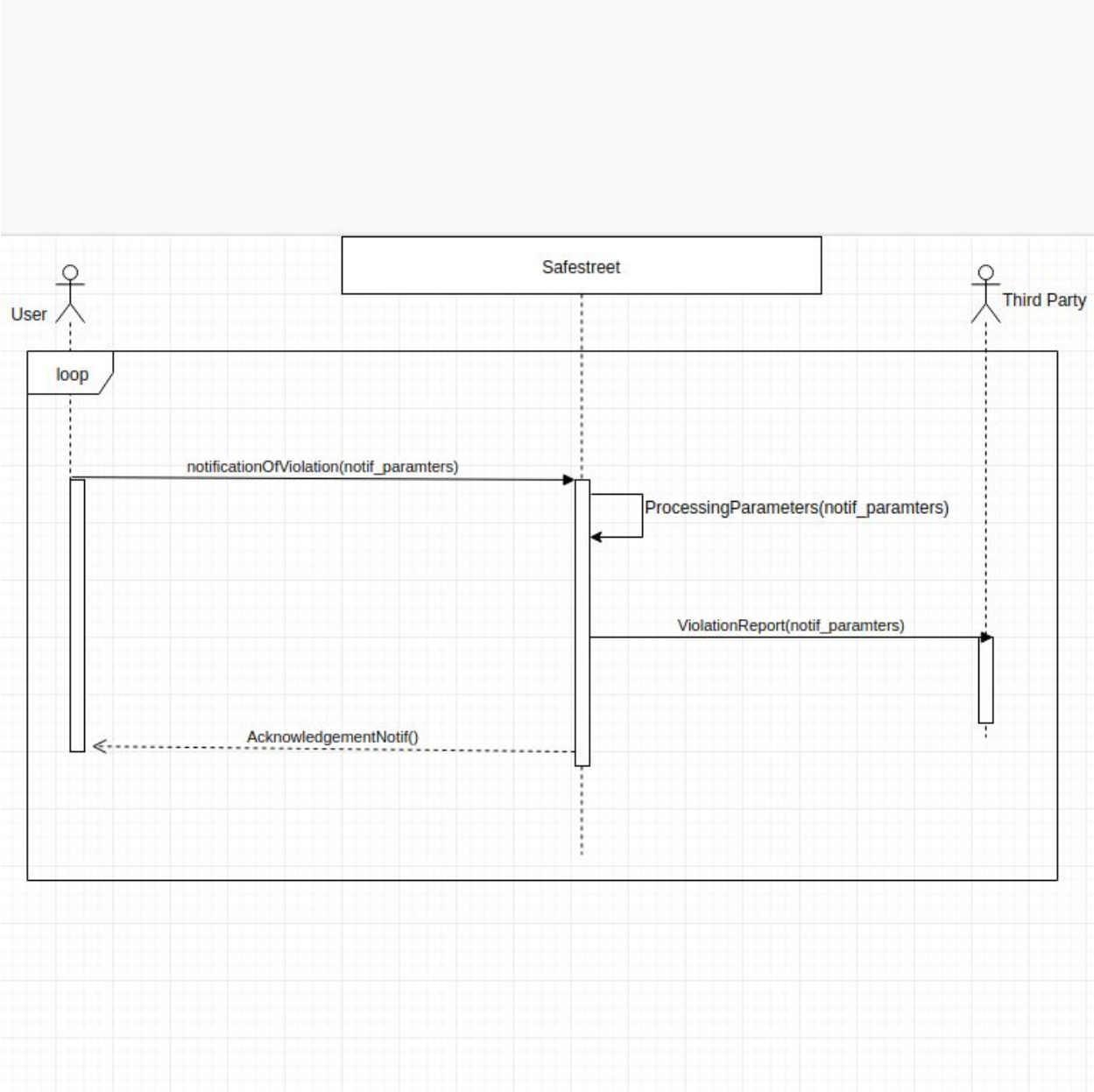


Figure: 15 Sequence Diagram: User

3.2.2 Third-Party

Scenario

Scenario 3

Pietro parks his car on the wrong side of the road as he is in a hurry to meet his friend. Meanwhile, a lady sees this and reports it through SafeStreet's NotifyViolation. Officer Hill is on duty and receives the notification. He immediately contacts the other officers near to Pietro's area to tow away his car. He also checks if Pietro's car is there in the list of vehicles that commit most violations.

