Requirement analysis and specification document Safestreets

Luca Massini   Daniele Nicolò

release date to be defined

Contents

1 Introduction

The purpose of this document is to represent the Requirement Analysis and Specification Document (RASD). This document shows what are the goals and the requirements of the software. It has to represent how the application can be useful for the users that will use it and why they are fundamental to improve the quality of the service offered. Secondly, this document can be also used as a support for the testing of the system, for the verification activities and also the validation ones. The RASD can be also used to guide the changes in a already existing system.

1.1 Purpose

1.1.1 Description of the given problem:

SafeStreets is a software useful to help people to be safer when they are on the street.
The users can send to the municipality pictures of violation occurring in public streets: the reporting can concern violation on the road, in a parking and so on.
The software allows the users to send detailed information about the violation, such as the hour, the date, the type of violation and the position (they can be captured with GPS).
Furthermore, the service can provide, both to the user and to the municipality,information about the streets in which the user is around, such as the number of violations per street and consequently the level of danger of the street.
In addition, the user can have a different service based on the category to which he/she belongs.
The user and the municipality can also find on the application the most "dangerous" vehicles, that are the ones with the highest number of reports from the users.
The service must be different in base of the type of user, such authorities, motorists, motorcyclists, bikers, pedestrians, disabled people, etc, so it must be easy to use.
Finally, the users must be able to receive recommendations from the system to avoid using streets, parking lots that are risky in general or at a specific hour or date.

1.1.2 Goals:

- [*goal*1]: The service allows the users to report a violation to the municipality.

    - [*goal*1.1]: The user can send a picture of the violation.
    - [*goal*1.2]: The user can specify the date and the hour when the violation has happened and the type of violation.

- [*goal*1.3]: The municipality must be able to receive the reports

- [*goal*2]: The service allows the users to have detailed information about the violations and the street safety.

  - [*goal*2.1]: The users can know which are the most reported streets, areas or parking.
  - [*goal*2.2]: The users can see which are the vehicles that commit the highest number of traffic violations.
  - [*goal*2.3]: The user, in base of his/her position, can be recommended to use the safest streets/areas, etc or avoid the dangerous ones, that are highlighted in different ways on the map.

- [*goal*3]: The service offers different types of interfaces and accessibility in according to the type of user (biker, pedestrian, rider, driver, disabled person).

  - [*goal*3.1]: The user that want to park his/her car/motorbike can receive a suggestion of where to do it based on the safest streets around him/her.

- [*goal*4]: A user can specify the category to which he/she belongs (motorcyclist, biker, pedestrian, car driver, disabled person) in order to improve the service quality.

- [*goal*5]: A driver or a motorcyclist can add to his personal profile the license plate of his vehicle.

- [*goal*6]: Safestreets can send to the municipality suggestions of what to do in order to improve the safety on the streets for every type of Safestreets user.

  - [*goal*6.1]: The municipality must receive every suggestion sent by Safestreets.

- [*goal*7]: The user must be able to sign-up into the safestreets platform.

- [*goal*8]: The user must be able to log-in into the system after having done the sign-up.

- [*goal*9]: The user must be able to log-out of the system if logged-in.

1.2 Scope:

In this section is introduced the so called "world and machine phenomena".
We distinguish the world and the machine. The world is related to every phenomena that take place outside of the region of events that the system to be developed (called machine) is able to control or eventually observe. Instead, the machine is related to phenomena which happen inside the machine and so they're completely controllable. Some of the totality of the phenomena can be observed by the machine but they're controlled by the world and vice versa. These are called shared phenomena. In this section there will be described only the world phenomena and the shared one.The machine phenomena will be described later in the next chapters.

World phenomena:

From the user point of view:

- The user sees one or more violations.

- The user wants to report a violation to the municipality.

- The user wants to know the area/streets where the major violations occur.

- The user wants to know where is safer to park his/ her car/bike/motorcycle.

- The user wants to know what are the vehicles that commit the major number of violations.

From the municipality point of view:

- The municipality wants to know what to do to improve the safety of the streets.

- The municipality wants to discover which are the most dangerous zones of the city.

Shared phenomena:

Shared phenomena controlled by the world and observed by the machine:

- The user signs-up into the platform.

- The user logs-in to the platform.

- The user logs-out from the platform.

- The user sends to the system a picture of a violation and all the other data related to it.

- The user sends some data request to the system.

- The municipality sends a data request to the system in order to receive suggestion to improve the safety of the streets.

- The municipality sends a data request to know what are the vehicles that have committed the major number of violations.

Shared phenomena controlled by the machine and observed by the world:

- The user receives ,by the system,all the data related to the safest and most dangerous streets .

- The user receives, by the system, all the data related to the vehicles that have committed the major number of violations.

- The municipality receives suggestions from SafeStreets to provide a better safety on the streets.

- The municipality receives the list of the most dangerous vehicles .

Machine phenomena:

- The system stores the user's data.

- The system stores all the data about the municipality.

- The system stores all the data about the violation.

- The system analyses the received picture in order to retrieve the car plate.

- The system checks if the street in which the violation occur is an existing street.

- The system runs an algorithm to retrieve suggestions from users' reports in order to provide them to the municipality.

- The system refuse a request by the user because of insufficient data.

1.3   Definitions, Acronyms, Abbreviations

1.3.1   Acronyms:

S2B	System to be
GDPR	General Data Protection Regulation
FC	Fiscal Code sign-up
WP	World Phenomena
MP	Machine Phenomena
SP	Shared Phenomena
GPS	Global Positioning System

1.3.2 Definitions:

- **Violation** = Any kind of infringement of the highway code. Here we distinguish violations concerning the traffic and violation concerning parking.

- **Personal data** = Data belonging to the user, needed in the moment of the signing-up. They are name, surname, date of birth, e-mail and FC.

- **S2B** = System to be developed. It will contain all the services provided by SafeStreets.

- **Private user request** = A user can ask for information about the safety of an area or a parking lot around him/her. He/She can also request data about the most dangerous vehicles in that area.

- **Report of a violation** = A user can use SafeStreets to report an infringement he has seen. In particular he can add the type of violation, the date and the hour, the street (that can also be retrieved automatically) and he can also add a picture of the violation.

- **Municipality request** = The municipality can ask for the same things aforementioned for the user. They will receive more accurate information rather than a private user, including for example data about the violators. They can also request suggestions (provided by SafeStreets) to improve the quality of their service.

