POLITECNICO
MILANO 1863

Computer Science and Engineering

A.A. 2019/2020

Software Engineering 2 Project:

"SAFE-STREET"

Desgin **D**ocument

December 7, 2019

Prof.Rossi Matteo Giovanni

Amirsalar Molaei

karim Zakaria Saloma

Erfan Rahnemoon

Contents

Figures

Tables

1 Introduction

The Software Design Document is a documentation of the intended system design used to convey the expected output of the development phase. In this section, an overview of the content and intended use of the document is discussed.

1.1 Purpose

The Software Design Document is built to describe a detailed description of the *Software To Be* from the architectural and technical aspects. This document specifies the manner in which the software shall be built through the use of narrative and graphical tools to aid in the communication of the necessary information to the concerned audience.

This document is intended to provide a clear and complete description of the system to the persons who shall be developing the system. In order to, assist in the understanding of how the system should be built and how the end product should function in accordance with the previously decided upon requirements and specification of the system.

1.2 Scope

In this section, the scope of the system previously described in the *Requirements and Specification Document* shall be revisited. As well as, consider some more in-depth aspects of the system.

As previously stated in the *Requirements Document*, the *SafeStreets* system shall be providing four main functions to various users; in this section, the system boundaries and scope used to define the limitations and different responsibilities of the S2B.

The first of the main functionalities is the enabling of users to report traffic violations. Regarding this, some phenomena are regarded as world phenomena not viewed by the system due to its limitations such as the fact that the system does not directly detect a violation. However, it can be accounted for by the system through a traffic report made by the users. Moreover, another functionality that has to do with the users is the publishing of collected data to be viewed by the users in a refined representation to help them consider the safety of various areas based on traffic violations. The data is also communicated to the authorities but with different levels of details.

The other two main functions have to do with the *SafeStreets* system providing services to government authorities. The domain limitations of the system affecting this interaction are also discussed in this section. Such as, the fact that the system is only able to make suggestions for preventive measures to the authorities based on the accident data that have been communicated. Meaning, that the system does not have any knowledge of accidents unless they are reported by the authorities and that the system can only suggest interventions and neither put them into place nor can detect them being applied. Moreover, a second function to the authorities would be the communication of traffic reports received from users to be later used by government officials to give out traffic tickets, the system responsibilities to support this process is to prevent the users from tampering with images *digitally* and to provide the collected reports to the authorities proactively. In other words, physical tampering with license plates to mislead authorities and the actual process of giving out tickets is not part of the application domain.

Moreover, in this document, some more technical issues regarding the functioning of the system need to be discussed. Primarily, the security aspect of the system; more specifically, data security. Since the system will be dealing with the collection and communication of sensitive data while performing more than one of the main functions; the system must, at all times ensure the safe transmission and storage of data and the application of measures to prevent any means of data tampering. The data being discussed includes but is not limited to, user personal data and detailed data of traffic reports.

1.3 Definitions,Acronyms,Abbreviations

1.3.1 Acronyms

S2B *Software To Be*

GPS *Global Positioning System*

UI *User Interface*

GDPR *General Data Protection Regulation*

UML *Unified Modeling Language*

RASD *Requirements Analysis and Specification Document*

OS *Operating System*

SMS *Short Message Service*

1.4 Revision history

Revision	Date	Author(s)	Description
0.0	24/11/2019	karim Zakaria Saloma, Amirsalar Molaei, Erfan Rahnemoon	First document issue

1.5 Reference Documents

References

[1] *Standard for Information Technology–Systems Design–Software Design Descriptions IEEE 1016.* 2009.

[2] UML,
`https:omg.org/spec/UML`

[3] Specification Document. *SafeStreets Mandatory Project Assignment.*

1.6 Document Structure

The Software Design Document(**DD**) is comprised of six main chapters. Which shall be described in this section of the document:

Chapter 1 (Introduction): provides an overview of the document as a whole; describing, the various sections constituting this document, as well as, the intended use of this document.

Chapter 2 (Architectural Design): a detailed description of the architecture to be developed during the implementation phase of the system; spanning from a high-level component view to a detailed run-time description of the different modules of the system. This is used as a guideline for the development team in order to have a clear idea of how the system should be built.

Chapter 3 (User Interface Design): an overview of the design of the different interfaces that the users of the system shall be interacting with the system through; in order to utilize the functionalities of the system according to their needs. This overview is concerned with the visual aspects of the user interfaces.

Chapter 4 (Requirements Traceability): provides a link between the design decisions in this document and the requirements of the system described in the *Requirements and Specification Document*. This is done by providing an explanation of how the system design described in this document fully satisfies the requirements the system must abide by.

Chapter 5 (Implementation, Integration and Test Plan): describes the approach to be followed during the development and testing phase of the system. This is also provided as a clear guideline for the development team to follow.

Chapter 6 (Effort Spent): summarizes the efforts of the team members in developing this document in terms of time spent on each of the sections of the document.

