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 Overview

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 golas 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 constraint and the assumption of the application and the classes of utilizators

3. The third part mainly defines the requirements and the domain assumptions related to each golas. It also provides an analysis for the non-functional requirements

4. The fourth part consist in a set of scenarios that describes real uses cases of the application

5. In the fifth chapter we propose of different type of UML diagrams that describes different aspects of the service. In particular we offer use cases diagrams, class diagrams, statechart diagrams and sequence diagrams

6. This part contains the Alloy model that ... TO BE FINISHED

7. 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 add some textual information about the event. When this information are 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 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.

3.3 Design constraints

3.3.1 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.

3.3.2 Hardware limitations

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

3.3.3 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.

3.4 Software system attributes

3.4.1 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.

3.4.2 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.

3.4.3 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.

3.4.4 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.

3.4.5 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).