- **Fiscal code** = It's a 16-characters code identifying an Italian citizen.

- **Credentials** = username and password used by a user to log-in to the system.

1.3.3 Abbreviations:

- Gn = n-th goal;

- Dn = n-th domain assumption;

- Cn = n-th constraint;

- Rn = n-th requirement;

- MP = Machine Phenomena;

- WP = World Phenomena;

- SP0 = Shared Phenomena that are managed by the Machine but that are observable by the World;

- SP1 = Shared Phenomena that are managed by the World but observable by the Machine.

1.4 Revision History:

- Work started 13/10/19;

1.5 Reference Documents:

- RASD Slides by professor Matteo Rossi (Polimi);

1.6 Document Structure:

This document is organized in the following way:

- **Overall Description:** This section contains a deeper analysis of the world and machine phenomena already described in the scope. It also contains class diagrams which describe the relationship between actors, state chart diagrams used to describe the state of the system from a dynamic point of view, the description of the stakeholders and their needs and finally the key functional requirements and the domain assumption.

- **Specific requirements:** In this section there is a more detailed description of the requirements already described in the previous section. This is thought to help the develop team to understand what the system must ensure.In particular in this part there is a description of the external interfaces requirements,of the functional requirements, of the performance requirements, of the design constraints and finally of the software system attributes.

- **Formal analysis:** This section contains the formal analysis written in alloy. Alloy is a formal notation used to specify model of systems and software. Thanks to alloy it is possible to write a formal models with their own requirement, domain assumptions and goals and then check the correctness of the model.
There is another very important aspect concerning this section. Such aspect is the fact that alloy gives the possibility to clarify aspects that could be ambiguous. The ambiguity is usually and intrinsically due to natural language.

2 Overall Description

This section is necessary to give a better and deeper description of the shared and world phenomena. In this section, the system will be described also with the help of the class diagrams and state diagrams.

2.1 Product Perspective:

The user sees one or more violations: It's the situation in which the user sees a violation which can concern traffic or parking. The user knows what happens and also where and when. Such violations could be for example: a car/motorcycle parked in a red zone or on a sidewalk area or however any kind of traffic or parking violation.

The user wants to report a violation to the municipality. In this case the user would like to have the possibility of report a violation of which he/she is aware.

The user wants to know the areas/streets where the major number of violations occur. In this type of phenomena the user would like to know what are the most dangerous areas in his city/town in terms of traffic or parking violations.

The user wants to know where is safer to park his/ her car/bike/motorcycle: In this case we analyse the case in which the user has a vehicle that needs to be parked. He/she would know what are the safest streets in order to decide in which street to park. He/she wants to know the safest street whose distance from him/her is less than or equal to a certain value (given by the user).

The user wants to know what are the vehicles that commit the major number of violations: The user would like to know the vehicles that commit the highest number of violations. Obviously, the user is not an authority and so he can see a limited quantity of information due to GDPR regulation.

The municipality wants to know what to do to improve the safety of the streets: In this case the municipality recognizes that in a certain area or street there is a problem with the too high number of violations. In this case the municipality would like to have some suggestions in order to provide a solution to make the number of violations decrease.

The municipality wants to discover what're the most dangerous zones of the city. The municipality would like to know which are the most dangerous streets in order to monitor them with more attention. This could be useful for the municipality if it wants to improve its service quality.

The user signs-in to the platform: In this case the user compiles all the mandatory fields which are: name,surname,username,password,fiscal code. Driving license data are not mandatory because this system is designed for people who use the app also to ride a bike or walk.

The user signs-up to the platform: In this case the user compiles all the mandatory fields which are: name,surname,username,e-mail,password and FC. Driving license data are not mandatory because this system is designed for people who use the app also to ride a bike or walk.

The user logs-in to the platform: The user can log-in to the system by entering username and password.

The user logs-out from the platform: In this case the user quits the platform if he/she logs-out. The user will not be able to receive any service by the platform until he/she does a new log-in.

The user sends to the system a picture of a violation and all the other data related to it: In this case the user sends a report of a park violation or a traffic violation to the authorities (the municipality in this case). The user can take a picture of the violation and add the mandatory information which are needed which are: the hour and the date. The place of the violation can be detected automatically or manually (it's a user's choice).

The user sends some data request to the system: In this case the user could want to retrieve some information from the system. Such information, could concern the most dangerous zones or who commits the major number of violations, the most safe streets where to park and so on. The system will provide these information in different ways, such as a list of the users or a maps with the highlighted streets.

The municipality sends a data request to the system in order to receive suggestion to improve the safety of the streets: The municipality sends a data request to the system in order to have suggestions to improve the safety. These suggestions are simple phrases which describe what it's possible to do to overcome problems.

The municipality sends a data request to know what are the vehicles that have committed the highest number of violations: In this case the authorities would like to understand which are the ones who commit the major number of violations, so they send a data request to retrieve those information. The municipality has to specify the types of vehicles (motorbikes, cars or every kind of vehicle that has a license plate), the maximum number of instances to obtain with the query.

9

The user receives from the system all the data related to the safest and most dangerous streets : In this case the user receives all the data about the streets that are considered the most dangerous or the safest. He/She receives a map with the highlighted streets which are at a certain distance from a point specified by the user.

The user receives from the system all the data related to the vehicles that have committed the major number of violations: In this case the user receives the list of license plates that have committed the highest number of violations. The user receives a number of license plates less than or equal the number he/she has specified as input.

SafeStreets provides the suggestion in order to give them to the authorities: SafeStreets runs an algorithm to retrieve information and suggestions about the future improvement of the streets safety from the users' reports. These data will be sent to the municipality so that they will be able to use them to enhance the safety.

The municipality receives suggestions to keep the streets safer: The municipality receives list where each element is composed on one side by the problem and on the other by the suggestion to overcome it.

The municipality receives the list of the most dangerous vehicles: The municipality receives all the license plates belonging to the vehicles that have committed the highest number of violations. The authorities select the maximum number of plates to receive (That they can choose each time as an input) and they obtain by the system a less than or equal number of license plates.

The system stores the user's data: The system will store permanently and safely all the user data. They will be stored in the system databases in such a way that data corruption is rather difficult.

The system stores all the data about the municipality: The system will also store all the data about the municipality in its databases.