2 Architectural Design

2.1 Overview

2.2 Component View

2.3 Deployment View

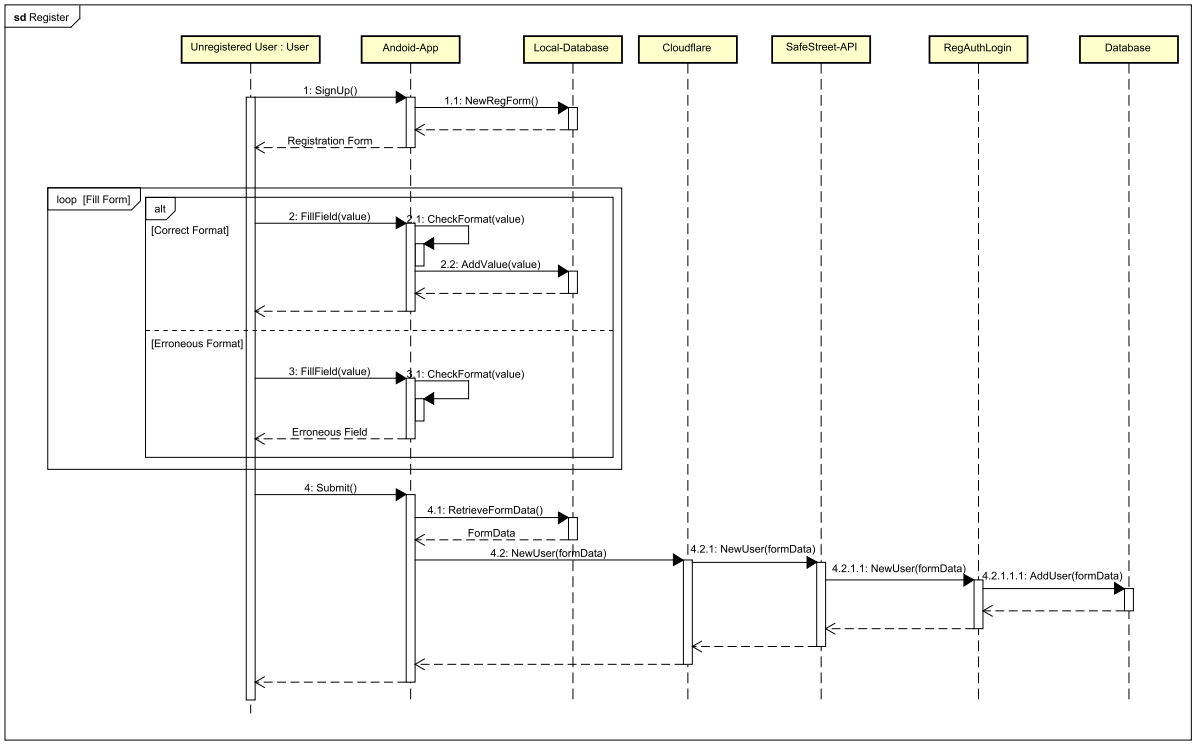
2.4 Runtime View

This section of the document is concerned with the dynamic interactions between the various components of the system during runtime to achieve the desired functions of the system. The section is constituted of four major subsections grouping together various runtime views of different parts of the system. These subsections are as follows *general functions, violation reports, view safety* and *municipality interaction.*

2.4.1 General Functions

This first subsection considers the runtime behavior of system components for the delivery of basic general functions. Specifically, *registration and verification, login* and *profile viewing and info editing.* The first two runtime diagrams below describe the user registration and verification process. As can be seen in the diagram a new user fills in the registration form with his information the submits the form. The data is relayed to the *Safestreets-API* which saves the user data and prompts the *Firebase-Authentication* server to verify the new user. Which in turn, sends the verification message to the user and returns feedback of the user confirmation.

