1. Introduction
   1. Purpose
      1. General Purpose

This document represents the Requirement Analysis and Specification Document (RASD). In this document we will explain SafeStreets. This will be done by a detailed presentation of the proposed solution and its purpose, listing its goals, and the requirements and assumptions through which they will be achieved.

SafeStreets is a public interface aimed to public-spirit citizens who want to help keeping the streets clear. This S2B intends to provide users with the possibility to notify authorities when traffic violations occur. This materializes using a platform through which users can upload pictures of streets violations, in particular parking violations.

There could also be other types of issues that a customer can report, for example speed violations, accidents, non-respected traffic lights. They’re reported in different ways.

The S2B also has a map, based on Google Maps, on which some areas are highlighted with different colors according to the number and types of violations reported (for example first the user can choose the type of violation, then the map shows different areas with different colors: a red area means that a lot of the chosen violation have occurred, yellow is medium quantity, green one means very few).

Once the violation is sent, its data are stored in SafeStreets center and analyzed by the software, in order to retrieve information to update the map.

The customers of the application are both singular users and authorities, for example the Police Department, that can find the S2B useful in order to maintain the public order.

An important point is that a user can eventually report fake violations. First, the application allows the user to report a violation even if the user’s geographical position and the violation’s positions are different (for example the user sees an illegal parked car while he is jogging but doesn’t want to stop his run to make the signalization, so he updates the violation once he gets home. This means that he could give a non-accurate position of the illegal parking). Secondly, there could actually be some users that find funny to make reportings of wrong violations, for now there is nothing we can do to fix this.

Finally, SafeStreets wants to offer a service exploiting the information of the municipality, if it allows users to retrieve the required information. The application crosses the information given by the municipality (only accident info), which are reliable, and the ones given by the users, which are not. Then it updates the map and make suggestions regarding possible solutions to prevent violations (only in red areas).

* + 1. Goals

-[G1] The application must allow private users to send reports of streets violations. In particular users can send reports even if they aren’t in the position where the infraction occurred.

-[G2] The application will have to store the information about violations and complete them with suitable metadata.

-[G3] The application must allow both end users and authorities to mine the information stored. This is done by coloring the map based on the number of violations.

-[G4] The system must update the map after every reporting.

-[G5] The system must be able to cross information received from municipality with its own data.

-[G6] The system must suggest to municipality possible interventions to prevent accidents.

-[G7] The system must be able to send the verified parking violation to the municipality.

-[G8] The system must allow authorities to send accidents information to SafeStreets application in order to update the map.

* 1. Scope

As already mentioned, the SafeStreets system is made to provide users a service to report streets violations and a map where is possible to watch the areas in a safeness key.

The users, once they sign in with a proper username and password, are in the system and can use all the services offered by the application.

As first, the system will ask to choose either to take a look on the map or to report a street infringement. In the first case, it will be asked if the user wants to share the GPS signal with the application, so to be more precise about the areas close to him. As mentioned before, on the map are highlighted different areas with different colors, based on the frequency of violations. Then the user can interact with the map. If the user taps on a highlighted area, the number of infringements is shown.

It is also possible to select the type of violation and the interval of time (today, last week, last month) in which the user is interested in watching the map. By default, if the user doesn’t select any type of violation or interval of time, it will be shown the map of all types of violations together occurred in the last month.

Regarding to the reporting of violations, the systems allows to select the type of reporting (parking, speed, traffic light violation or accident), only in case of parking is possible to send a picture. The system will ask for a picture with visible license plate, if the user doesn’t send one, he can type the license plate by himself (not mandatory).

After any reporting, the user is asked if he wants to keep anonymous his reporting or not. If he chooses to be anonymous, on the map won’t be shown his username along with his reporting, otherwise his username will be shown.

To prevent any misuse of the reporting violation system, if the user sends a picture with visible license plate and the application analyzing pictures algorithm fulfill in recognizing the it, then that car will be shared with authorities. Otherwise, either if the user only types the license plate without sending a picture of it or the application analyzing pictures algorithm can’t recognize the plate, it is considered not reliable, so the car won’t be shared with authorities. It will only be added on the map.

For speed infraction, traffic light violation or accident reporting, is not possible to send a picture, in order to prevent the use of phones while driving. The user can only choose the type of violation.