Scenario 4

The municipality of Milan decides to offer a service that will give the details of traffic rule offenders, parking rule violators, etc. SafeStreet decides to use this information for their PreventViolations feature so that they can cross-verify this data with that of their own. This will help them to provide more accurate results when users search for risk rates in a particular area.

Use Case Diagram

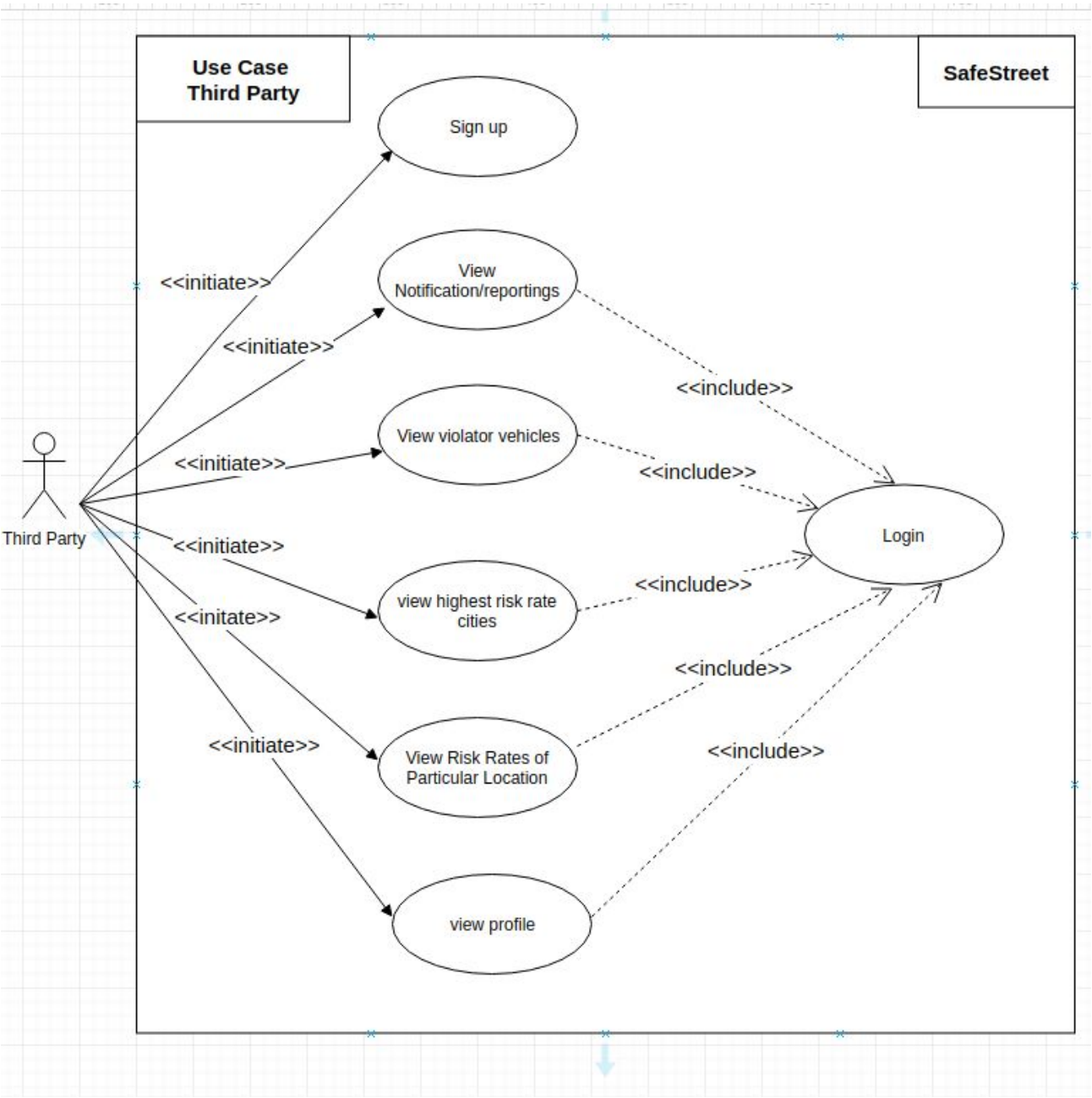


Figure: 16 Use Case Diagram: Third Party

Use Cases

Name	Sign up
Actor	Third Party
Entry Conditions	The Third party should open the application on the device.
Events Flow	1. Third party chooses the sign up option. 2. Fill the sign up form with necessary data. 3. Confirm sign up. 4. The application saves the data.
Exit conditions	The Third party is registered and the data is stored in the application.
Exceptions	1. The Third party is already registered. In this case application recognizes the email id and will suggest login. 2. When Third party fills the form with invalid data, application will indicate it be registering the user.

Name	Login
Actor	Third Party
Entry Conditions	The Third party has already created an account.

Events Flow	1. The Third party chooses login option. 2. Fills the email and password fields. 3. Confirm Login.
Exit conditions	The Third party is logged in moves to other features of the application.
Exceptions	1. Third party is not registered. 2. Third party enters a wrong email id 3. Third party enter a wrong password In all the cases, the user will be notified about what went wrong.

Name	View Profile
Actor	Third Party
Entry Conditions	The Third party is logged-in to the system
Events Flow	The Third party will be able to see the details on selecting the profile option from the main menu.