Figure 2: Registration Runtime View



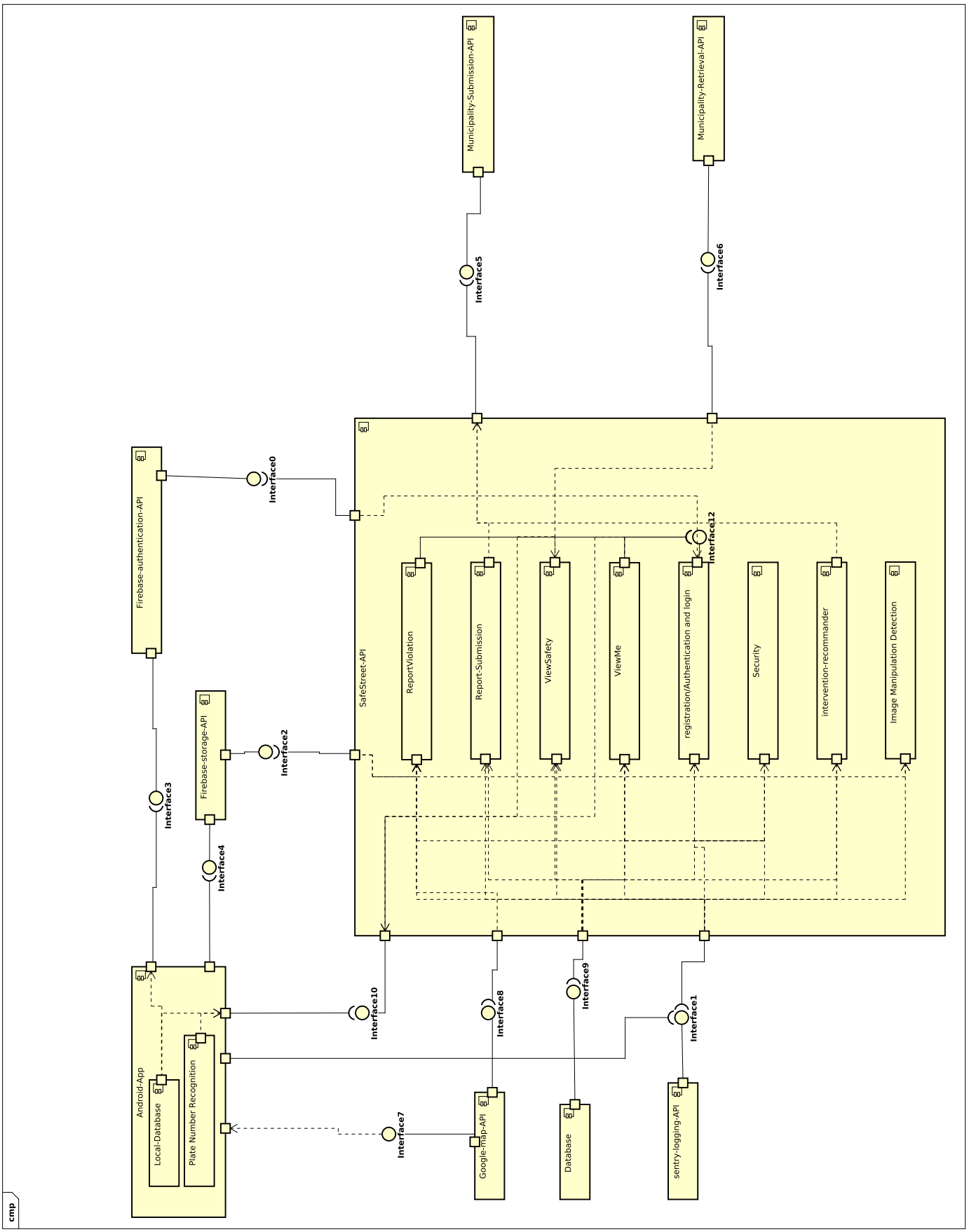
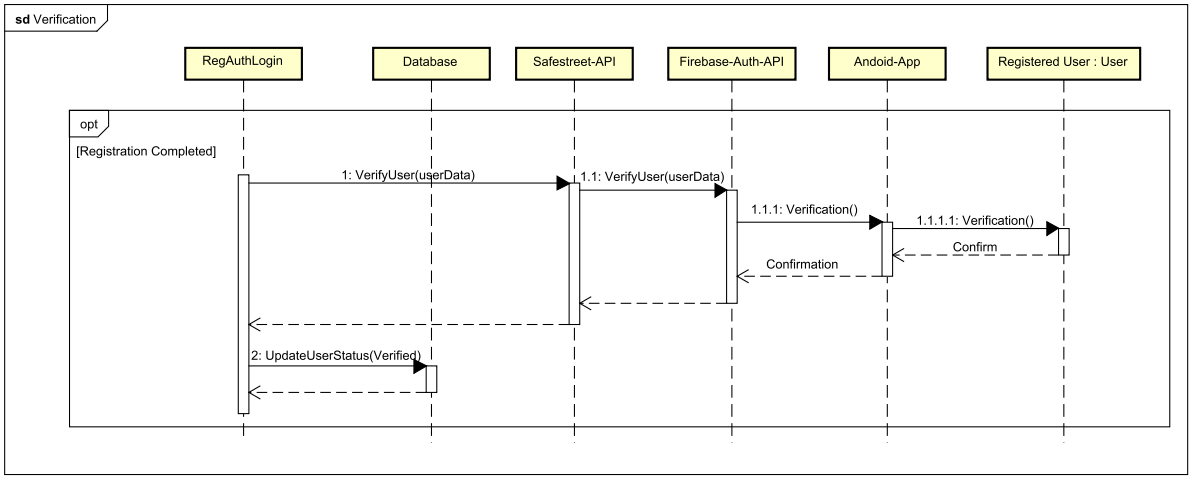


