Comprehensive Design and Analysis Project

(IT 405)

Location based crisis analysis and day planning with artificial intelligence and IOT for smart city

Project ID   : 19-045

Supervisor: Hansika Mahaadikara

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to be describe about the functional and nonfunctional requirements as well as the to provide a detailed description of all the requirements and process features of "**Location based crisis analysis and day planning with artificial intelligence and IOT for smart city**" application. This SRS Document the Scope, Process, Constrains, Main features of the system, Mockup interfaces and other necessary information are detailed.

This document will be the main document for the QA Engineers for their tests and for the Developers for their further implementations and to solve problems regarding the system.

1.2 Scope

A large number of people moving towards urban cities, by 2030 more than 60% of the population will live in urban environment. With this accelerated rise of the urban population, cities throughout the world are facing many risks and concerns. This situation has created an urgency for finding smarter ways to manage the challenges.

Introducing our app is a great step towards a better and faster smarter city. Because with our app people will be able to travel here and there without any trouble because they will be able to plan ahead. It will cause people to be more efficient and time saving in the busy world

1.3 Definitions, Acronyms, and Abbreviations

1.3.1 Definitions

Terms	Definition
Location based crisis analysis and day planning with artificial intelligence and IOT for smart city	Name of the proposed system.
Software Requirement Specification	A document that completely describes all of the functions of a proposed application and the constraints under which it will operate.
Database	A set of data or information retrieved and stored by the system.
SRS	A fully descriptive document of the software system to be developed which contains all the functional and non-functional requirements.

1.3.2 Abbreviations

Acronym	Definition.
App	Application
SRS	Software Requirement Specification.
DB	Database
GUI	Graphical User Interface.
UI	User Interface.

1.4 Overview

Nowadays people are so busy with their day-to-day work. They have a very limited time gap to complete all the responsibilities. So, people will use anything that they can to maximize their efficiency and save their time to do daily tasks. In the present people tend to use technology to get things done than manually. Even the older technology is getting thrown away with the newer ones.

Here people can use the app to update disasters, crimes and crisis just by uploading a picture under the correct department that we have listed in the app. Also, people can plan ahead by using day planning with the weather and traffic analysis. Knowing these things in the busy world this will be much more helpful.

The proposed system is basically based on using Internet Of Things (IOT) and so that it leads to use technology on fulfilling analysis effectively and efficiently.

2. Overall Descriptions

This Document is mainly focused on the System requirements and the user requirements of the app as there are more researches to be done. A detailed Description of the mobile app will be provided in the Literature review.

2.1 Product perspective

Although accidents, crimes and disasters are a common occurrence in roads and urban areas, a quick report submission, identification and notification capabilities are lacking in most of the existing road-based solutions as elaborated in the literature. The proposed Smart City Application is a solution which enhance the user experience.

Even though there are existing proposed applications available, they do not address most of the problems that the proposed system is going to address. Proposed mobile application consists of many features as a solution for the main issues people faced in day to day life due to traffic problems.

Features	Google Map	Culture and new digital technologies transforming world cities	Proposed App	Pick Me
Analyze the time taken for reach the destination	✓	✓	✓	✓
When we submit the time to reach the destination show the start time	✗	✗	✓	✗
When we start travelling show us the weather report on the passing areas	✗	✗	✓	✗
Identifying the closest department relevant to the report	✗	✗	✓	✗
Shows the closes route according to the weather and traffic data	✗	✗	✓	✗

