Requirement analysis and specification document
Safestreets

Luca Massini   Daniele Nicolò

release date to be defined

Indice

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 Descripton 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]: Safestreet 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 in into the safestreets platform.

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

- [*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 viceversa. 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-in to 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 commited 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 suggestion 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 analyze the received picture in order to retrieve the car plate.

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

1.3   Definitions, Acronyms, Abbreviations

1.3.1   Acronyms:

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

1.3.2   Definitions:

da fare durante il progetto.

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:

da fare durante il progetto.

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 describes the relationship between actors, state chart diagrams used to describe the state of at least one component 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 contain the formal analysis done 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 important aspect concerning this section. Such aspect is the fact that alloy gives the possibility to clarify aspects that could be ambiguos. 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 happened 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 also any kind of traffic 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 area/streets where the major 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 violations or parking violations.

The user wants to know where is safer to park his/ her car/bike/motorcycle: In this case we analyze the case in which the user has a vehicle that needs to be parked. He/she would know what are the most safety streets in order to decide in which street to park. He/she wants to know the most safety 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 major number of violations. Obviusly, the user it's not an authority and so he can see a limited quantity of information due to privacy policies.

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 this kind of problems regarding the city.

The municipality wants to discover what're the most dangerous zones of the city. The municipality would like to know what 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 sign-in to the platform: In this case the user compiles all the mandatory fields which are: name,surname,username,e-mail,password. Driving license data are not mandatory because of the fact that this system is designed for people who use the app also to ride a bike or walk.

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

The user log-out from the platform: In this case the user quit 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 in an automatic way 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 commis 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 them that it wants to get.

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 between all the other. He 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 major number of violations. He receives a number of license plates less than or equal the number the user has specified.

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 system will provide a number of plates less than or equal to those indicated by the authorities.

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's databases in such a way as to make data corruption almost impossible (the impossibility doesn't exsists in real world).

The system stores all the data about the municipality: The system will store also all the data of the municipality in its databses. These data are both given from the municipality to the system and also used to manage the overall service.

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 reported from at least 2 different user.

The system analyze 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. In such a way the vehicle is uniquely identified.

The system checks if the street in which the violation occur is an exsisting street: if the place of the violation is manually entered by the user there exists the possibility that such place doesn't exsist in the real world. In this case the system must automatically search the reported streets and report an error to the user. Viceversa,in the case in which the street exsists the system will store it in the database together with all the other information related to the violation. ≪≪≪< HEAD =======

2.2   Product functions:

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

- **Report of violation:** Safestreets allows the user to report a violation. To do that he/she has to take a picture, indicate the hour and if the user wants, also the location where the violation occurred. The place where the violation occured is taken in an automated manner in case the user does not want to insert it manually. After that the violation will be reported to the municipality that will take action based on 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 have the effect of show a list of license plates to who have made such request. The request can be personalized by express: a maximum number of vehicles that must be exhibited. 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 the can be retrieved specying the maximum number of vehicles that the request has to show.
  Who is making the request has also to give the area of the streets he/she wants to analyze. This is done by specifying the GPS coordinates of a fixed point and a maximum distance from that point. In this way, the user specifies a set of streets that he/she wants to analyze. 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 where it wants to have suggestions. The system will return a list containing in 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 the obtain where to park his/her vehicle:** This function allows the user to explictly ask the system to suggest places where park a 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 in 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. The procedure in which the area is given to the system as input is identical to the explanations given in the previous points.

2.3   User characteristics:

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

- Car Drivers.

- Motorcyclists.

- Cyclists.

A private user can be 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. All the private users are supposed to have downloaded the app from the store,to have done the sign in entering all the mandatory data and finally to have accepted all the permissions.

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 smartphones.

- 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 photoes in real time.

2.4.3 Dependencies:

- The system will use the GPS services provided by the smartphones.

- The system will use the internet services offered by the smartphones.