POLITECNICO
MILANO 1863

SafeStreets project
Manuel Pedrozo, Tomás Perez Molina

Design Document

Deliverable:	DD
Title:	Design Document
Authors:	Manuel Pedrozo, Tomás Perez Molina
Version:	1.0
Date:	November 27, 2019
Download page:	https://github.com/lethanity/PedrozoPerez
Copyright:	Copyright © 2019, Manuel Pedrozo, Tomás Perez Molina – All rights reserved

Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose

The purpose of this document is to dive into technical details concerning the SafeStreets system. In the RASD, the system and its functionalities were introduced in an abstract manner. The Design Document focuses more on implementation, explaining the decisions that had to be made and the reasoning behind them. The topics covered in this document are the following:

1.2 Scope

The SafeStreets system is designed to provide users with the ability to report and get information of reported traffic violations through an application. Any user with a device capable of running the application can sign up to the system, which enables them to access its functionalities. In order to submit a report, the user needs to fill a form. In it, they have to enter the license plate number of the vehicle committing the violation, the type of violation and at least one photo of the scene, where the license plate of the vehicle can be easily recognized. This data, along with metadata retrieved from the user's device (geographical position, date and time) is then sent to the system. The system is responsible for analysing the validity of the report. To achieve this, a license plate recognition algorithm is utilized. When faced with difficulties in the detection, the photo must pass through community review, in which users reach a consensus on the validity of the report photo. The data collected by the system in relation to reports is to be queried by its users. There are two distinct targets of this functionality: standard users and the municipality. The main difference between the two is that the municipality can access information that should not be freely accessible to everyone because of security and privacy concerns. Through the application, users are capable of visualizing a city map showing where the violations happened. Furthermore, a public API is made available, facilitating data analysis and system integration.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- Traffic violation: An action performed by a driver of a vehicle which is against the local traffic regulations.
- Report: Information submitted by a user to notify the system and, by extension, authorities of a traffic violation.
- Compromised report: A report that has been modified by an unauthorized agent outside the system boundaries.
- Authority: A local agency whose purpose is, as indicated by the current law, to enforce traffic rules. For example: the police.
- Ticketing system: A government database containing information about issued traffic tickets.
- License plate registry: A government database connecting a license plate with the car registered to it and information about it such as the make, model and color.
- False-positive report: An invalid report that is considered valid by the system.
- False-negative report: A valid report that is considered invalid by the system.

1.3.2 Acronyms

- GPS: Global Positioning System
- API: Application Programming Interface

1.3.3 Abbreviations

Gn	: n-th goal
Dn	: n-th domain assumption
FRn	: n-th functional requirement
SRn	: n-th security requirement

1.4 Revision history

1.5 Reference Documents

1.6 Document Structure

2 Architectural design

2.1 Overview

2.2 Component view

2.3 Deployment view

2.4 Runtime view

2.5 Component interfaces

2.6 Selected architectural styles and patters

2.6.1 Architecture patterns

- **Three tier architecture:** As already mentioned, the SafeStreets system is divided into three tiers: Presentation, Business Logic and Persistence tier.

- **Client-Server architecture:** Computing model in which the server hosts, delivers and manages most of the resources and services that are consumed by the clients.

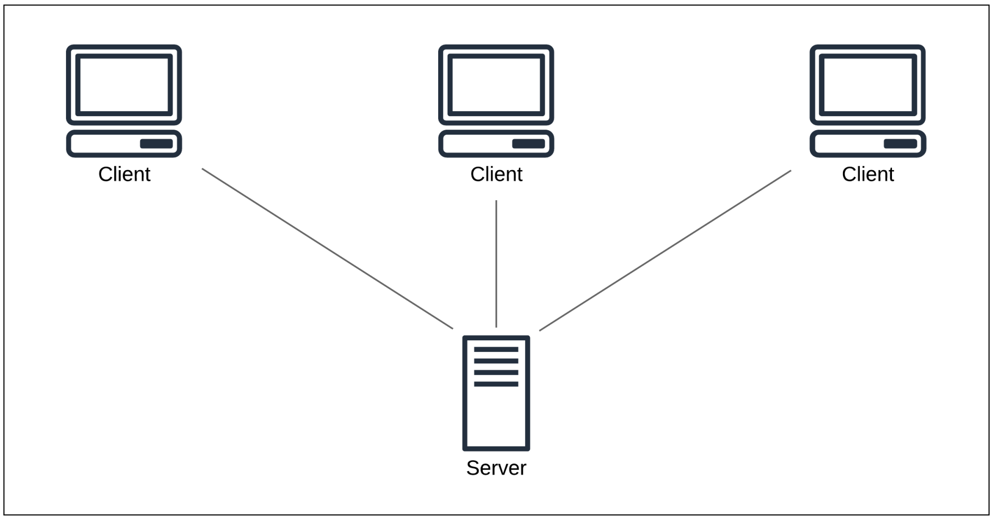

Figure 1: Client-Server architecture.

