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 that we wish to develop. 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, class/sequence 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's also a document that can be given to potential customers to get them an idea of what the software will be like.

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 driving 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 of 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 the authorities in order to check the violations recieved. It's important to note that the picture sent by the user must contain the car plate in order to let the police officers know which is the real vehicle that committed the violation. Another functionality of the application offers to both users and authorities 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 accidents SafeStreets can cross this data with its own data in order to get a better idea of the potentially unsafe areas and therefore suggest some possible interventions. Ultimately 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 immediately report it.

1.2.2 Current system

SafeStreets is a new service that it's entering the market right now,so all the architecture will have to be built from scratch. There aren't any legacy systems that needs 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.

1.2.3 Goals

- [G1]: Allow a person 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

    - [G2.1]: Allow the user to send along with the picture the plate number

- [G3]: Allow the user to watch the history of his reports and their status

- [G4]: Allow the user to watch on a 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 technician by giving him an administrator account

- [G.6]: Allow the authorities to visualize the reports submitted by the users

- [G.7]: Allow the authorities to mine the data to make analisys and retrieve statistics

- [G.8]: If the municipality offers a service that provides data about accidents then the system must be able to cross this data with its own 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

1.3 Definitions, acronyms and abbreviations

1.3.1 Definitions

- *Most Dangerous Street*: 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 sees a violation, so our goal was to design an architecture that can offer a good level of performance with respect to scalability and portability. The decision that has been made was to divide the system 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 cross-platform front-end mobile client; with this approach the client side is relieved from the computation and the result is a fast application for the user. With this mobile application the user 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 then 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 request 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 is done by sending the picture taken by the user to this external service that finds the plate number in the image and send 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 and ultimately it provides the functions to compute (on request) the MDS and other useful metrics. The backend can also be configured to interact with the municipality service that offers data about accidents. In this case the backend compares the data sent with the data saved in the database and then tries to identify the potentially unsafe areas and then suggests possible interventions. The Web application takes advantage of the in-

teractions with the remote database offered by the backend to provide a simple interface for the authorities in order to let them consult all the data. Via this web interface they can see all the reports, filter them by some specific criteria 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 into 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 phone 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 sent 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 form 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. So to extract the plate number from an image we thought to use an external service that does the job. 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 we should build indoors 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 photo) 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 plate number string (if recognized/provided) and all the other information sent in the report. Is important to highlight the fact that the photo won't be discarded after the recognition as they can be still used for legal purposes.

2.2.3   Information mining

The system offers to both the users and the authorities the possibility to analyze the registered data although the level of visibility differs with respect to the utilizator. The users can only see the streets with the highest frequency of violations. They can visualize those streets via the map present in the mobile application. The authorities instead have a wider accessibility to the data, in fact they can access to all the reports registered. Via the Web application they can perform queries on the database (only selection query) to mine the desired data and they can also compute statistics and metrics for a deeper analysis. With this role based approach the users will only see the MDS as we think this is the only information really useful for them, while for the authorities we leave access to all the reports as they can be used to mine all sorts of statistics useful for the police.

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 accidents happened in the municipality area. If this external service exists for the specified municipality than SafeStreets can compute a sort of "merge" between the data offered and the reports related to the municipality area in order to find out the potential unsafe areas. The system can then suggest possible interventions in order to improve the safety of this areas. Is important to note that the data offered by this service

needs to be structured in a way that the backend system can recognize otherwise the information crossing won't 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/ municipality) 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 report. 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   Constraints

2.4.1   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.

2.4.2 Hardware limitations

- Mobile applications:

    - Any kind of modern smartphone;

    - Internet connectivity;

    - Camera;

    - GPS.

- Web application:

    - Browser (any HTTP client);

    - Internet connectivity.

2.4.3 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.

2.5 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.

2.5.1 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 standard 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) streets with the highest frequency of violation reports (called "most dangerous" streets (MDS) from now on);

  - Authorities can mine information concerning all the stored information 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.

2.5.2 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 legacy system allows SafeStreet to retrieve information about the car accident among a certain area;

- [D11] The image recognition service is able to detect and transcribe license plates;