Exit conditions	Personal data will be displayed.
Exceptions	\

Name	View Notifications
Actor	Third party
Entry conditions	The third party has been successfully logged-in.
Events Flow	1. The third party chooses the option to view the notifications. 2. Particular data will be displayed.
Exit conditions	The notifications of violation reporting made by the user will be displayed.
Exceptions	\

Name	View highest risk rate cities
Actor	Third party
Entry Conditions	The Third party has been successfully logged-in.
Events Flow	1. The Third party chooses the option to view the cities with highest violation/risk rates. 2. The stats will be displayed
Exit conditions	The data is displayed
Exceptions	The comparison of data to calculate the risk factor has not taken place. In that

	case status may not be accurate.

Name	View Risk Rates of Particular location
Actor	Third party
Entry conditions	The third party has been successfully
Events Flow	1. The third party chooses the option to view the risk rate of particular location. 2. The third party gives the particular location. 3. The risk rate of that particular area will be displayed.
Exit Conditions	The risk rate of a particular area will be shown.
Exceptions	The comparison of data to calculate the risk factor has not taken place. In that case stats may not be accurate.

Name	View violator vehicles
Actor	Third party
Entry conditions	The third party has been successfully
Events Flow	1. The third party chooses the option

	to view the details of vehicles that commit violations. 2. The details of the vehicle will be shown
Exit Conditions	The details of the vehicle that commit most number of violations will be shown.
Exceptions	The comparison of data to calculate the violation factor has not taken place. In that case status may not be accurate.