The mobile devices running the application are the clients, which interact with the business logic tier (Application server); it is here that the heavy computation of the system is done. The "Thin Client Server architecture" can also be referenced.

2.6.2 Design patterns

- **Model-View-Controller:** Divides the program logic into three interconnected elements.
  - Model: Central component. Manages the data, logic and rules of the application.
  - View: Visual representation of the model.
  - Controller: Accepts inputs and converts it to commands for the model or view.

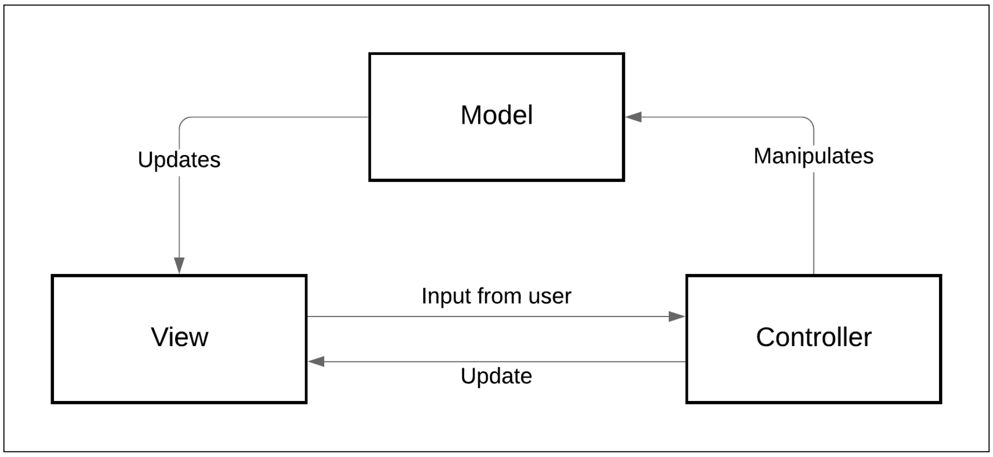

Figure 2: Model View Controller pattern.

This pattern is used in the development of the mobile application, which is the part of the system the user directly interacts with.

- **Facade pattern:** Provides a simple interface to a larger body of code. This is used in the backend to expose the API to external users. The router component acts as the facade, interacting with the lower level components and exposing the appropriate endpoints.

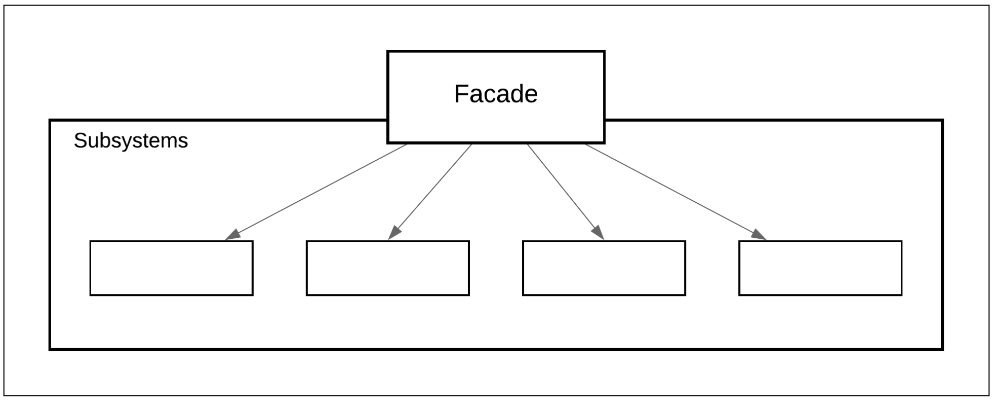

Figure 3: Facade pattern.

- **Dependency injection pattern:** A technique whereby one object supplies the dependencies of another object.
  Utilized in the backend to solve the dependencies between services.

2.7 Other desing decisions

3 User interface design

The following mockups represent the final look of the application in great detail. They are based on the sketches provided in the RASD. Material design principles were used as inspiration in the design process.