The user has also to choose the time and geographical position of the infraction, which is independent from the type of violation. It’s either possible to select the user’s position (using the GPS) or selecting a position on the map or typing the address.

The user can’t open the gallery on his smartphone to update a picture. This is for preventing the update of fake parking pictures that were shot in a time we can’t know. The user can open the camera through SafeStreets application and, either take the picture and send it, or take a picture and, if he doesn’t have time to send it, the S2B puts a timer of 2 hours on the photo, during which the user can re-open the application and update the picture. If the timer expires the picture gets eliminated.

If the municipality offers a service through which is possible to retrieve information about the accidents, SafeStrets analyze them and crosses them with the SafeStreets’s ones. This is done in order updated the map, which means identify unsafe areas (SafeStreets sees the municipality information as they were reliable reportings) and to give suggestions to the municipality itself to prevent more violations (for example add barrier between the bike lane and the part of the road for motorized vehicles to prevent unsafe parking).

* 1. Definitions, acronyms, abbreviations
     1. Definitions

·User: the costumer of the application that send reports. It could be a private citizen or an authority like municipality. In this case the use of the application will be different.

· application analyzing pictures algorithm: the algorithm that SafeStreets uses for recognizing the license plate of the car object of the violation.

·municipality: this is the municipal police section that collaborates with SafeStreets for preventing accidents and violations.

* + 1. Acronyms

·RASD: Requirement Analysis and Specification Document

·API: Application Programming Interface

·GPS: Global Positioning System

·S2B: Software To Be

* + 1. Abbreviations

·Gn: nth goal

·Rn: nth requirement

·Dn: nth domain assumption

* 1. Revision history

·Version 1.0:

-First release.

* 1. Reference documents

- Specification document: “SafeStreets Mandatory Project Assignment”.

- IEEE Std 830-­‐1998 IEEE Recommended Practice for Software Requirements Specifications.

- Examples documents:

- RASD Sample from A.Y. 2015-2016.pdf

- RASD Sample from A.Y. 2016-2017.pdf

* 1. Document structure

The RASD is composed by 5 sections.

* Section 1:

it is the introduction of the RASD in which the problem is presented informally with natural language. It provides base information such as the product to develop and the application domain. The scope part is an analysis of the world and the shared phenomena.

* Section 2:

it presents an overall description of the project. It describes external interfaces, summary of major functions, constraints, assumption and dependencies of the S2B. Furthermore, a class diagram and some state diagrams are provided to make stakeholders better understand the project, but even for giving more details on shared phenomena and the domain model.

* Section 3:

this is the body of the document. It first describes the interfaces requirements. Then it lists some scenario to show how system works in real life situations and functional requirements. Lastly, we have nonfunctional requirements such as performance requirements and design constraints. This section will be useful for the development team.

* Section 4:

here we have the Alloy formal description of the problem that includes all the relevant details.

* Section 5:

it presents the effort spent for every member of the group.

1. Overall Description
   1. Product perspective

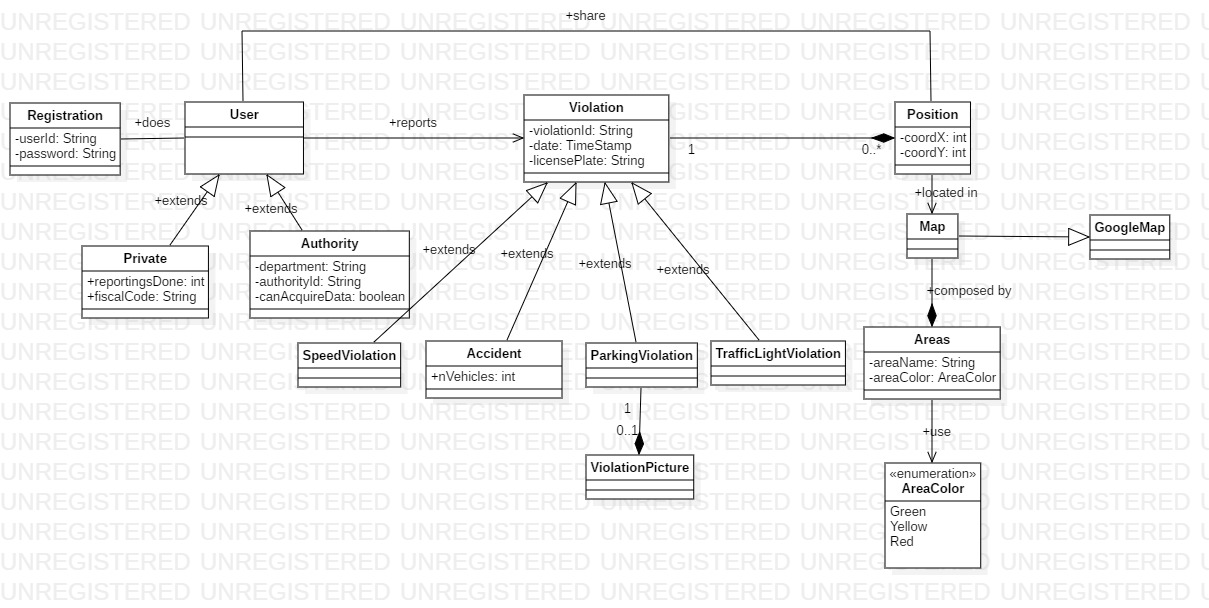
The SafeStreets system is a completely new software application, designed from scratch. It is intended to be used as a mobile application.

