**Purpose**

This document is aimed to provide system for an authorities and they want to develop software-based services in which people are allowed to notify authorities when traffic violations occur. Atually the app will give every citizen of the country an opportunity to become a virtual traffic agent. SafeStreets is expected to be an answer for various limitations faced by the authorities in controlling traffic violations.

The SafeStreets software tools would provide a service in which it can report traffic violations and send the data from the mobile phone. This document will try to find the requirements for setting up such a system.

This document will also provide an analysis about a service tool called Advanced functions 2, Basic a service which can takes the information about the violations coming from SafeStreets, and generates report it to authorities and show some statistical diagram and finally provide an overview of Safestreet will describe the system and its requirements.

**Scope**

**Description of the given problem**

SafeStreets is an application that intends to bringing public participation into reducing traffic offences, in particular parking violations. The application allows users to send pictures of violations, including their date, time, and position to system. There are 2 possiblie servic basic servic and advanced servic.for the basic servic SafeStreets stores the information provided by users. In particular, when it saves the report of vilation the user recognize the plate number of vehicle and stores the retrieved information with the type of violation, which select by users and the name of the street where the violation occurred and also can be retrieved from the geographical position of the violation. In addition, the application allows both end users and authorities to mine the information that has been received, for example by highlighting the streets (or the areas) with the highest frequency of violations, or the vehicles that commit the most violations. Of course, different levels of visibility could be offered to different roles.

**Advanced functions 2**: In addition, the municipality (and, in particular, the municipal agent) could offer a service that takes the information about the violations coming from SafeStreets, and generates traffic tickets from it. In this case, mechanism should be put in place to ensure that the chain of custody of the information coming from the users is never change, and the information is never altered (e.g., if a manipulation occurs at any point of the image showing the violation, for example to alter the license plate, the application should discard the information). In addition, the information about issued tickets can be used by SafeStreets to build statistics.

**1.2.2. Current System**

There are two applicatios which their names are TrafficEye, PublicEye that people can report road crashes & traffic violations without disclosing the sender identity.

**World and machine phenomena:**

The occurrences of Violation

The report of Violation by public

Takeing photo and write details of Violation scene

Upload the photo

The update of a database entry

The creation of a new object of class traffic Violation

Access to the camera of smartphone

### **Goals**

[G1] Allow a Visitor to become registered User after providing personal information in sign up form.

[G2] Allow user to report a violation

[G2.1] Giving Access to use camera of his/her smart mobile phone.

[G2.2] Providing a form for some additional information, such as the license plate number (it would be better to has this item which user can write license plate number in form), type of violation, details about location and vehicle’s position, and other description.

[G2.3] Give access user to location (it is possible by employing Map service to find the location of user and violation on the map and sending it to authorities).

[G3] Allow user to get useful information about general information and news about streets situation.

[G3.1] Highlighting and warning most accident potential areas.

[G4] Give access the municipal agent details of violation

[G4.1] the image must be clear and without noises.

[G4.2] the image must focus on area that violation exactly happened and ignoring the rest.

[G4.3] information of violation which is sent by user, should be represented in a table.

[G5] Give access to municipal agent a panel for general announcements.

[G5.1] A section for representing the result of data mining processes on stored statistics and dividing and labeling city areas to Green: approximately without violations, Yellow: low present of violations, Red: high present of violations.

[G5.1.1] An option for categorizing type of violations.

[G6] Guiding municipal agent for future determinations.

[G6.1] Complete access to (recent and old) reported information in statistic and searching way.

[G7] The system must allow user to report new violation immidately after submited one before.

[G8] The municipal agent must reject the violation report if the photo is in poor quality.

[G9] The system must reject the violation report if the place is not exact.

[G10] The system must reject the violation report if user alter the report after submitting.

[G11] The system must not accept requests of user to change report after submit it.

[G12] The system must comunicate to the municipality system.

[G13] The system must be able to show zone with hight rate of violation based on type.

[G14] The system must be able to show car number with high rate of parking violation.

## **Definitions, acronyms and abbreviations**

### **Acronyms**

* RASD: Requirement Analysis and Specification Document.
* API: Application Programming Interface

### **Abbreviations**

• [Dn]: n-domain assumption.

• [Gn]: n-goal.   
• [Rn]: n-functional requirement.

**Definitions**

• User: he is a user of the app. when he requests to sing up to app he should insert the following information: Name, Family, Telephone number.

• Traffic Violation: Traffic violations occur when drivers violate laws that regulate vehicle operation on streets.

Type of violation:it can be 5 type of parking violation: Double parking,

Double parking: is a traffic offence categorized under parking violations. It occurs when a car parks parallel to another vehicle parked against a curb.

Handicap parking: is a traffic offence categorized under parking violations. It occurs when a car parks on that spots are set aside for people with disability.

Residential parking: is a traffic offence categorized under parking violations. It occurs when a car parks where parking zones are controlled or designated for use by residents living nearby.

Sidewalk parking: is a traffic offence categorized under parking violations. It occurs when a car parks in sidewalks are meant for use by pedestrians.

Special parking: is a traffic offence categorized under parking violations. It occurs when a car parks a place that are designated for specific vehicles.

• System: it is the new system we will create with the database.

• API: application programming interface; it is a common way to communicate with another system.

## **References**

• Specification Document: “Assignments AA 2016-2017.pdf”.

• GPS Performances

• IEEE Std 830-1993 - IEEE Guide to Software Requirements Specifications.

## **Overview**

## This RASD is composed by six parts and an appendix:

## 1. In the first part an introduction to the problem is given listing all the identified goals and providing some base information in order to better understand the other sections of the document.

## 2. The second part consists of an overall description of the system in which its boundaries are identified, and the actors involved in the system’s usage lifecycle are listed. The boundaries are given providing all the necessary assumptions: both the ones required in order to better understand the customer’s specifications given and the ones that will hold into the system and now on considered as true.

## 3. The third part is composed by the specific requirements identified, both functional and non-functional.

## 4. In the fourth part a list of eight scenarios is provided; each of them describes a particular situation with the system might have to cope with.

## 5. The fifth part is entirely composed by the UML diagrams that model the system in details.

## 6. Sixth part is embodied with the Alloy model of the system and includes all the relevant details; a proof of consistency and an example of the generated world are also provided.

## 7. The last part is accessory and contains a list of the tools used to redact this document and its contents, and a detailed report of the hours spent to do so.

**User characteristics**

**Actors**

* *Visitor:* a person using PowerEnJoy (through the mobile app or the website) without being registered. The only thing he/she can do is proceeding with registration.
* *Registered User / User:* a person passed through a successful registration process and now able to use all the safestreet services. He/she can login to the system and, after that, use all the platform's functionalities.
* *municipal agent:* an employee of local trrafic police able to maintain and update the system. Registration for this kind of users is not possible and they have to be added directly during system's installation process.

