Characterization of Traveling Data Using GPS-enabled Smart phones

By

Muhammad Azeem

Supervised by

Madam Memoona Afsheen

Department of Computer Sciences

Quaid-i-Azam University

Islamabad

In the name of Allah who is the most Merciful and Compensate.

A report submitted to the department of computer science, Quaid-i-Azam University, Islamabad as a partial fulfillment of the requirements for the award of the degree of master in computer science

DEDICATED

TO

My loving parents, respected teachers, siblings and my beloved friends their love and inspirations are my sources of strength, guidance and achievement for me.

Acknowledgement

First of all I would like to express my sincere and humble gratitude to Almighty Allah whose blessing and guidance has been a real source of all the achievements in my life. I am thankful to my parents, brothers and my sister for their prayers, moral & financial support during my educational career.

I would like to express my sincerest gratitude to my supervisor, Madam *Memoona Afsheen*, who has supported me throughout my project with her patience and knowledge whilst allowing me to work in my own way. I attribute the level of my Master degree to her encouragement and effort, and without her this work, too, could not have been completed.

It is my bounden duty to pay tributes to my worthy teachers and staff members of Department of Computer Science Quaid-i-Azam University Islamabad who have molded a lump of clay and made me into what I am today. Special thanks to Dr. Afzal Bhatti, Dr. Onaiza Maqbool, Dr. Shuaib Karim, Dr. Mubashir Mushtaq, Dr. Rabeeh Ayaz Abbasi, Mr. Umer Rashid, Mr. S.M. Naqi, Dr. Ghazanfer Farooq, and Madam Iffrah Farukh khan for their kind help and guidance.

In the end I would like to thanks all of my class fellows.

Abstract

This project is related to the development of a platform in which data is obtained by GPS, and uses this data to analyze user's travelling routine and generate report which will help the user to decide which time and path is better for its journey to the place where he/she wants to go.

Contents

Chapter 01

1.1 Introduction

This chapter describes why the proposed system is needed and what its advantages are. It also describes the motivation for this project and focuses on its applications. In short, that this chapter provides an overview of proposed system.

1.2 Problem Definition

Nowadays, time is the most precious commodity. People have to go to work or ought to do a lot of important things. But sometimes they get stuck in the traffic because the path they chose was not smooth enough, it may be through a bazaar where there is a lot of traffic or through an area where there are schools especially on certain timings let’s say when school time is over and pupils are going home etc.

Because of that people get late and in certain cases a person drives very fast to reach his/her destination on the right time, which is dangerous for his/her self and for the people or things in the surrounding environment.

A person therefore should know which time he/she should leave for his/her destination so that he/she does not get late. Thus save his/her time and also protects against any possible dangers/accidents etc.

1.3 Related Work

There are different applications which are using GPS and use the data obtained from GPS for different purposes e.g. to navigate user of his destination, tell the user his/her speed, altitude etc. However, they do not analyze user’s commute and help user to analyze his travelling habit. Some popular android applications working today are:

- **TeleNav GPS Navigator**

  This app helps user in driving. Telenav GPS is taking a backseat, making way for newer and better Scout with:

  -Free Voice Turn-by-Turn Directions
  -My Traffic

- **Waze Social GPS Maps and traffic**

  This application analyzes user’s location and provides user with information like road is block, or a traffic jam etc. based on the data uploaded by different users on social media website and may suggest alternate path.

1.4 Motivation

The motivation behind this application is, to aware the user about the route he chooses and timings of his/her travelling habit which will eventually help the user to plan his schedule. Not to put his/her self and other around him/her in danger by driving dangerously fast and thus save the user from any possible loss.

1.5 Proposed Solution

The proposed system will get the data from GPS. Use this data to perform computations and analyze user’s commute. Then break-up the user’s trip into multiple parts to tell the user about certain characteristics of his/her journey. Such as:

- User can see which part of his journey was at dangerously high speed.
- User can see which part of his journey was spent in congested traffic. That helps the user to see if his commute is causing excessive fuel usage due to sub-optimal driving conditions.
- User can perform a time-of-day correlation with the quality of his commute to optimally choose such timings which suites his commute so that his commute is fast, smooth and efficient.

1.6 Scope

The application helps user to identify characteristics of his commute. User should be able to see which part of his journey is in congested area, normal traffic area and where he drives at high speed. User's speed will be used to identify congested, normal traffic and high speed area in user's journey/trip. The application will record the data from GPS then save it into SQLite Database and then uses this data to characterize user's trip/journey. The application will generate reports which the user can view and delete whenever he wants.

Chapter 02

2.1 Requirements Specification

Requirements specification for a software system is a complete description of the behavior of the system to be developed and may include a set of use cases that describe interactions the users will have with the software.

The requirements specification contains all necessary requirements that are required for the project development. To derive the requirements we need to have clear and thorough understanding of the product to be developed.

2.1.1 Objectives

Following are the objectives of the application:

- To make user aware of timings that are better for his/her journey.
- To make user aware about the best possible path/route to reach his/her destination.
- To help the user to schedule travelling in a better way.
- To help user avoid any loss either financial or physical or both.

2.1.2 Constraints

Following are some constraints which restrict developer's options:

- The system is completely dependent on GPS. If the GPS is not available the application will not work.
- An android based smart phone is mandatory to run the application.
- Internet connection should be enabled in order to see the full features of the application.

2.2 Functional Requirements

Following are major function which system must provide:

- Application must check whether GPS service is enabled or not.
- Application must capture the value of longitude, latitude and speed associated with the user's commute.
- Application must store the data obtained from GPS into SQLite database.
- Application must tell the user which part of his/her journey is at dangerously high speed

- Application must tell the user which part of his/her trip is in congested traffic.
- Application must tell the user which time his/her commute was most efficient and smooth.

2.3 Non-functional Requirements

Non-functional requirements impose constraints on the design or implementation. Following are the non-functional requirements of the system:

- The application should not use too much memory on user's phone.
- The application should not consume/drain user's phone battery.