2.1.1 User Interface
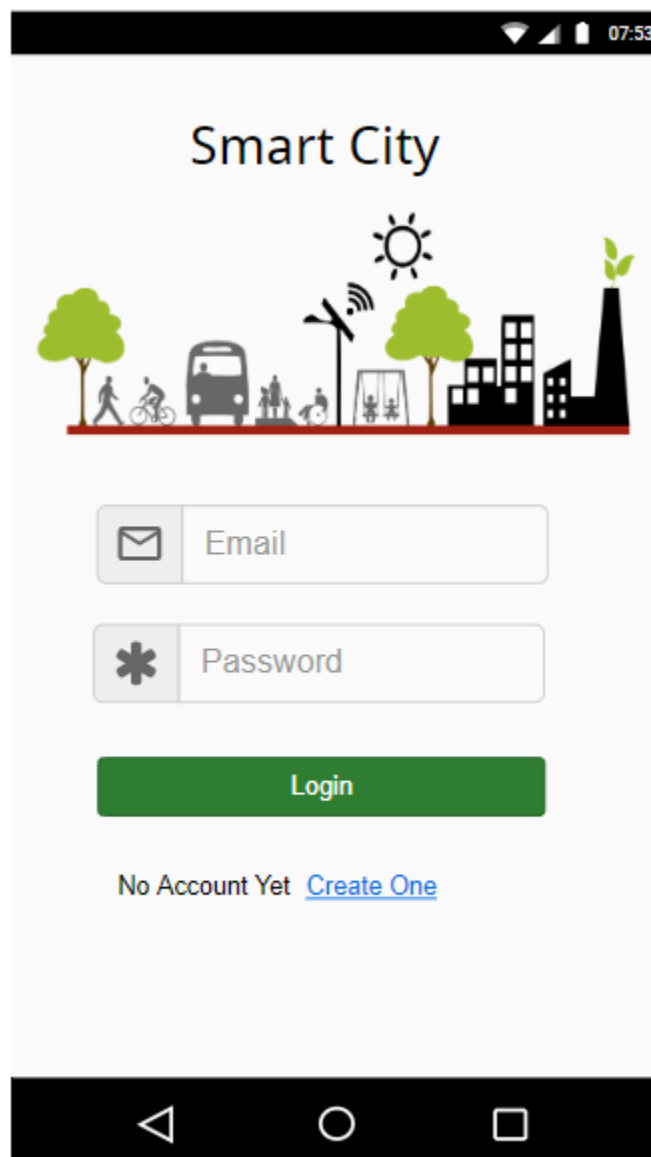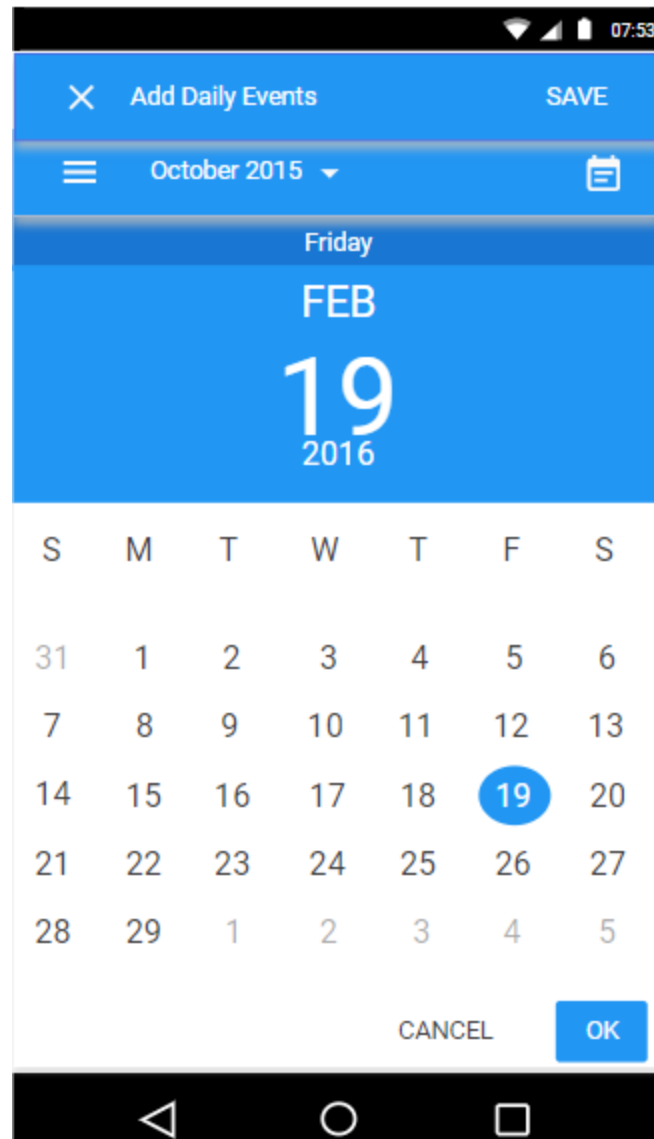