**Constraints**

**Regulatory policies**  
The system will ask for users' payment informations and obviously, in addition to store them safely.  
Moreover, the system will have to ask for users' permission in order to retrieve and use their positions (at least in a first implementation) storing them.

The system will have to ask for users' permission in order to access to camera and use their it (at least in a first implementation) storing them

**Hardware limitations**

- iOS or Android smart phone

- 2G/3G/4G connection  
- GPS

• Web App  
- Modern browser able to retrieve user's location

**Interfaces to other applications**

…?

**Functional requirements:**

**Assumptions, dependencies and constraints.**

**Domain assumptions**

* Credentials that a visitor has to provide to become a registered user are: name, surname, telephone number and username.
* In order to access to the system, he has to provide the username and password associated to him/her.
* Every special parking area is a No parking area.
* A No parking area is a special parking area that no one can park a car there.
* A restricteced parking area is a special parking area that people with disabiliteis one can park a car there.
* As soon as the system send violation report the user can create another violation report again.
* fining a car to the system consists in adding its license plate, owner and model, and in polic system to communicate with the safestreet.
* The username must be unique.
* Smartphone of user is equipped with cammera.
* When the system shows a report of violation in a certain position it means that it's actually there.
* The registration number of the vehicle involved in the violation must be legible in the photo.
* The app has a provision to take photos of the violation
* The violation can be captured on the phone camera or the photo can be uploaded from the gallery.
* The plate number,type, time and place of violation will be sent to the police
* when the location setting (GPS) in the device is on active mode.
* Once a traffic violation is successfully submitted, an electronic fine-ticket will be generated by the municipal agent.
* Using the registration number of the vehicle, the municipal agent collect the details of the vehicle owner to issue the fine.
* For every report violation, details about the violation are correctly entered.
* For every report, photo about the violation are clearly captured.
* Accurate violation scene locations are known by GPS.
* All the GPS always give the right position.
* Current Italian car registration plate scheme is: "AA 000 AA"; on left the common EU design with the country code, on right the year of registration and the provincial code.

**Dependencies:**  
[D1]: The safestreet will depend on Map service API’s to define the place of violation.

[D2]: The app is dependent on the camera, on the smartphone and the smartphone, to be able to roll out their services. For that reason, it is important to have a close relationship with them.

**Constraints:**  
[C1]: Users should capture photo with high quality and in right angel.

2.2 Product functions:

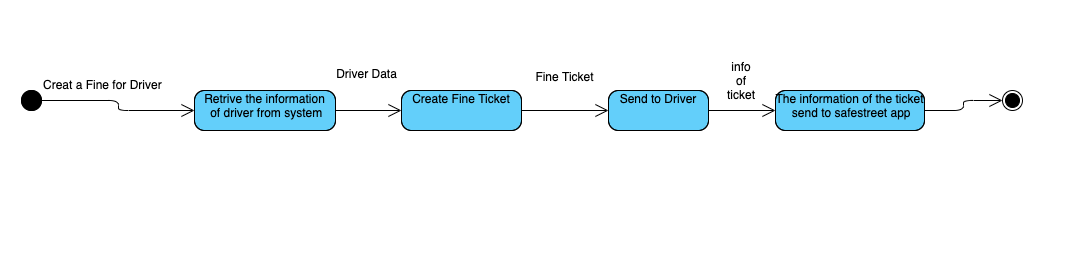
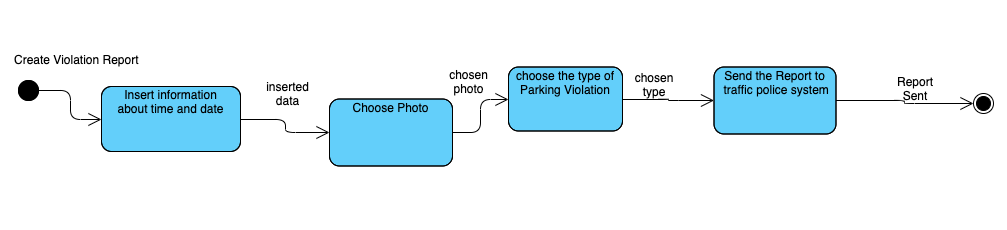
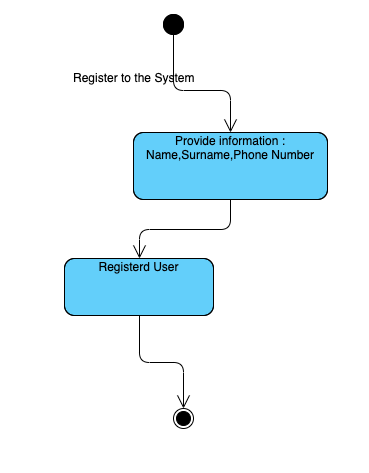
Considering all of the presented goals of the SafeStreet system, most of the functions of the system can be divided into 2 services. In the next section, they are listed and more accurately defined, on the basis of the goals mentioned earlier.

BASIC:

In this service, the user will be able to register themselves. Once they have registered, the users  
will be able to log-in on the SafeStreet application which will provide the user on their smart-  
phone a ability to report a traffic violation, also user can see the area where is place with most parking violation, also traffic polic can acsess to the places with hight rate of violation and also filiter the by car that has large number of violation.

ADVANCED\_2:

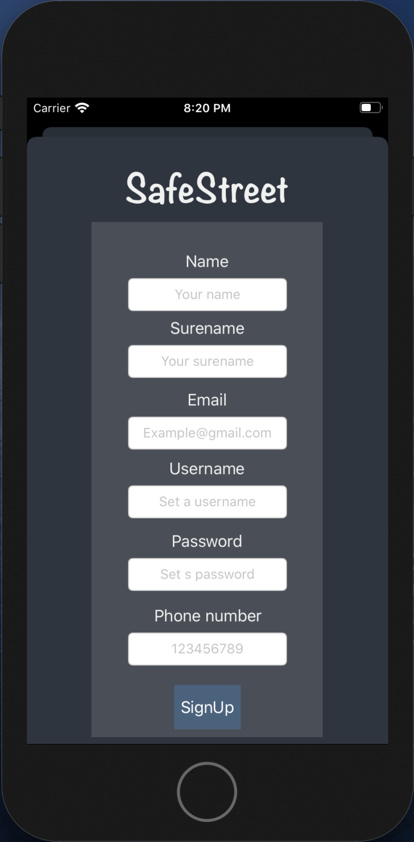
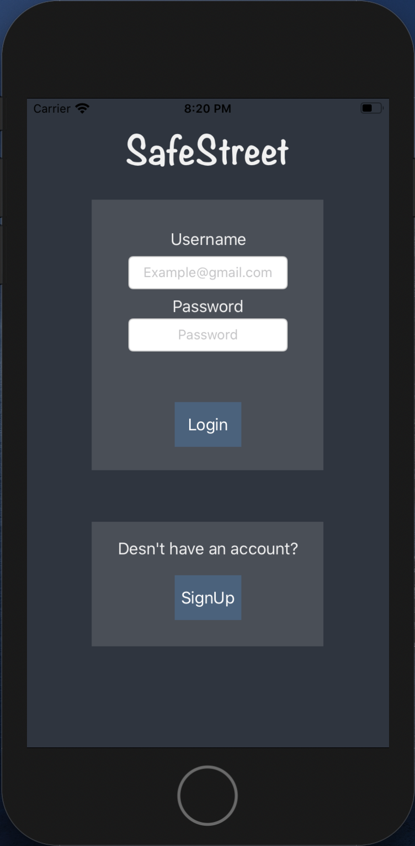
In this part of the system municipal agent can define a fine based on the data that retrived from Safestreet App, and also municipal agent is sure the custody that have been sent to system, did not under go on changes, The safestreet system will be able to provide connection with the external parties to fine driver. municipal agent will be able to not take action if the report and recived photo of the driver be a below a certain safisfaction. The system shares a certain amount of fine data and the location data with the safestreet. The system of the safestreet will analyse the data and it will provide a statistical report to the hightlight area and type of violation.

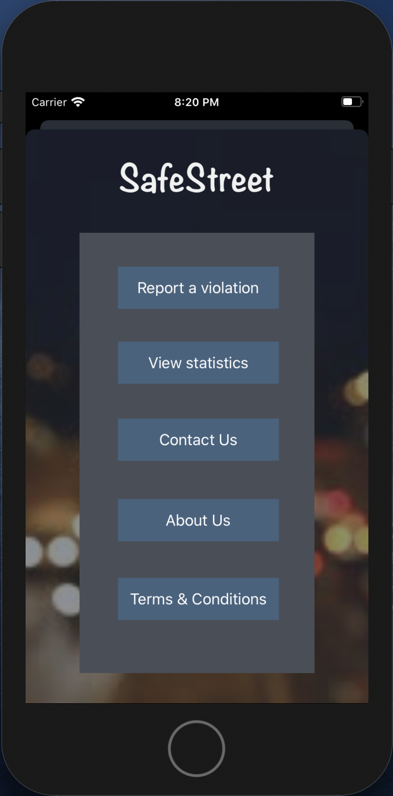
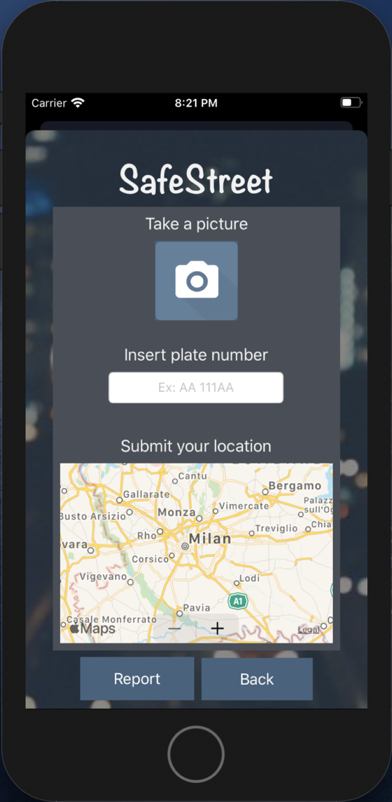


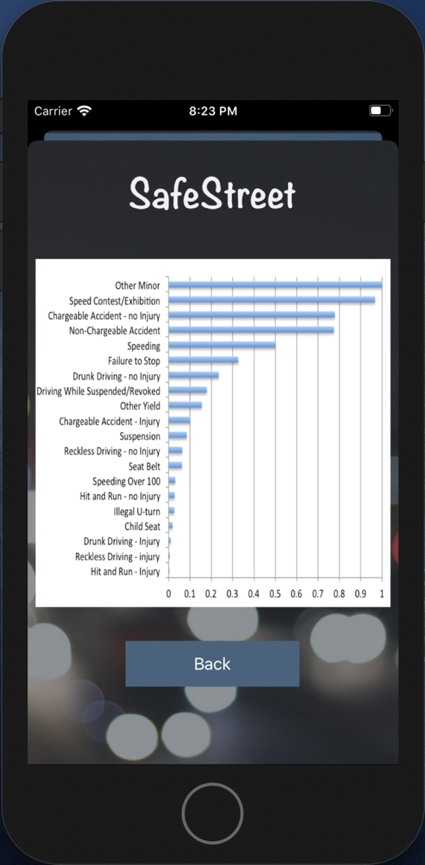
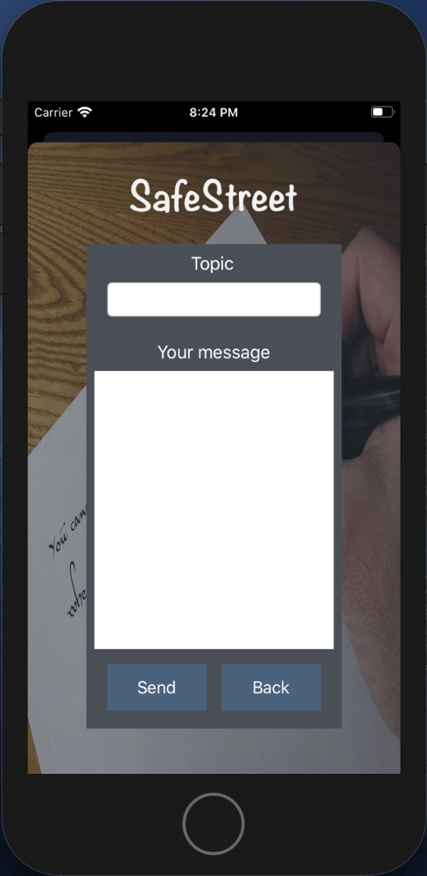
**SPECIFIC REQUIREMENTS**

