POLITECNICO DI MILANO

SOFTWARE ENGINEERING II

SafeStreets - Design Document

Version 1.0

Authors
Iacopo Marri
Manuel Salamino
Steven Alexander Salazar Molina

December 1, 2019

Contents

1 Introduction

1.1 Purpose

This document provide an overview of *SafeStreets* application, explaining how to satisfy the project requirements stated in the RASD.
It's mainly intended for developers and testers and they can find a functional description of the components of the System, their interactions and their interfaces, and also how they will be implemented.

Finally, all the requirements expressed in the RASD document will be dealt with what is expose in this one, describing how the components presented can satisfy them.

1.2 Scope

As explained in the RASD, *SafeStreets* is an application that was created with the intention of monitoring the compliance with traffic regulations.
Its main goal is to allow Users to notify traffic violations.
Then the application must collect the reports sent by the Users and allows Authorities to verify if ecah violation is sanctionable and, if it is, provide them the possibility of generate Traffic Ticket.

Moreover, by crossing its collected data with the Municipalities ones, is also possible to provide to Users and Authorities statistics about the effectiveness of *SafeSteets* and suggestions about how to avoid reccurrent accidents.

The application provides Users a way to send data about a violation (most common are parking violations ones, e.g. vehicles parked in reserved places), like the kind of violation, pictures of the involved vehicles, and the date and the position in which the violation occurred. The User can also specify the license plate of the vehicle and the name of the street, but in case he/she doesn't, the application is equipped with algorithms capable to obtain those and other metadata by pictures and position.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- **Guest**: a person who has not logged in or registered yet and cannot use the functionalities of the System.
- **User**: a person that uses the application to send notifications of traffic violations.
- **Authority**: a municipality worker that is able to create traffic tickets depending on the violation that a person has committed.

1.3.2 Acronyms

- DBMS: Data Base Management System
- UI: User Interface
- API: Application User Interface
- OS: Operating System
- REST: REpresentational State Transfer
- HTTP: HyperText Transfer Protocol
- UX: User eXperience

1.3.3 Abbreviations

- [**R.i**]: i-th functional requirement.

1.4 Reference Documents

- RASD document
- Project assignment specifications

1.5 Document Structure

Architectural Design: shows the main components of the System and their relationships. This section also focus on design choices and architectural styles, patterns and paradigms. **Algorithm Design:** presents **User Interface Design:** contains ...

Requirements Traceability: shows ...

Implementation, Integration and Test plan: explain ...