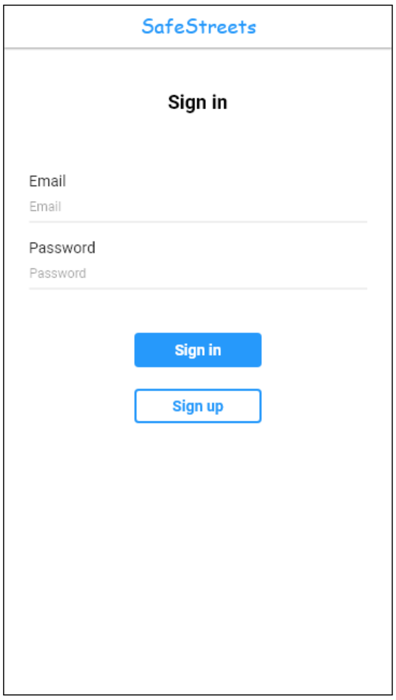

Figure 4: Mockup - Sign in.

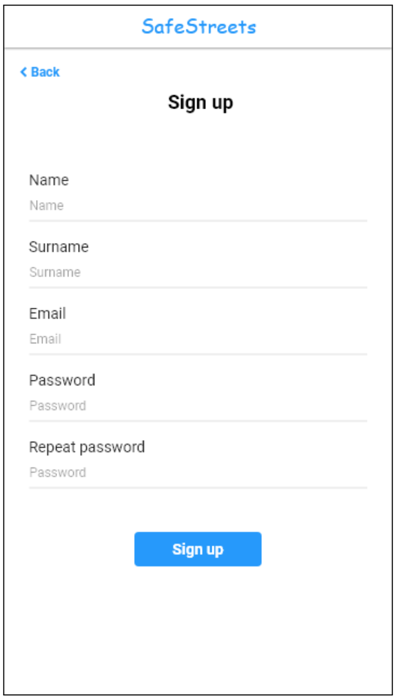

Figure 5: Mockup - Sign up.

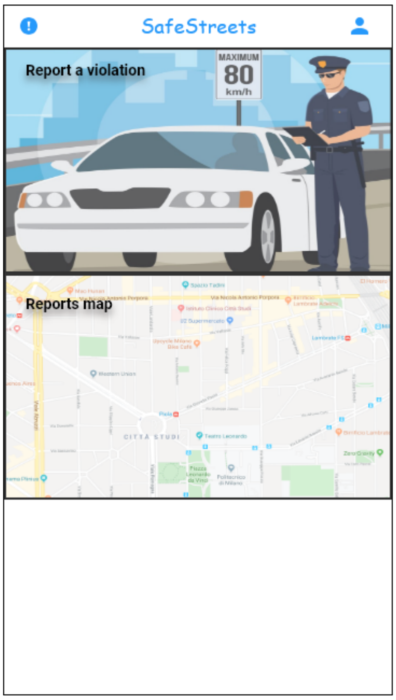

Figure 6: Mockup - Home.

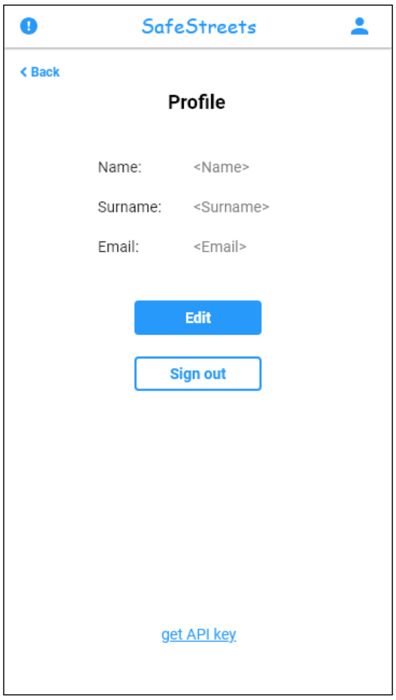

Figure 7: Mockup - Profile.

