Contents

Introduction

1.1 Purpose

The purpose of this Requirement Analysis and Specification Document (RASD) is to provide a detailed description about the software SafeStreets. In particular, this document is focused on important aspects that are useful during the design of the software architecture like: scope, functional and non-functional requirements, use cases and scenarios, constraints and assumption, UML diagrams, limitation and interfaces with other softwares. Overall the document is a useful guide for the developers that will have to follow and implement all the necessary requirements; nevertheless it is also a document that can be given to potential customers to get them an idea of what the software will be like.
The software features have been identified in the following list of goals.

1.1.1 Goals

- [G1]: Allow a visitor to become a registered User after submitting his credentials for the registration to the service;
- [G2]: Allow the user to send a violation report consisting in a picture and some metadata;
- [G3]: Allow the user to watch the history of his reports and their status;
- [G4]: Allow the user to watch on map areas and streets with an high frequency of violations;
  - [G4.1]: The user gets the latest violations near its current location;
  - [G4.2]: The user gets the latest violations near the location that he specifies;

- [G.5]: Allow the local system administrator of the police station to create accounts for the police technicians;
- [G.6]: Allow the authorities to visualize the reports submitted by the users;
- [G.7]: Allow the authorities to mine data to make analisys and retrieve statistics;
- [G.8]: If the municipality offers a service that provides data about car accidents then the system must be able to cross this information with its owns data in order to identify possible unsafe areas;
  - [G.8.1]: In this case the system must be able to suggest possible interventions;
  - [G.8.2]: The users must be able watch this unsafe areas;
  - [G.8.3]: The authorities must be able to consult the crossed data.

1.2 Scope

1.2.1 Description of the given problem

SafeStreets is a service that aims to improve the safety of the streets via the help of the users. They can notify violations or any illegal behaviour related to street parking to authorities. In particular users can iteract with the service via an application that can be used to send the violation reports; the latters mainly consist in a picture, taken by the user, of the vehicle responsible for the violation. Moreover users can send, along with the picture, location, date and additional information related to the infringement. The system also provides a Web interface that can be used by authorities in order to check the violations recieved. It is important to note that the picture sent by the user must contain the car plate in order to let police officers know which is the real vehicle that committed the violation. The application also offers the possibility to see areas/streets with the highest violations rates thanks to the data collected over time. As an advanced functionality the application can interact with services offered by the municipality; in particular if a service offers data related to car accidents, SafeStreets can cross this information with its owns, in order to get a better idea of the potentially unsafe areas and therefore suggest some possible

interventions. Finally the application will have to be scalable and easy to use in order to provide a fast and efficient utilization for users that see a violation and want to report it immediately.

1.2.2 Current system

SafeStreets is a new service that it is entering the market. There aren't any legacy systems that need to be integrated into the application, with the exception of the third party services that offers some functionality required by the application (like Maps and plate recognition). Those services will be better explained in the following paragraphs.

1.3 Definitions, acronyms and abbreviations

1.3.1 Definitions

- *Most Dangerous Streets*: Streets with the highest frequency of violation reports;

1.3.2 Acronyms

- EU: European Union;
- CET: Central European Timezone;
- API: Application Programming Interface;
- HTTP: Hyper Text Transfer Protocol;
- GPS: Global Positioning System;
- GDPR: General Data Protection Regulation.

1.3.3 Abbreviations

- [Gn]: n-th goal;
- [Dn]: n-th domain assumption;
- [Rn]: n-th functional requirements;
- MDS: Most Dangerous Street;

1.4 References

1.5 Revision History

1.6 Document Structure

1. In the first part an overall vision of the system is given. We provide a rather summarized description of the purpose of the application and its scope as wall as a list of fundamental goals that the application must accomplish;

2. In the second part are described the general architecture of the software and its principal functions. Moreover in this section are also defined the classes of utilizators;

3. The third part is the core of the document. It defines: the requirements and the domain assumptions for each goal, a set of scenarios that describes real uses cases of the application as well as different types of UML diagrams related to different aspects of the service like use cases diagrams and sequence diagrams. It is also proposed a list of assumptions and design constraint and an analysis of the non-functional attributes of the system;