The system stores all the data about the violation: The system will store all the information concerning the violation. Such information are the picture, the data, the hour, the license plate of the vehicle, and an unique identifier to distinguish identical violations eventually reported by different users.

The system analyses the received picture in order to retrieve the car plate: The system, after having received a picture of the violation, will run an algorithm to recognize the license plate of the vehicle that committed the

violation so that it is uniquely identified. The algorithm takes the picture sent by the user as an input and works on it to retrieve the license plate. If it is recognizable from the picture, the system saves the data about it in its database. Otherwise it only keeps the information about the violation.

The system checks if the street in which the violation occur is an existing street: if the place of the violation is manually entered by the user there is the possibility that such place doesn't exist. In this case the system must automatically report an error to the user. Otherwise the system will store the street in the database together with all the other information related to the violation.

The system refuse a request by the user because of insufficient data: In the case that the data on the databases in insufficient the system abort and will report the fail to the user.

2.1.1 Class diagrams:

This is the user class diagram:
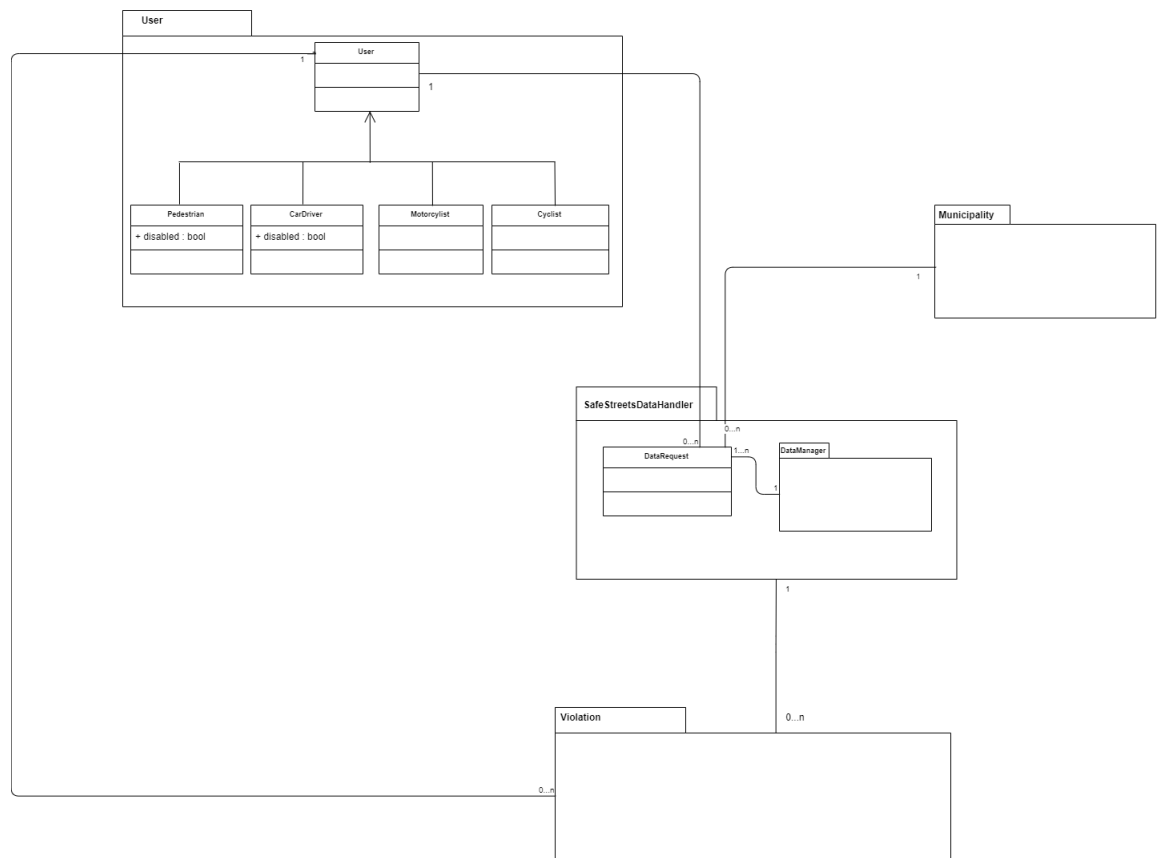
Figure 1: User class diagram

This is the violation class diagram:
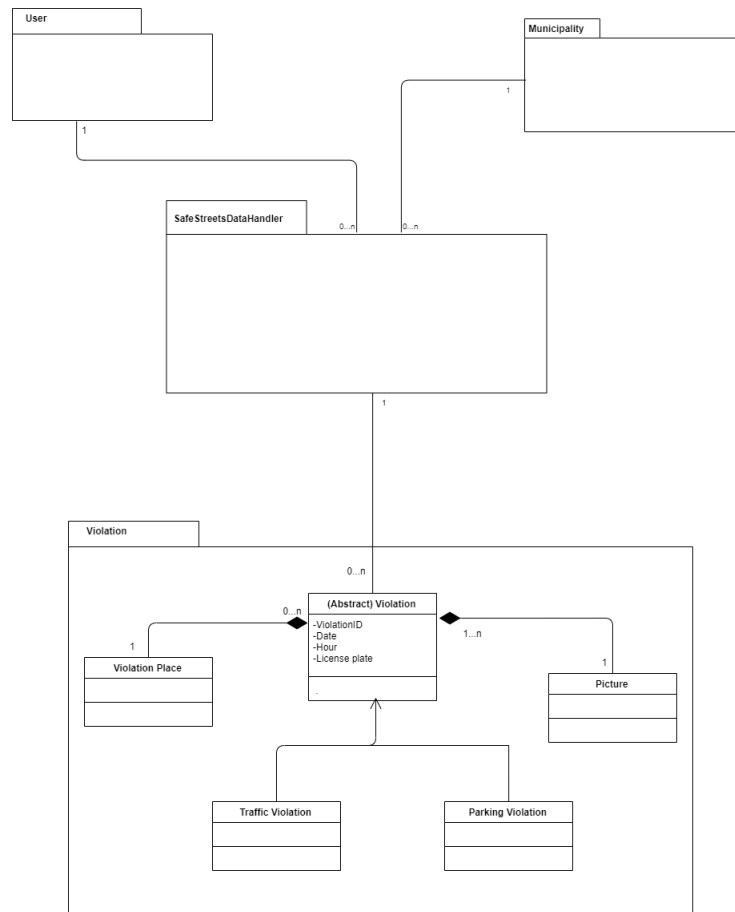
Figure 2: Violation class diagram