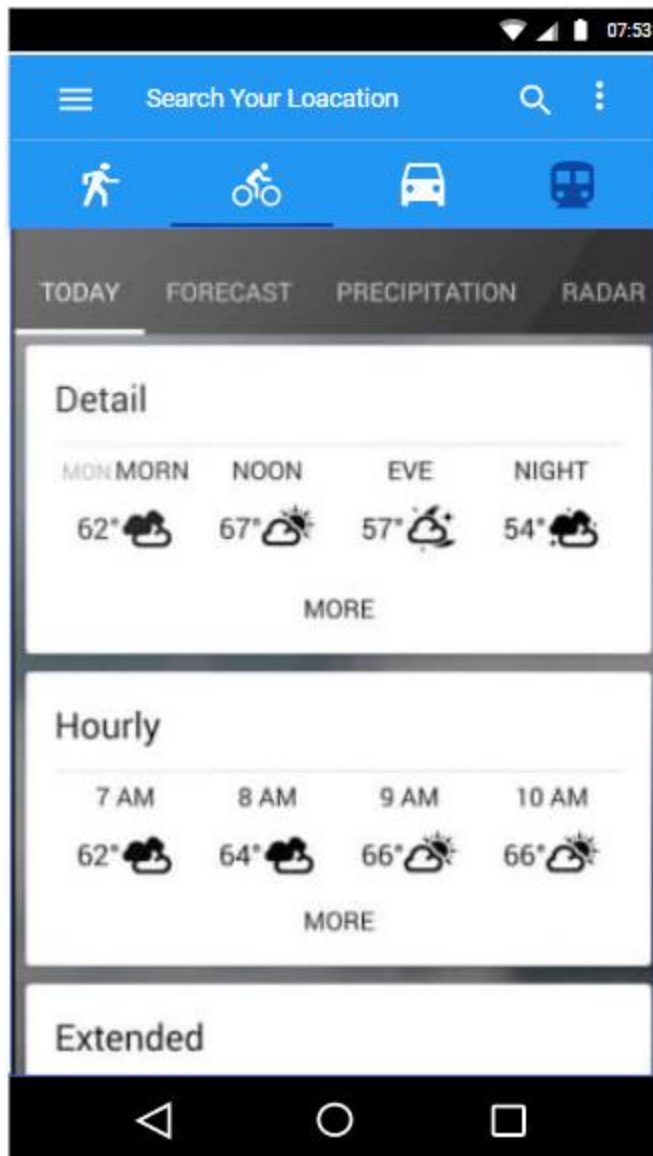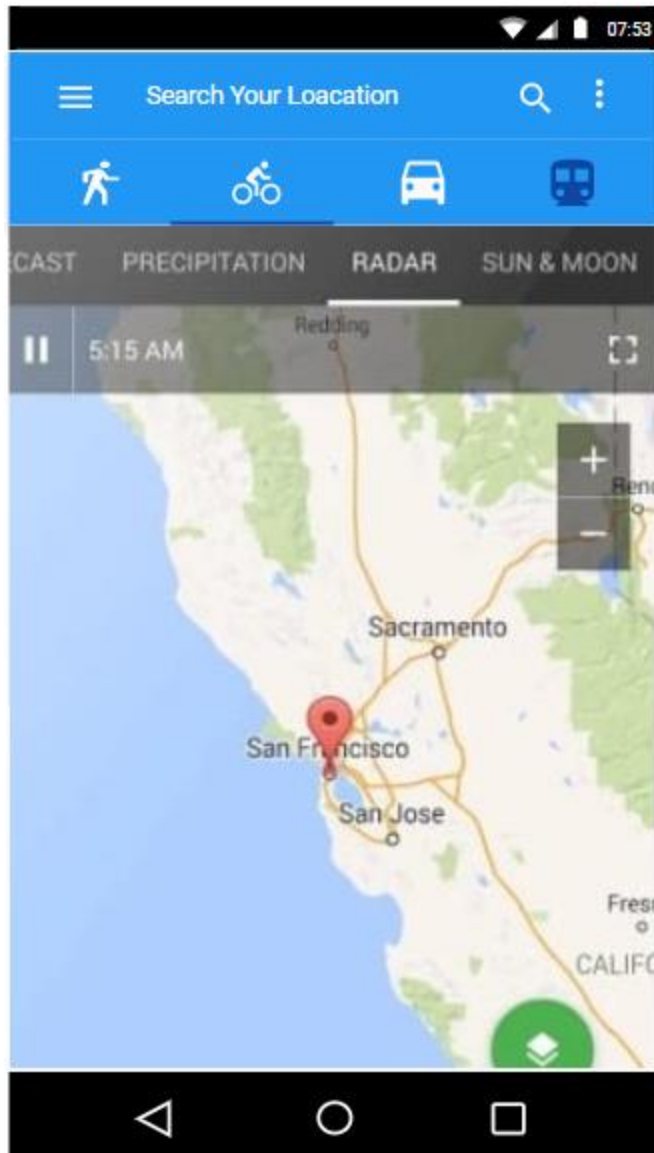
2.2   Production Functions.