2.4 Use Case Analysis

"A use case describes the manner in which an actor (in the context of user interface design, an actor is always a person) interacts with system" (Pressman, 2009).

We specify requirements in the form of use cases. A use case diagram shows all use cases and their interaction with mostly primary actor. After use case diagram, use case description is discussed. Here we also discuss the identification of classes and their relationships.
Use case analysis is performed in order to identify the use cases of the system.

2.5 Use Case Diagram

A use case diagram shows the user of the system and his/her interaction with the use cases. A use case diagram depicts actors, use cases, and the relationships among them. Use case is Software engineering term. With some simple extensions, use case diagram can be used to represent a usability professional's task model.

The following is the use case diagram of the system:

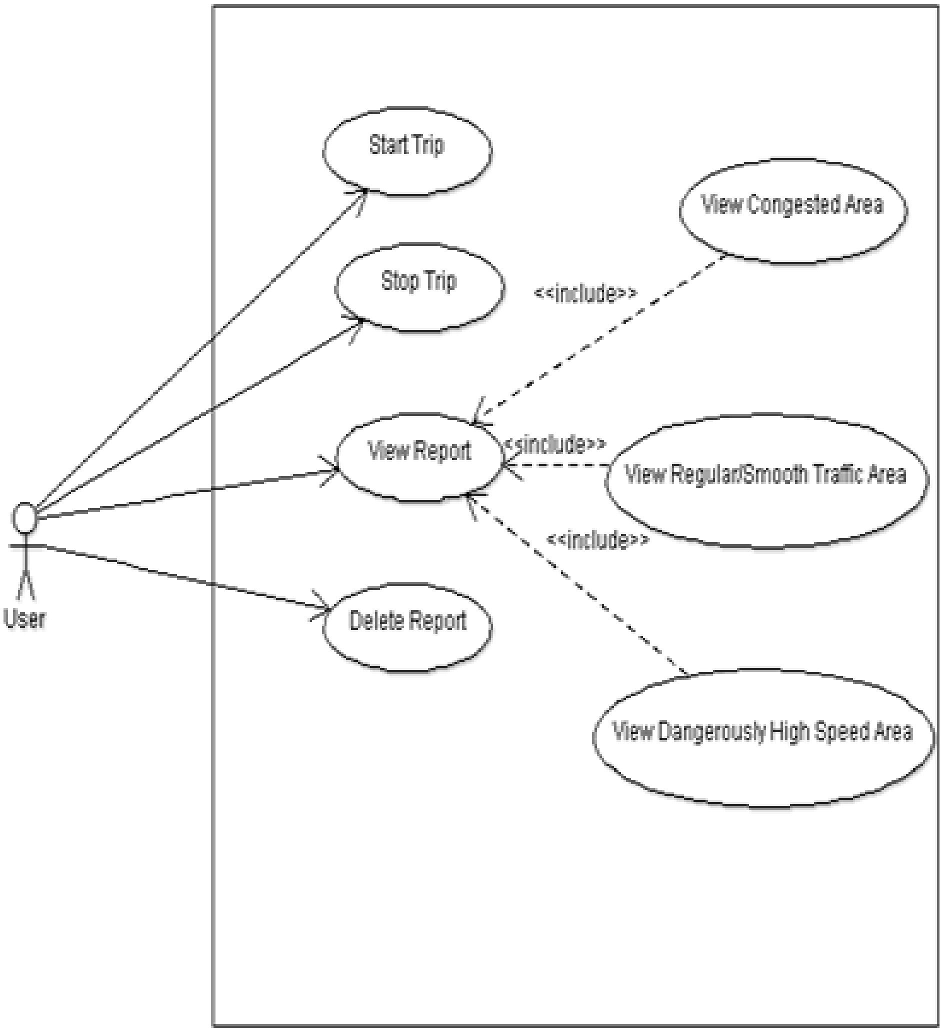


Figure-2.1 Use case diagram

2.6 Use Case Description

All use cases are written in detail which includes use cases description, primary actor, pre-condition, post-condition, success scenario and extensions. (Larman, 2002)

Use Case # 01	View Report
Primary Actor:	User
Pre-condition:	The application Should be running.
Post-condition:	The user successfully viewed his/her trip report which he/she wants to see.
Scenario:	1. The application provides user with a menu. 2. User selects the View Report option. 3. The application displays all the reports.* 4. The user selects the report he/she wants to see. 5. The system presents the report to the user.
Extension:	*3. There is no report available to present to the user

Use Case # 02	Delete a Report
Primary Actor:	User
Pre-condition:	There is at least one report available to delete
Post-condition:	The user successfully deletes the report.
Scenario:	1. The user starts the application. 2. The application displays a menu to the user. 3. User selects the View Report option. 4. The application display report(s) to the user. 5. User selects the report he/she wants to delete 6. User selects the delete option. 7. The application removes the report from its database.

Use Case # 03	Stop the trip
Primary Actor:	User
Pre-condition:	The application should be running and analyzing user's trip.
Post-condition:	The user successfully stops the trip, and the application stops reading data from GPS.
Scenario:	1. The application displays user a menu which contain stop trip option. 2. The user selects the stop option from the menu. 3. The application stops reading data from GPS.
Extension:	None

Use Case # 04	Start Trip/Commute
Primary Actor:	User
Pre-condition:	The user has started the application and GPS and Internet of the user’s device is ‘ON’.
Post-condition:	The application successfully starts analyzing user’s trip/commute.
Scenario:	1. The application displays user with a menu containing start trip option. 2. The user selects the start trip option from the menu. 3. The application displays a form to the user. 4. User enters the information about the trip in the form and select save option *. 5. The application save the trip information and start analyzing user’s trip.
Extension:	*4. User selects the cancel option. In this case the application will take the user to the initial screen.