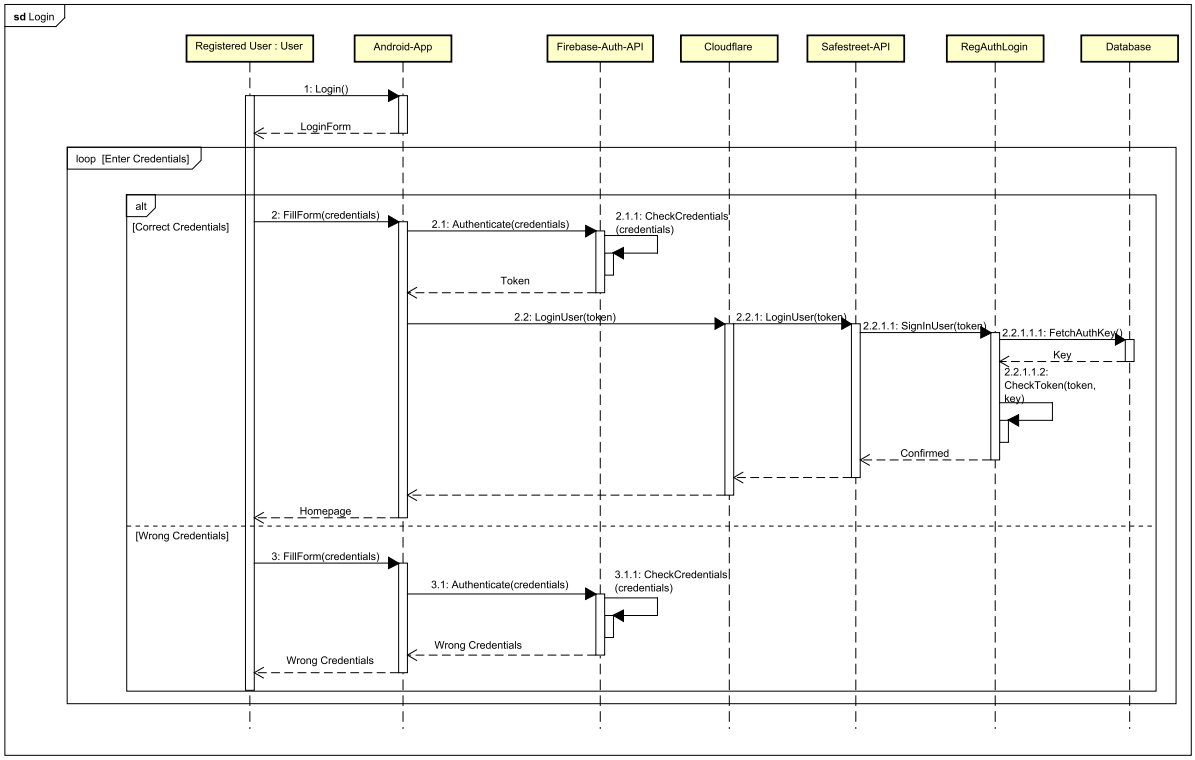
Figure 1: Entity Class Diagram

Figure 3: Verification Runtime View



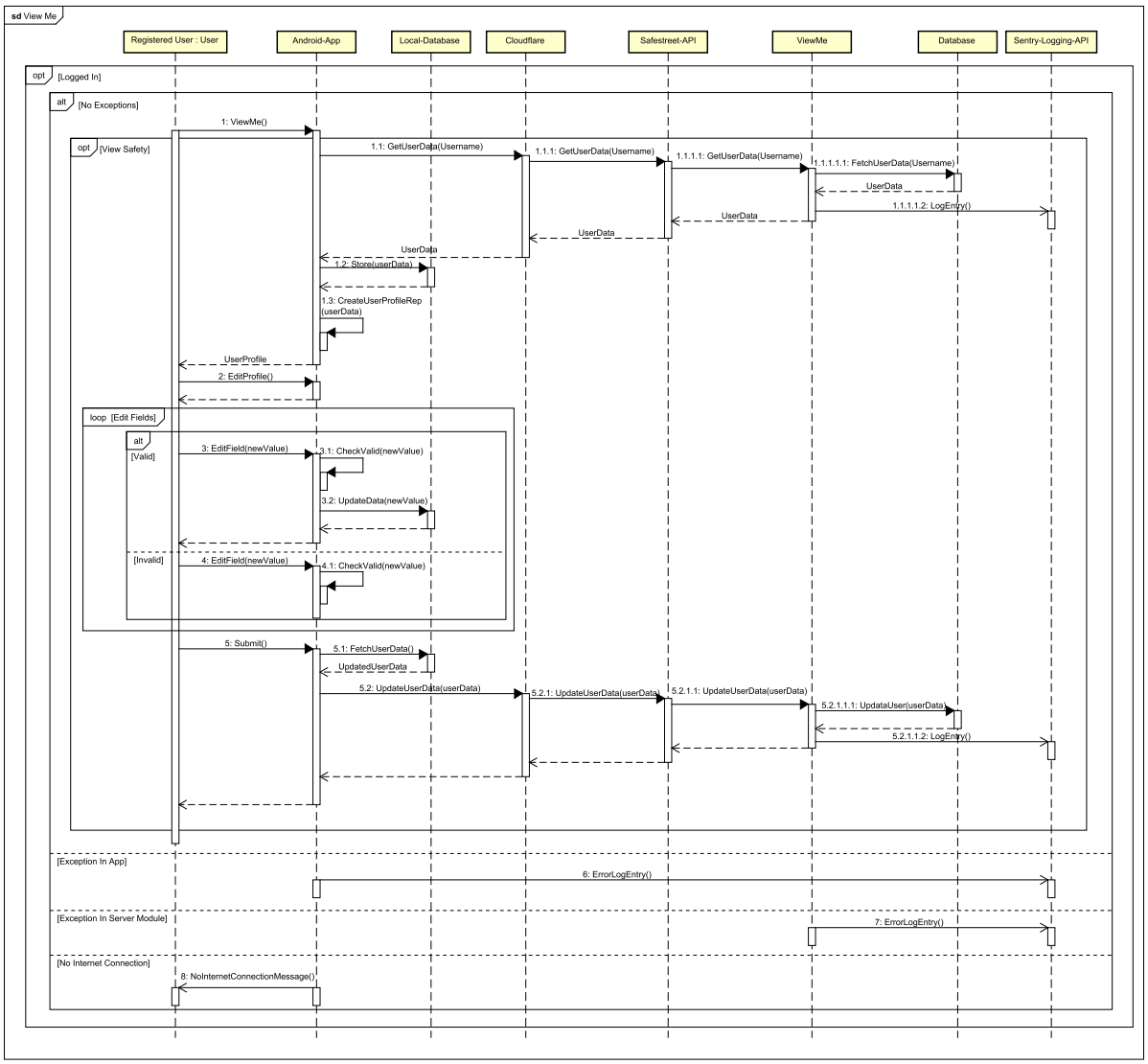
The below figure describes the user login process. Which again, relies on the *Firebase-Authentication* server to authenticate the user credentials and issuing a token to be sent to the app server to confirm the authentication process.