2.1.2 State charts:

Here you can find a state chart diagram of the system:
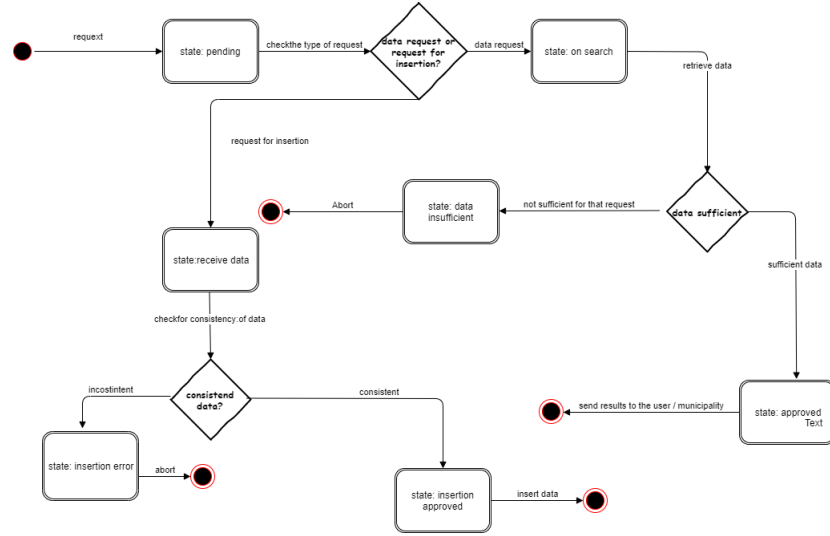
Figure 3: System State chart diagram

2.2 Product functions:

In this section there is a description of the main functions that the system will make available to the final user.
These functions are the following:

- **Report of a violation:** Safestreets allows the user to report a violation. To do that he/she has to take a picture, indicate the hour and, eventually, the location where the violation occurred. The place where the violation occurred is taken automatically in case the user does not want to insert it himself, using the smart-phone GPS signal. After that the violation will be reported to the municipality that will handle the problem.

- **Request of the list of vehicles that have committed the highest number of violations:** The users and the authorities can make a request and ask what are the vehicles that have committed the highest number of violations. This request (if no exception happen) will show a list of license plates to the user. The request can be personalized by expressing: a maximum number of vehicles that must be showed. If the number is not given as input the system will show a fixed number of results.

- **Get the list of most dangerous streets:** The most dangerous streets are the ones with the highest violations and they can be retrieved specifying the maximum number of vehicles that the request has to show.
Who is making the request has also to write the area of the streets he/she wants to analyse. This is done specifying the GPS coordinates of a fixed point and a maximum distance from that point. In this way, the user

14

specifies a set of streets that he/she wants to analyse. The result of that request will be a map with the most dangerous roads highlighted.

- **Send a request to obtain suggestions to make the streets safer:** This function is only designed for the municipalities. In this case the municipality has to insert the place about which it wants to have suggestions. The system will return a list containing on one side the problem and on the other side a possible solution to overcome it. Obviously, the list is not static but it will be updated over time thanks to the reported violations.

- **The user send a request to know where to park his/her vehicle in a safe area:** This function allows the user to explicitly ask the system to suggest places where to park his vehicle in a secure way. The user must specify what type of vehicle he/she has and how far he/she wants to park from his/her position or from a position given as an input. The system will not guarantee the fact that in the proposed place there will be at least one parking lot but it only will suggest which roads are considered as more secure than others to do so.

- **The user will receive the suggested streets where to park:** After having requested the safest streets where to park, the system will provide them to the user in a map form. In this map there will be highlighted the streets considered more secure than the others in the specified area given by the user as input.

2.3   User characteristics:

In this subsection there is a description of the application users and an analysis of their needs. We will categorize the people who use the app in order to understand their necessities in a better way.

Description of the users of the app:
The user of the app can be divided in a first categorization which is the following:

- Private user;

- Municipality or authority.

Description of the private users:
It is possible to divide again the private users in other classes. For every class there will be a different concept of dangerousness.

- Pedestrians;

- Motorists;

- Motorcyclists;

- Cyclists.

A private user can be also a disabled person belonging to the pedestrians category or to the car drivers one. Because of this fact these type of users will require a different type of service and a different concept of dangerousness. For example they will have information about cars parked in a handicap parking or on ramps for disabled people.

All the private users are supposed to have downloaded the app from the store,to have done the sign-up entering all the mandatory data and finally to have accepted all the permissions.

The private user, according to his category, will take advantage of a different type of service.

A car driver, for example, will have information about parking lots or parking violation, that is not required in a biker service.

A cyclist will have informations about bicycle paths, while pedestrians will have detailed data about sidewalks or pedestrian crossing, with all the possible violations related to them.

Furthermore, the application can encounter:

- Users that have downloaded the app but that haven't already signed-up in the system;

- Users that have already signed-up the system but that haven't logged-in yet. Users logged-out from the system belong to this category;

- Users also logged-in the system.

2.4 Assumptions, dependencies and constraints:

2.4.1 Domain assumptions:

- [**D1**]: The user is supposed to submit correct e-mail, and data matching his/her fiscal code.

- [**D2**]: The user creates one and only one account.

- [**D3**]: Every municipality which has an account on the system certifies its own account as valid.

- [**D4**]: The GPS signal has a relative error of 10 meters.

- [**D5**]: The memory where the data is stored is persistent.

- [**D6**]: The user allows the app to have access to his/her position and the camera of his/her device.

- [**D7**]: The internet connection has to be enabled when the app needs it.

2.4.2 Constraints:

- The system must treat the users' personal data in according to the GDPR regulation.

- The system will be designed and implemented for smart-phones.

- Since the system uses the fiscal code to identify the user it is supposed that it will be used only in Italy, so the application will have to be available only in the Italian app stores.

- The pictures must be taken only through the app camera and not imported from the device gallery in order to take photos in real time.

2.4.3 Dependencies:

- The system will use the GPS services provided by the smart-phones.

