****

**Department of Computer Science and Engineering**

**Design Document (DD)**

**SafeStreets**

- v1.1 -

*Authors:*

|  |  |
| --- | --- |
| **Dlamini,** Taras | 10451348 |
| **Quran,** Abdullah | 10657807 |
| **Abdelaziz,** Hesham | 10654249 |

November 10th, 2019

[**1.Introduction**](#_x0v3ierbp558) **3**

[1.1 Purpose](#_pfjmyw1uifyh) 3

[1.2 Scope](#_t312k77givan) 4

[1.3 Definitions, acronyms, abbreviations](#_9rkswrcbbvv) 4

[1.3.1 Definition](#_h3iuqnsz3xdr) 4

[1.3.2 acronyms](#_q5vxle9kg7zg) 4

[1.3.3 abbreviations](#_z47yg3qo3e7j) 5

[**2. Architectural Design**](#_5le2qvcroiuf) **7**

[2.1 Overview](#_vpcdi9g967e8) 7

[2.2 Component view](#_x6w76hmir873) 7

[2.3 Deployment view](#_kiqmo7hibxbk) 7

[2.4 Runtime view](#_7rkqhwamrzjy) 7

[2.5 Component interface](#_6lj7bc1pkxxp) 7

[2.6 Selected architectural design and patterns](#_2tudg9vab4q3) 7

[2.7 Other design decisions](#_ekr9sbxhuonj) 8

[**3. User Interface Design**](#_n02qbuov8u8e) **8**

[**4. Requirement Traceability**](#_4ughh8y5i653) **10**

[**5. Implementation, Integration and Test Plan**](#_p0lnprj970j1) **11**

[**6. Effort Spent**](#_tk6lsspr7wkm) **12**

# 

# 

# 

# 1.Introduction

## **1.1 Purpose**

The requirements elicitation and analysis activities concerning SafeStreets system were presented in detail in the RASD document (see the References section). After doing so, It is then proper to start exploring the design phase. This document aims at providing the technical details, architectural, and design choices that are, and should be, considered in order for SafeStreets’ system to have the capability to fulfill the goals specified in the RASD, and to be used as a guide for the implementation and testing processes afterwards.

Below is the list of the aspects this document discusses:

* Overview of the high level architecture
* The main components and their interfaces provided one for another
* The runtime behavior
* The design patterns
* Graphical user interface
* Functional requirements mapping into the actual component of the system
* Implementation, integration, and testing plan

## **1.2 Scope**

### 

## **1.3 Definitions, acronyms, abbreviations**

### 1.3.1 Definition

### 1.3.2 acronyms

* API: Application Programming Interface
* DD: Design Document
* RASD: Requirements Analysis and Specifications Document
* GUI: Graphical User Interface
* HTTP: HyperText Transfer Protocol
* TSL: Transport Layer Security
* JSON: JavaScript Object Notation
* SQL: Structured Query Language
* MVC: Model-­‐View-­‐Controller

### **1.3.3 abbreviations**

* [Rn]: n-functional requirement.

### 

**1.4 Revision History**

* 9/12/2019 - Initial release

**1.5 Referenced Document**

* Specification document: “SafeStreets Mandatory Project Assignment AY 2019-­2020”
* IEEE Std 830-­1998 IEEE Recommended Practice for Software Requirements Specifications
* Requirements Analysis and Specifications Document: SafeStreets -V1.2-

**1.6 Document Structure**

Chapter **one** is and introduction of the design document. It describes the document’s purpose and scope. It also includes definitions and abbreviations used in this document.

Chapter **two** represents the core part of the document. It includes an overview of the architectural design of the SafeStreets’ system. It describes the system’s architecture from different paradigms, in particular; component view, class view, deployment view, run time view, and exact interfaces and their actions. Then it includes a justification on the choice of the design architecture and pattern

Chapter **three** complements the user interface section discussed in the RASD section. However, here all the pages of the SafeStreets web and mobile applications are included

Chapter **four** includes mapping between the functional requirements thoroughly discussed in the RASD into the design elements explored in this current document.

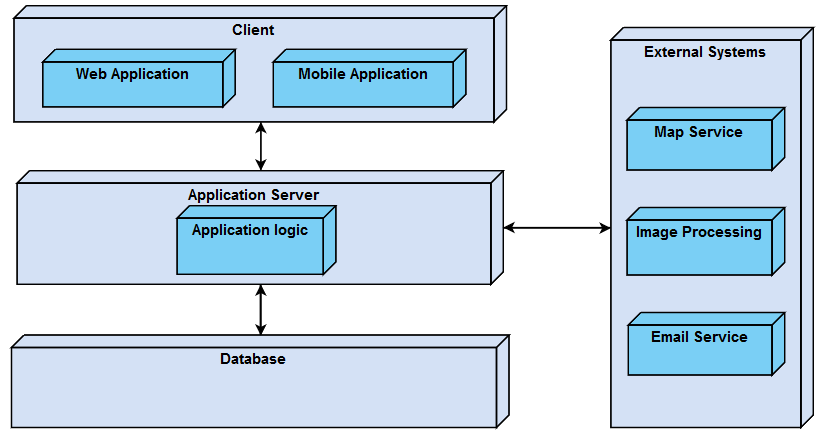
Chapter **five** presents the implementation, integration, and test plan of the system. It describes how these three phases are to be tackled.