Figure 4: Login Runtime View



The last of the general functions in the figure below is the *ViewMe* functionality. Where the user can view their profile and edit their personal information.

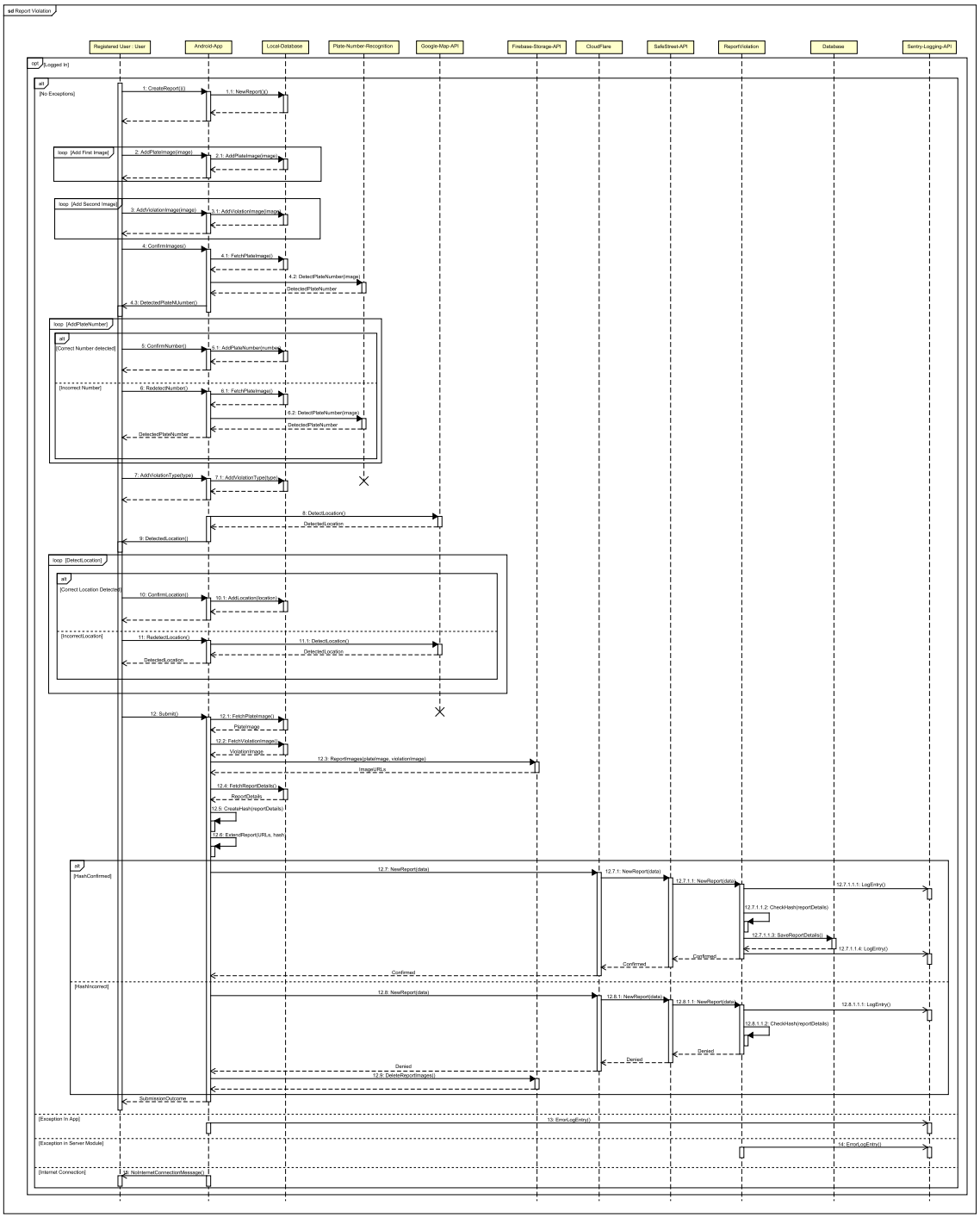
Figure 5: ViewMe Runtime View



2.4.2 Violation Reports

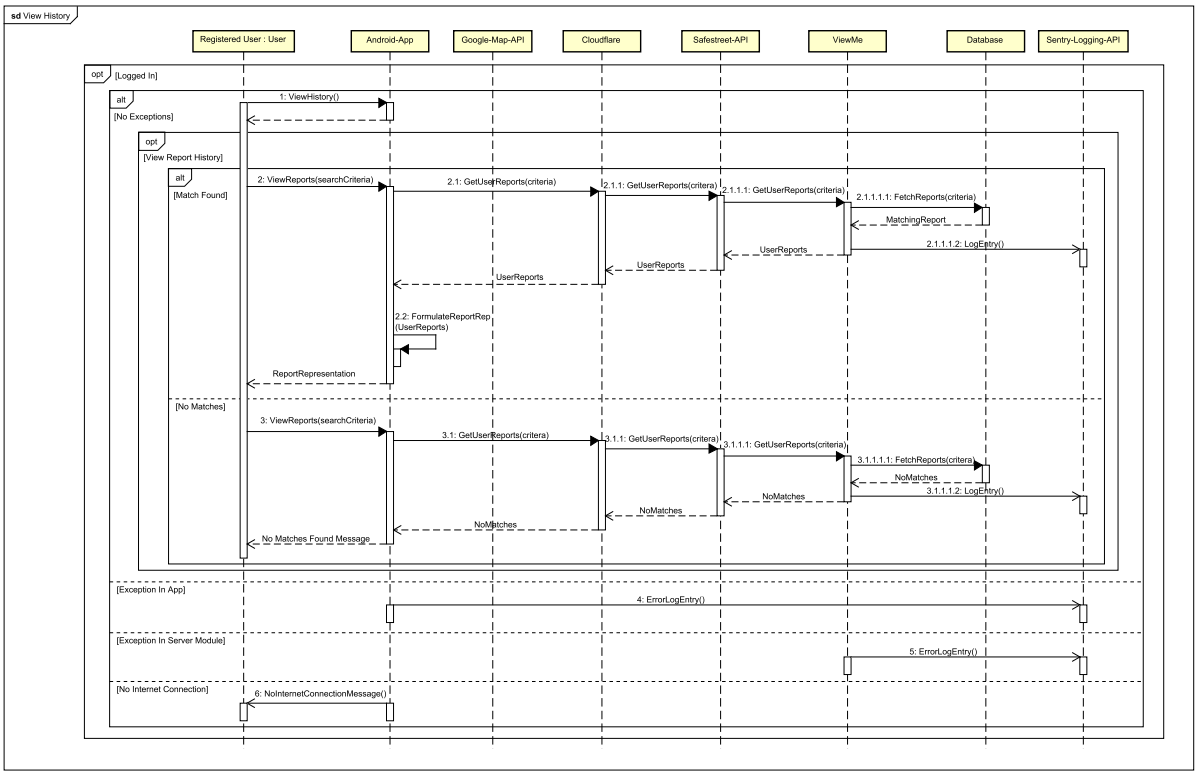
The process of *Violation Report* submission performed by the users is constituted of the user report formulation including images showing the violation, the location, and other useful information. The images are saved on the *Firebase-Storage* and their respective URLs are used to access them for any future use. Finally, the report details, image URLs and a computed hash which is verified to ensure data integrity are sent to the server. A detailed breakdown of the step-by-step process of violation report submission is described in the following figure.

Figure 6: Report Violation Runtime View



The report history function described in the following figure is a function offered to the user to search through reports that were previously submitted by them according to criteria that they define. Such as, reports that were submitted in a certain time period or a certain type of violation reports.

