



POLITECNICO
MILANO 1863

A.A. 2019/2020
Computer Science and Engineering
Software Engineering 2

SafeStreets

DD
Design Document

Bonatti Andrea Buttironi Monica
Caruso Marco Giuseppe

09/12/2019

Version: 1.0

Contents

| | | |
|----------|--|----------|
| 1 | Introduction | 2 |
| 1.1 | Purpose | 2 |
| 1.2 | Scope | 2 |
| 1.3 | Definitions, Acronyms, Abbreviations | 3 |
| 1.3.1 | Definitions | 3 |
| 1.3.2 | Acronyms | 3 |
| 1.3.3 | Abbreviations | 4 |
| 1.4 | Revision history | 4 |
| 1.5 | Reference documents | 4 |
| 1.6 | Document structure | 4 |
| 2 | Architectural description | 6 |
| 2.1 | Overview | 6 |

1. Introduction

1.1 Purpose

This document is the Design Document (DD) of the SafeStreets system. Its purpose is to completely describe the logical and physical architecture of the system.

It's written for project managers, developers, testers and Quality Assurance people.

This document describes the components forming part of the system with the related run-time processes.

All the design choices are listed and motivated. In particular, this document contains:

- High-level architecture
- Main components, their interfaces and deployment
- Runtime behavior
- Chosen design patterns
- Details about the user interface and its flow
- Mapping of the requirements on the architecture's components
- Implementation, integration and testing plan

1.2 Scope

SafeStreets is a crowd-sourced application that intends to provide users with the possibility to notify authorities when traffic violations occur, and in particular parking violations.

The system addresses to:

- **Generic people:** Registered and Unregistered User
- **Authorities:** Local Officer and Municipal Employee

An unregistered user must be able to register to the service, becoming a registered user.

A registered user must be able to send reports about parking violation (which include date, time, position, type of violations and a picture of the vehicle), to see his/her past reports and to see valid violations occurred in a certain area (unsafe area).

An authority must be able to mine reports concerning his/her municipality and retrieve statistics about reports and issued tickets. In particular, a municipal employee must also be able to get informations about possible improvements to be realized on the municipal territory (starting from reports and accidents occurred) and to eventually set them or some of them as "done", while a local officer must be able also to validate reports.

SafeStreets data about valid reports are shared with the municipal ticket service in order to issue tickets, so the system must guarantee that the chain of custody of the information coming from the users is never broken.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- **Report:** Collection of data that represents a violation, in particular
 - Picture: a photo of the vehicle that has been found committing a violation
 - Date: the date when the picture has been taken
 - Time: the hour when the picture has been taken
 - Position: the place, formatted using GPS location, of the vehicle that has been found committing a violation.
- **Improvement:** a possible road intervention finalized to the development of the road and to achieve a safer environment
- **Token:** string provided to the client as an answer to the login request, which contains information about the user and will always be part of the following requests

1.3.2 Acronyms

- UU = Unregistered User;
- RU = Registered User;
- ME = Municipal Employee;
- LO = Local Officer;
- S2B = Software to Be;
- TS = Ticket Service;
- MS = Map Service;

- MAS = Municipal Accident Service;
- OCRS = OCR Service;
- VT = Violation type;
- DMZ = demilitarized zone;
- DBMS = database management system;
- I&T = implementation and testing;
- DMS = degrees minutes seconds;

1.3.3 Abbreviations

- [Rn] : n-th functional requirement

1.4 Revision history

- Version 1.0
 - Initial release

1.5 Reference documents

- *Specification document*: "Mandatory Project Assignment AY 2019/2020"

1.6 Document structure

The DD document is composed by seven chapters:

Chapter 1 is an introduction and describes the purpose of the document and defines the scope of the system. It also contains the description of definitions, acronyms and abbreviations used inside the document.

Chapter 2 describes the architectural design of the system. It's the core section of the document and contains a description of the high-level components and their interactions, the deployment of the system along with some dynamical views of the components, the selected design patterns and architectural styles and other design decisions.

Chapter 3 describes a UX model for the user interface (presented in RASD) for normal users and authorities

Chapter 4 provides the requirements traceability, specifying how the requirements identified in the RASD are linked to the design elements (components and architectural decisions) defined in this document

Chapter 5 includes the description of the implementation plan, the integration plan and the testing plan, specifying how all these phases are thought to be executed

Chapter 6 shows the effort which each member of the group has spent working on the project

Chapter 7 contains eventual references used during the writing of the document

2. Architectural description

2.1 Overview

The product is a distributed application based on the three logic layers of

Presentation manages the user's interaction with the system

Application handles the logic of the system

Data manages the information.

Those three layers are divided onto four different physical tiers. As shown by the following image, Presentation and Data levels reside on a single tier, while Application level is split into two tiers. The first one is the Web Server, responsible for the forwarding of requests from the clients and for the caching of static contents, while the second one is the Application Server, which contains the logic of the system.