The S2B receives reporting of accidents and parking, speed and traffic lights violation. It stores the information on the application data center. Then updates the map every time a reporting occurs. The map as already said is based on Google Maps’ APIs because it offers a very large library of APIs.

The application also works along with authorities. Authorities can only acquire reliable data from SafeStreets. A reporting is reliable only if it is a parking violation reporting, with a picture, and the algorithm to read the license plate fulfils.

Anonymity is an important aspect that must be respected. When authorities acquire data, the username of the costumer that made the reporting won’t be sent. Only the reporting along with the picture, geographical position and time will be sent.

The municipality can also send information about accidents to SafeStreets. The information will be used as they were reliable reportings.



*Figure 1 - Class Diagram*

The class diagram (figure 1) shows the front-end side of the application. Users can be either private costumers or authorities, they must make a registration. Each private costumer has an attribute that stores the number of reportings that he has done.

A user can make a reporting violation. The violation will have a violationID, a date and a license plate, which is not mandatory. A violation occurs in one position, and in one position can occur many violations.

Only with a parking violation a picture can be sent, in the other cases the probability of the user is driving while noticing a violation is very high (almost every accident is sees while driving, for example on the highways. Also speed violation: while driving it’s easy to notice a car that speeds, for example if it overtakes you. Taking a picture while driving is very risky, besides being illegal).

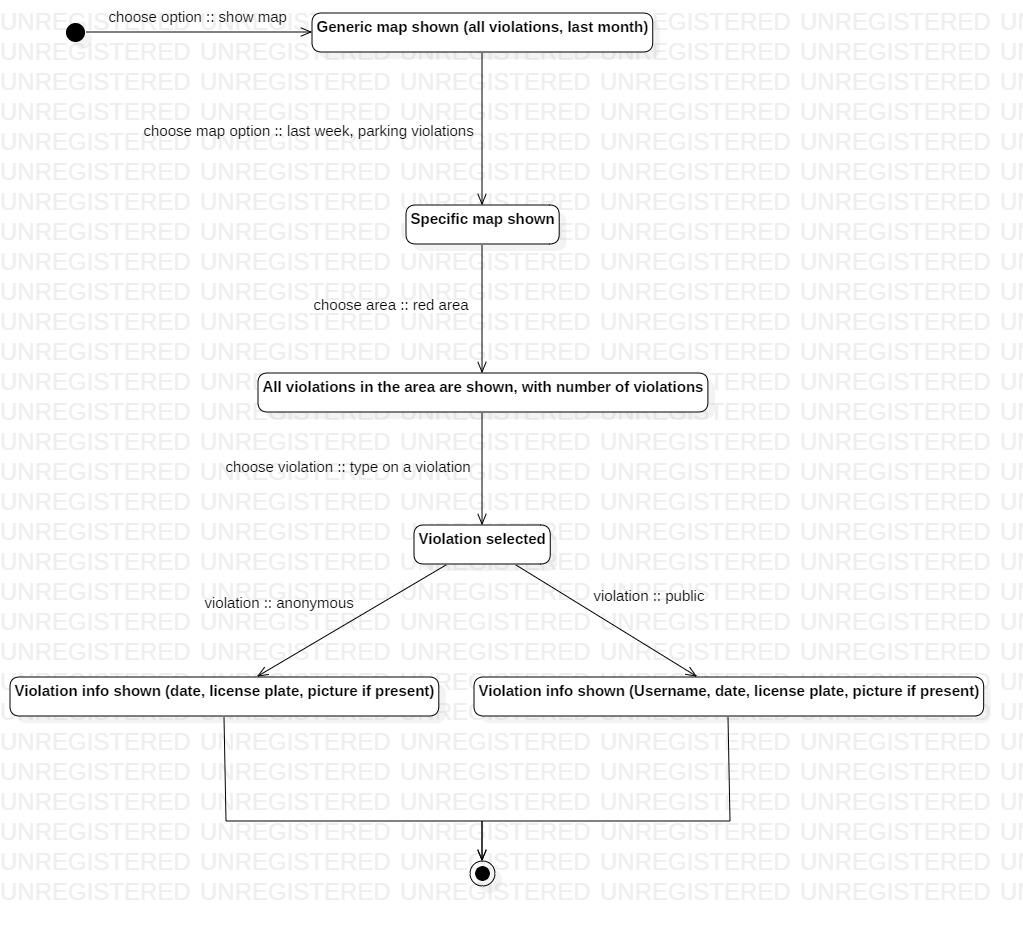
A user can share the position of an infringement (using GPS, if he allows to). All the violations positions are shown on the map, which areas are differently colored (green, yellow, red) based on the frequency of the violations.