Use Case # 05	View Congested Area
Primary Actor:	User
Pre-condition:	User has analyzed a trip using the application.
Post-condition:	User sees congested areas on his route on a Map.
Scenario:	1. User starts the application. 2. The application provides user a menu containing ‘view reports’ 3. User selects the ‘view reports’ option. 4. The application displays user all the recorded reports. 5. User selects the desired report. 6. The application displays report information and provides options to enable user to see congested area. 7. User selects the ‘congested area’ option. 8. The application opens a map which shows the congested route in the user’s path *.
Extension:	*8. User hasn’t travelled in congested area. In this case, application displays user a message that will inform the user.

Use Case # 06	View Normal/Regular Traffic Area
Primary Actor:	User
Pre-condition:	User has analyzed a trip using the application.
Post-condition:	User sees regular/normal traffic areas on his route on the Map.
Scenario:	1. User starts the application. 2. The application provides user a menu containing ‘view reports’ 3. User selects the ‘view reports’ option. 4. The application displays user all the recorded reports. 5. User selects the desired report. 6. The application displays report information and provides options to enable user to see normal/regular traffic area in his trip. 7. User selects the ‘normal/regular traffic area’ option. 8. The application opens a map which shows the normal/regular traffic area route in the user’s path. *
Extension:	*8. User hasn’t travelled in the range of normal threshold for speed. In this case, application displays user a message that will inform the user.

Use Case # 07	View Dangerously High Speed Area in Commute/Trip
Primary Actor:	User
Pre-condition:	User has analyzed a trip using the application.
Post-condition:	User sees areas where he/she has driven at dangerously high speed during his commute, on a Map.
Scenario:	1. User starts the application. 2. The application provides user a menu containing ‘view reports’ 3. User selects the ‘view reports’ option. 4. The application displays user all the recorded reports. 5. User selects the desired report. 6. The application displays report information and provides options to enable user to see where he/she has driven at dangerously high speed. 7. User selects the ‘Dangerously High Speed’ option. 8. The application opens a map which shows where the user has driven at dangerously high speed during his/her commute/trip. *
Extension:	*8. User hasn’t travelled at dangerously high speed during the trip/commute. In this case, application displays user a message that will inform the user.

2.7 Identification of Classes

During the system analyses following conceptual classes are identified.

2.7.1 Candidate Classes

- Trip
- Commute
- DataSource
- User
- Map
- Report
- Location
- Time
- Journey
- GPS

2.7.2 Identify Classes

Discard those which are obviously not good candidate classes for any one of the variety of reasons. (Mohapatra)
Discard: Commute and journey, because they are referring to same thing as Trip.

2.7.3 Identified Classes

List of probable classes:

- Trip
- DataSource
- Map
- Report
- GPS
- Location
- User

2.8 Domain Model

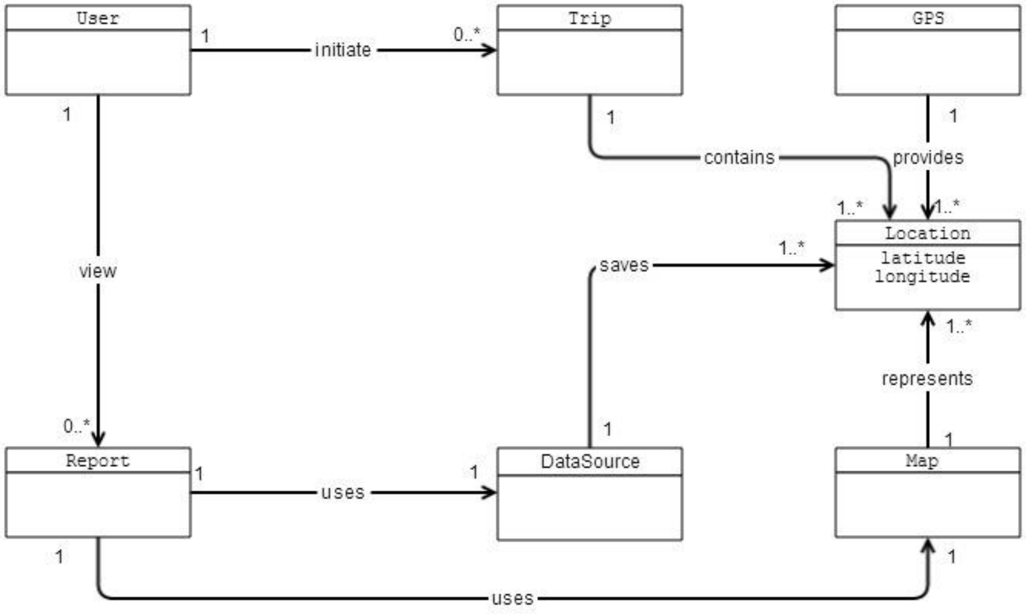


Figure-2.2   Domain model

Chapter 03

3.1 Introduction

System design describes the design of the proposed system. It defines the architecture, flow of the application and class diagram. In system design I am describing major parts of design phase.

1. System Architecture.
2. Activity Diagram.
3. Class Diagram.
4. Sequence Diagrams.

3.2 System Architecture

System Architecture defines the infrastructure of system, that is, it tells that which elements are related to one another and in what manner. How the communication inside a system is being handled? I am following the “Three Tier Architecture”. Use of “Three Tier Architecture” makes the application more understandable and easy to maintain and modify.

3.2.1 Three Tier Architecture

Three Tier Architecture indicates a physical separation of components. It generally contains UI or Presentation Layer, Business Logic Layer and Data Access Layer.

Architecture diagram shows the relationship between different components of the system. This diagram is very important to understand the overall concept of the system. Architecture is use for creating and building the complex system (Rechtin, 1991).