Sequence Diagrams

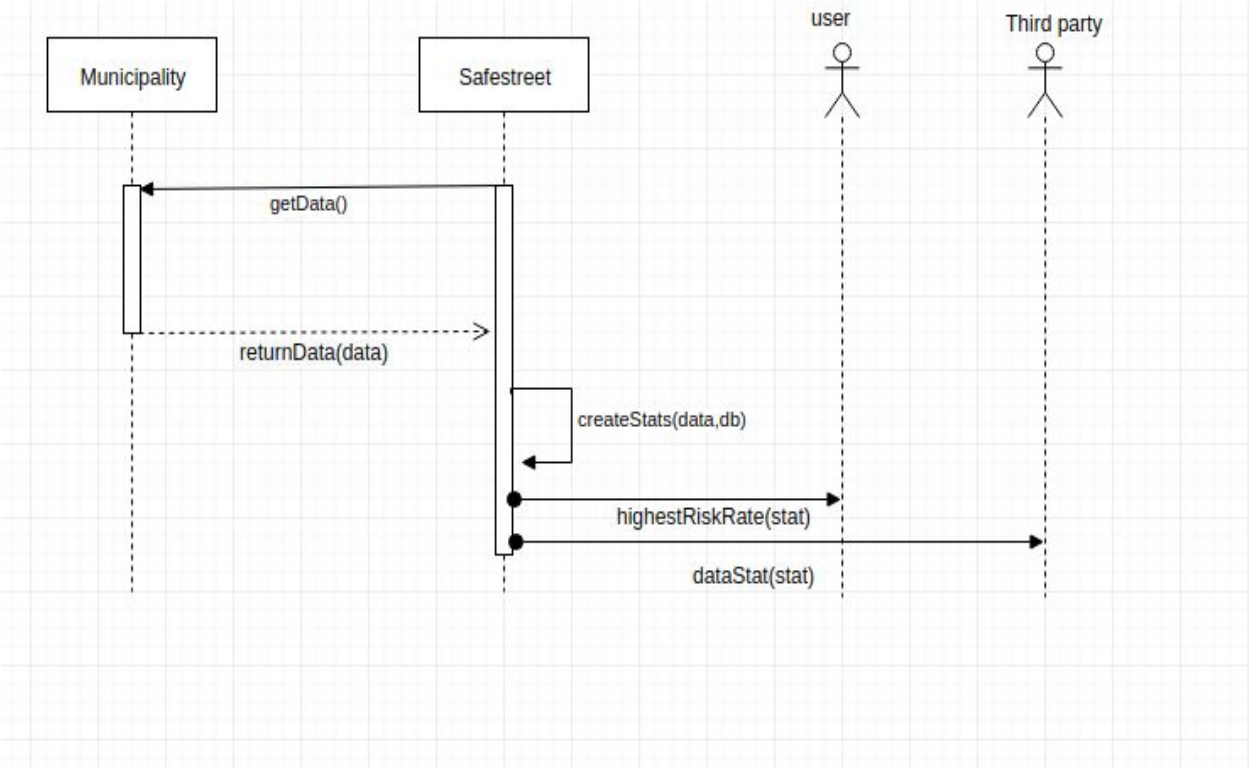


Figure: 17 Sequence Diagram: Third Party

3.2.3 Requirements

In this section, we show that the requirements ensure that the satisfaction of the goals with respect to the domain assumptions. The list of requirements and domain assumptions for each goal are given below.

G1: The application must allow users to report a violation by uploading pictures and metadata of the violation.

- **R1:** The application stores the details of the violation report.
- **R2:** The application has collected the details of the user at the time of registration
- **D1:** The location of the area is provided with an error of at most 5 meters.
- **D2:** Each fiscal code number is unique.
- **D4:** The images uploaded by the user are of good quality.
- **D5:** Third parties that are notified of violations are government organizations.
- **D6:** The internet connection works properly without failure.

G2: The application must notify the authorities about the violation reported.

- **R1:** The application stores the details of the violation report.
- **R3:** The application notifies the third parties who are concerned authorities like the local police.
- **R4:** The application must allow the third party to access the contact details and identity documents of the user if needed.
- **D3:** The government ID provided at the time of organization (third party) registration is unique.
- **D5:** Third parties that are notified of violations are government organizations.
- **D6:** The internet connection works properly without failure.

G3: An algorithm is run on the picture provided by the user to detect the license plate. The license plate number is stored along with other data such as type of violation, date, location is stored in the database.

- **R5:** The application runs the algorithm to retrieve the license plate number from the images uploaded by the user.
- **R6:** The retrieved licence plate number is sent as part of the violation report.
- **D4:** The images uploaded by the user are of good quality.
- **D6:** The internet connection works properly without failure.

G4: The application must provide third parties with the details of vehicles that commit the most violations.