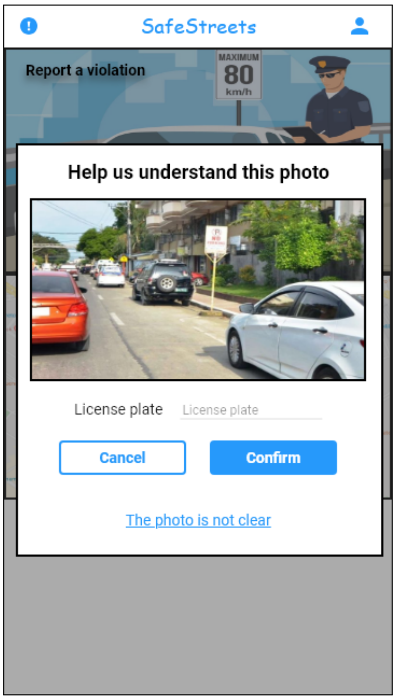

Figure 8: Mockup - Photo review.

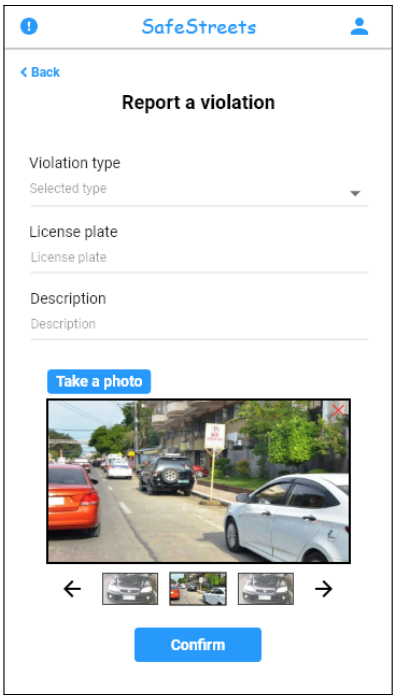

Figure 9: Mockup - Report violation.

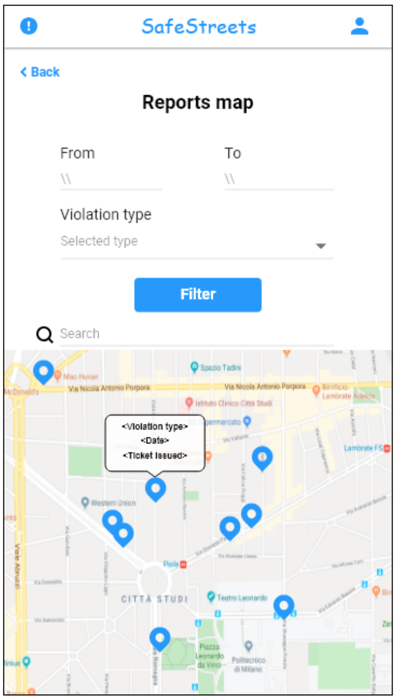

Figure 10: Mockup - Reports map.

Here is a description of each screen and its functionalities:

- Sign in: Form for the user to access the system.
- Sign up: Form for the user to register in the system.
- Home: Main screen of the application, from here the user can access every functionality.

- Profile: Shows the user's personal information and allows them to modify it. Also allows the user to request an API key and sign out.
- Photo review: The user is required to input the license plate being shown in the photo, or specify that is not clear enough.
- Report violation: Form to submit a report violation. Includes a preview of the photos taken where they can be discarded.
- Reports map: A map with the reports is shown, the user is capable of filtering the results by date, type and location. When pressing on a report, a popup with details appears.

The flow of the application can be seen in [figure n], where the arrows represent a transition between two screens. As it can be appreciated, every functionality can be accessed from the home screen.