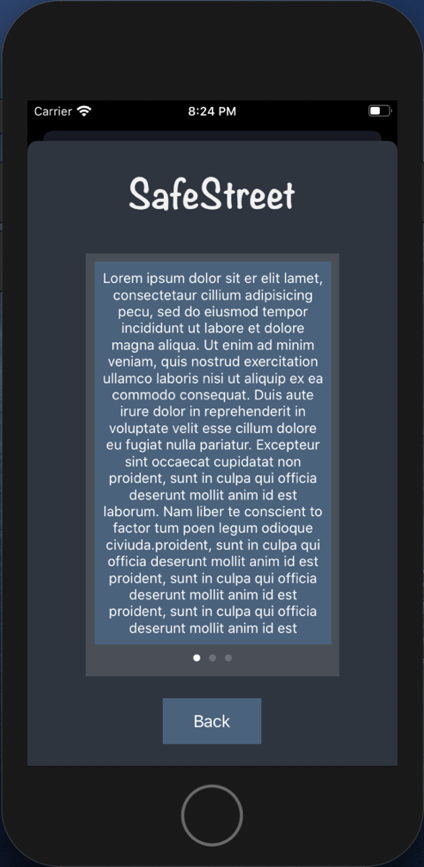
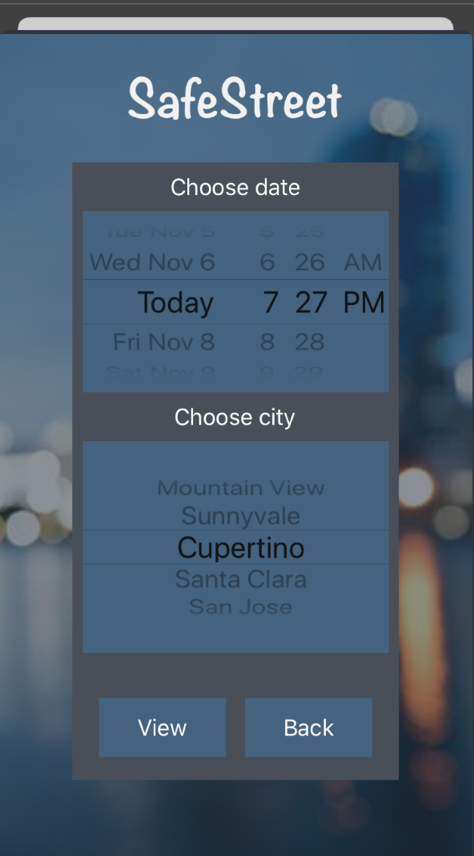
**A. External Interface Requirements**

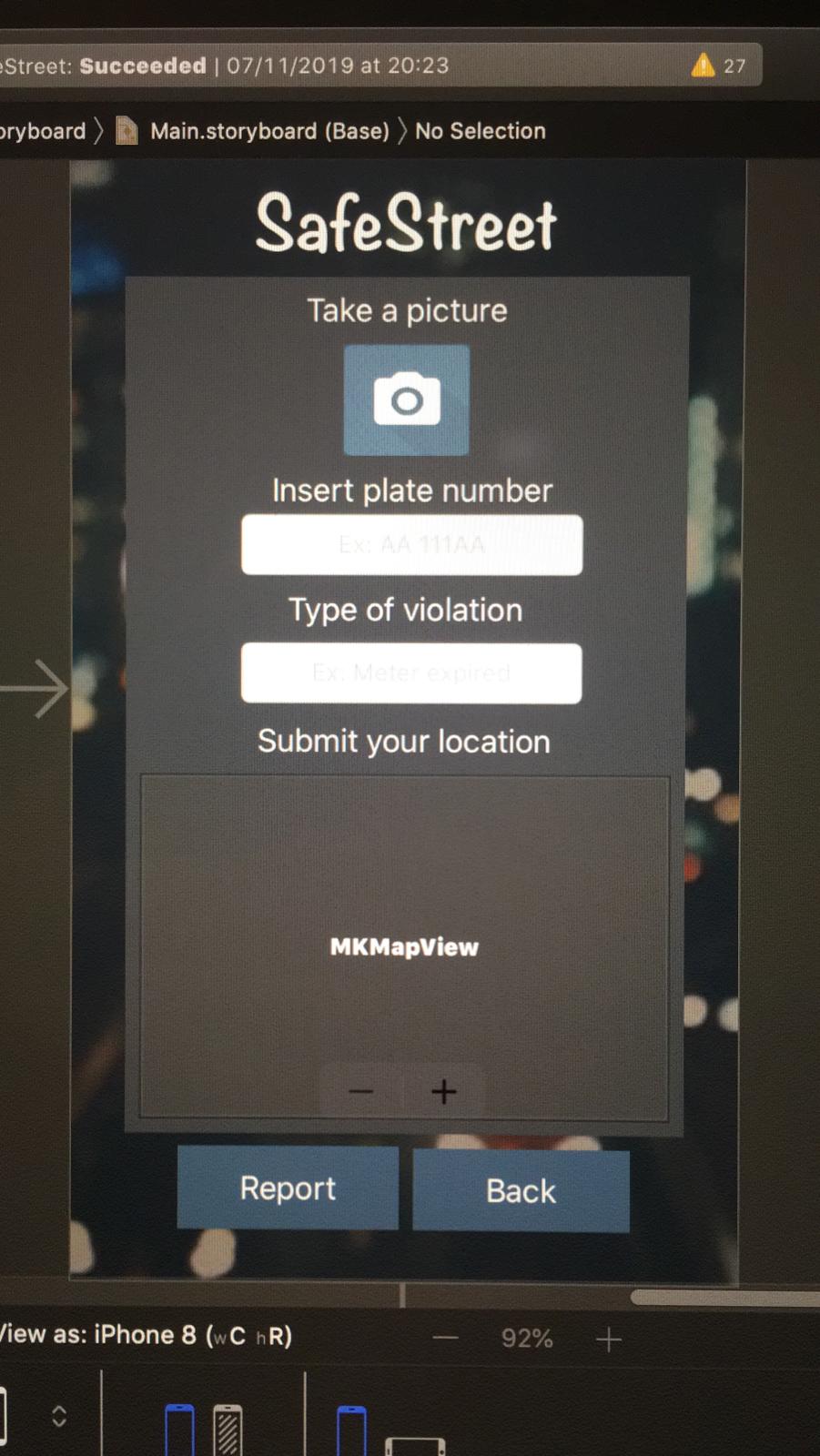
**A.1 User Interfaces**

** **

****

****

****



**A.2 Hardware Interfaces**

It is just a recommendation for other versions of the software. Municipality should assign each car an RFIDs which is located inside the car behind the front window. Regarding this, the reporter has to scan the RFID code and attach it to the report. So that the municipality can check if the number of the car is matched with the rfid code or not and find out the validation of the report.

**A.3 Software Interfaces**

This application does not need a software interface.