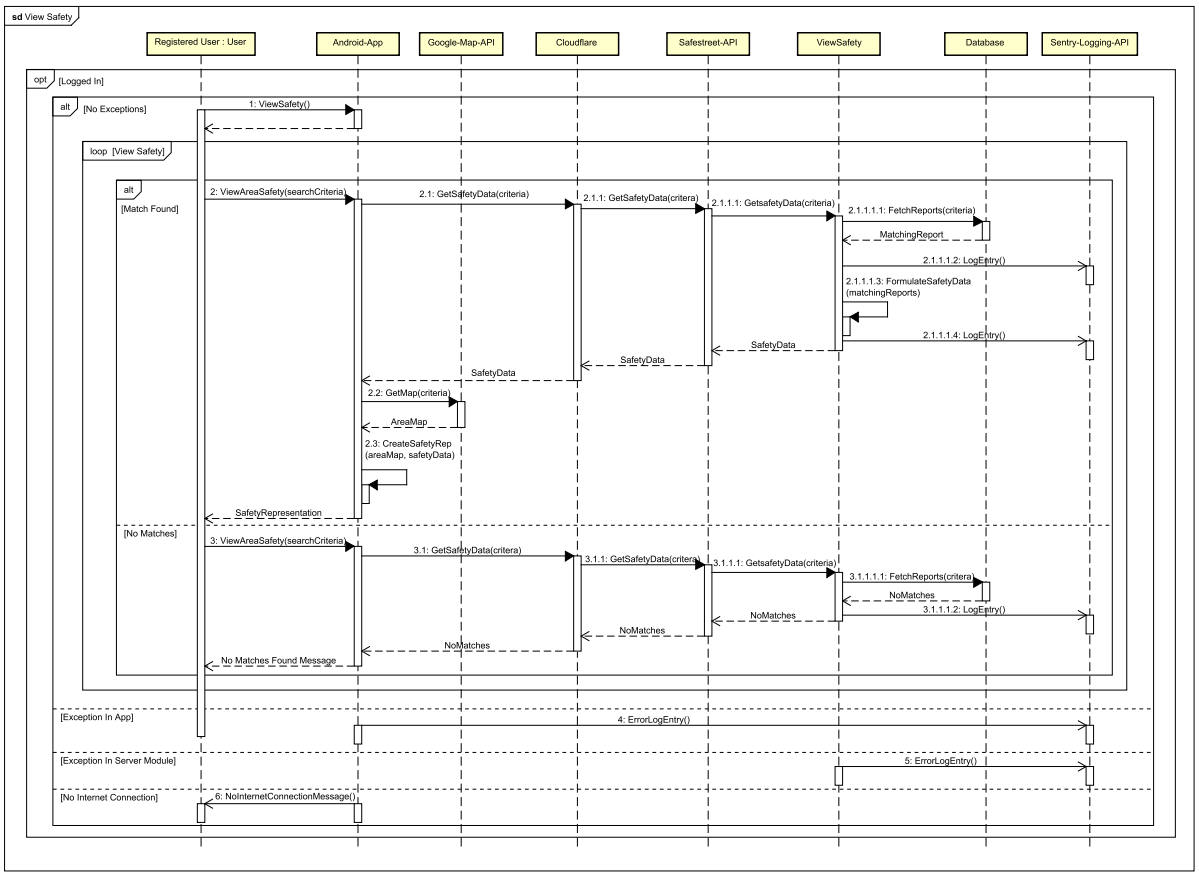
Figure 7: Report History Runtime View



2.4.3 View Safety

This subsection discusses one main feature of the *Safestreets* that is the *View Safety*. This feature entails the graphical representation of the safety of the various areas according to the user-specified search criteria. Where the user would search for a certain area and the app would retrieve report data of both violation reports filed by the users and accident reports retrieved from the municipality and formulate a geographical representation using the *Google Map API* as can be seen in the figure below.

Figure 8: View Safety Runtime View



2.4.4 Municipality Interaction

This last subsection is concerned with the various interaction between the *Safestreet* system and the municipality. To be precise, there are kinds of interactions between the system and the municipality; which are as follows, firstly, the communication of user-generated violation reports to the authorities through periodic submissions to the municipality submission API. Second of the three features, the retrieval of accident reports from the municipality retrieval API to be used in the *View Safety* feature and in the intervention suggestion which is the last feature described in this subsection. In the intervention suggestion process, the user-generated reports and the retrieved accident reports are processed together to gain insight into the traffic violations which are causing a high accident rate in various areas and suggest appropriate measures to resolve these issues. The following figures provide a more detailed description of the aforementioned processes.

