POLITECNICO
MILANO 1863

# *SafeStreets*

Software Engineering 2 Project - Prof. Matteo Rossi

**DD Document**

Salvatore Fadda - 944786
Adriano Mundo - 944684
Francesco Rota - 948714

A.Y. 2019/2020
Version 1.0

December 9, 2019

# Contents

# 1 Introduction

## 1.1 Purpose

This document represent the *Design Document* (DD). It aims at providing an in-depth description of the architecture below *SafeStreets* application and its services. It will present a section related to the architectural design with different perspective on the components of the *System*, how they interact and how they will be implemented. All the requirements of the RASD document are mapped with the components to explain how they will be satisfied. Finally, a section for the testing plan for Q&A team is provided.

## 1.2 Scope

*SafeStreets* is a crowd-sourced application that aims at keeping safe the city's streets. The idea behind this service is to allow *Users* to notify the *Municipality* when a violation occur on the streets under its jurisdiction. The *User* can notice and notify the violation by sending a photo of the violation including date, time and position. *SafesStreets* stores all the data and uses a plate recognition algorithm to recognise the image content.

The **Basic Service** allows *Users* and *Authorities* to mine the information collected by the service, so they can access statistics built from the data.

As **AF1**, the application *SafeStreets* identifies potentially unsafe areas and suggests possible interventions to *Authorities* to solve the founded issues.

As **AF2**. the application *SafeStreets* allows the *Municipality* to generate traffic tickets directly from the application data. Also, using the data of issued tickets the *System* can build statistics and find insights to suggest to *Municipality* in order to improve their service.

## 1.3 Definitions, Acronyms, Abbreviations

### 1.3.1 Definitions

- **Client:** a piece of computer hardware or software that accesses a service made available by a Server.
- **Server:** a computer program or a device that provides functionality and handle the requests of other programs or devices, called Clients.
- **N-Tier:** or multilayer is an architecture in which presentation, application processing, and data management functions are physically separated in n-layers.

### 1.3.2 Acronyms

- **GPS:** Global Positioning System
- **API:** Application Programming Interface
- **RASD:** Requirements Analysis and Specification Document
- **DBMS:** Data Bases Management System
- **GDPR:** General Data Protection Regulation
- **MVC:** Model-View-Controller
- **REST:** REpresentational State Transfer
- **Q&A:** Quality and Assurance
- **UI:** User Interface
- **HTTPS:** Hyper Text Transfer Protocol Secure

### 1.3.3 Abbreviations

- **[Rn]:** n-th Functional Requirement
- **A1:** Advanced Function One
- **A2:** Advanced Function Two

## 1.4 Revision history

| Version | Date | Description |
|---|---|---|
| 1.0 | 09/12/2019 | First Delivery |

Table 1: Revision History

## 1.5 Reference Documents

- Mandatory Project Assignment
- RASD Document of *SafeStreets* application

## 1.6 Document Structure

The other sections of the Design Document (DD) are organised in this way:

- **Architectural Design** (Section 2): an in-depth description of the System's architecture. It defines the main components, the relationship between them and the deployment of components. There are different views and levels of analysis of the components plus some subsection useful for identifying how the components interact and the architectural styles and patterns.
- **User Interface Design** (Section 3): a complementary section of what was included in the RASD. It includes the definition of the UX process through a model that represents the flows of the interfaces.
- **Requirements Traceability** (Section 4): a complementary section of what was included in the RASD. It contains all the identified requirements and show the relationship between them and design choices in order to satisfy them.
- **Implementation, Integration and Test Plan** (Section 5): shows the order of the implementation and integration of all the components and subcomponents, providing how the application will be tested.
- **Effort Spent** (Section 6): a section containing a table for identifying the hours and the effort spent by each team member to deliver the DD.

# 2 Architectural Design

## 2.1 Overview

## 2.2 Component view

## 2.3 Deployment view

## 2.4 Runtime view

## 2.5 Selected architectural styles and patterns

## 2.6 Other design decisions

# 3 User Interface Design

# 4 Requirements Traceability

| Component (DD) | Requirements (RASD) |
|---|---|
| Component Name | • |
| Component Name 2 | • |

Table 2: Requirements Traceability

# 5 Implementation, Integration and Test Plan

## 5.1 Overview

## 5.2 Implementation

# 6 Effort Spent