Immagine che contiene testo, mappa

Descrizione generata automaticamente

*Figure 2 - State diagram 1: parking violation*

This first state diagram (Figure 2) shows how the application works on a parking violation reporting. The user chooses the type of violation, then inserts the information about it. Once it is sent to SafeStreets, the data is stored, then if a picture is present, the license plate recognizing algorithm is ran. If the reporting is reliable (algorithm fulfills) the data are sent to authorities, otherwise it is only used to update the map.



*Figure 3 - State diagram 2: map usage*

Figure 3 describes how the map works along with selection of the users. Firstly, the map shows all types of violations together which occurred since a month ago from now. Then, the user can choose a specific type of violation, parking violations, which occurred in a chosen time interval, last week. Now the areas on the map regards parking violations that occurred last week, and the user can select by typing on it a colored area, in this example a red one. The area shows the exact positions of the violations (the position selected by the costumer that sent it) and the number of them. The user can type on a specific violation, information about it will be shown, including the date and time of the reporting, along with the picture if present. As already said, after every reporting, the user is asked if wants the reporting to be an anonymous or not. If he chooses it to be anonymous, his username won’t be shown to other users that select his violation on the map.

* 1. Product functions
  2. User characteristics
  3. Assumptions, dependencies and constraints

1. Specific Requirements
   1. External interface requirements
      1. User interfaces
      2. Hardware interfaces
      3. Software interfaces
   2. Functional Requirements

