

AA 2019/2020

Computer Science and Engineering

Software Engineering 2 Project

SafeStreets

**DD**

Design Document

version 1.0 --- 24/11/2019

|  |  |
| --- | --- |
| Authors: | Professor: |
| Aida Gasanova | Elisabetta Di Nitto |
| Alexandre Batistella Bellas |  |
| Ekaterina Efremova |  |

**Table of Contents**

[1 Introduction 3](#_Toc26111469)

[1.1 Purpose 3](#_Toc26111470)

[1.2 Scope 3](#_Toc26111471)

[1.3 Definitions, acronyms and abbreviations 4](#_Toc26111472)

[1.3.1 Definitions 4](#_Toc26111473)

[1.3.2 Acronyms 4](#_Toc26111474)

[1.3.3 Abbreviations 5](#_Toc26111475)

[1.4 Revision history 5](#_Toc26111476)

[1.5 Reference documents 5](#_Toc26111477)

[1.6 Document structure 5](#_Toc26111478)

[2 Architectural design 7](#_Toc26111479)

[3 User interface design 8](#_Toc26111480)

[4 Requirements traceability 9](#_Toc26111481)

[5 Implementation, integration and test plan 10](#_Toc26111482)

[6 Appendices 11](#_Toc26111483)

[6.1 Used tools 11](#_Toc26111484)

[﷟6.2 Hours of effort spent](#_Toc26111485) 11

1. Introduction
   1. Purpose

This document represents the Design Document (DD). The main purpose of this document is to fully describe the technical details of the SS application.

In comparison with the RASD, while it has the purpose to describe the functionalities, the requirements and the goals of the system, the DD specifies the design aspects of the system, defining the architecture, the main components and your interfaces, the runtime behavior, the design patterns, and the information about implementation, integration and testing. With these aspects, this document gives a wide knowledge about the description of the components forming the system to be developed.

* 1. Scope

Reviewing from the RASD, the SS service is a crowd-sourced application offered to common users and authorities that want to follow the violations occurred on the municipality territory.



**Figure 1.** SS as a sharing information system

The system has various directions of action. First, the largest mass of users are ordinary people. Everyone can download the application to their phone and use it to the benefit of the welfare of their city. authorities

Every time when the citizens see the violation, they can take a photo and upload it to this application. Based on this information, a register of violations will be compiled for mapping threats on the streets of the city.

When the app receives a picture, it runs an algorithm to read the license plate, and stores the retrieved information with the violation, including also the type of the violation and the name of the street where the violation occurred. In addition, the application allows both end users and authorities to mine the information that has been received, by highlighting the streets (or the areas) with the highest frequency of violations, and the vehicles that commit the most violations. In this case there are more user's levels lake the municipality and authorities, and different levels of visibility are offered to different roles.

Another, also a very important part of users is the municipality. Its role is divided into two parts. First, the municipality can upload accident information to the application, thereby complementing existing databases. The application mixes information from users and from the municipality. As a result, we get a new map with more relevant information about the city. Secondly, the municipality can receive information about offenses from the application. This application initially checks the accuracy of the information (including date, Photoshop using, etc.). Based on this information, the municipality, in cooperation with the police, may issue fines.

Police could also offer a service that takes the information about the violations coming from SS and generates traffic tickets from it. In this case, mechanisms should be put in place to ensure that the chain of custody of the information coming from the users it never broken, and the information is never altered.

* 1. Definitions, acronyms and abbreviations
     1. Definitions
* **User:** the “normal” customer of the application that send the information about the violations to authorities or extract the information that have been received (to use it for useful purposes);
* **Authorities:** the customer of the application that receive the information about violations that have been received from “normal” customers;
* **Customer:** general SS customer;
* **Municipality:** a town or district that has local government;
* **Violation:** general traffic violation, and in particular parking violation;
  + 1. Acronyms
* **API:** Application Programming Interface
* **GPS:** Global Positioning System
* **HTTP:** HyperText Transfer Protocol
* **JSON:** JavaScript Object Notation
* **SQL:** Structured Query Language
* **UI:** User Interface
* **AI:** Artificial Intelligence
* **DD:** Design Document
* **RASD:** Requirements Analysis and Specifications Document
* **SS:** SafeStreets
  + 1. Abbreviations
* **Rn:** n-th functional requirement
  1. Revision history
* 1.0.0 – Release version
  1. Reference documents
* Specification document: “Mandatory Project Assignment AY 2019-2020”
* IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications
* UML diagrams: <https://www.uml-diagrams.org/>
* Alloy doc: <http://alloy.lcs.mit.edu/alloy/documentation/quickguide/seq.html>
  1. Document structure

The DD is composed by five chapters, as outlined below.

**Chapter 1** is an introduction to the design document. Its goal is to explain the purpose of the document and to highlight the differences with the RASD, whilst showing the link between them.

**Chapter 2** aims to provide a description of the architecture design of the system, it is the core section of the document. More precisely, this section is divided in the following parts:

* + - Overview: High-level components and their interaction
    - Component view
    - Deployment view
    - Runtime view
    - Component interfaces
    - Selected architectural styles and patterns
    - Other design decisions

**Chapter 3** specifies the user interface design. Actually, this part is already contained in the RASD in the mockups’ section, but it is reviewed in this document.