2.2.1   Use Case Diagram.
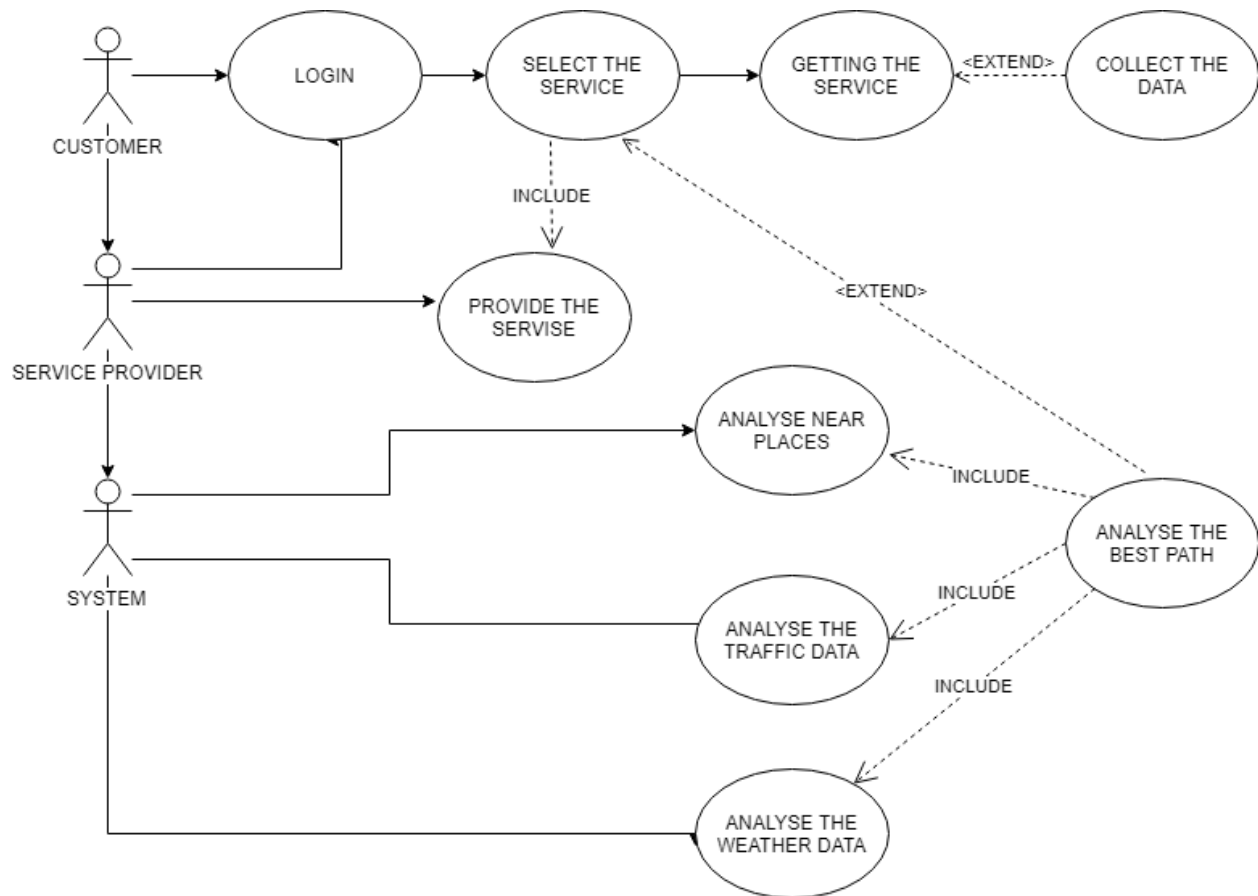
Figure 3: Use case diagram.

2.3 User characteristics

Use case Name	Feedback system for customers
Pre –Condition	Customer should be logged into the system and should get a service from the selected service provider.
Post-Condition	Recording and publishing the feedback
Actor	Customer
Main Success Scenarios	1. Selecting a specific service provider for a specific work. 2. Getting the service from the service provider. 3. Completing the transactions or finalizing the service they had. 4. Rating the service provider and adding comments.
Extension	1. a. Selected service provider is unavailable and select another service provider. 1. c. Should select a valid payment method. 1. d. Rated service provider is not allocated to this domain.

Table 3: Use Case Scenario 1.0

Use case Name	Creating the daily plan
Pre –Condition	Login to system
Post-Condition	Confirm the Trip
Actor	Customer
Main Success Scenarios	1.Select the daily plan tab. 2.Select the start and end point. 3.Identify the time that the user wants to be in the destination.
Extension	1. a. User not giving access to GPS data. 1. b. User not connected internet.

Table 4: Use Case Scenario 2.0

Use case Name	Feedback system for service providers
Pre –Condition	Service providers should be logged into the system and provide services for the selected customers.

Post-Condition	Rating and commenting about the customers.
Actor	Service Provider (API)
Main Success Scenarios	1. Agreeing with a specific customer to provide a specific work. 2. Providing the service for the selected customer. 3. Rating the customer and adding comments. 4. The closest service provider search in app and the system routes them to customer location. 5. Service provider accept the request and customer is notified that a service provider is on the way 6. System notifies user via text message when service provider is one minute away from the meet location 7. service provider and customer meet at meet location