4. The fourth part propose the Alloy formalization of the system composed by its fundamental functionalities;

5. The last part states the hours of work division and the tools used to create all the part of this RASD document.

Overall Description

2.1 Product perspective

SafeStreets is a software that needs to be fast and reliable in order to allow the user to send a report immediately after he detects a violation. The goal is to design an architecture that can offer a good level of performance with respect to scalability and portability. The system is devided into three separate applications: a mobile front-end application for the users, a Web application for the authorities and a back-end system that manages all the operations. More in detail, the user application is a light-weight crossplatform mobile client; with this approach the client side is relieved from the computation and the result is a fast application for the users. With this mobile application the users can perform the main functions related to him, such as: registration, login, send reports, consult the history of his reports and watch a map with the MDS. The application is connected to the backend service which is the part of the architecture that handles all the main operations regarding the elaboration of data. In fact the backend is the core of the architetcure: it manages the incoming requests and handles the interaction with the third party system for the plate recognition. This is fundamental because the authorities need to receive the plate number in order to immediately identify the right veichle. This operation is done by sending the picture taken by the user to an external service that finds the plate number in the image and sends it back to the system as a string. Moreover the backend provides an interface that interacts with the cloud-based database in which the data will be saved. Finally it provides the functions to compute (on request) the MDS and other useful metrics. As an optional feature the backend can be configured to interact with the municipality service that offers data about car accidents. In this case, the backend compares the information sent with the data stored in the database. Therefore tries to identify the potentially unsafe areas and then suggests possible interventions. The Web application takes advantage

of the interactions with the remote database offered by the backend to provide a simple interface for the authorities in order to let them consult the stored data: they can query all reports, filter them by some specific criterias and analyze the data to get the statistics that they desire.

2.2 Product Functions

2.2.1 Send a report

This is one of the most important function of the service. After the user has successfully registered/logged in to the application, he can choose to send a report of a violation to the officers. This is performed by taking a picture of the car that committed the violation. After the device took the picture the application asks the user to choose from a list of common violations the category that best describes the event (i.e. No parking zone, Double parking). When this information is provided the application gathers some other data in order to have a correct and useful report. In particular it gets:

- The data related to the user who wish to send the report (User ID);
- The local time and date gathered from the phone OS;
- The information about the street where the user took the picture (Obtained by the GPS coordinates of the user).

The user can also optionally add the car plate numbers and the name of the street in case, for example, the application misrecognized the street from the GPS coordinates. When all the necessary data are correctly gathered, the report is sent to the backend system that will handle the request. The user can then consult the history section of the application to monitor the status of its report.

2.2.2 Receiving a report - Plate recognition

The specification document states that the license plate needs to be extracted from the image in order to save it as a correct report. In order to extract the plate number from an image an external service is used. This choice was made because, for this kind of application, an established and robust, high accuracy recognition service will certainly work better than a recognition system that has

to be built from scratch only for this kind of purpose. Therefore the backend system runs a task that listens for new reports and as soon as it receives a new one it immediately sends the image contained in the message to the plate recognition service. After the elaboration, the service returns the response of the recognition algorithm that should contain the text transcription of the plate. Note that if for some reason (i.e. bad picture) the plate recognition system can't recognize the plate, the backend system will instead use the plate number inserted by the user if provided; if not provided the violation won't be associated to any plate number. After this recognition step the backend system will interact with the database to create a record that contains the licence plate (if recognized/provided) and all the other information sent in the report. It is important to highlight the fact that the photo won't be discarded after the recognition as they can still be used for legal purposes.

2.2.3 Information mining

The system offers to both users and authorities the possibility to analyze the registered data although the level of visibility differs with respect to the utilizator. Users can only see the streets with the highest frequency of violations via the map present in the mobile application. Authorities instead have a wider access to data, that is to say they can access to all registered reports. Via the Web application they can also perform queries on the database (only selection queries) to mine the desired data and compute statistics and metrics for a deeper analysis. The role based approach has been designed in order to guarantee the privacy of car owners with respect to the users of the application. Therefore they will only be allowed to consult for the MDS.

2.2.4 Crossing data with the municipality

The backend system offers the possibility to interact with an external service, offered by the municipality, that provides data regarding car accidents happened in the municipality area. SafeStreets can define the potential unsafe areas of the municipality by merging car accidents data and viaolation reports stored by the application. The system can then suggest to the local system administrator possible interventions in order to improve the safety of this areas. It is important to note that the data offered by this service need to be structured in a standard way recognizable by the backend system, otherwise the information crossing will

not provide any results.

2.3 User Characteristics

2.3.1 Actors

- *Visitor*: a person without a SafeStreet account. Visitors can only have access to the homepage and the registration form of the mobile application;
- *User/ Mobile user*: a person correctly registered to the SafeStreet account service. Users/Mobile users can perform any of the actions made available by the SafeStreet mobile application;
- *Recognized authority*: a recognized authority (Police station) which submitted to SafeStreet and can interact with it through its web application interface;
- *Local system administrator/Police corporal*: a person which belongs to a recognized authority, in charge of dealing police technician accounts and scheduling patrols;
- *Police technician*: a policeman encharged of dealing with the violations reports . He/She patrols the unsafe areas and gives fines to the reported cars in case of violations;
- *Third party recognition service*: an image recognition service which allows SafeStreet to extract car plates from the violation report's images.

2.4 Assumptions and dependencies

As far as the specification document is concerned, it is necessary to specify some details and to state clearly a few ambiguous points. In order to better clarify those situations, the following assumptions are intoduced.