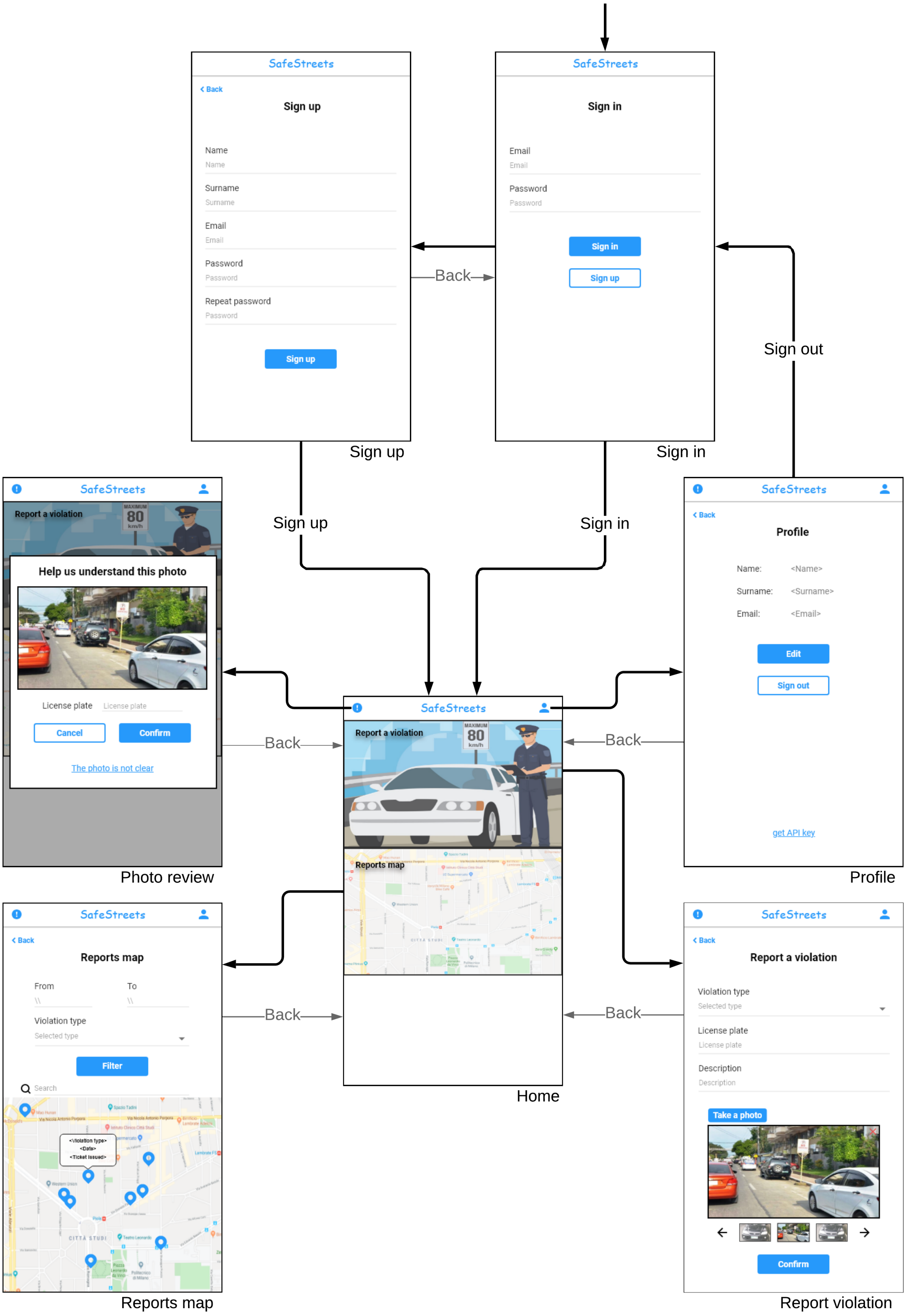

Figure 11: Mobile application flow.

4 Requirements traceability

Every design decision made and presented in this document was decided upon with the goals previously specified in de RASD in mind. From these goals, requirements were extracted; the following list provides a mapping between the goals and requirements, and the system components that help to accomplish them.

4.1 Functional requirements

[G1] - The user is able to report a traffic violation to authorities.
Requirements [FR1] - [FR2] - [FR3].

- <>

[G2] - The user is able to visualize reports in a specified area and time.
Requirements [FR4] - [FR5] - [FR6] - [FR7].

- <>

[G3] - It is possible to query report information in an easily parsable format to allow for data analysis.
Requirements [FR9] - [FR10] - [FR11] - [FR12] - [FR13].

- <>

[G4] - Only authorities are able to access report pictures and license plates.
Requirement [FR14] - [SR1].

- <>

[G5] - Compromised reports are detected and discarded.

- <>

[G6] - Authorities are able to access a curated list of reports which are considered to have a higher than average level of accuracy and reliability.
Requirements [FR15] - [FR16] - [FR17] - [FR18] - [FR19] - [SR2].

- <>

[G7] - It is possible to determine if a particular report contributed in issuing a traffic ticket.
Requirements [FR20] - [FR21].

- <>

5 Implementation, integration and test plan

5.0.1 Technologies

5.0.2 Implementation and integration

5.0.3 Testing

6 Effort Spent

6.1 Manuel Pedrozo

Task	**Hours**
-	-

Table 1: Effort spent by Manuel Pedrozo

6.2 Tomás Perez Molina

Task	**Hours**
-	-

Table 2: Effort spent by Tomás Perez Molina

- “SafeStreets Mandatory Project Assignment”