- The system will use the internet services offered by the smart-phones.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- **SafeStreet Interfaces**

  - **SafeStreets Logo:**
    SafeStreets has a simple but explicative logo. It represents a street, because the main theme of the application concerns public roads. It also have a green tick, that represents the correct functioning and the safety of the streets.

Figure 4: SafeStreets Logo

– **SafeStreets loading page:**

Figure 5: SafeStreets Loading

– **SafeStreets log-in page:**
  This is the page a user sees when he downloads the app and opens it
  for the first time or when he has done a log-out. In this page there is
  the possibility to log-in (inserting username and password) or to sign-
  up. The user can also recover his credential using the funcion below
  the log-in button. When logged-in, the user will be redirected to the
  Home Page, that will be different for private users and authorities.
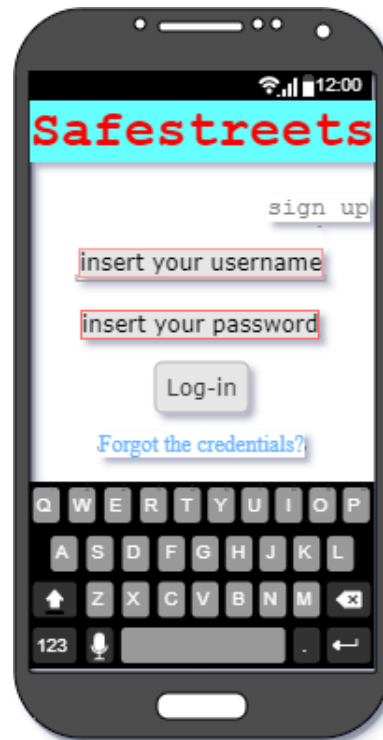
Figure 6: SafeStreets Log-in

– **SafeStreets sign-up account choice:**

Here the user can choose the type of account he wants to create. There is the account reserved to the municipality and the one available to a private user. By choosing one or the other, the user will be redirected to the second step of the sign-up.
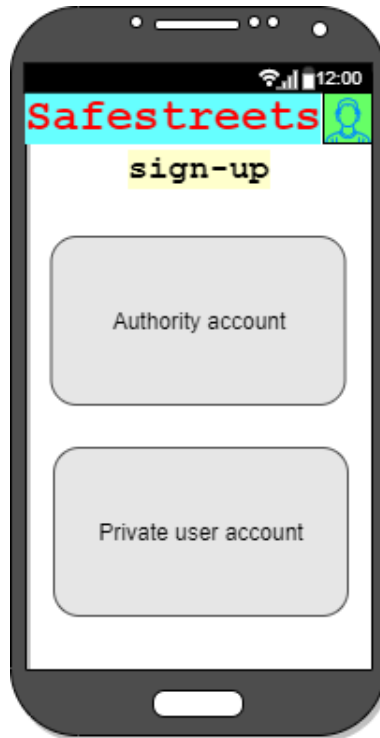
Figure 7: SafeStreets Account Choice

– **SafeStreets private user sign-up:**

This is the first part of the private user's sign-up. Here the user has to put his first name, his last name, the username he will use in SafeStreets (that will also identify him uniquely) and the password, that has to comply the security policy. Then he can continue to the second part of the registration.

Figure 8: SafeStreets private user sign-up, first part

– **SafeStreets private user sign-up (second part):**

This is the second and last part of the private user's sign-up in the SafeStreets platform. Here the user has to fill the fields with his personal e-mail and his FC. Furthermore he has to choose which category he belongs to and he can also indicate if this account will be created for a disable person or not. Finally, he has to insert his date of birth and optionally he can add the id of his drive license if he drives a vehicle that needs it. After doing that the sign-up is complete and the user will be redirected to the log-in page.

Figure 9: SafeStreets private user sign-up, second part

– **SafeStreets municipality sign-up:**
   If the user has chosen the municipality sign-up he will be redirected
   here. The authorities must indicate the city of their competence and
   must insert an ID code to prove that they have the authorization to
   represent the municipality. Then they have to insert a username and
   a password, in the same way as explained for private users. Then
   they will be redirected to the log-in page.

Figure 10: SafeStreets Municipality sign-up

– **SafeStreets Private User's Home Page:**

This is the Home Page reserved to private users. It's a very basic interface, because we want that SafeStreet will be used by as many types of people as possible. Its functions are very easy to understand and they are the possibility to report a violation to the system, the chance to see which are the most reported streets in a selected area, the possibility to see the condition (in terms of safety) of a selected street, parking lot or area and the chance to check which are the vehicles that are reported the most. Finally there is also the log-out button.
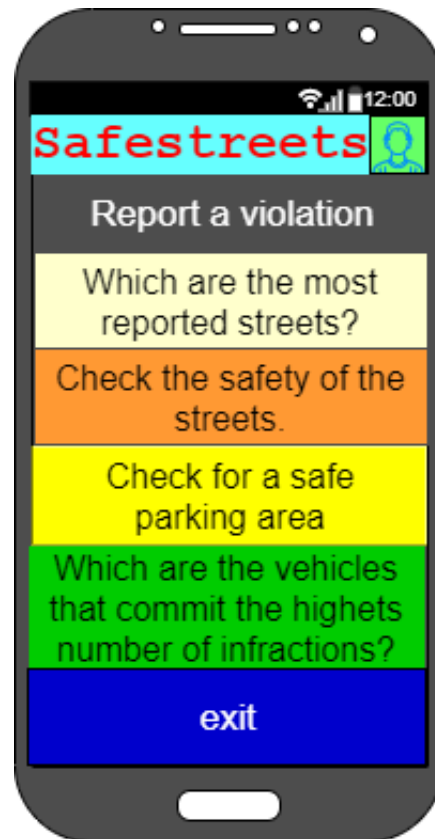
Figure 11: SafeStreets Private Users' Home Page

– **SafeStreets Private User's Home Page:**

This is the first function presented in the Home Page. Here the user can report a violation that he has seen. The first field to fill is the one that allows the user to describe the violation, indicating the type of violation that he has seen (for example motorcycle parking areas occupied by cars, or traffic accidents, and so on). Then he has to indicate the date and the hour of the violation. Furthermore a user has to indicate (manually or with GPS signal) the street and the city of the violation and finally he can take a photograph of the violation (but this is mandatory).