2.4.1 Text Assumptions

- **Violation report**

  - The information sent by user in the violation report includes: the license plate image and an optional textual transcription, its position coordinates (extracted automatically from the smartphone GPS) and the violation metadata;
  - The suitable metadata descripted in the specification document is intended as a choice of the law infringment and a textual description of the event;
  - Given the coordinates of the violation, the system is able to retrieve the address from which the report was sent;
  - Every time a report is received, the system will ask the image recognition service for the license plate textual transcription;
  - A license plate textual transcription is considered correct if it fits into one of the common standards of the EU states. In case of wrong textual transcription of the license plate from the image recognition service:
    * The textual transcription provided by the user is taken into account. After a previous check, the provided license plate will be considered as correct;
    * If the user has not sent a textual transcription, the violation is not associated to any license plate number;

- **Mining information**
  - End users are allowed to mine information about the violation reports that they sent. Also, they can have access only to the (map visualization or list) MDS;
  - Authorities can mine information concerning all the stored reports in the SafeStreet system, such as MDS or the list of cars with an high number of violation reports;

- **Suggested intervention**
  - Only authorities are allowed to have access to the suggested intervention on unsafe areas identified by the system.

2.4.2 Domain assumptions

- [D1] A picture taken with a smartphone is performed with a quality sufficient for the image recognition service to transcribe it;
- [D2] The reported picture contains only the license plate of the car that committed the violation and not others;
- [D3] The GPS information collected from the smartphone of a user has an accuracy of less than 5 meters;
- [D4] The timestamp collected from the smartphone is synchronized with the CET standard;
- [D5] The users can only report violations occured in Europe;
- [D6] Reports are only sent through a secure connection channel;
- [D7] The users reports all the violation that they detect;
- [D8] The users send report containing correct information about the detected violation;
- [D9] The municipality system stores correctly all the information concerning car accidents;
- [D10] The municipality system allows SafeStreet to retrieve information about the car accident among a certain area;
- [D11] The municipality system sends data accordingly to a common standard which is comprehensible by the SafeStreet system;
- [D12] The image recognition service is able to detect and transcribe license plates.

Specific requirements

3.1 External interface requirements

3.1.1 User Interfaces

This section shows a mockup version of the main functionalities of the mobile application and the web application of the final version of SafeStreets.

- Mobile application:
  - LogIn interface
  - Violation report form
  - History section
  - Map section
- Web application;
  - Login interface
  - Violation reports page
  - Specific report page
  - Report mining page
  - Statistics page

3.1.2 Hardware Interfaces

SafeStreets do not require any specific hardware device, thus there are no hardware interfaces to other existing systems

3.1.3 Software Interfaces

SafeStreets is a standalone service and does not share any API to external systems. However, the system requires the interaction with the Image recognition service and with the municipality's legacy system to perform its tasks.
The software requires a standard format in order to retrieve car accident data from the municipality's system. Such standard will be specified more in details in the DD.

3.1.4 Comunication Interfaces

External services are not allowed to interact actively with SafeStreets, therefore the system does not offer any comunication interface to external systems.

3.2 Functional Requirements

3.2.1 Scenarios

Scenario 1

John is a very sportsman person who likes to ride his bicycle a lot. So as every Sunday afternoon he decides to go for a ride along the bicycle lane near his house. After some time he sees a car parked right on the bicycle lane that limits the passage of bicycles. Therefore John decides to open up the SafeStreets application on his smartphone and takes a picture of the car. After that he selects the "No parking zone" as a violation category and he even adds the car plate number to the report just to make sure that it will get correctly processed by the officers.

Scenario 2

Albert's son Freddy has recently been operated to his leg so the municipality released to him the pass that allows Albert to park his car in places reserved for people with disabilities. Albert has to bring Freddy to the doctor for some checks every week. Sometimes he manages to find the parking spot in front of the hospital reserved for people with disabilities free but some other times the place is occupied by some other car that does not have the pass. When this happens he needs to find another parking spot which may not be close to the

hospital and therefore he his forced to make his son walk to the centre which can be dangerous as he has just been operated. After some weeks in this conditions Albert decides to use the SafeStreets service in order to prove if the service can be really useful or not. Therefore he sends a report every time he finds this kind of situation. After some weeks he notices that the parking spot is either free or regularly occupied indeed he checks his report history to find out that all of his requests have been take into account by the police.

Scenario 3

Like every morning Kevin wakes up to go to work. Around 7 o'clock he leaves home to get to the parking slot where he left his car yesterday. When he arrives he notices that the car parked behind his has crushed into his rear bumper leaving a noticeable dent. He does not have time to call the police because he is in a hurry for an important meeting at work so he decides to immediately send a crush report by taking a picture of the unpleasant event, after that he rapidly gets on his car in order to arrive in time at work. When he returns from work he opens up the SafeStreets app once again to check the status of his report. Fortunately he finds out that his request has been taken into account by a local officer.