2 Architectural Design

2.1 Overview

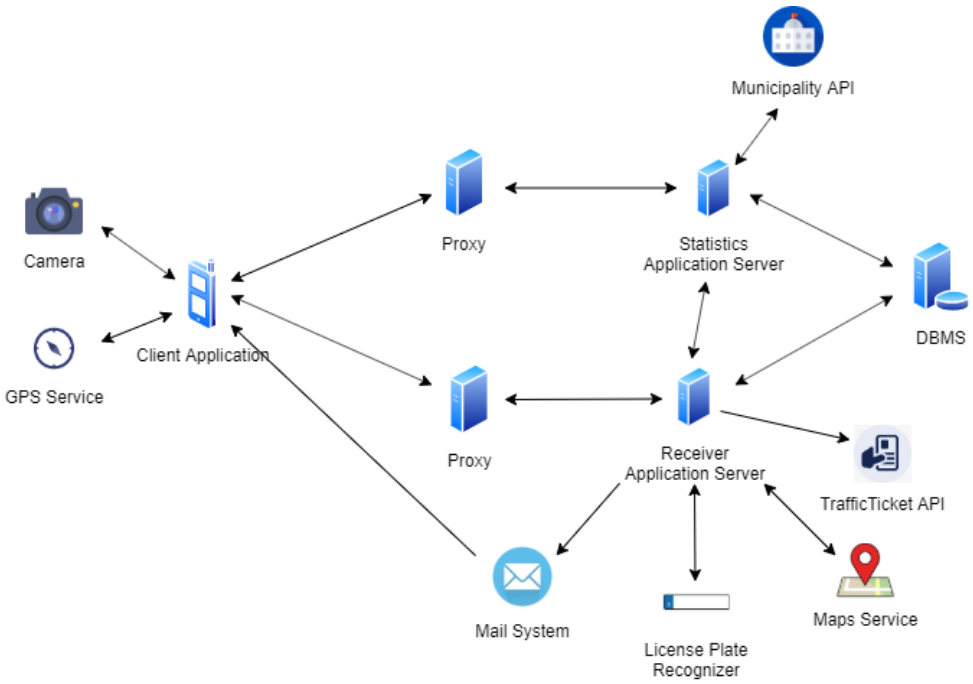

Figure 1: System overview

The above figure shows an high level overview of the System's architecture. We can immediately identify that is divided in two parts: the Client Application part and the Application Servers part.
The former shows the services used by the Client Application and its communication with the Application Servers.
The other one indicates how the System work with the external functionalities in order to accomplish the required Goals.
Further details on the System components and their interactions will be explained in detail in the following sections.

2.2 Component view

2.3 Deployment view

The distribution of components capturing the topology of the system is illustrated below by using a deployment diagram.
The system is structured in a multitier architecture.

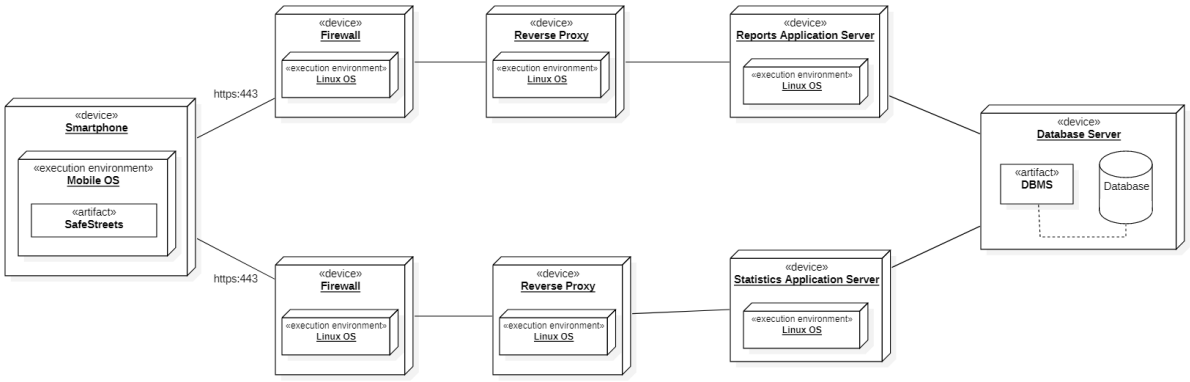

Figure 2: Deployment Diagram

Smartphone
This node represents the smartphone of the User or Authority, both can access to the functionalities of SafeStreets by using their own smartphone.
Firewall
This component allows to protect the Reverse Proxy by filtering the packets received from *internet*.
Reverse Proxy
We chose to deploy a reverse proxy in order to increase parallelism and scalability. Moreover, the reverse proxy is responsable of the load balancing since the component receiving all the request from internet (filtered by the Firewall). Another reason for the deployment of The Reverse Proxies is because it increases security and anonymity by protecting the identity of our backend servers.
Receiver Application Server
This component contains all the logic used to receive reports from Users and the forwarding of those to the corresponding Authorities. This component may also be replicated in order to balance the workload.
Statistics Application Server
In order to decouple all the logic behind the functionalities offered by SafeStreets, it has been decided to use a seperate Server that allows to compute satistics by accessing to local information and municipalities' public information.
Database Server
This part of the System is equipped with a relational DBMS, it allows to store and retreive all the data needed by the Application Servers.