- **R2:** The application has collected the details of the user at the time of registration.
- **R5:** The application runs the algorithm to retrieve the license plate number from the images uploaded by the user.
- **R6:** The retrieved licence plate number is sent as part of the violation report.
- **R7:** The application must provide the details of vehicles that have committed traffic violations.
- **D2:** Each fiscal code number is unique.
- **D4:** The images uploaded by the user are of good quality.
- **D5:** The third parties are acknowledged organizations.
- **D6:** The internet connection works properly without failure.

G5: The application must provide the users and third parties with the details of cities/areas with the highest rate of violation.

- **R8:** The application must provide both third parties and users with the statistics of cities with the highest violation rates.
- **D1:** The location of the area is provided with an error of at most 5 meters.
- **D5:** The third parties are acknowledged organizations.

- **D6:** The internet connection works properly without failure.

G6: The application must provide the risk rates of location as per user's request on enabling the PreventViolations feature.

- **R9:** The application must allow the users who have enabled the PreventViolation feature to get risk rates of particular location upon request.
- **R10:** SafeStreet should cross its data with the information provided by the authorities about the accidents that occur on the territory of the municipality to give potential unsafe areas.
- **D1:** The location of the area is provided with an error of at most 5 meters.
- **D6:** The internet connection works properly without failure.

3.3 Performance Requirements

The application must be able to serve a great number of users and third parties simultaneously. It must guarantee quick, reactive and accurate responses. It is important to make sure that the notifications of violation reports are sent to the correct third parties.

3.4 Design Constraints

3.4.1 Standards Compliance

With regard to the privacy of data, the entire project is subject to the General Data Protection Regulation (GDPR), a regulation in EU law on data protection and privacy for individuals within the European Union (EU) and the European Economic Area ((EEA).

3.4.2 Hardware Limitations

Every device with an internet connection and GPS is enough to use the application. To be concise all the hardware requirements are mentioned below.

- Connection to internet.
- GPS
- Camera (phone or other)

3.5 Software System Attributes

3.5.1 Reliability

The application must run continuously without any interruptions or errors. The application must be ensured as fault tolerant. It should be able to serve all the users and the third parties.

3.5.2 Availability

A fault tolerant architecture should be ensured for the application. As the application serves a large number of individuals and third parties it should be available to all. All the features of the application must work independently.

3.5.3 Security

As the data provided by the user contains personal information, the security aspect of the system is of prime importance. The central database where the data reside must be protected from both internal and external attack. Encryption should be used while sending data.

3.5.4 Maintainability

The application must be such that it is easy to fix any part of it in the future in case of any issues. It must be feasible to accomodate any changes or modifications without much hassle.

3.5.4 Compatibility

The application offers multiple services to both users and third parties. Hence it should be compatible with as many devices and technologies possible.

4. Formal Analysis Using Alloy

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:

1. Only if the user has reported a violation a notification must be sent to the third party
2. The username and fiscal code must not be the same for two users. It must be unique for each person
3. There must not be any duplicate reports created of the same violation. In order to check that we check if two reports have the same date, license plate number, and location.

```
sig Fiscalcode{}
```

```
sig Username{}
sig GovtID{}
sig Password{}
sig Date{}
sig LicenseNo{}

sig Registration{
        username: one Username,
        password: one Password
}

abstract sig Customer{
        registration: one Registration
}

sig User extends Customer{
        fiscalcode: one Fiscalcode,
        report: lone ViolationReport
}
sig Thirdparty extends Customer{
        govtid: one GovtID,
        notificationreported: set ViolationReport
}

sig Location{
        latitude: one Int,
        longitude: one Int
}
```

```
sig ViolationReport{
        sender: one User,
        date: one Date,
        location: one Location,
        licenseNo: one LicenseNo
}



fact FiscalcodeUserConnection{
        all fc: Fiscalcode | some u:User | fc in u.fiscalcode
}

fact UsernameRegistrationConnection{
        all u: Username | some r: Registration | u in r.username
}

fact PasswordRegistrationConnection{
        all p: Password | some r: Registration | p in r.password
}

fact ViolationReportNotification{
        all u: User, vr: ViolationReport | (vr.sender=u iff u.report=vr)
        and (one tp: Thirdparty | vr in tp.notificationreported)
}



fact NoSameFiscalCode{
        no disj u1,u2: User | u1.fiscalcode = u2.fiscalcode
```

```
}
fact NoSameUsername{
	no disj c1,c2: Customer | c1.registration.username=c2.registration.username
}

fact NoReportwithSameLocationAndDate{
	no disj r1,r2: ViolationReport | r1.date=r2.date and r1.location=r2.location and
r1.licenseNo=r2.licenseNo
}

fact Notificationsentifandonlyif {
	all u: User,tp: Thirdparty | {
	(#tp.notificationreported>1) iff (u.report=1)
	}
}



assert NoDifferentThirdPartyReceivingTheSameViolationReport{
	no disj tp1,tp2: Thirdparty | some vr: ViolationReport | vr in
tp1.notificationreported and vr in tp2.notificationreported
}

check NoDifferentThirdPartyReceivingTheSameViolationReport for 3
```