Scenario 4

Marco has been invited to his friend Luigi's graduation party organized to celebrate his newly graduated master degree in Computer Science and Engineering obtained at Politecnico di Milano. Since he lives in a small city with very limited public transportation solutions to go to Milano he decides to go with his car to event. When he is almost near to the designated place he starts to think where to park his car. After some time he remembers that Luigi told him to download the SafeStreets app in order to check which are the most unsafe streets in the area. He stops for a second and opens the mobile application, then he navigates to the map section and finds on the map the location where the place is held. The application then enlightens the most dangerous streets near this location so that Marco can avoid parking his car there.

Scenario 5

Leon is a newly hired police technician for the Berlin police department. For his first days of work the corporal assigned to him some routine work mainly represented as patrols or giving tickets. In order to check where he needs to go today he logs in with his credentials in to the SafeStreets Web application from his computer located in the police station. He then consults the new violation reports submitted in the city in order to find out their locations.

Scenario 6

Daniele is the local system administrator of the Genova police department and he is responsible of dispatching the patrols. He decides that he wants to schedule the officers in a more intelligent way. To do that he consults the SafeStreets report database by logging in the web application. He would like to send the more experienced officers to areas in which there is a high number of violations categorized as "crash violations" whereas he would like to assign regular officers to other streets with an high frequency of violation of any category. In order to find out those streets he write down a query and execute it via the web application; after some seconds the interface displays the desired results, therefore he is able to schedule the patrols as he wants. Daniele would also like to communicate to all the officers which are the 10 cars that committed the highest number of violations in the city. To do that he once again performs a query on the SafeStreets web application searching for the 10 car plates with the highest rate of violations.

3.2.2 Use cases

Name	Sign Up
Actors	Visitor
Entry Condition	The visitor has opened the application on his device
Event flow	1. The Visitor chooses the Sign Up option; 2. The Visitor fills the mandatory fields regarding his personal data (i.e. email, password); 3. The Visitor fills the optional fields (i.e. Name,Surname, age) 4. The Visitor confirms the registration; 5. The system saves the information and creates the User account.
Exit condition	The User is registered.
Exception	1. The user was already registered
Special requirement	TBD.

3.3 Performance requirements

In order to promptly tackle violations that are reported, SafeStreets has to be a low latency service: police technicians have to intervene quickly, in order to fine the transgressors before they leave the scene of the violation. Also, the system has to guarantee an efficient service to an high number of users and police technician connected simultaneously.

3.4 Design constraints

3.4.1 Regulatory policies

The user information is stored accordingly to the GDPR policies in order to guarantee the privacy of individuals.

- No data is shared with third parties for commercial purposes;
- All reported images and violation data are stored safely through encryption methods. The third party system used for image recognition can not store any information/image which identifies;
- Any additional information, such as GPS position or Camera access, is promptly asked to the user before performing the specific operation, accordingly to the Android/IOS standards.

3.4.2 Hardware limitations

- Mobile applications:
  - Any kind of modern smartphone;
  - Internet connectivity;
  - Camera;
  - GPS.
- Web application:
  - Browser (any HTTP client);
  - Internet connectivity.

3.4.3 Interface to other applications

- A certified external image recognition service deals with the car plate recognition through report's images;
- Report's information are stored in a cloud database which guarantees data encryption and information retrieving through authentication;
- The municipality information center is periodically asked to provide data about the car accidents that happen among the municipality's territory.

3.5 Software system attributes

3.5.1 Reliability

The system needs to always be online in order to correctly store and manage violation reports and to be robust in case of failures. These requirements can be guaranteed through a virtual replication of the backend system: a possible solution could be implemented through Docker containers managed by a container orchestrion such as Kubernetes or DockerCompose. The container architecture automatically restarts containers in case of failures (fault tollerant) guaranteeing the system functioning in every situation.

3.5.2 Availability

As previously stated, the system must always guarantee an operating service of violation reports. Thus, the system has to be available under stressing condition of an high number of users.

3.5.3 Security

As the data transferred and stored in the SafeStreets system deals with private citizen's data, it must be guaranteed an high level of security. In order to meet the GDPR conditions, all data has to be correctly encrypted through several security methods before being sent and stored.

3.5.4 Maintainability

The software maintainability is one of the most critical aspect in the development of this software. In order to allow cheap and easy fix, SafeStreets is developed by using several design patterns that make the software flexible and scalable. Such patterns can be recognized both in the previous UML class diagram and in further documents.

3.5.5 Portability

SafeStreets is intended to work on every mobile device, thus the technologies used for its development have to be cross-platform (such as Flutter or any other language which can generate applications for IOS and Android systems).