3.2.1 Private

**Scenarios**

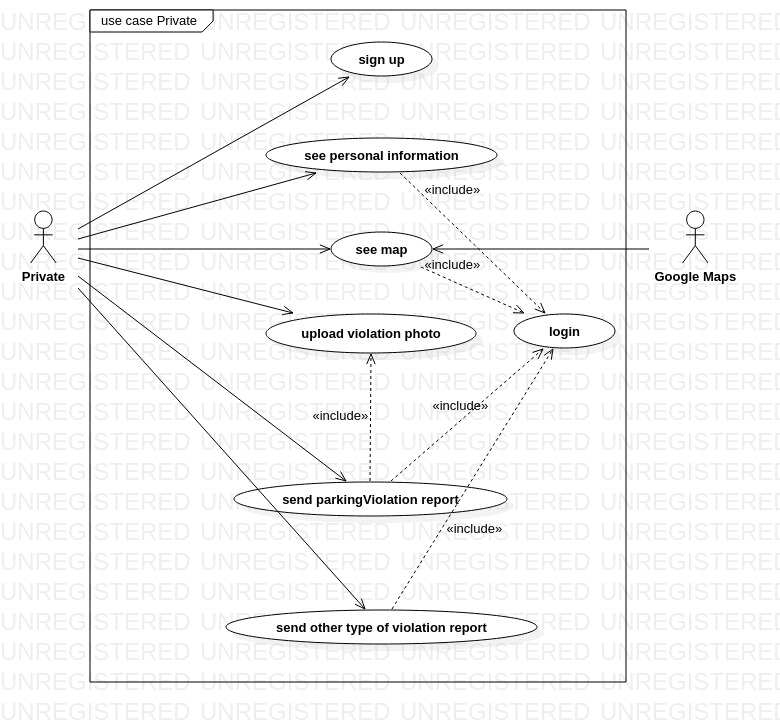
Scenario 1

Mhysa, a foreign student studying Computer Science at Politecnico di Milano, needs to go to the student office to bring some documents. She has a great sense of civic duty, so on her way from Lambrate to the office she notices a car parked on the crosswalk impeding an elderly woman with the groceries to cross the street safely. Mhysa decides to report the car with a parking violation. She takes a picture of the vehicle and types the license plate on the reporting she’s about to send. Then shares her GPS position (which agreed to shared at her signup) and time, deciding to remain anonymous for this reporting. Finally she chooses crosswalk as type of parking violation.

Scenario 2

Alfonso lives in the Navigli area in Milan. On Saturday he has an exam at Politecnico di Milano Leonardo, in Città Studi. He knows there is a public transportation strike organized during the week end, so he chooses to go by car at the exam. He first looks at Google Maps which gives him two possible alternative routes of the same duration. So, to decide, he opens SafeStreet application on his smartphone and takes a look at the map. He knows that SafeStreet colors the map based of the violations reported from users therefore he selects accidents as type of violation, and today as time interval. He notices there are reportings accidents on one route he could have possibly chosen. So he decides to take the other one.

**Use Case Diagram**



**Use Cases**

|  |  |
| --- | --- |
| Name | Sign Up |
| Actor | Private |
| Entry condition | The user has opened the application on his smartphone. |
| Events flow | 1. The user chooses the “sign up” option. 2. The user fills all the mandatory fields and provide the necessary information. 3. The user click on confirm option. 4. The system checks all user fields. 5. The system saves the user data. 6. The system asks to the user if he wants to share his GPS position. 7. The system keeps track of user position if he accepts. |
| Exit conditions | The user is registered and he is able to use the application. |
| Exceptions | 1. The user is already registered. The system warns the user to do the login. 2. The user inserts not valid information in one or more fields. 3. The username is already taken. 4. The mail is already taken   Except point 1, the system handles others exceptions returning at the start of point 2 of the events flow, so the user re-enters all the fields (mandatory or not). |

|  |  |
| --- | --- |
| Name | Login |
| Actor | User |
| Entry condition | 1. The user has already download the application. 2. The user has already done the “Sign up” activity. |
| Events flow | 1. The user chooses the “Login” option. 2. The user enters his email. 3. The user enters his password. 4. The user clicks on confirmation option. |
| Exit conditions | The user is logged in and he can use the application services. |
| Exceptions | 1. The users enters the wrong email. 2. The users enters the wrong password.   In each case the system tells the user what field is wrong and let him re-enter the wrong field. |

|  |  |
| --- | --- |
| Name | See personal information |
| Actor | User |
| Entry condition | The user has already done the “Login” activity. |
| Events flow | 1. The user clicks on the “Personal information” button. 2. The user select the type of data he wants to consult. 3. The system shows all the information stored at the registration of the user. |
| Exit condition | The user sees what he has requested from the system. |
| Exceptions | None |

|  |  |
| --- | --- |
| Name | See map |
| Actor | User |
| Entry condition | The user has already done the “Login” activity. |
| Events flow | 1. The user clicks on “See map” button. 2. The system shows the map to the user, with the default options. 3. The user selects the type of violation he is interested in. 4. The user selects the interval of time he is interested in. |
| Exit condition | The system shows to the user the updated map with the options chosen by the user, who can taps on the area and he can sees other information. |
| Exceptions | None |

|  |  |
| --- | --- |
| Name | Send parking violation report |
| Actor | User |
| Entry condition | The user has already done the “Login” activity. |
| Events flow | 1. The user clicks on the “send violation reporting” button. 2. The user clicks on the “parking violation” option. 3. The user selects the type of parking violation he wants to report (crosswalk, bus zone, etc). 4. The user inserts the position of the violation. 5. The user inserts the date of the violation. 6. The user inserts the license plate of the vehicle. 7. The system asks to the user if he wants to upload a picture of the violation. 8. The system asks to the user if he wants to remain anonymous. |
| Exit condition | The user sends the reporting to SafeStreets. |
| Exceptions | 1. If SafeStreets can’t retrive user position. it asks to the user to insert his position manually. 2. If SafeStreets read an invalid field, it asks to the user to re-insert the data. |