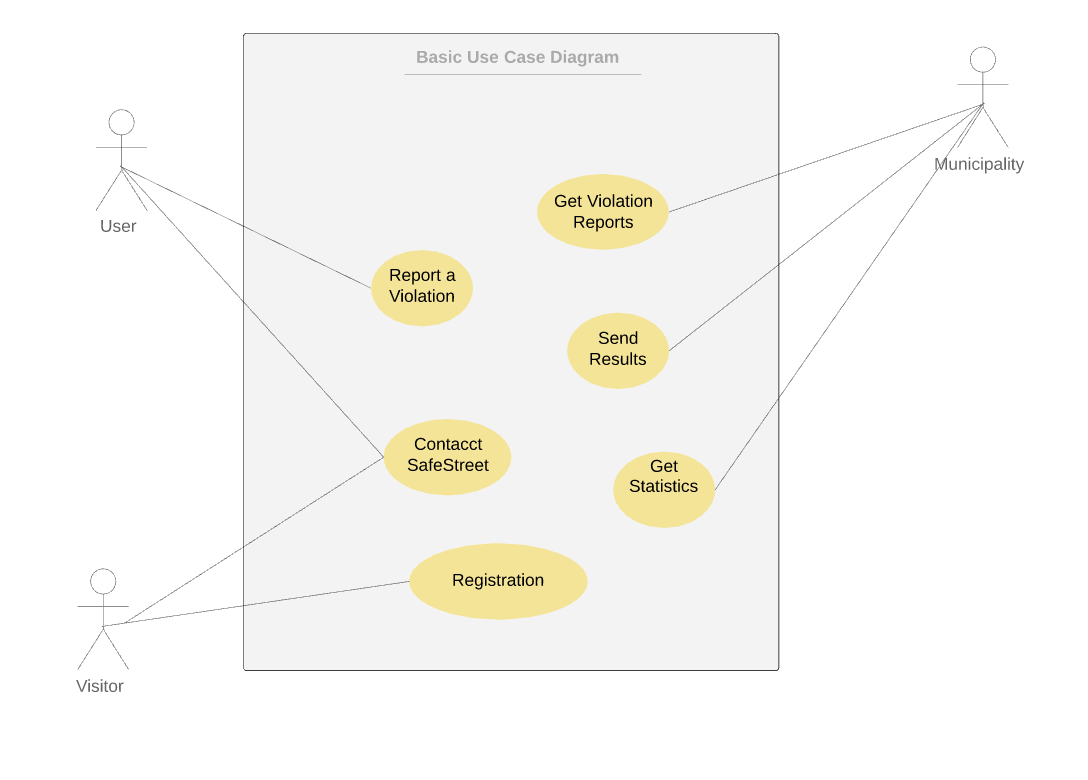
**A.4 Communication Interfaces**

For communication the application uses internet connection.

**B. Functional Requirements**

**B.1 Goals and requirements**

**B.2 Usecase Diagrams**



**Use Case Scenarios:**

**Scenario one: Registration**

Jack is a public transport bus driver. It has occurred a lot when he arrives at a bus station, finds a car parked in that bus station. He has heard about SafeStreet and decides to take advantage of this app to notify authorities about drivers who are not law obedient, so he signs up for the app. While he stops at bus station and waiting for passengers to board, he carelessly takes a picture of a car that parked in that bus station. Since the picture doesn’t fulfill the required conditions of what app has specified, jack’s report is denied.

|  |  |
| --- | --- |
| **Name** | Registration |
| **Goals** |  |
| **Actors** | Visitor |
| **Input Conditions** | There are no entry conditions |
| **Event Flow** | 1.The Visitor on the home page clicks on the “Sign UP” button to start the registration process.  2.The Visitor fills all the mandatory fields and provides his/her payment information.  3.The Visitor clicks on the “Confirm” button.  4.The system saves the data.  5.The system sends an SMS to the new User with the password. |
| **OutputConditions** | The Visitor successfully ends the registration process and become a new User. From now on he/she can log in to the application providing his/her credentials and start using SafeStreet. |
| **Exceptions** | 1.The Visitor is already a User.  2.The Visitor inserts not valid information in one or more mandatory fields.  3.The Visitor chooses a username that has already been taken by another user.  4.The Visitor chooses an email that has been associated with another user.  All exceptions are handled notifying the issue to the Visitor and taking back the Event Flow to the point 2. |

**Scenario two: Report a vilation**

Alex is a handicap driver. It’s crucial for him to be able to park his car in handicap parking slots since he has to go to his office by his car every day. Unfortunately, most of the time there are careless drivers who doesn’t respect this matter and park their car in those parking slots. So, Alex decides to exploit SafeStreet application do notify authorities about those rough drivers. One day he faces this situation so after Singin he takes a picture of situation along with that car’s plate and successfully reports this minor offence.

|  |  |
| --- | --- |
| **Name** | Report a Violation |
| **Goals** |  |
| **Actors** | User, Municipality, Car |
| **Input Conditions** | The User is already on the home page. |
| **Event Flow** | 1.The User inserts Login information in the SafeStreer.  2.The User clicks on the “Log In” button in order to access.  3.The User fills all the mandatory fields and take a picture of the violation.  4.The User clicks on the “Confirm” button.  5.The system saves the data.  6. The system processes the data and extracts the properties of the car.  7. The system makes a report.  8.The system sends the report to the Traffic Police. |
| **OutputConditions** | The User is successfully redirected to the Home Page. |
| **Exceptions** | 1. The User inserts a not valid username.  2. The User inserts a not valid password.  All exceptions are handled notifying the issue to the Visitor and taking back the Event Flow to the point 2. |

**Scenario three: Provide API**

Sofia is a traffic control operator in municipality of Milan, and she has been tasked to detect streets with highest violation of double-parking cars where it’s prohibited so that municipality can plan to place elevator car parking to address the issue. She has heard about SafeStreet application which can indicate such streets by the help of users who report such violations such as double-parking. So, she signs a contract with SafeStreet corporation to receive such information by the help of an API which SafeStreet provides.