3 User Interface Design

The mockups of the application were already presented in *Section 3.1.1* of the RASD. This section illustrates the User Experience flow by using two User Experience diagrams. Both diagrams are pretty similar since some interfaces are the same for both common User and Authority. As we can see from *Figure 5* and *Figure 6* the leftmost part is merely devoted to allow users to log into the application:

- Splash Screen
- Login
- Register
- Recovery Password

The Main Menu is different for User and Authority since an Authority is able to:

- View the violations notified by Users and generate traffic tickets
- View the own traffic tickets generated
- Get Suggestions

And the User is able to:

- Report a traffic violation
- View the own reports list

However, both share also some other interfaces in addition to the ones related to the login:

- Statistics
- Area with higest frequency of violations
- Settings
- About SafeStreets
- Contact SafeStreets

Settings. About and *Contact SafeStreets* are accesssible through the button in the top-left corner from the Main Menu. Those 3 activities weren't presented in the RASD because we didn't consider them as really important for the RASD.

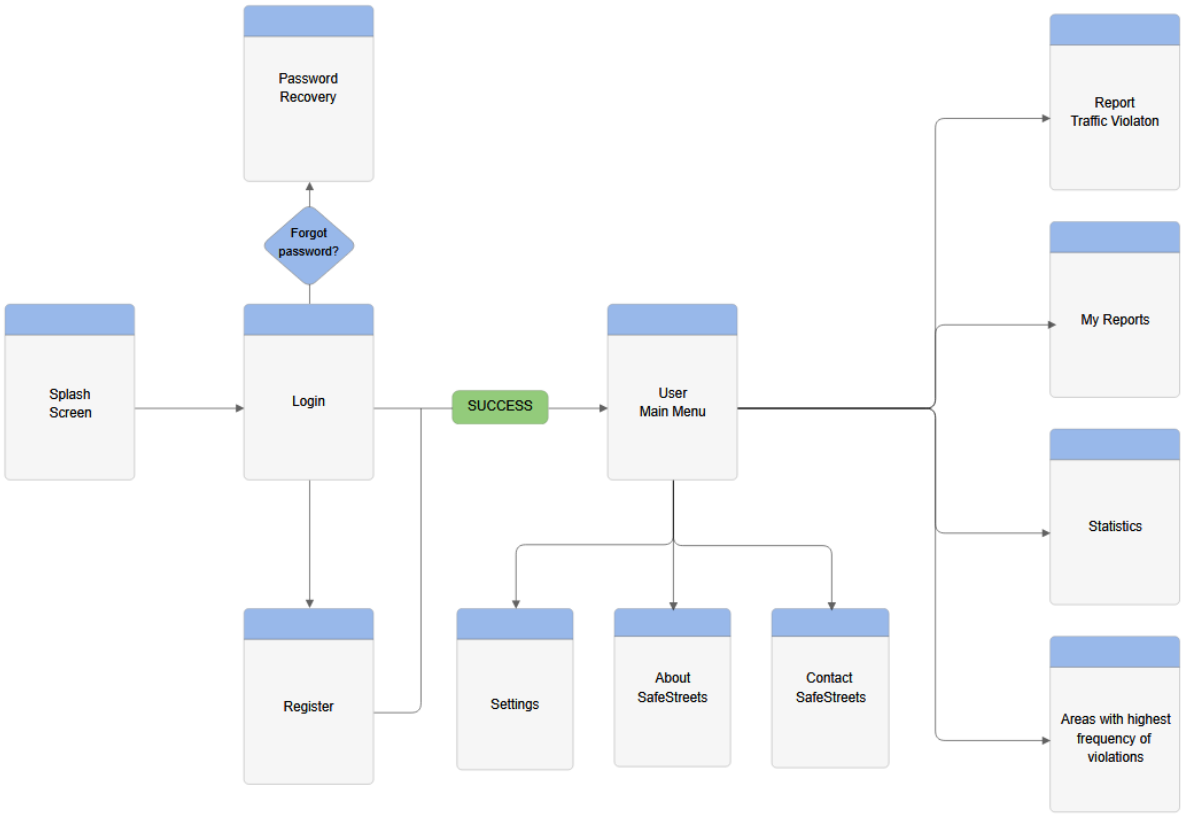

Figure 3: User UX Diagram

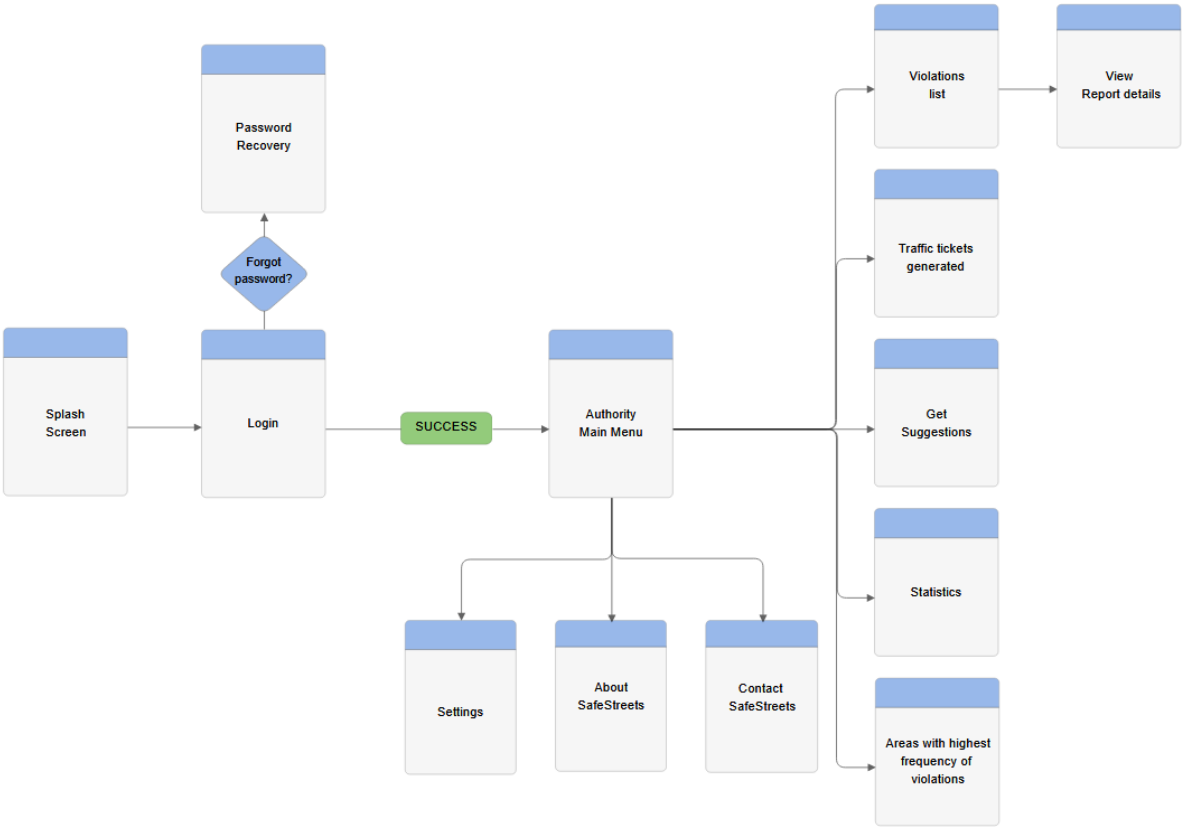

Figure 4: Authority UX Diagram

4 Requirement Traceability

5 Implementation, Integration and Test Plan

6 Effort Spent

The following tables summarize the effort spent by each member of the team to create the RASD document.

6.1 Marri Iacopo

Description of the task	**Hours**
First meeting + Github setup	4
Purpose, Scope and Definition	5
Product Perspective	5
Product Functions	3
User Characteristics and Assumptions, dependencies and constrains	2.5
Specific Requirement	8
Alloy	4

6.2 Salamino Manuel

Description of the task	**Hours**
First meeting + Github setup	4
Purpose, Scope and Definition	6
Product Perspective	8
Product Functions	2
User Characteristics and Assumptions, dependencies and constrains	2
Specific Requirement	4
Alloy	10

6.3 Salazar Molina Steven Alexander

Description of the task	**Hours**
First meeting + Github setup	4
Purpose, Scope and Definition	5.5
Product Perspective	5
Product Functions	4
User Characteristics and Assumptions, dependencies and constrains	4.5
Specific Requirement	6
Alloy	5

7 References