|  |  |
| --- | --- |
| Name | Send other type of violation report |
| Actor | User |
| Entry condition | The user has already done the “Login” activity. |
| Events flow | 1. The user clicks on the “send violation reporting” button. 2. The user clicks on the “other type of violation” option. 3. The user selects the type of violation he wants to report. 4. The user inserts the position of the violation. 5. The user inserts the date of the violation. 6. The user inserts the license plate of the vehicle. 7. The system asks the user if he wants to remain anonymous. |
| Exit condition | The user sends the reporting to SafeStreets. |
| Exceptions | 1. If SafeStreets can’t retrive user position. it asks to the user to insert his position manually. 2. If SafeStreets read an invalid field, it asks to the user to re-insert the data. |

**Sequence Diagrams**

* + 1. Authority

**Scenarios**

Scenario 1

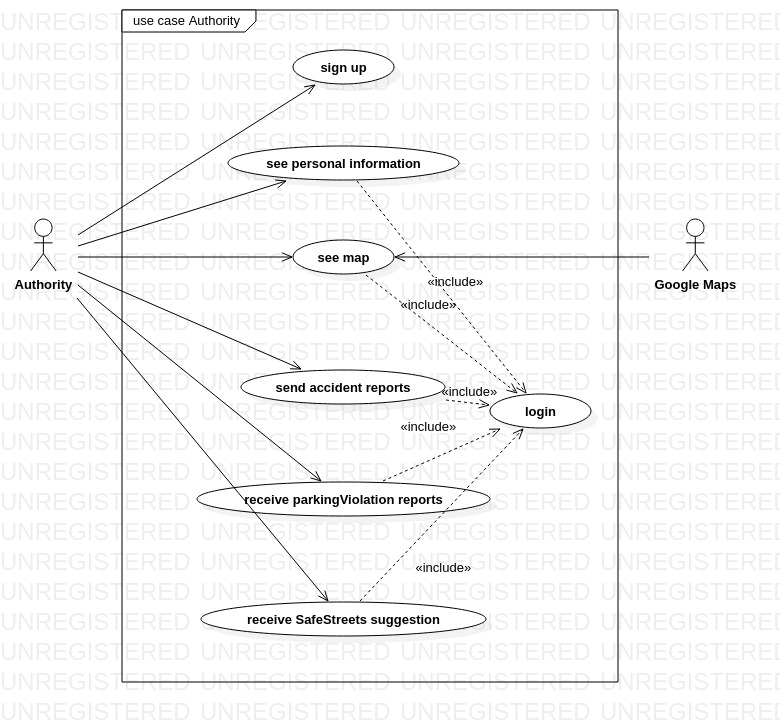
Donald, an employee of the Milan municipal police, is in charge of choosing the rules and the routes that the traffic auxiliaries will have to follow in order to finish their day of work. Donald consults the reports sent by SafeStreets. The reports are divided into two sections. The first concerns parking violations and the second, if present, concerns suggestions on how to prevent this type of violation. Thanks to our application Donald will be helped in the decisions to take regarding the work of the auxiliaries.

Scenario 2

The municipality of Milan offers to share his accidents information with SafeStreets.

SafeStreets agrees to send its data regarding parking violations to the municipality, while the municipality will send its accidents reportings, which SafeStreets will automatically read with its own tool. Donald, who cares a lot about the collaboration between the two parties, aware of the potential that can be developed by this agreement , takes the commitment of sending all the police reports of the last month and send them to SafeStreets.