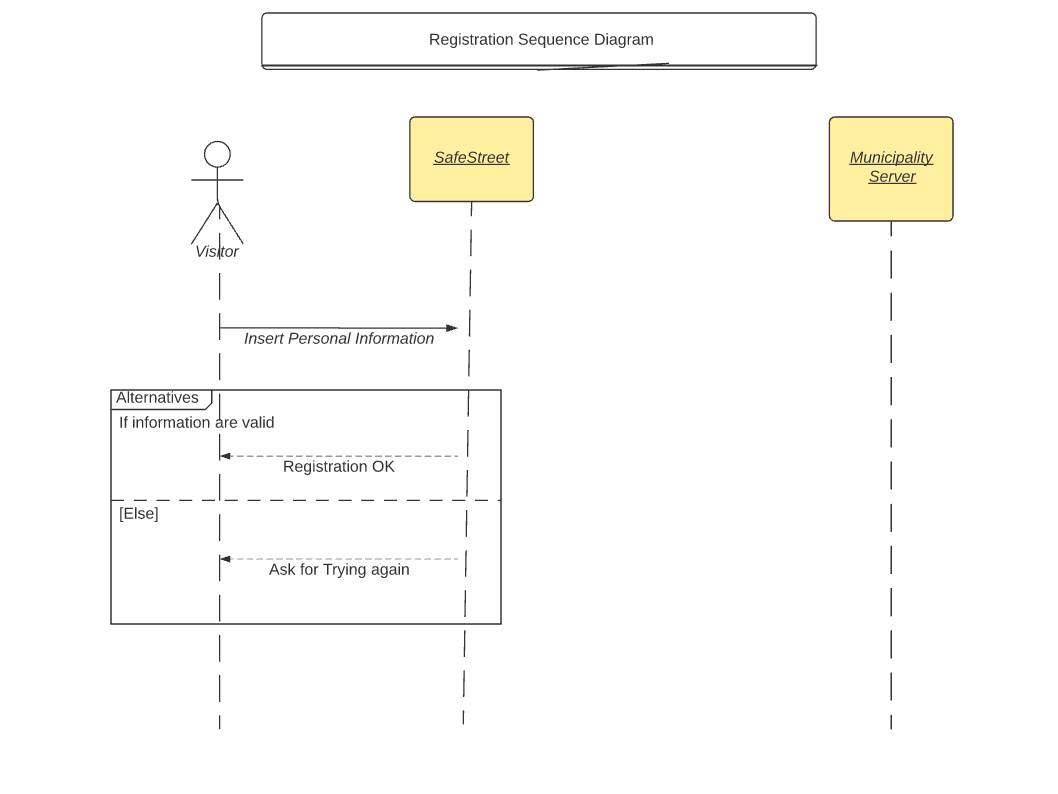
|  |  |
| --- | --- |
| **Name** | Provide API |
| **Goals** |  |
| **Actors** | Third-Parties |
| **Input Conditions** | There are no entry conditions |
| **Event Flow** | 1.The Visitor on the home page clicks on the “Contact Us” button to start  2.The Visitor can see the contact information of the SafeStreet Owners.  3.The Visitor clicks on the “Back” button to redirect to the home page. |
| **OutputConditions** | No output. |
| **Exceptions** | No expectation. |

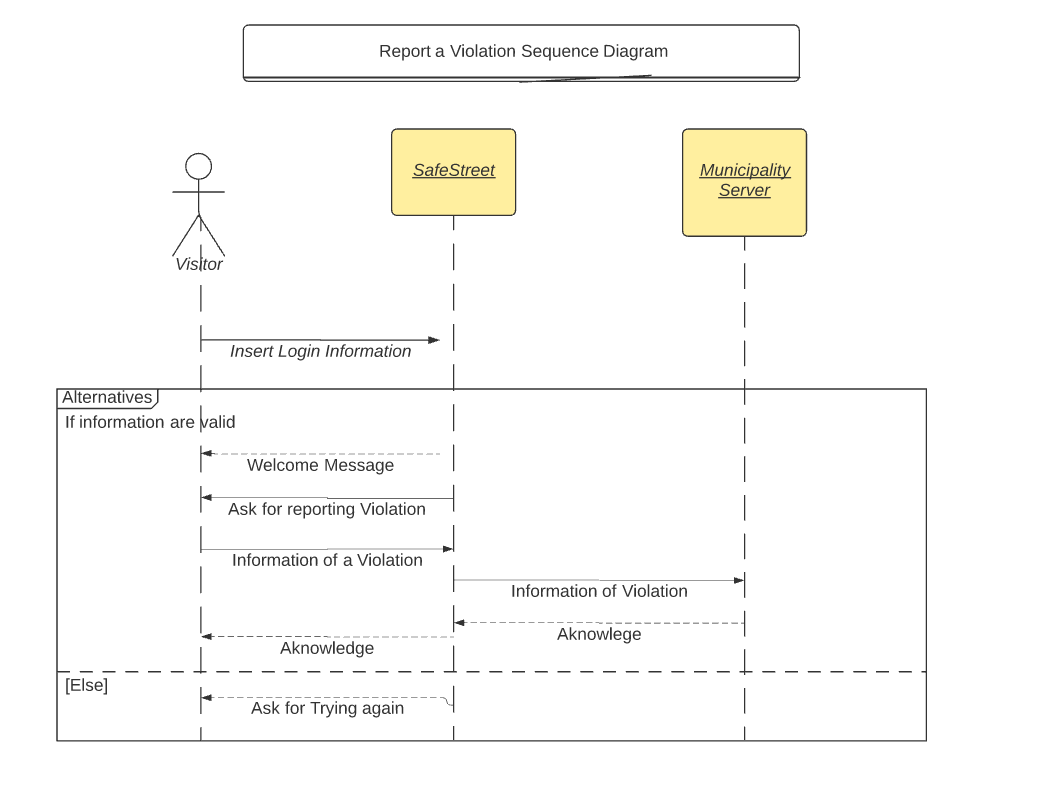
**Scenario four: Notify Municipality and Get ticket info**