**Chapter 4** provides the requirements traceability, namely how the requirements identified in the RASD are linked to the design elements 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 spent working on the project.

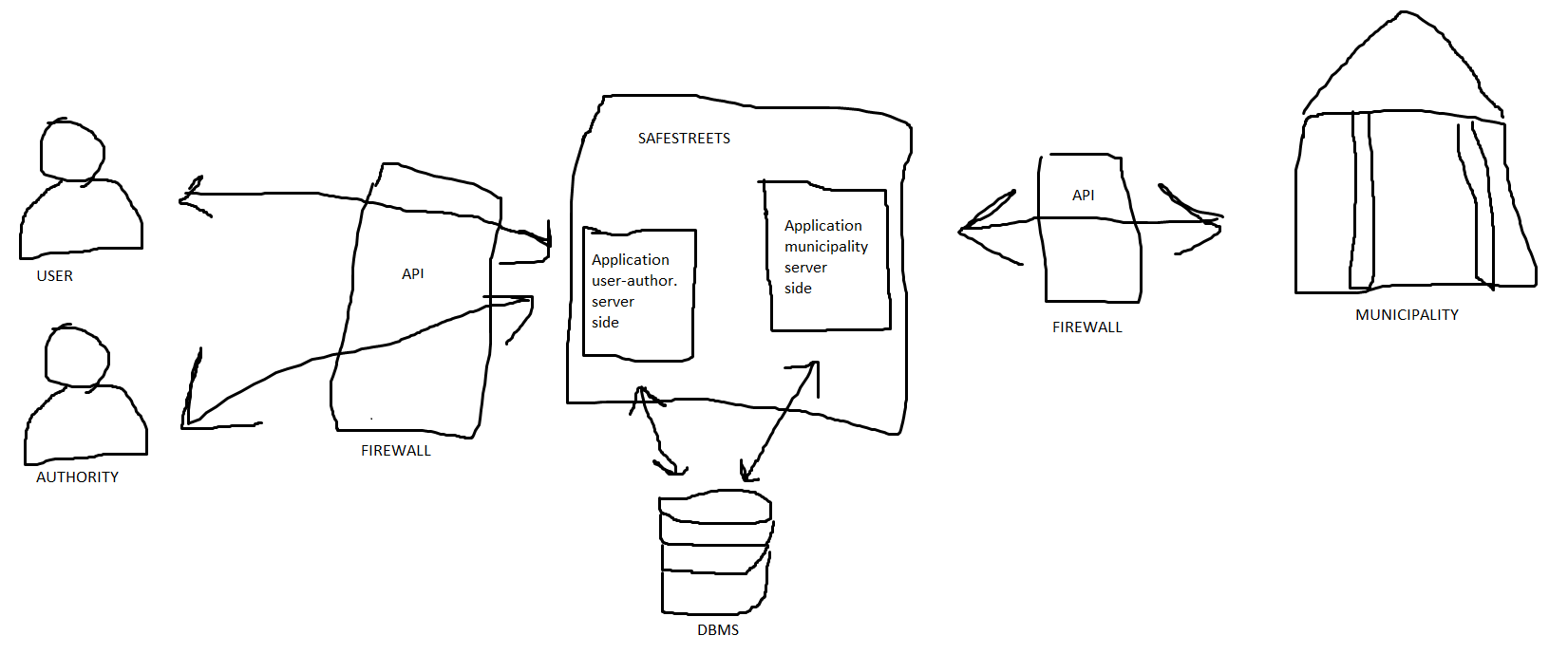
1. Architectural design
   1. Overview: High-level components and their interaction

The SS application will be based on the three-tier client server architecture, in such a way that will be separated in

* (C) Costumers layer, that handles the users and authority interaction with the system;
* (A) Application layer, that handles the business logic of the application and its functionalities;
* (D) Database layer, that contains the database itself and manages the access to it.

These separations are made aiming scalability and flexibility for the application, since that with the application layer, data can be easier processed with security and assertiveness, even more when dealing with sensible data. Besides this, the layer has one node dealing with the requests from users and authorities, and another node dealing with the interaction with the municipality’s system. That’s why there’s no specification of layer about municipality: it is not included on the project development. By now, it is supposed to be already implemented by another development team and will only be used as a ready part of the complete project.

The image below shows a high level architecture of the system for simple representation.



**Figure 1** kjhkg

* 1. Component view
  2. Deployment view
  3. Runtime view
  4. Component interfaces
  5. Selected architectural styles and patterns
  6. Other design decisions

1. User interface design
2. Requirements traceability
3. Implementation, integration and test plan
4. Appendices
   1. Used tools

The tools used for the development of this document were those ones listed below.

* Microsoft Office Word Professional Plus 2016
* draw.io
* GitHub
  1. Hours of effort spen

The hours spent by the group are listed below, differentiating for each participant.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Hours spent** | | |
| **Aida Gasanova** | **Alexandre**  **Batistella Bellas** | **Ekaterina Efremova** |
| Introduction |  |  |  |
| Architectural design |  |  |  |
| Requirements traceability |  |  |  |
| Implementation, integration and test plan |  |  |  |