**Use Case DIagram**



**Use cases**

|  |  |
| --- | --- |
| Name | Sign up |
| Actor | Authority |
| Entry condition | The authority has opened the application on his working device. |
| Events flow | 1. The authority inserts the “Sign up” option. 2. The authority clicks on the “are you an authority” written. 3. The authority enters his department code. 4. The authority enters his department email. 5. The authority chooses the “Confirm” option. |
| Exit condition | The authority is registered and can use the application. |
| Exceptions | 1. The user is already registered. The system warns the user to do the login. 2. The user inserts not valid information in one or more fields. 3. The mail is already taken.   The excpetions are handled returning to the starting point of the “Sign Up” option. |

|  |  |
| --- | --- |
| Name | Login |
| Actor | Authority |
| Entry condition | The user is already registered in the system. |
| Events flow | 1. The user clicks on the “login” option. 2. The user clicks on the “are you an authority” written. 3. The user inserts his department code. 4. The user inserts his email. 5. The user clicks on the confirmation option. |
| Exit condition | The user is logged in and can use the application services. |
| Exceptions | 1. The user inserts the wrong department code. 2. The user inserts an email that not correspond to the ones stored under the department.   These exceptions are handled by allowing the user the possibility to re enter the wrong field. |

|  |  |
| --- | --- |
| Name | See personal information |
| Actor | Authority |
| Entry condition | The user has already done the “Login” activity. |
| Events flow | 1. The user clicks on the “Personal information” button. 2. The user selects the types of the data he wants to see. 3. The system shows all the information requested. |
| Exit condition | The user sees what he has requested from the system. |
| Exceptions | None |

|  |  |
| --- | --- |
| Name | See map |
| Actor | Authority |
| Entry condition | The user has already done the “Login” activity. |
| Events flow | 1. The user clicks on the “See map” option. 2. The system shows the map to the user, with the default option. 3. The user selects the type of violation he is interested in. 4. The user selects the interval of time he is interested in. |
| Exit condition | The system shows to the user the updated map with the options chosen by te user, who can taps on the area and he can sees other type of information. |
| Exceptions | None |

|  |  |
| --- | --- |
| Name | Send accident reports |
| Actor | Authority |
| Entry condition | 1. The user has already done the “Login activity. 2. The user has the permits to send the accident reports. |
| Events flow | 1. The user clicks on the “send accident reports”. 2. The user selects the reports he wants to send. There is a default option that give the user the opportunity to send automatically the reports occurred in the last week. 3. The user selects the confirmation option. |
| Exit condition | The reports are sended to SafeStreets. |
| Exceptions | The user doesn’t select any report. So the system warns the user to redo the operation. |

|  |  |
| --- | --- |
| Name | Receive parking violation Reports |
| Actor | Authority |
| Entry condition | 1. The reports are sended periodically from SafeSTreets to the municipal police, who stores them in a queue. 2. When the queue is full, a message to the authorized personnel is sended. 3. The authorized personnel has already done the “Login” activity. |
| Events flow | 1. The user clicks on the “Download SafeStreets reports”. 2. The reports are downloaded and stored in the municipal police software. 3. A SafeStreets tool tells to the municipality if they have been downloaded correctly. 4. The user clicks on “end operation” button. |
| Exit condition | The reports are downloaded correctly |
| Exceptions | The reports information are damaged, in these case there are two operations that the authiorities can do. One is to reboot the system and try to download the reports again. The other is to contact SafeStreets which will give them suggestions on how to download the reports manually. |

|  |  |
| --- | --- |
| Name | Receive SafeSTreets suggestions |
| Actor | Authority |
| Entry condition | 1. The reports are sended periodically from SafeSTreets to the municipal police, who stores them in a queue. 2. When the queue is full a message to the authorized personnel is sended. 3. The authorized personnel has already done the “Login” activity. |
| Events flow | 1. The user clicks on the “Download SafeStreets suggestions”. 2. The suggestions are downloaded as a word document. 3. The user click on “Confirm and exit” button. |
| Exit condition | The user has downloaded the SafeSTreets suggestions. |
| Exceptions | 1. The report can be the same as the one received last week. In this case it means that municipality has to wait until the application creates the new one. 2. The report can be empty, it means that an error occurred. The user should re download the document or look for SafeStreets number and asks for help. |

**Sequence Diagrams**

* + 1. Requirements

**Traceability Matrix**

* 1. Performance Requirements

The application must be able to save data and violations via database. It also must be able to analyze the data by querying the database. Our software must ensure communications with other messaging applications in order to receive data from authorities and send it to them. In particolar, it must adapt to legacy systems managed by the public administration and smartphone devices.

* 1. Design constraints
  2. Software system attributes
     1. Reliability
     2. Availability
     3. Security
     4. Maintainability
     5. Compatibility

1. Formal analysis using Alloy
2. Effort spent