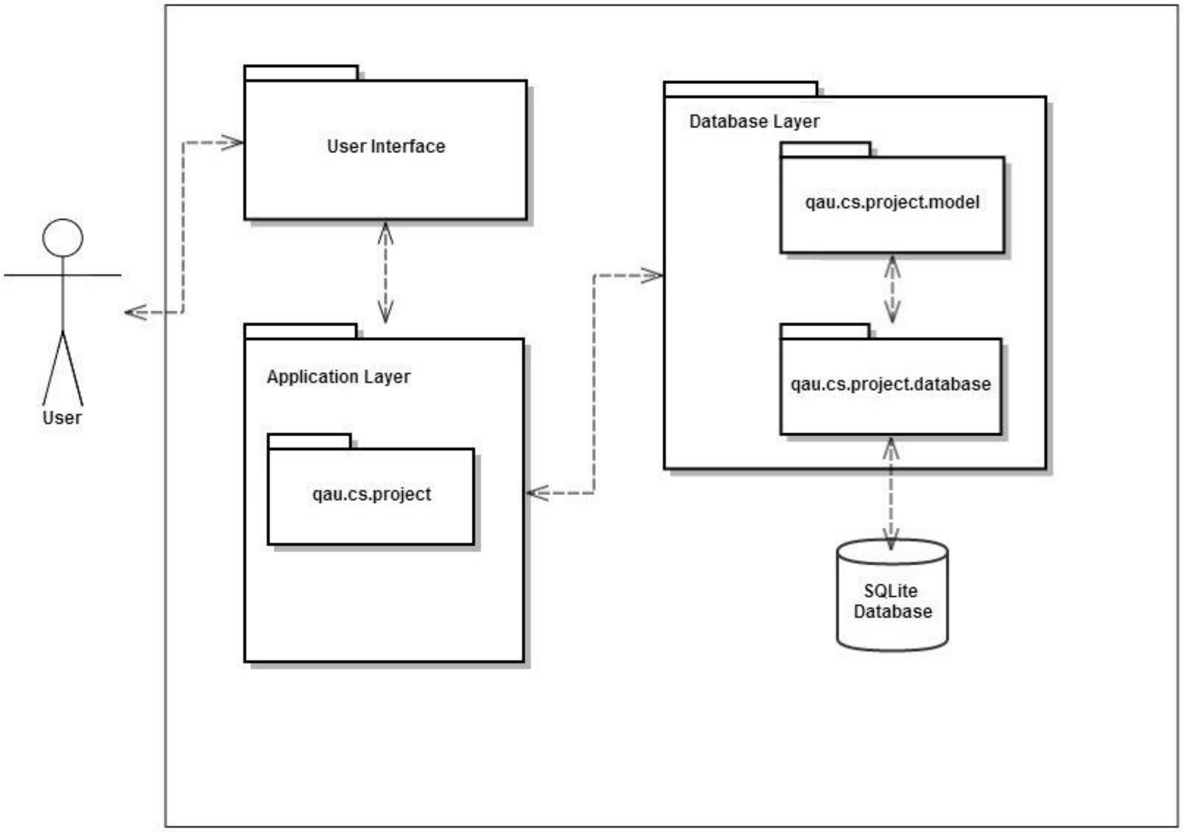
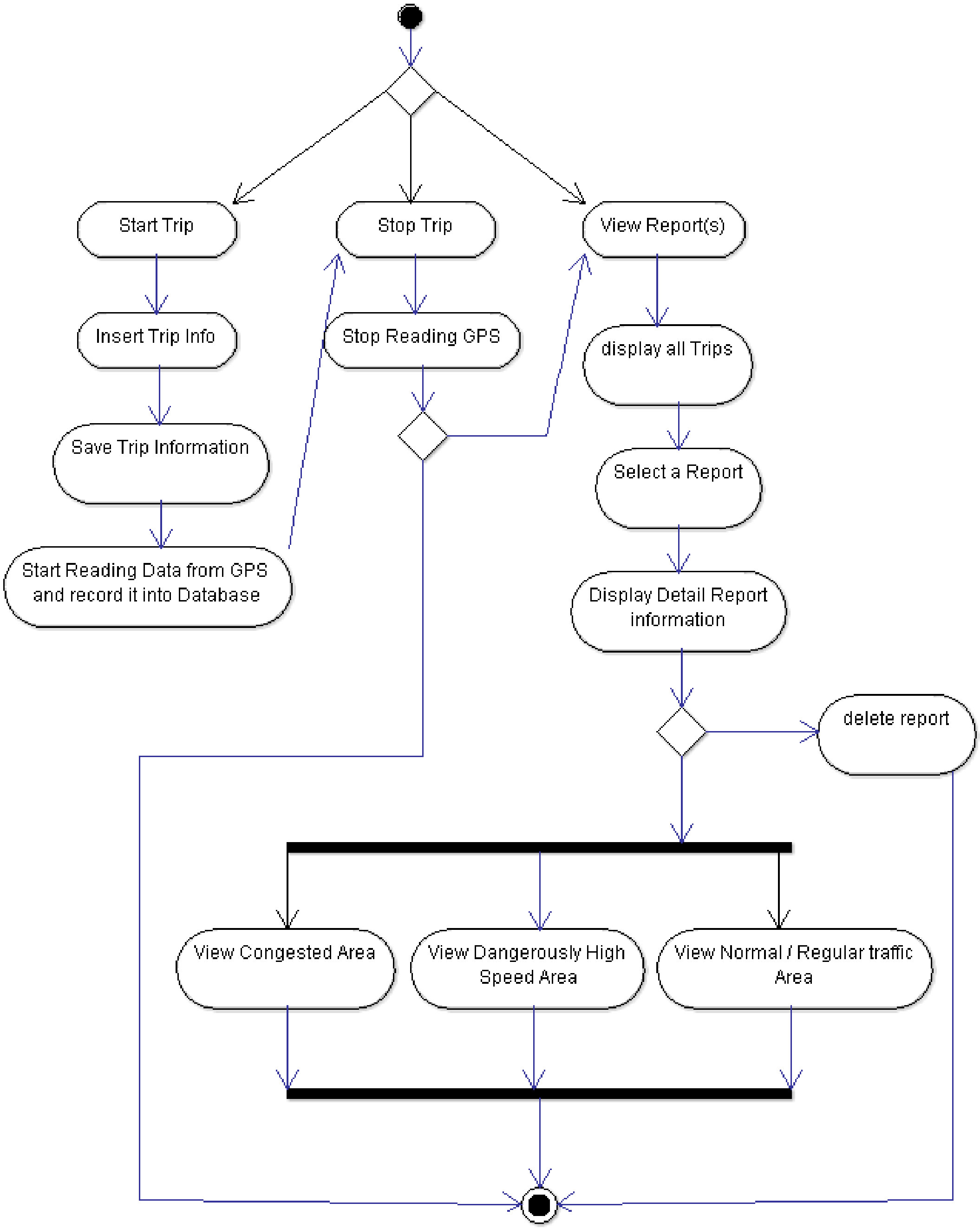


Figure -3.1 System Architecture diagram

3.2 Activity Diagram

Figure 3.2 shows the activity diagram of the system.