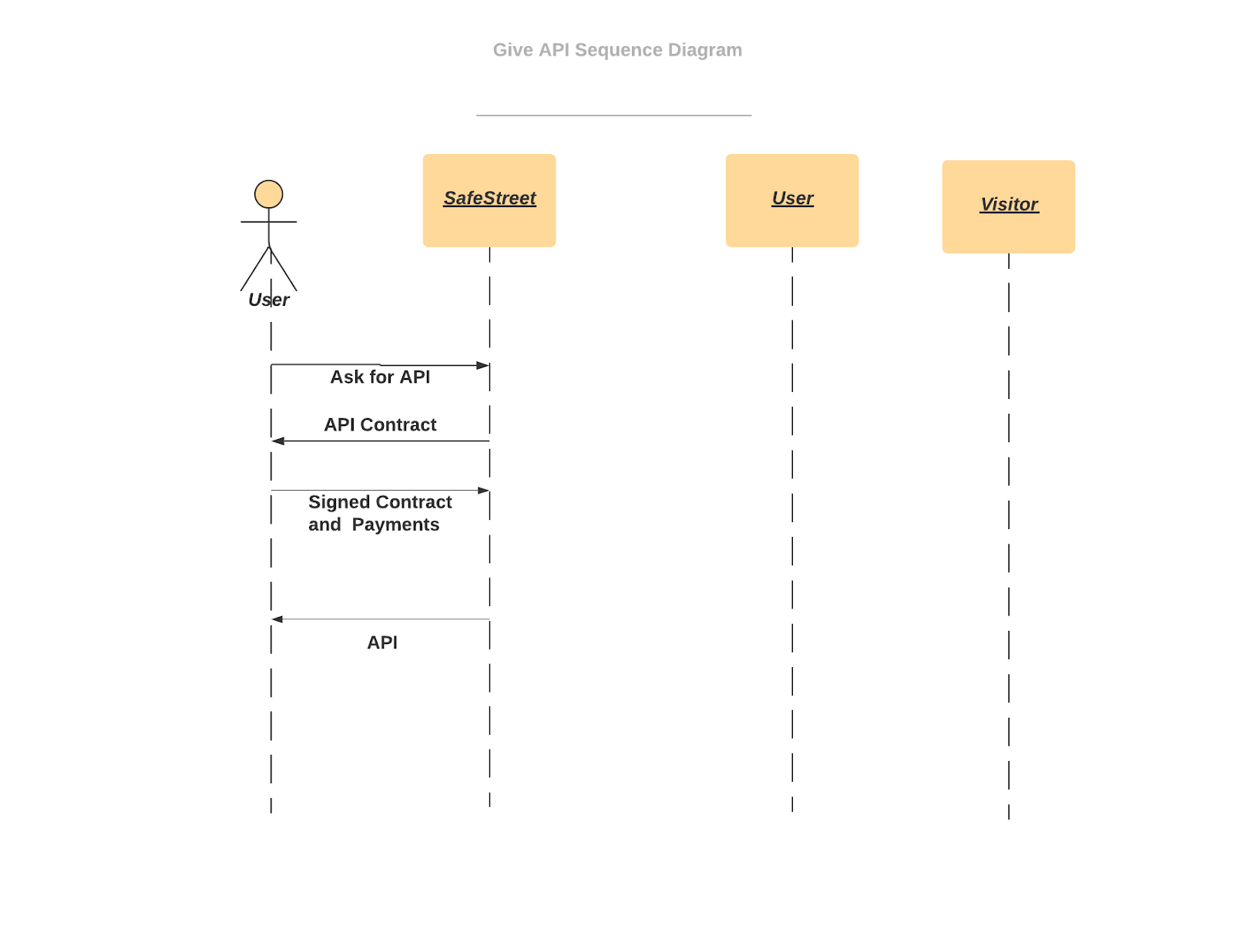
Due to a contract singed by SafeStreet corporation and municipality of Milan, SafeStreet is now responsible to provide municipality with reported traffic violations collected by its users and in return municipality is responsible to provide Safestreet with the tickets they issued for rough divers. People at SafeStreet plan to retrieve these thickest so that they can indicates how many of those violations reported by its users resulted in issuing thickets by municipality so that they can underline reliability of their user’s reports and further planning to improve the overall performance of SafeStreet application.

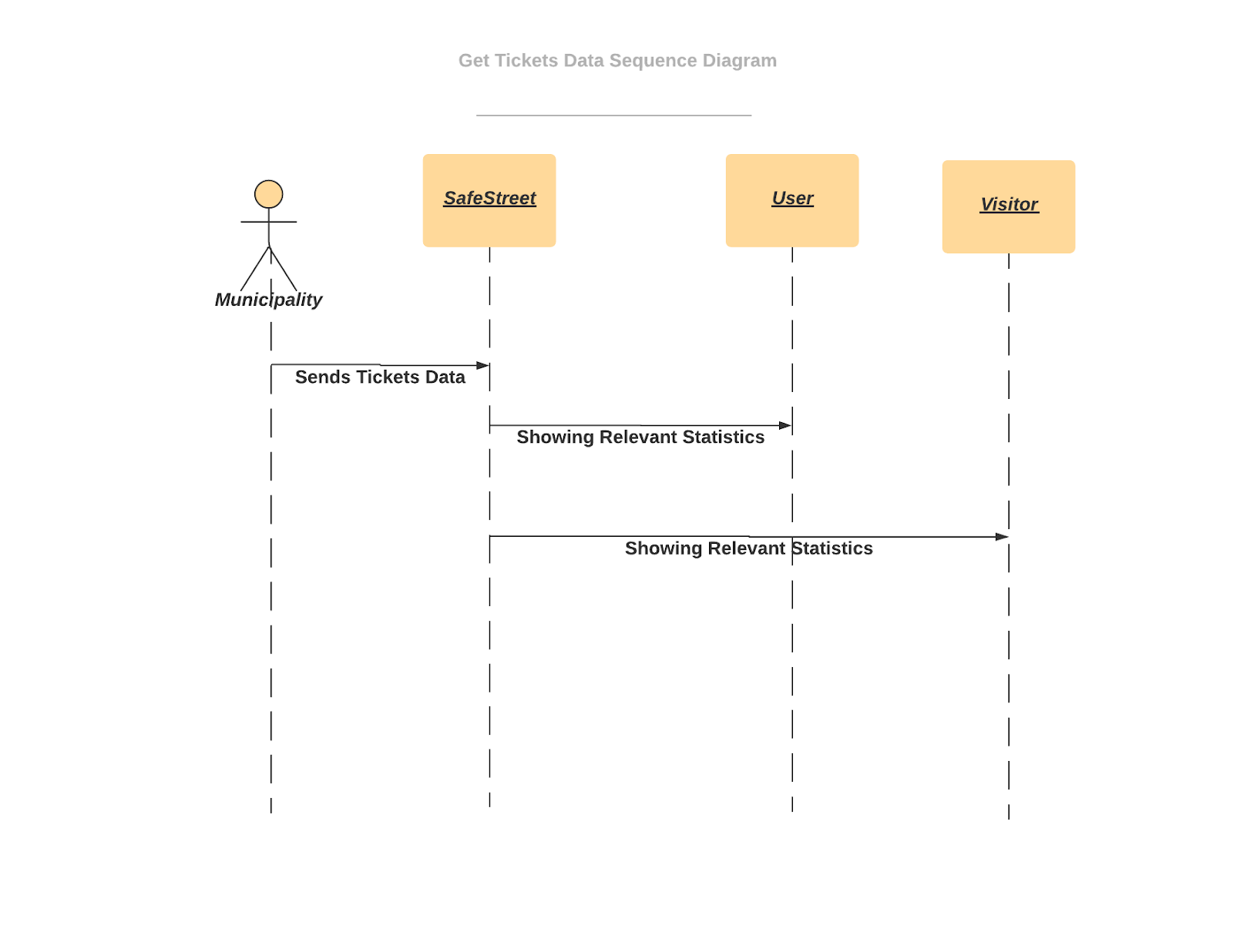
|  |  |
| --- | --- |
| **Name** | Notify Municipality and Get ticket info |
| **Goals** |  |
| **Actors** | Municipality |
| **Input Conditions** | 1.Municipality have received data from SafeStreet  2.Municipulity considered the data and had a reaction on each record |
| **Event Flow** | 1.The Municipality sends new data to SafeStreet  2.SafeStreet saves the data.  3.SafeStreet mine the data and update the statistics  4.Visitors and Users can see the updated statistics |
| **OutputConditions** | The data from municipality saves in the Safestreer Database. |
| **Exceptions** | 1. The process of receiving data fails,  In this case the received data is deleted and the Municipality redirects to line 1 of event flow. |