Figure 12: SafeStreets Violation Reporting interface

– **SafeStreets Most reported streets interface :**

This interface allows the user to check which are the most reported streets in a city. The city can be inserted manually by the user or retrieved automatically by the system using the GPS signal of the user's smartphone. The user has to specify how many streets the list should be made up. Then he has to submit the request and he will receive the list.

Figure 13: SafeStreets Most reported streets interface

– **SafeStreets safest streets interface :**

Here the user can check which is the condition of a selected street. He can specify the street manually, inserting its name and the name of the city. The street can also be retrieved automatically by the system with the GPS signal of the user's smartphone. Then the user has to insert a radius (expressed in KM) within which the system will provide the condition of the contained streets, highlighting the safest and the most dangerous ones.
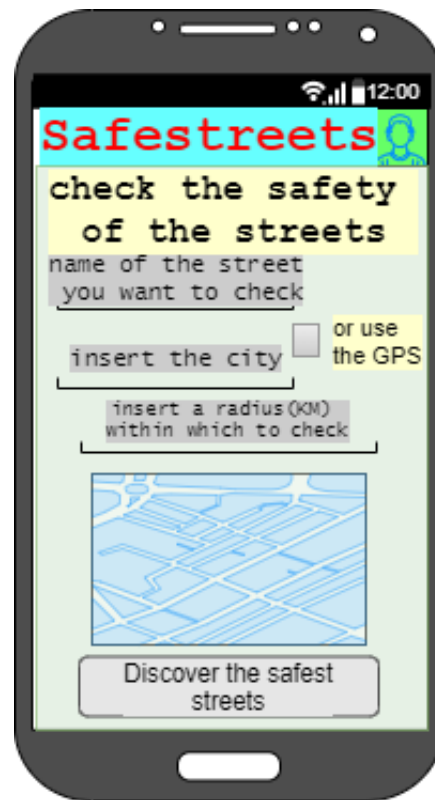

Figure 14: SafeStreets safest streets interface

– **SafeStreets safest parking areas :**

The user can use this functionality to find the safest place where to park his vehicle. He has to specify his position, that can be done manually (inserting the name of the city and the street) or, as in the other functionalities, automatically (using the GPS signal of the smartphone). Then the user has to insert the maximum distance (expressed in kilometres) from the selected position to which he is willing to get. After have inserted all the needed information, the user will see the safest parking area where to go, highlighted differently from the less safe ones.
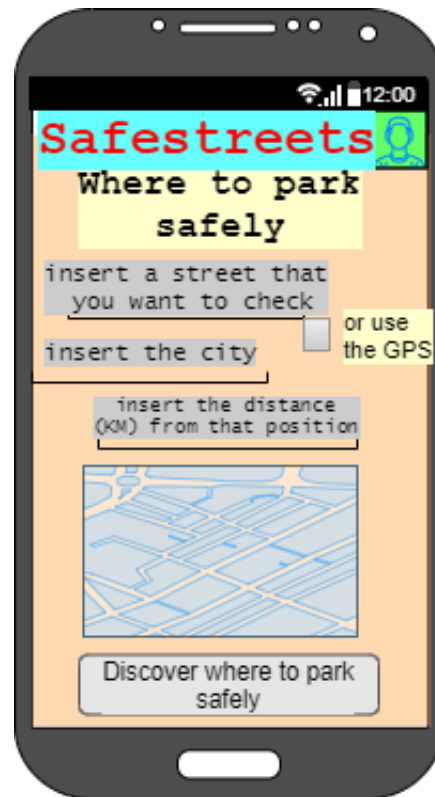

Figure 15: SafeStreets safest parking area interface

– **SafeStreets Most dangerous vehicles :**

Here the user can check which are the most reported vehicles in a selected street. The street can be inserted as in the previous functionalities, manually or automatically and it's also mandatory to indicate the maximum number of vehicles that will make up the chart. Then the user will have the list of the selected vehicles so that, for example, he will avoid to park near them, and so on.
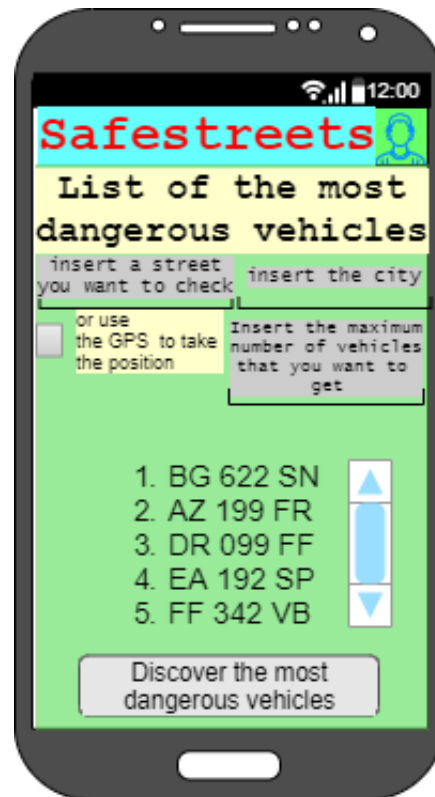

Figure 16: SafeStreets Most dangerous vehicles

– **SafeStreets Municipality Home Page :**

This is the Home Page reserved to the authorities. Here the municipality has the possibility to choose between four functionalities: three are structured in the same way of the ones designed for the private users, and are the possibility to check the most reported streets and the most reported vehicles in the city. The interfaces for the requests are the same than the ones for the users. The municipality also has the possibility to do a request to obtain suggestions in order to improve the quality of their service and, so, the safety of the city.