Start Trip
Stop Trip
View Report(s)
Insert Trip Info
Stop Reading GPS
display all Trips
Save Trip Information
Select a Report
Start Reading Data from GPS and record it into Database
Display Detail Report information
delete report
View Congested Area
View Dangerously High Speed Area
View Normal / Regular traffic Area

Figure - 3.2 Activity Diagram

3.3 Class Diagram

In an object-oriented design each object is referred to as an instance of class. We can find an association between objects, and these associations are represented through a diagram in UML notation known as” class diagram”.

A “class diagram” in the Unified Modeling Language (UML) is type of static structure diagram that describes the structure of a system by showing the system’s classes, their attributes, operations (methods), and the relationship among the classes.

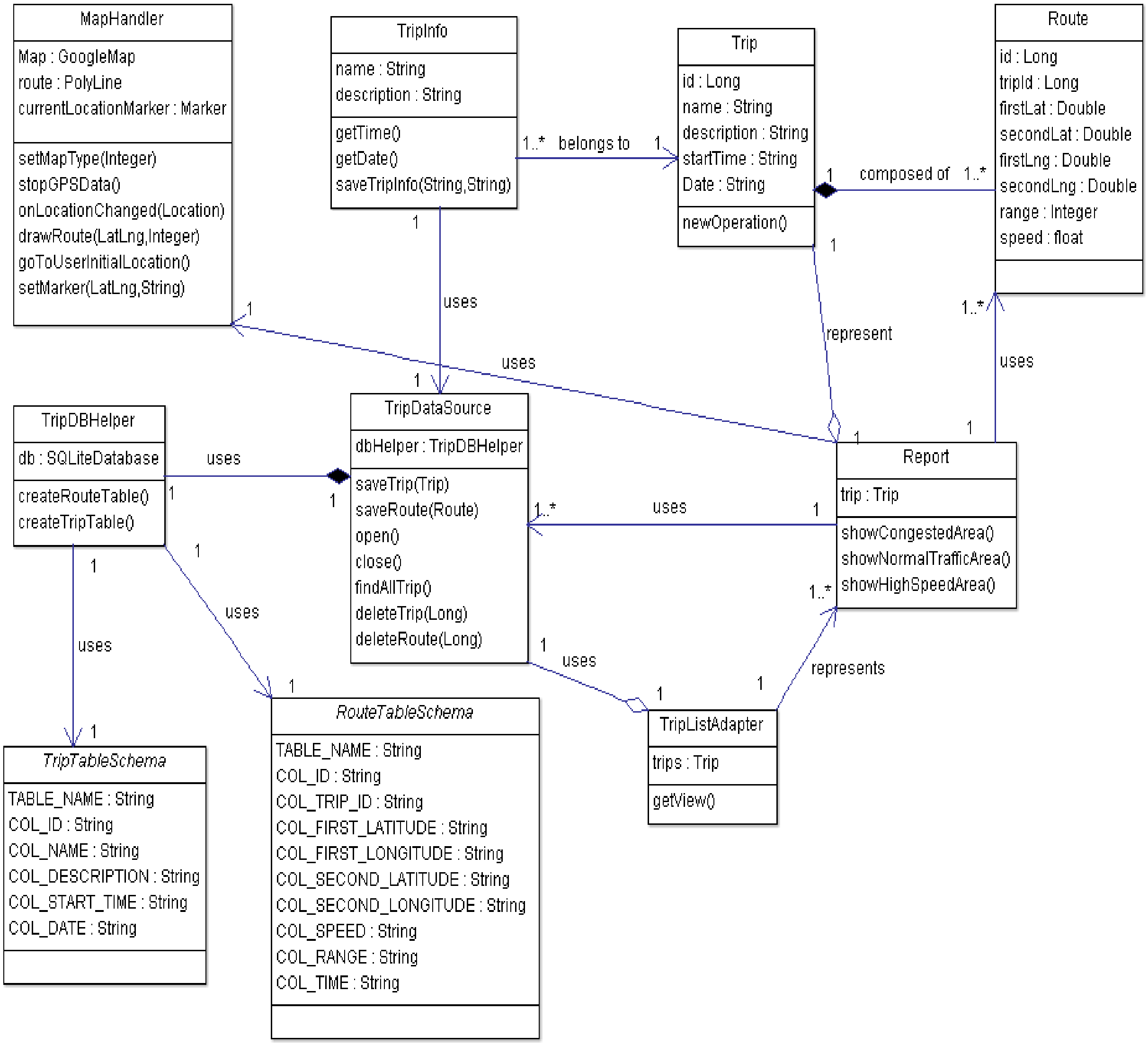


Figure – 3.3 Class diagram

3.4 Sequence Diagram

A sequence diagram shows the sequence of method calls of different objects based on a time sequence. It shows how the objects interact with one another in a particular scenario of a use case.