**B.3 Sequence Diagrams**









**C. Performance Requirements**

**This application deals with sensitive information and the municipality may charge people by the reports of the application. It has to consider especial methods for preventing people to report not valid violations. Therefore, we allocate a number to each person. This number changes according to the reaction of the municipality to the reports of that account. To explain more, if the municipality decide the report from user x is valid the number for the user x increase. As a result we rank the reports based on this numbers when we are sending the reports to the municipality.**

**D. Design Constraints**

**D.1 Hardware limitations**

**This application does not need cutting-edge hardware technologies. The only requirement is having a smartphone.**

**•Android or Iphone IOS for smartphones.**

**•GPS service.**

**•3G/4G connection.**

**.Camera**

**D.3 Any other constraint**

There are no any other constraints for the application.

**E. Software System Attributes**

**E.1 Reliability**

The application provides a reliable service, as it limits the users to take a picture of the violation at the moment and does not let them to upload pictures from the past so that we are sure the pictures are from the moment and not edited. All the data are sent to the municipality without any additional manipulation so that they can trust the records.

**E.2 Availability**

The availability of the application depends on the device state, whenever it connects to the internet, the reports which are saved on the device are sending to the SafeStreet servers and save in a buffer. Each night on 12am the buffer is sent to the server of the municipality. Since communicating data using this application is not a real time service, we do not need a high tec server just need enough space for saving records and also not the pictures.

**E.3 Security**

The systems uses hash functions to store users data. In addition, in the reports to the municipality the personal information of the users are not sent.

**E.4 Maintainability**

The application will be implemented and designed with a complete documentary which makes it easier to maintain and understand the application for the future. Additionally, the user interface designed simply so that every user and third parties in the near future will figure out the usage. Furthermore, as the system sends all the data of the day to the municipalities, not much data will be loss in case the system breaks down. Also the system takes backups each night so the statistics will be safe again.

**E.5 Portability**

The application will be able to run on Android or Iphone systems. SafeStreet wants to focus on both, because this application must be available for all the people in the citizens.

<https://www.lucidchart.com/invitations/accept/9985e121-afce-440c-95b9-20c407b95c4f>

<https://www.lucidchart.com/invitations/accept/40a9f25e-9bd6-44ab-943b-937072677617>

<https://www.lucidchart.com/invitations/accept/0dfa03e5-d8d7-4cf0-926a-7939e519f9b5>

<https://www.lucidchart.com/invitations/accept/9ce54e43-49b2-4ca3-a76e-65aa08ee5b61>

<https://www.lucidchart.com/invitations/accept/19d19ef5-4add-48a0-9e6e-1cfeb0b54135>

open util/boolean

sig string {}

sig Name, Surname{}

sig Email, Password{}

abstract sig User {

name: one Name,

surname: one Surname,

email: one Email,

password: one Password,

accessLevel: one Bool,

minedInfo: some MinedInfo,

}

sig EndUser extends User{

userLocation: one Location,

}{

accessLevel = False

}

sig Authority extends User{

tickets: Ticket,

}{

#tickets >= 1

accessLevel = True

}

sig Location {

latitude: one Int ,

longitude:one Int

}

sig Photo {}

sig Violation {

location: one Location,

addr: some ReverseGioCoding,

reporter: one EndUser,

//type: one string,

photo: one Photo,

licensePlate: one ALPR,

date: one Date

}

sig ReverseGioCoding {

loc: some Location,

//addr: one string

}{#ReverseGioCoding = 1}

sig ALPR { //remember to add somethin to tell it's only one

picture: some Photo ,

// licenseP: one string

}{ #ALPR = 1}

sig Date {}

abstract sig MinedInfo {

violations: some Violation,

}

sig MinedStreet extends MinedInfo{

/\*

// name: some string,

frequency: some Int,

location : one Location,\*/

}

sig MinedOffender extends MinedInfo{

/\*

n\_Violations: one Int,

licensePlate: one Int,

uuid: one Int\*/

}

sig Ticket {

violations: one Violation,

}

fact NoSameGPSForDifferentUsers {

no disjoint u1, u2 : EndUser |

u1.userLocation = u2.userLocation

}

fact NoSameGPSForDifferentReverseGio {

no disjoint revGio1, revGio2: ReverseGioCoding |

revGio1.loc = revGio2.loc

}

fact NoSamePhotoForDifferentViolation {

no disjoint v1,v2 : Violation |

v1.photo = v2.photo

}

fact NoSameViolationForDiffReporter{

no disjoint v1, v2 : Violation |

v1.reporter = v2.reporter

}

fact EachTicketOneAuthority {

all t: Ticket | one au: Authority |

au.tickets = t

}

//Two diﬀerent users can’t have the same email

fact NoSameEmailForDiﬀerentUsers {

no disjoint u1, u2 : User |

u1.password = u2.password

}

fact NoSamePasswordForDiﬀerentUsers {

no disjoint u1, u2 : User |

u1.email = u2.email

}

fact NoSameMinedInfoForDiﬀerentUsers {

all disjoint u1,u2: User |

u1.minedInfo != u2.minedInfo

}

fact EqualUserAndLocation{

#EndUser = #Location

}

//fact {all u: User | some n: Name | u.name = n}

//fact {all u: User | all n: Surname | u.surname = n}

fact EndUserRelateOnlyMinedStreet {

all user: EndUser |

user.minedInfo = MinedStreet

}

//fact { all mined: MinedInfo | one us: User | us.minedInfo = mined}

//fact { all us: User | some mined: MinedInfo | us.minedInfo = mined}

fact EachPhotoBelongsAlpr {

all ph: Photo|

one alpr: ALPR |

alpr.picture = ph

}

////

fact EqualLocationForEndUserAndGio {

one revGio: ReverseGioCoding|

one u: EndUser |

revGio.loc = u.userLocation

}

fact EqualLocationForViolationAndEndUser {

all viol: Violation |

some u:EndUser|

viol.location = u.userLocation

}

fact {all d: Date, viol: Violation | viol.date = d}

fact EachViolatioContainsOneTicket {

one t: Ticket , v:Violation |

t.violations = v

}

pred show {

}