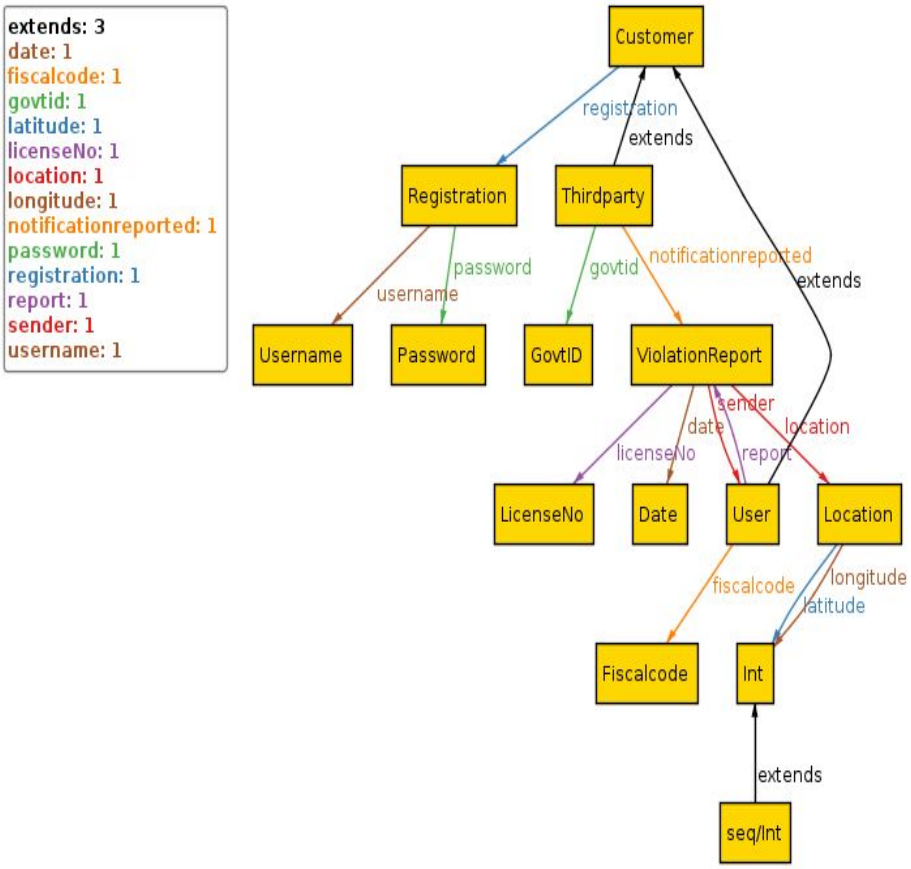


Figure: 18

5. Effort Spent

ST1

Description of the task	Hours

Introduction	6
Product Perspective	5
Product Functions	12
Domain Assumptions	2
External Interface Requirements	8
Functional requirements	15
Non-Functional Requirements	1
Formal Analysis using Alloy	20

ST2

Description of the task	**Hours**
Introduction	4
Product Perspective	15
Product Functions	17
Domain Assumptions	2
External Interface Requirements	8
Functional requirements	15
Non-Functional Requirements	1
Formal Analysis using Alloy	5