Start Trip:

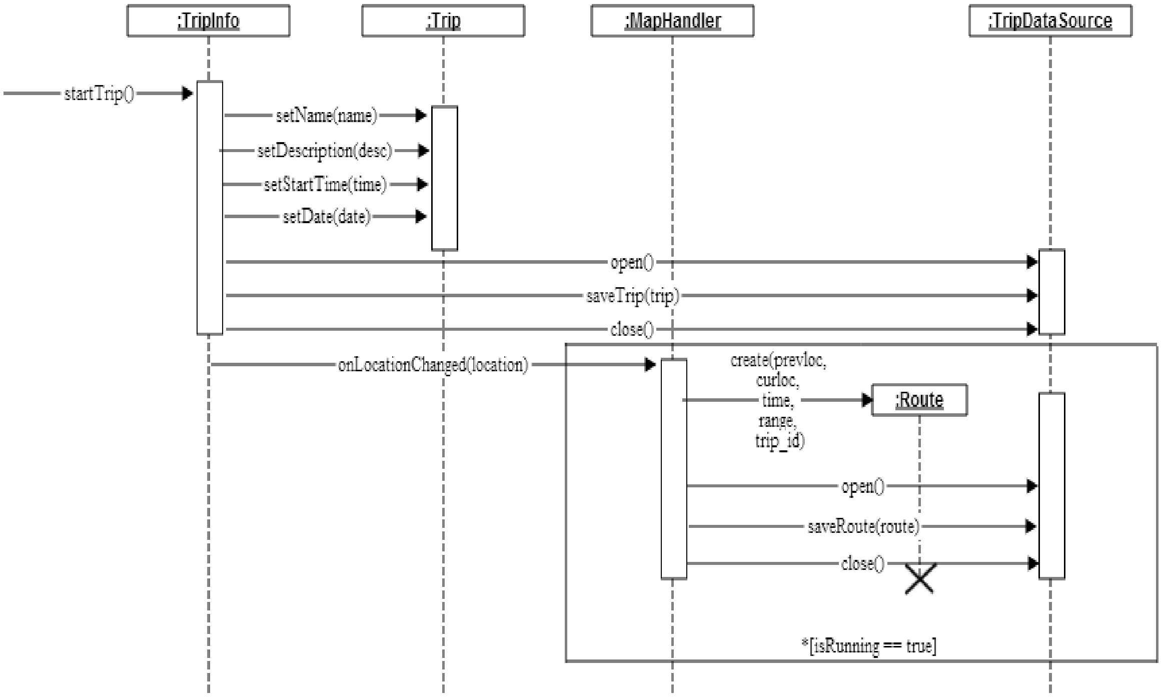


Figure – 3.4 Start Trip Sequence Diagram

Stop Trip:

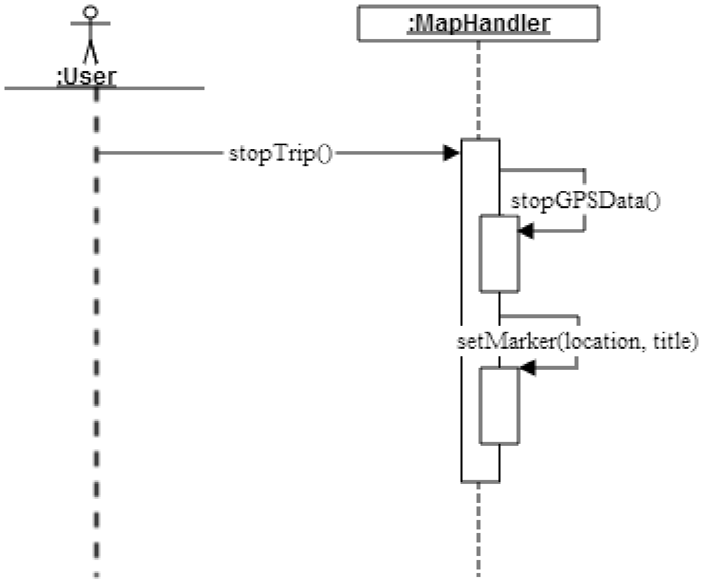


Figure – 3.5 Stop Trip Sequence Diagram

View Report:

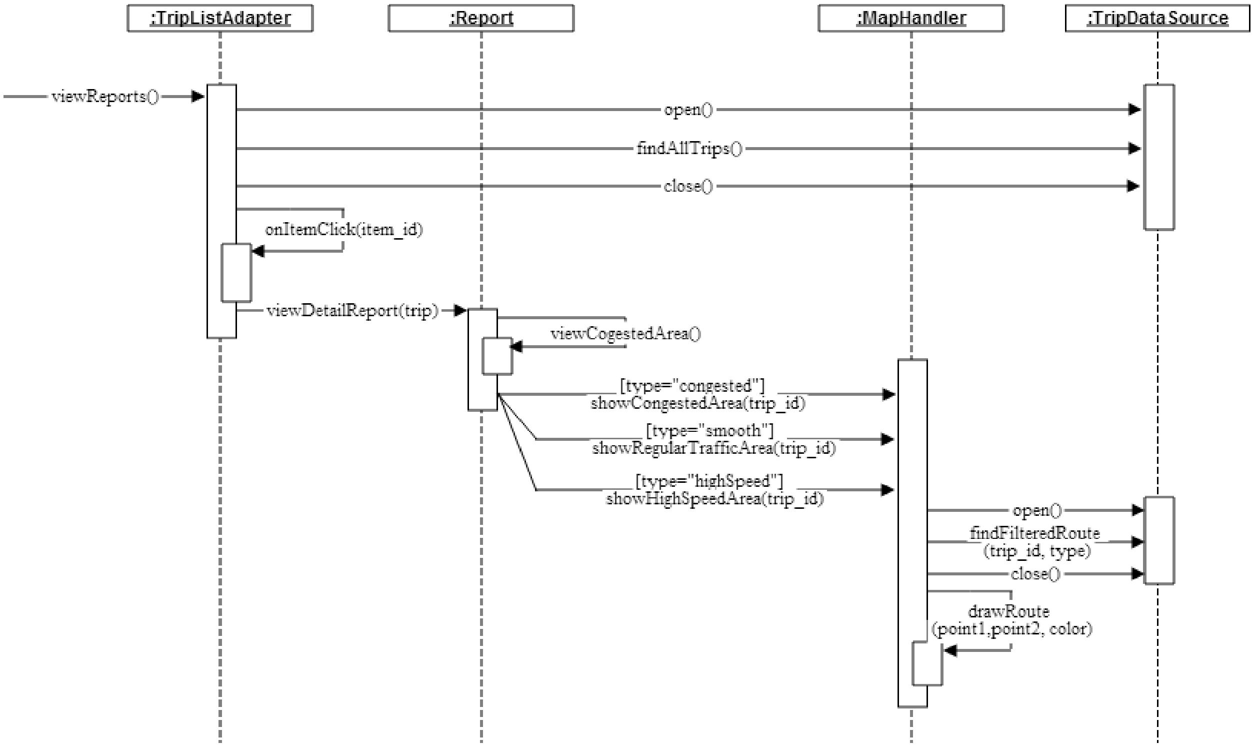


Figure – 3.6 View Trip Report Sequence Diagram

Delete Report:

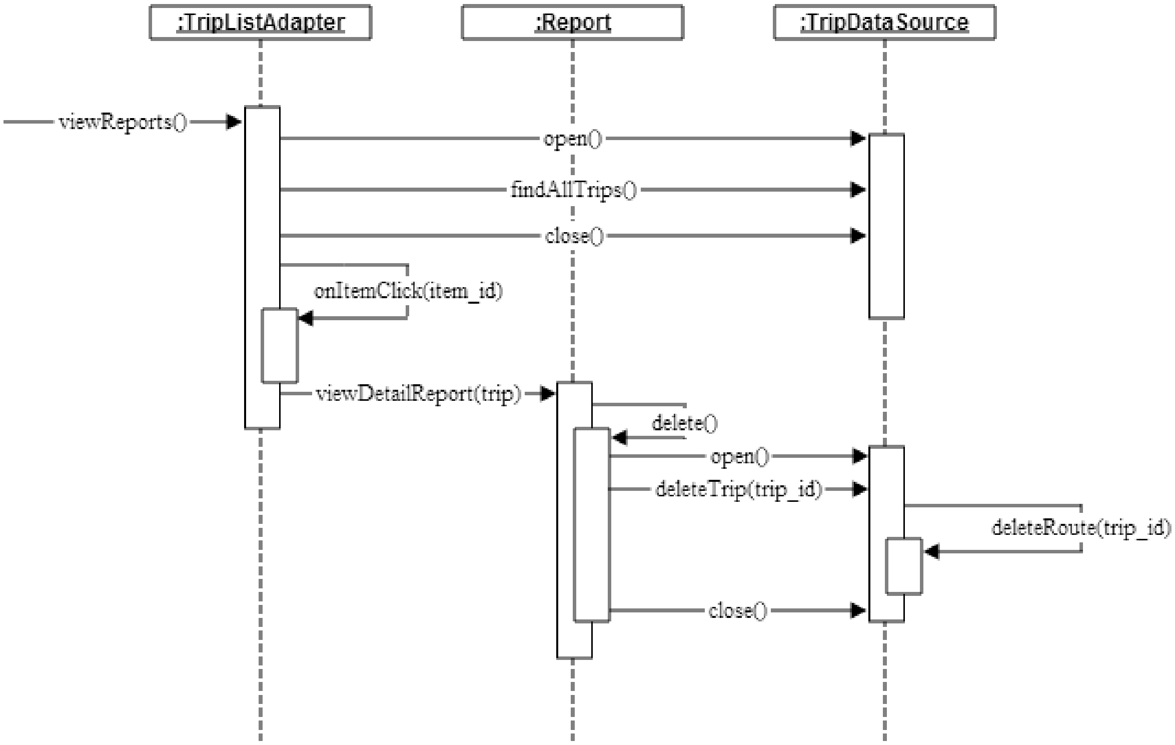


Figure – 3.7 Delete Report Sequence Diagram

Chapter 04

4.1 Introduction

After the design phase comes the implementation phase. The implementation phase takes the requirement and design phase products and implements them using appropriate technologies. The purpose of the implementation is to transfer the design of the system into working software.

4.2 Coding Standards

Following coding standards and conventions have been followed.

- All the packages names are in small alphabets.
- All the constant variable names are written with capital alphabets.
- All the words in class names are in Upper Camel Case, with the first letter of every word capitalized.
- Single line, multi lines, and java documentation comments have been used.
- Proper indentation has been used for code blocks.

4.3 Android Development Environment

There are majorly two options for developing android application i.e. by using android studio or by using Eclipse IDE with Android Development kit.

Android Studio is a new Android development environment based on IntelliJ IDEA. Similar to Eclipse with the ADT Plugin, Android Studio provides integrated Android developer tools for development and debugging. On top of the capabilities you expect from IntelliJ, Android Studio offers:

- Gradle-based build support.
- Android-specific refactoring and quick fixes.
- Lint tools to catch performance, usability, version compatibility and other problems.
- ProGuard and app-signing capabilities.
- Template-based wizards to create common Android designs and components.
- A rich layout editor that allows you to drag-and-drop UI components, preview layouts on multiple screen configurations, and much more.
- Built-in support for Google Cloud platform, making it easy to integrate Google Cloud Messaging and App Engine as server-side components.

For developing application in Eclipse IDE, a good option is to use Android SDK. Android SDK bundle includes the essential Android SDK components and a version of the Eclipse IDE with built-in Android Development Tools, ADT, to streamline your Android app development.

The ADT Bundle includes everything you need to begin developing apps:

- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- The latest Android platform
- The latest Android system image for the emulator

Since Android Studio is in its early development stages and there can be bugs in it, therefore I chose to develop this application using Android SDK bundle.

4.4 SQLite Database

SQLite is a tiny yet powerful database engine created by Dr. Richard Hipp in 2000. It is arguably the most widely deployed SQL database engine in the world. SQLite is an Open Source Database which is embedded into Android. Some characteristics of SQLite that are unusual and which make SQLite different from many other SQL database engines are zero-configuration, server less, single database file and compact.

Since Android platform already have the support of SQLite database, therefore, I am using SQLite database in the application.