Extension	1. a. Selected customer is unavailable and select another customer. 1. c. Rated customer is not allocated to this domain.

Table 5: Use Case Scenario 3.0

Use case Name	Closest route defining
Pre –Condition	Getting the closest route according to the data
Post-Condition	When user finishes the destination with the provided route
Actor	System
Main Success Scenarios	1. Get data for the closest route 2. Identify start point and end point 3. Provide the user with the closest route on the map.
Extension	1. a. User not giving access to GPS data. 1. b. User not connected internet.

Table 5: Use Case Scenario 4.0

2.4   Constraints

- It is a mobile application so an internet access is a must.
- Uses Firebase to manage the db.

2.5 Assumptions and dependencies

2.4.1 Assumptions.

- Assume that all constraints are given to the system before generating any routes or plans.
- Assume that a Traffic data and Weather data is 100% optimized.
- Assume that the data in the DB is 100% accurate.

2.4.2   Dependencies

- Accuracy and efficiency of the final output depends on the traffic data and weather data analyzer.
- Stored data will affect to generate a better daily plan route for the users

3. Specific Requirements

3.1   External Interface Requirements.

3.1.1   User Interfaces.

Login Interface:

Login requires the user to enter registered user information, first a user name and then a password. This information is entered into a login window on a GUI.

Weather analysis Interface:

In here we generate the weather analysis data for the selected routes

Traffic Analysis Interface:

In here we generate the traffic analysis data for the selected routes

3.1.2   Software Interfaces.

Database:

Firebase is easy to work with android online apps. Therefore, it is being used as our Database.

Code / text Editor:

Android Studio will be our main code editor

3.2   Classes / Objects

```
                        ┌──────────────┐
                        │   Delay      │
                        └──────────────┘
                               │
                              ╱ ╲
                             ╱ISA╲
                            ╱─────╲
                           ╱       ╲
              ┌────────────┘         └────────────┐
              │                                   │
              ▼                                   ▼
┌──────────────────────────┐      ┌──────────────────────────┐
│        Weather           │      │         Traffic          │
├──────────────────────────┤      ├──────────────────────────┤
│                          │      │                          │
│ + Weather Type: String   │      │ + Traffic Type: String   │
│ + Weather Culture : String│     │ + Traffic Culture : String│
│ + Weather Duration: integer│    │ + Traffic Duration: integer│
│                          │      │                          │
├──────────────────────────┤      ├──────────────────────────┤
│ +Calculate Delay()       │      │ +Calculate Delay()       │
└──────────────────────────┘      └──────────────────────────┘
              │                                   │
              └─────────────┐      ┌──────────────┘
                            │      │
                            ▼      ▼
┌──────────────────────────┐      ┌──────────────────────────┐
│         User             │      │        Vehicle           │
├──────────────────────────┤      ├──────────────────────────┤
│ - UserId:String          │      │ + Vehicle Type: String   │
│ + UserName:String  ──────┼─────▶│ + Vehicle Speed : Integer│
│ +Email : String          │      │ + Vehicle Size: integer  │
│ +Destination :String     │      │                          │
│ +StartingLocation :String│      │                          │
│ +Gender : String         │      ├──────────────────────────┤
├──────────────────────────┤      │                          │
│ +Select Location()       │      └──────────────────────────┘
│ +Select Arrival Time()   │
└──────────────────────────┘
```
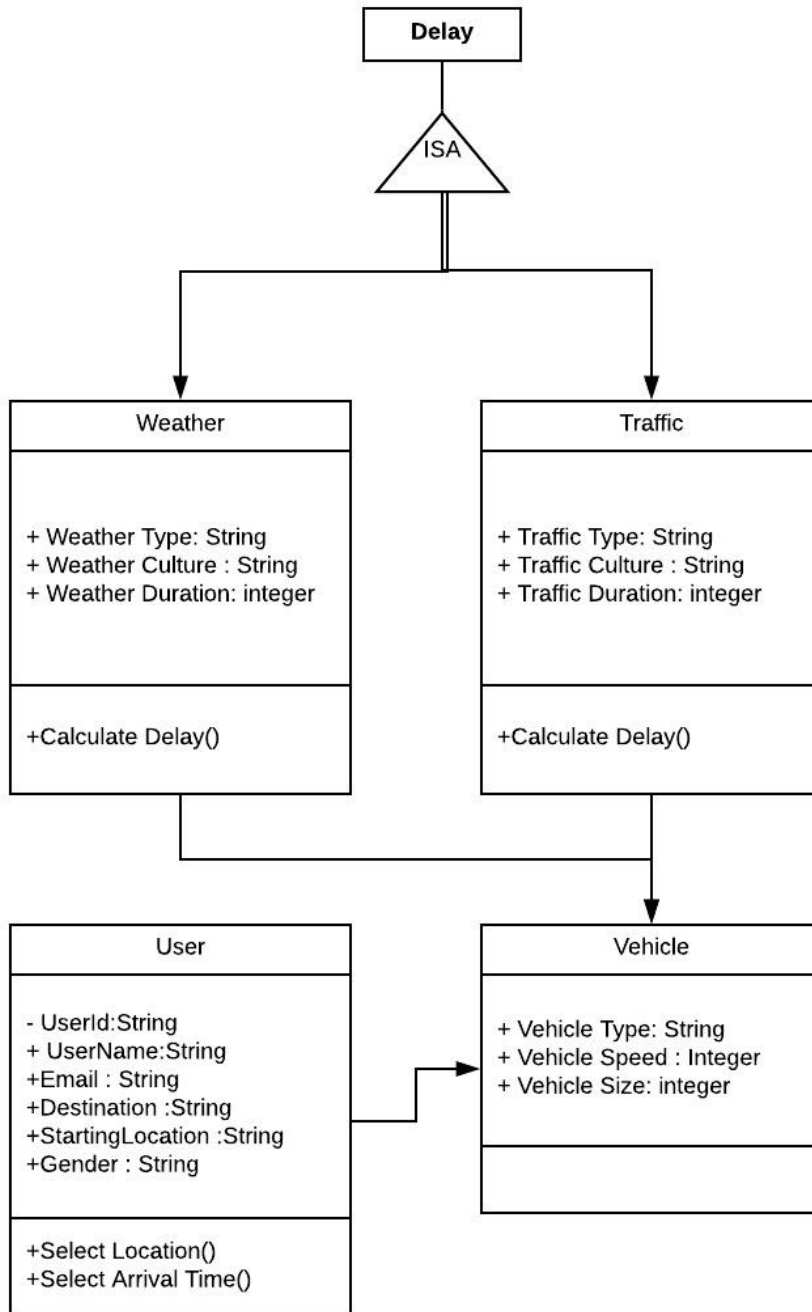
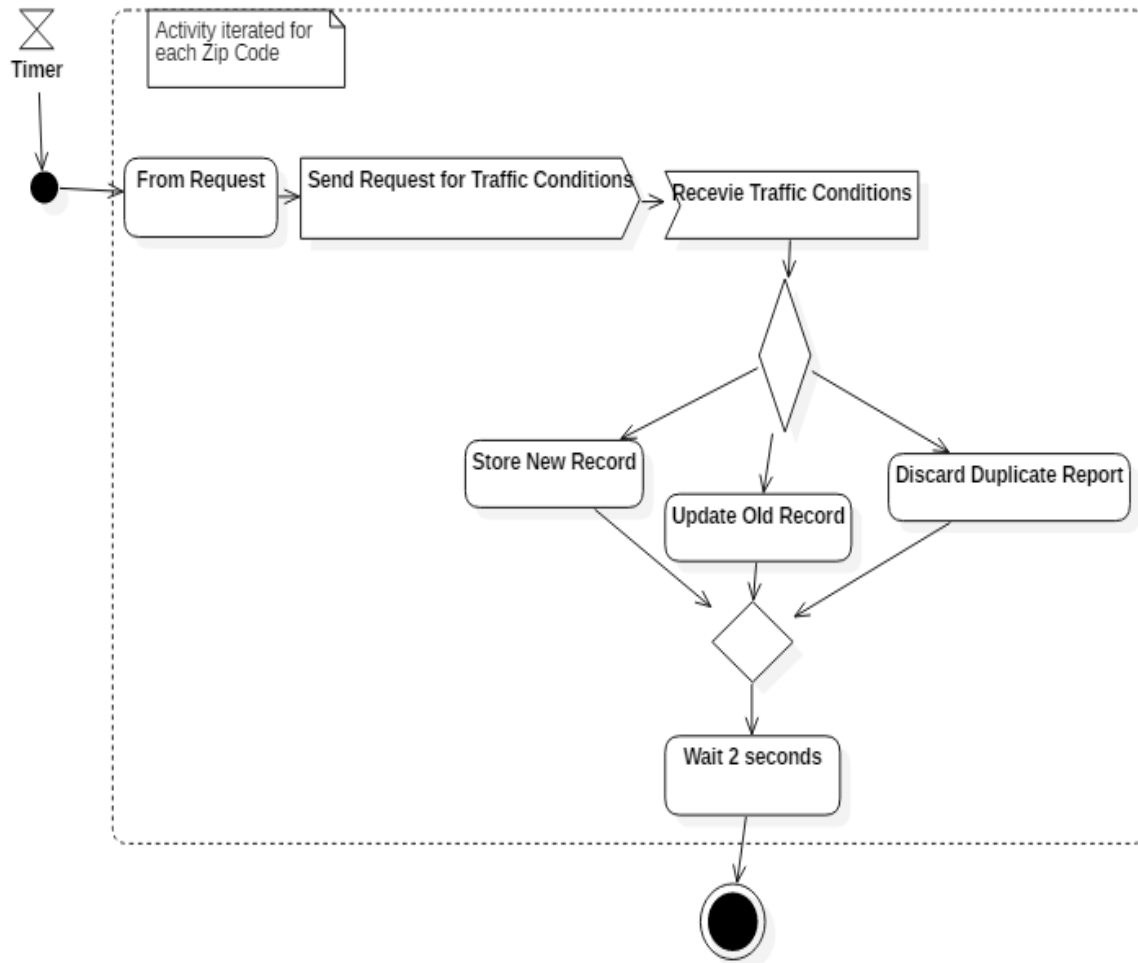Figure 4: Class Diagram

3.3   Activity Diagram.

Figure 5: Activity Diagram

3.4　Design constraints

Our app uses both GPS and Internet. Therefore, to use the app users need to get registered. All registered users can get the services but only the users with permissions can edit/update the data.

3.5　Software System Attributes

3.5.1　Reliability.

Our app should be reliable and should not crash or give out false data that might cause the users to get lost in somewhere they might not even know. Therefore, It should be optimized to provide the user with better data.

3.5.2　Availability.

It should be available at any time which mean the user should be able to access the system at any time 24/7 X 365.

3.5.3　Security.

The system should be at its best security level. Authority and authentications are given for accessing the system.

3.5.4　Maintainability

OOP concepts are used for to implement this system so maintainability is easily covered by following the OOP concepts.