



S

**SAFESTREETS**

Design Document

S

edoardo putti

9 December 2019

Table of Contents

[**1. Introduction 4**](#_Toc25332202)

[**1.1 Purpose 4**](#_Toc25332203)

[**1.2 Scope 4**](#_Toc25332204)

[**1.3 Definitions, Acronyms, Abbreviations 4**](#_Toc25332205)

[**1.3.1 Definitions 4**](#_Toc25332206)

[**1.3.2 Acronyms 4**](#_Toc25332207)

[**1.4 Revision history 4**](#_Toc25332208)

[**1.5 Document Structure 5**](#_Toc25332209)

[**2. Architectural Design 5**](#_Toc25332210)

[**2.1 Overview 5**](#_Toc25332211)

[**2.2 Component view 6**](#_Toc25332212)

[**2.2.1 ER diagram 6**](#_Toc25332213)

[**2.2.2 To check 6**](#_Toc25332214)

[**2.2.3 See what type of managers do you need 6**](#_Toc25332215)

[**2.3 Runtime view 6**](#_Toc25332216)

[**2.3.1 Sequence diagrams see how many you need 6**](#_Toc25332217)

[**2.3.2 A 6**](#_Toc25332218)

[**2.3.3 S 6**](#_Toc25332219)

[**2.4 Component interfaces 6**](#_Toc25332220)

[**2.5 Selected architectural styles and patterns 6**](#_Toc25332221)

[**2.5.1 Client-server 6**](#_Toc25332222)

[**2.5.2 Service-Oriented Architecture (SOA) 6**](#_Toc25332223)

[**2.5.3 Model View-Controller (MVC) 6**](#_Toc25332224)

[**2.6 Other design decisions 6**](#_Toc25332225)

[**2.6.1 Password storage 6**](#_Toc25332226)

[**3. Algorithm Design 6**](#_Toc25332227)

[**3.1 Need to look 6**](#_Toc25332228)

[**3.2 Solution Calculator – How it works 6**](#_Toc25332229)

[**4. User Interface Design 6**](#_Toc25332230)

[**4.1 UX diagrams 6**](#_Toc25332231)

[**4.2 App Mockups 6**](#_Toc25332232)

[**5. Requirements Traceability 7**](#_Toc25332233)

[**6. Implementation, Integration and Test Plan 7**](#_Toc25332234)

[**7. Effort Spent 7**](#_Toc25332235)

[**8. References 7**](#_Toc25332236)

[**8.1 Software and Tools 7**](#_Toc25332237)

[**8.2 Reference Documents 7**](#_Toc25332238)

1. Introduction
2. Purpose

This document is the Design Document for the Safestreets application. Its aim is to provide a description of the system in terms of architectural components. DD contains a description of the Architectural design using component diagrams and sequence diagrams. It shows how each component is built, how it interacts with other components and with the externals actors involved. This document’s aim is also to provide a technical explanation of the behavior of some component using algorithms. It also shows interfaces through graphical screen representation.

1. Scope

Safestreets is an application in which the User can register and send report to the Authorities. The app allows, for the user, visualizing the traffic information, some information regarding relevant incidents or reports and see the status of the reports sent; for the Authority the app allows to see all the reports and also the statistics derived from them. The application contains a notification system and its role is to notify users about the closing of one of their reports

1. Definitions, Acronyms, Abbreviations
2. Definitions

Here is provided a list of definitions of words and expression used in the document

* **Closing notification:** a notification that the S2B sends when an Authority sets the status of a report to close

1. Acronyms

* **S2B:** System to Be
* **API:** Application Programming Interface
* **RASD:** Requirements Analysis and Specification Document

1. Revision history
2. Document Structure
3. **Introduction:** this serves as an introduction to the document to illustrate its purpose, scope, the conventions that will be used and its structure.
4. **Architectural Design:** After providing an overview of the system, in this section are included all the details of the architecture and the related design decision. Starting from the data model arriving to the description of the components and their role. After a static description, a runtime analysis of the interesting components is provided. Finally we describe the main component interfaces.
5. **Algorithm Design:** In this section are described the most interesting algorithms identified in the system, how they works and their context.
6. **User Interfaces Design:** After the technical description of the previous section, here are provided the indications on the User interactions with the app and mockups of the screens related to the main functionalities.
7. **Requirements Traceability:**  This section explains the rationale behind our design decision in terms of mapping between the goal/requirements defined in the RASD and the components illustrated in this component.
8. **Implementation, Integration and Test Plan:** In this last section is provided a plan for the whole development process, giving indications on the general approach, the priorities and the details of the process.
9. **Effort Spent:** here is included a summary of the effort spent.
10. **References:** this is the section in which are included details on the Software and tools used and the references Documents on which the work is based.
11. Architectural Design
12. Overview

Here is provided a high-level representation of client-server interaction and of the submodule of the servers. Moreover is included a brief indication of the deployment. The orchestrator is needed only on the client request: his role it to dispatch the request to the appropriate component based on the type of the request. After that, the component can communicate directly with the client. In the server, the orchestrator and all the other submodules are stales. I thought to use the elastic component architectural pattern for the components. Moreover, the orchestrator can eventually be duplicated using a fixed dimension pool whose size is configurable by the system administrator. The diagram only shows the overall interaction between client and server and the role of the orchestrator; interactions between the submodules are not shown. Further details on the submodules and the overall interactions will be provided in the following sections.

A screenshot of a cell phone

Description automatically generated

1. Component view

This section starts with the presentation of the Entity-Relationship diagram. In some of the diagrams following the ER, is included a fictitious component, App, that will be highlighted in green and serves the purpose of representing both the mobile app and the web app (through the web server), without adding complexity to the diagrams.

1. ER diagram

The following ER diagram represents the conceptual schema of the system database. The violation subclass presented in the schema are only two but could be more.

A picture containing text, map

Description automatically generated

1. To check
2. See what type of managers do you need
3. Runtime view
4. Sequence diagrams see how many you need
5. A
6. S
7. Component interfaces
8. Selected architectural styles and patterns
9. Client-server
10. Service-Oriented Architecture (SOA)
11. Model View-Controller (MVC)
12. Other design decisions
13. Password storage
14. Algorithm Design
15. Need to look
16. Solution Calculator – How it works
17. User Interface Design
18. UX diagrams
19. App Mockups
20. Requirements Traceability
21. Implementation, Integration and Test Plan
22. Effort Spent
23. References
24. Software and Tools
25. Reference Documents