Figure 17: SafeStreets Municipality Home Page

– **SafeStreets Municipality suggestions request :**

Here the municipality can make a request in order to discover which are the suggestions provided by SafeStreets to enhance the safety of the city. The authorities have to insert the name of the city, or providing it using the GPS signal of their device.
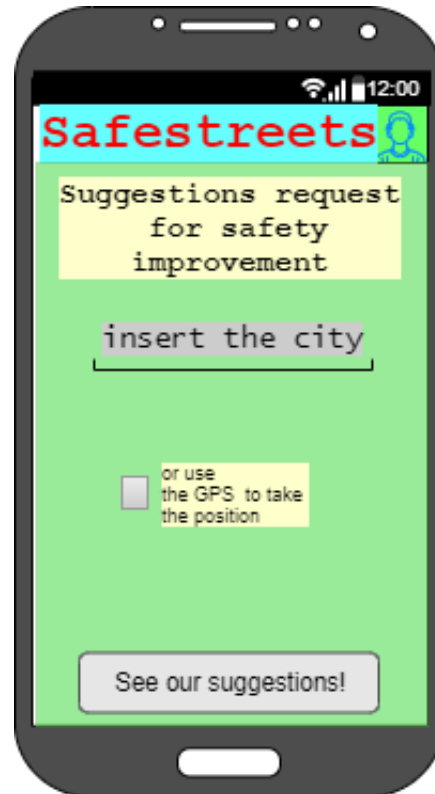

Figure 18: SafeStreets Municipality Suggestions Request

After having submitted the information, the municipality will receive the suggestions provided by SafeStreets concerning the indicated city. The data is retrieved by SafeStreets mining the information received by the users with their reports.
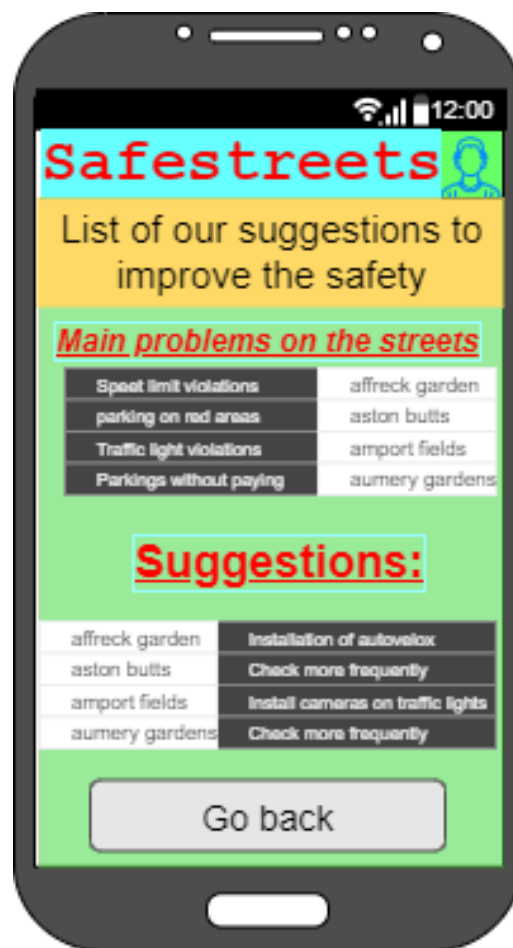

Figure 19: SafeStreets Municipality Suggestions List

3.1.2 Hardware Interfaces:

The system doesn't offer any kind of hardware interface to the user. In fact the user can use the application simply thanks to his/her phone.

3.1.3 Software interfaces:

The system offers his service through the use of the following software interfaces:

Operating systems:
The system will have to be designed for the most common smart-phones operating system:

- Android

- IOS

The choice of that decision is the fact that these operating systems are the most common use ones.

3.1.4 Communication interfaces:

Each smart-phone that will use Safestreets will communicate thanks to the HTTPS protocol. We've chosen this protocol because thanks to it, it is possible to obtain a secure connection between nodes. Furthermore it is a very very common use protocol for the connections over internet. Because of this it is very likely to not have any problem related to the protocol from several points of view.