Chapter **six** shows the effort each member of the group spent to generate this document.

# 2. Architectural Design

## 2.1 Overview

The architecture pattern to be adopted in SafeStreets system is decided to be 3-tier architecture which is basically client - server architecture consists of 3 physically separated tiers call presentation, application and data tiers. 3-tier architecture was chosen because it provides scalability since the application is thought to be crowdsourced application and flexibility if future development is needed. In addition the physical separation between tiers potentially results in hiding the logic from the users and this potentially advantageous in terms of information security. Below is a brief description of the high level component of each tier. Further elaboration on the system architecture is provided in the following sections of this chapter. Figure 3.1 shows the higher level components of the system and their interactions

****

**Figure 2.1: High level architecture of SafeStreets system**

The **Presentation** (client) tier is comprised of the mobile and web application. It is the tier which users (regular users and third parties) have access to through GUI. We will see that small but important part of the business logic is placed in the mobile application in order to ensure that the chain of custody is never altered (this was defined in the RASD documents in details).

Then comes the **Application** tier that carries most of the business logic the system has to do. The application tier is positioned in the middle between the data tier and the presentation tier and it responsible for performing detailed processing. Also this tier has an interface with the external systems.

The last tier is the **data** (database) tier which a [database](https://en.wikipedia.org/wiki/Database) comprising both data sets and the [database management system](https://en.wikipedia.org/wiki/Database_management_system) software that manages and provides access to the data. It communicates with the application server

## 2.2 Component view

## 2.3 Deployment view

## 2.4 Runtime view

Sequence diagrams but this time for component not users (link it with use cases)

## 2.5 Component interface

## 2.6 Selected architectural design and patterns

Which styles were used, why, how

## 2.7 Other design decisions

## 

# 3. User Interface Design

**UI of regular user of all the pages of as in the implementation**

**UI of Municipality of all the pages of as in the implementation**

# 4. Requirement Traceability

R1: The system must allow a user to register a new account

**User Authetnication**

R2: The system must allow a user to login to his/her account

**User Auth**

R3: The system must allow users to take images and to input details of violation

**Violation Manager**

R4: The system must allow users to upload images and data of violation to its database

**Violation Manager**

R5: The system must be able to apply a security mechanism able of detecting image tampering

**Verification Manager**

R6: The system must be able to store data of violation on the user side

**Database Manager**

R7: The system must be able to communicate with its database

**Database Manager**

R9: The system must be able to store data of violation in its database

**Database Manager**

R11: The system must allow users to send a request for data of violation

**Data Manager**

R12: The system must be able to anonymize data accessed by users

**Data Manager**

R13: The system must be able to send the data of violation to the email the user had registered with.

**Data Manager**

R14: The system must allow active municipalities to filter data by type, time, date, and location of violation

**Data Manager**

R15: The system must allow municipalities to access the data of violation

**Data Manager**

R16: The system must be able to communicate with active municipalities to acquire data of accidents

**Data Manager**

R17: The system must allow users to see a list unsafe areas

**Statistics Manager**

R18: The system must allow users to filter the list of unsafe areas by location, type, time, and frequency

**Statistics Manager**

R19: The system must allow users to enable/disable notification about the area they are currently in.

**NotificationManager**

R19: The system must allow users to send a request to subscribe to notification service

R20: The system must allow users to get safety notifications about the area he is currently in

R22: The system must be able to notify the user if the safety status of an area has changed

**NotificationManager**

R23: The system must allow a municipality to activate its account

**Mun\_Authentication Manager**

R24: The system must allow active municipality to access its account

**Mun\_Authentication Manager**

R25: The system must allow active municipalities to download generated file of violations

**Data Manager**

R26: The system must allow active municipalities to check the suggestions offered by SafeStreets

**Suggestions Manager**

# 5. Implementation, Integration and Test Plan

# 6. Effort Spent

|  |  |  |
| --- | --- | --- |
| **Taras Dlamini** | **Abdullah Quran** | **Hesham Abdelaziz** |
| **Task Hours** | **Task Hours** | **Task Hours** |
|  |  |  |
|  |  |  |