Figure 9: Municipality Report Submission Runtime View

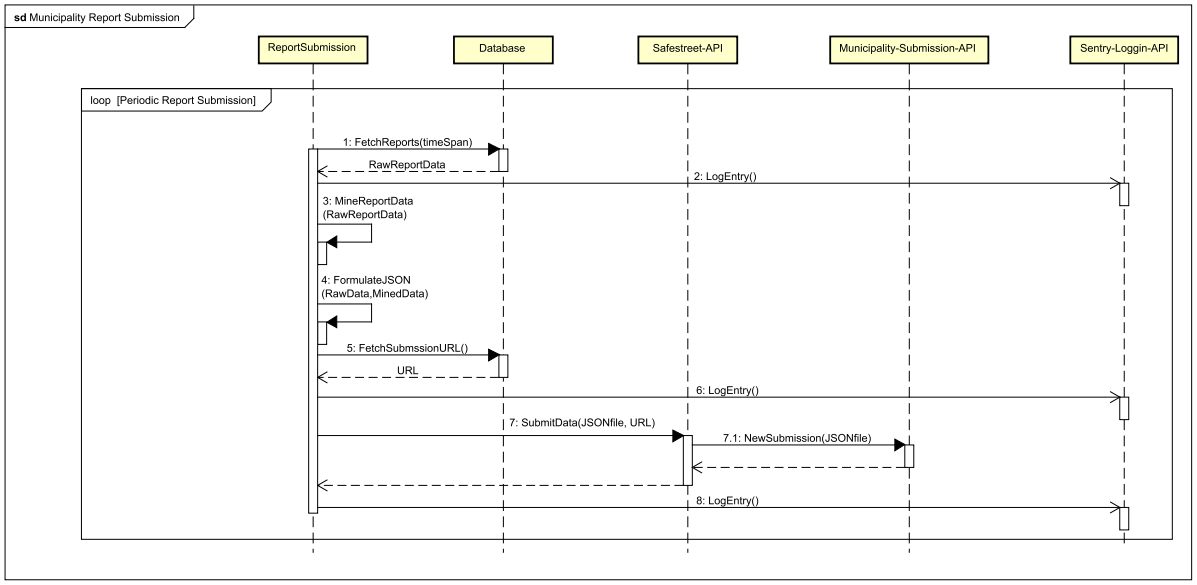


Figure 10: Accident Report Retrieval Runtime View

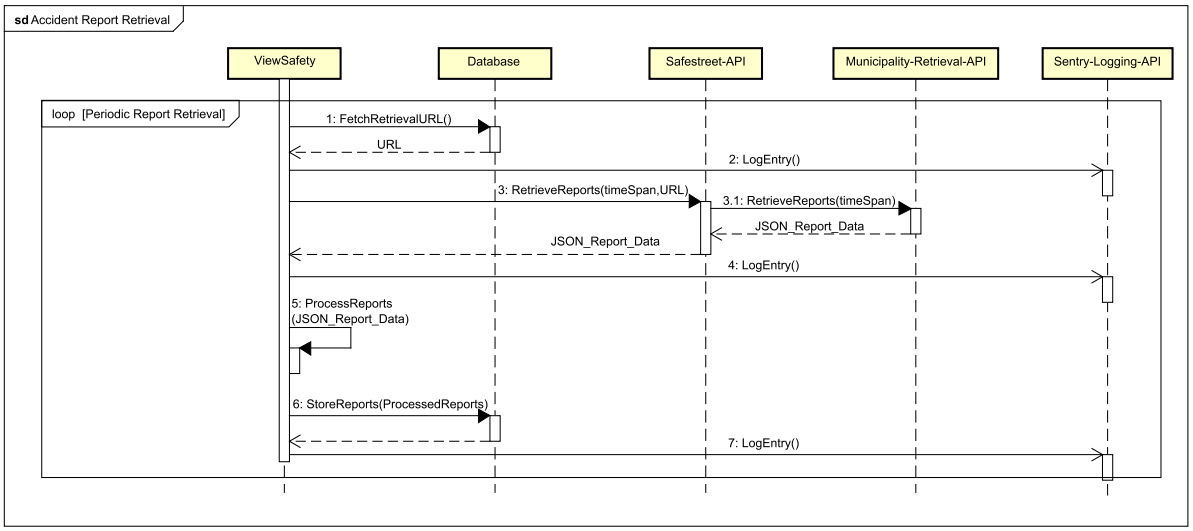
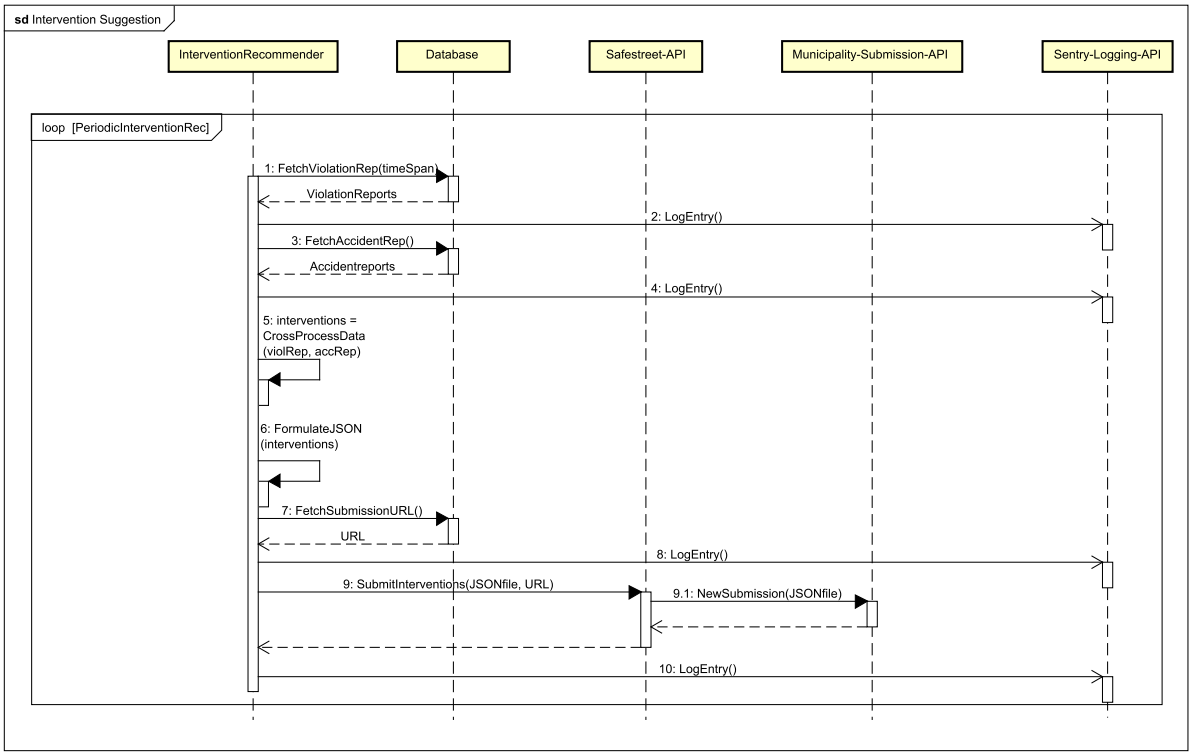


Figure 11: Intervention Suggestion Runtime View



2.5 Selected Architectural Styles And Patterns

3 User Interface Design

Figure 12: Entity Class Diagram

4 Requirements Traceability

5 Implementation Integration And TestPlan