In addition to SQLite database, Android preferences are also used in the application for saving user’s settings for the application.

4.5 Google Maps

Google Maps are perhaps the most widely used map application today. Google provide Google Map APIs to integrate Google Map in other applications. Google Maps APIs allow the developer to do many things like drawing on the map, interaction with map, and changing views.

Using Google Map APIs developer can draw polyline, circle and markers on the map. Google APIs provide user zoom control and allow scroll on the map and made user interaction easy. Google APIs provide four map view namely Normal, Satellite, Hybrid and Terrain.

4.6 Location API

Android provide location APIs which allow the developer to make its application location-aware. The location APIs make it easy for you to build location-aware applications, without needing to focus on the details of the underlying location technology. They also let you minimize power consumption by using all of the capabilities of the device hardware.

Chapter 05

5.1 System Testing

System testing contains User Acceptance Test (UAT) specification which is built against use cases stated in Requirement Analysis. Testing plays an important role, after completing the implementation of each module, testing is done to check whether the module is developed according to the client's requirements. Testing is the process of exercising a program with the specification of finding errors prior to delivery to the end user. Testing may also uncover requirement related issues. User Acceptance Test will provide actual function, input, expected result, actual result, procedure to perform test case, pass/fail status and date against test case.

5.2 UAT Test Cases

User Acceptance testing is the software testing process where system tested for acceptability & validates the end to end business flow.

Test Case # 01	Verifying GPS is Not Enabled
Pre – condition	GPS should not be enabled
Function	This test case verifies that whether the GPS is enabled or not.
Tester	User
Input	None
Expected Result	User should see a message to enable the GPS
Procedure	1. Start the application 2. Select the "Start Trip" option.
Actual Result	Massage: "Please, enable your GPS"
Status (Pass/Fail)	Passed

Test Case # 02	Verifying GPS is Enabled
Function	This test case verifies that whether the GPS is enabled or not.
Tester	User
Input	None
Expected Result	User should see a new form asking for trip name and description on his screen.
Procedure	1. Start the application 2. Select the "Start Trip" option
Actual result	User sees new form asking for trip name and description.
Status (Pass/Fail)	Passed

Test Case # 03	Getting data from GPS
Function	This test case will check whether the application is reading data from GPS or not.
Pre- condition	The application is start and his internet connection is also working.
Tester	User
Expected Result	User should see his location on the Map.
Procedure	1. Select the “Start Trip” option
Actual Result	The application displays the longitude and latitude values.
Status (Pass/Fail)	Passed

Test Case # 04	Check drawing on Map.
Function	This test case will check whether the application is reading data from GPS and drawing user’s path using that data.
Pre- condition	The application is start.
Tester	User
Expected Result	User should see blue line representing the path he followed.
Procedure	1. Select the “Start Trip” option 2. Enter “testing path” in name field 3. Enter “Testing drawing on Map” in the description field 4. Select “save” option. 5. Start moving.
Actual Result	The user sees the path he followed.
Status (Pass/Fail)	Passed

Test Case # 05	Check congested area on the user’s commute.
Function	This test case will check whether the application correctly stores the user trip data and then present it to the user later.
Pre- condition	The user should be a trip recorded by the application in which user drive at less the 30 km/h and named that trip “testing congestion”.
Tester	User
Expected Result	User should see yellow line on the map where he drove at less than 30 km/h.
Procedure	1. Select the “View Reports” option 2. Select “testing congestion” from the list. 3. Select “view congested area” option.
Actual Result	The user sees the path where his speed is less than 30 km/h.
Status (Pass/Fail)	Passed

Test Case # 06	Check normal traffic area on the user’s commute.
Function	This test case will check whether the application correctly stores the user trip data and then present it to the user later.
Pre- condition	The user should be a trip recorded by the application in which user drive at more than 30 km/h and less than 70 km /h and named that trip “testing regular traffic area”.
Tester	User
Expected Result	User should see green line on the map where he drove at more than 30 km/h and less than 70 km/h.
Procedure	1. Start the application. 2. Select the “View Reports” option 3. Select “testing regular traffic area” from the list. 4. Select “view regular traffic area” option.
Actual Result	The user sees the path where his speed is less than 70 km/h and more than 30km/h.
Status (Pass/Fail)	Passed

Test Case # 07	Check high speed area on the user’s commute.
Function	This test case will check whether the application correctly stores the user trip data and then present it to the user later.
Pre- condition	The user should be a trip recorded by the application in which user drive at more than 70 km/h and named that trip “testing high Speed”.
Tester	User
Expected Result	User should see red line on the map where he drove at more than 70 km/h.
Procedure	1. Start the application. 2. Select the “View Reports” option 3. Select “testing high Speed” from the list. 4. Select “view high speed area” option.
Actual Result	The user sees the path where his speed is more than 30km/h.
Status (Pass/Fail)	Passed

Chapter 06

6.1 Introduction

This chapter summarizes “Characterization of Travelling data using GPS-enabled Smart Phones” system regarding what functionality system is currently providing and what should be future enhancements.

6.2 Conclusion

The “Characterization of Travelling data using GPS-enabled Smart Phones” is an android based application, which extracts data from GPS and store this data using SQLite Database. This application then uses/process this data to analyze user’s journey and give some useful information to the user which will help the user to save resources like fuel, time and keeps him out of trouble like accidents etc**.**

6.3 Enhancement

This application can be enhanced further in a lot of ways.

- Visual representation of report can be included into the application.
- Add charts to the application.
- Messaging module can be add to the system so that if a child is over speeding than a message is send to his guardian / parent.
- Make the application more efficient (i.e. to improve its algorithm).
- Launch this application in Android Market place.
- Make User Interface attractive.

References

- Gliffy.com
- Craig Larman, (2002). Applying UML and Patterns
- Android Asset Studio
- Partap K.J.Mohapatra, Software Engineering lifecycle approach, new age international publishers.
- play.google.com
- www.androidhive.com
- developer.android.com
- Wikipedia.org