3.2 Functional Requirements:

3.2.1 Use case diagrams:

In this subsection you will find the use case diagram. It is very important because it allows the reader to understand the flow of events between actors and the system itself. We have done two different use case diagrams to make them more explanatory and less confusing.
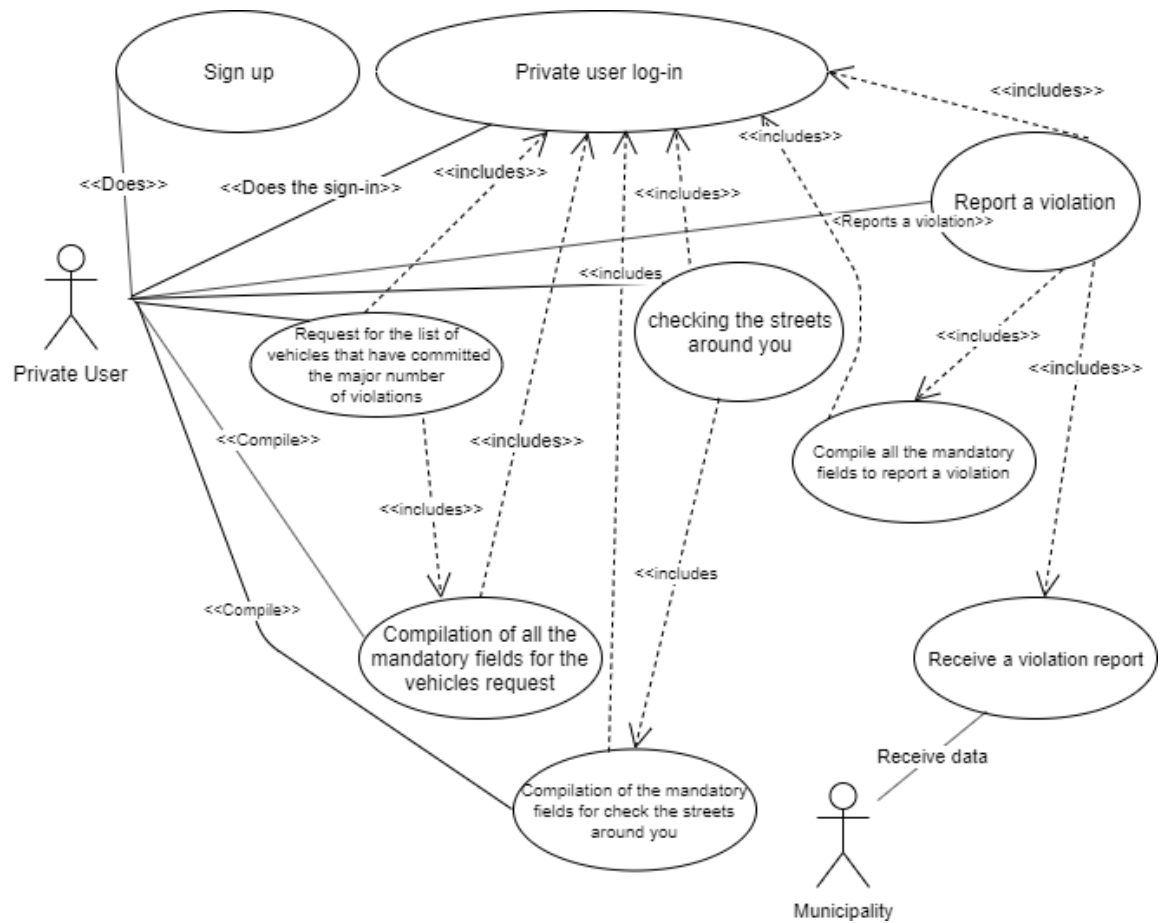
The first diagram:
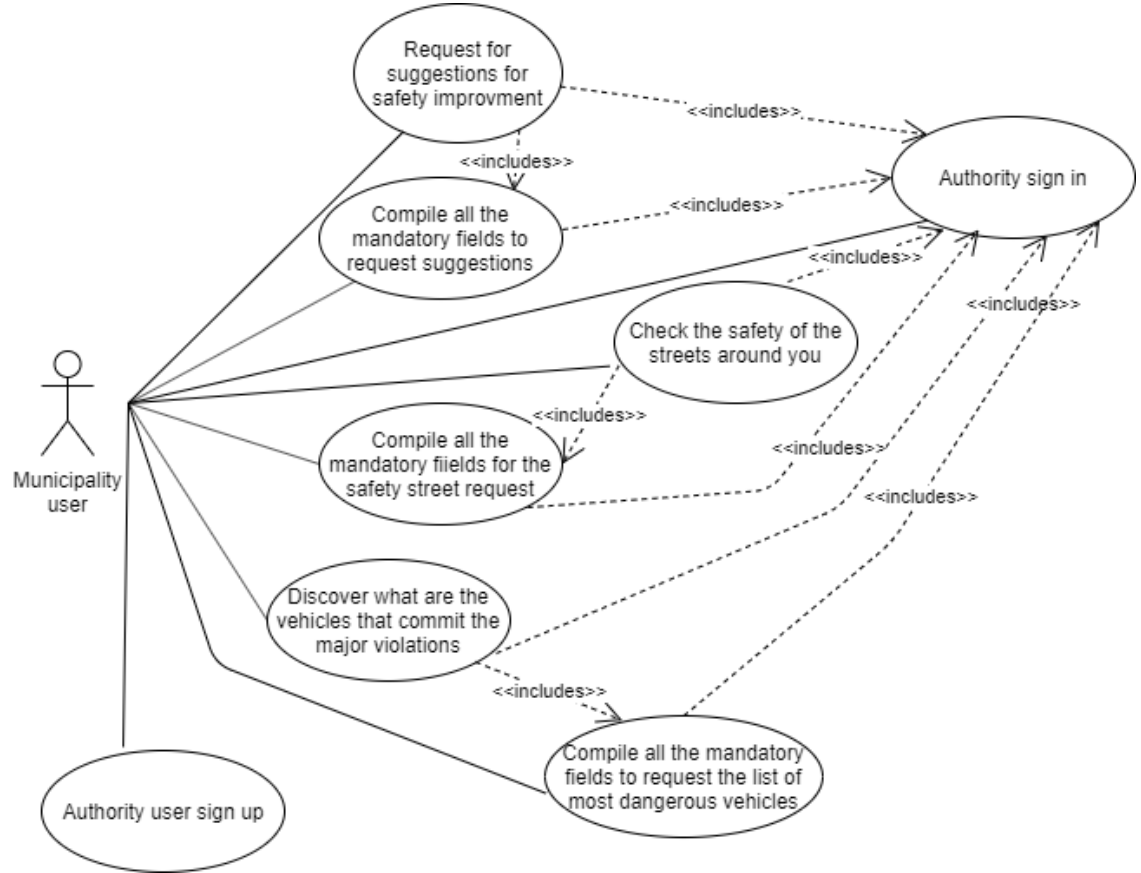

Figure 20: First use case diagram

The second diagram:

Figure 21: Second use case diagram

3.2.2   Scenarios:

From the private user point of view:

Scenario 1:   Luigi is a middle-aged man which lives in a very busy area of Milan. This district is very known to be a zone with a strong lack of parking. People usually park their vehicles in illegal zone. Because of this, he has found cars parked in front of his home gate several times. He has called the the municipality many times and now he is fed up about this situation. He usually calls the municipality with his phone but this process takes a lot of time because the telephone line is always busy. He would like the authorities to be have a better communication channel.
Luckily he has downloaded an application called Safestreets. He has read in the

description the fact that the application allow a user to report a violation via smart-phone.

Thanks to this mobile application he is now able to save time in reporting violations. Furthermore, he has observed that the municipality has started to do checks more frequently even without any report (because of the Safestreets suggestions to the municipality).

Scenario 2: Giovanni lives very close to the place where he works but he usually goes to work by car even if he would like to go there by bicycle. The problem is that the zone is very traffic and unfortunately accidents involving cyclists happen very often. Because of this, he does not feel like taking the risk. He notices an interesting application on the market called Safestreets. He is particularly interested in the function which suggest the most safety streets around the user from a point of view of the category in which the user has signed-up. This interest is caused by the idea of being able to choose a route that minimizes the danger of making an accident with a car taking in consideration the suggested streets given by the application.

Now, he is finally able to go to work by bicycle in a secure way.

Scenario 3: Andrea is a disabled person who strolls very often in his district. Unfortunately not all the roads are equipped with facilities for disabled people and also a lot of people parks their cars there.

He has downloaded the application which suggest to him the streets which are more "easy" and secure to be travelled for a person with his problem. The app allow Andrea to avoid roads which are almost sure to have cars parked on disabled facilities. He can compute himself the route from a place to another taking in consideration the suggestions given by the application. When he sees a violation he can report it to the municipality. Thanks to this feature Andrea is able to report violations, to do his strolls in the city almost in an autonomous way without the help of almost nobody.

Scenario 4: An employee of an insurance has the task of understanding which are the most dangerous areas of a Milan in order to change the price of insurance. He does not know where find the data necessary to do this task.Furthermore it has just been decided by the boss that a funding a research would be too costly. Fortunately the insurance company has just downloaded the Safestreets applications which can be used to retrieve dangerous areas. Thanks to Safestreets the company saved a lot of money by doing a data search at no cost.