FASTPOOL

A Carpooling Application designed especially for FASTians.

Software Design Specification

Version 1.0

November 6, 2016

Sauman Balkhi K14-2107

Hammad Khan K14-2145

Akif Jawed K14-2093

Instructor:

Ma'am Amber Khan

Revision History

Date	Description	Authors	Comments
6-11-2016	Version 1.0	Sauman Balkhi Hammad Khan Akif Jawed	None

Contents

1. Introduction

Everyday many students of universities or colleges around this city suffer due lack of proper transportation. They have to wait for hours for their ride and even when it arrives, the quality is very substandard. After seeing the sufferings of fellow students thought of designing such system that will try to remove everyday student's sufferings. The project will target FAST NUCES but can be implemented on any institutes which is associated with many professionals

1.1 Purpose

Aim of this software specification requirements document is to provide a complete description of all of the features that are planned to implement to system and define the expectations from the Carpool project. It also describes how the system operates and how users interact with the application. Besides external systems and interfaces which the application depends, are specified in this SRS document.

The potential audiences for this document are design and development team of the Carpool Project in order to specify software designs. Test team utilizes this software specification requirements document to define test scenarios according to the mentioned requirements. Besides project manager, quality manager and acquirer use this SRS document for reviewing purposes.

1.2 Scope

The Carpool Project is a web based and mobile based application which includes user interaction. Our project is going to be a web portal and an android application. It is going to provide communication environment for users (drivers and riders). Every user has their own profiles and they can have access with given password to the system.

The drivers can draw their routes from map in our mobile app. And riders can communicate with the driver via the messaging system and pick their path. After mutual agreement with each other, they record the transportation information to the system. At the end, users can assess each other via feedback system.

The system will bring many advantages. For instance, the drivers and riders spend less money on traffic. Moreover, traffic jam and air pollution will be decreased. And everyone benefits from these advantages.

In high level details, system will use Google Map API for retrieve location information, MySQL DBMS to store and manipulate the data, PHP for server side management, Android studio for developing an android app and GUI to interact with users.

1.3 Overview

Software Design Document (SDD) of Carpool provides necessary definitions to conceptualize and further formalize design of the software, whose requirements and functionalities were summarized in Software Requirements Specifications (SRS) Report. Aim is to provide guidance to a design which could be easily implemented by any programmer reading this report.

2. SYSTEM OVERVIEW

2.1 Product Perspective

Carpool project is independent and self-contained. The system has Google Map API as a subsystem. Google Map subsystem has their own web based and mobile based interface which is a map consists of roads and locations in a desired area and user can easily intersect with this system.

2.1.1 Design Method

The design of this product utilizes an object-oriented approach.

2.1.2 User Interfaces

This software product is developed for drivers and riders. Product will be deployed to web site and a mobile application and all users of the system will access the system through the web interface and mobile app which includes multiple pages according to the system functionality for example for login functionality there will be login page. To access the system, every user has unique user name and password. In addition, there will be a database who stores and manipulate all the data about the users. Website will only be the interface for all the user data which stored by database and the execution of provided functionalities.

After the sign up, user information will be transferred to database. In the sign up process, there will be e-mail verification to verify user information. After that point, users can register through the web interface. After log in, user will be able to log out whenever he or she wants.

2.1.3 Hardware Interfaces

The carpool system is web based and mobile based. The web app is compatible with all the browsers and can be run on any operating system and processor. For mobile app a mobile app is required which supports Android OS.

2.1.4 Software Interfaces

Database management system is required software product for Carpool system because all data about system for example user and route information must be stored in database for later use and system functionality.

MySQL database management system is used for that purpose and it has nice open source user interface which displays table and rows in well formatted form for developers to create and manage the whole database.

Another server that will be used is Google Map Server to provide geographical service and to visualize transportations.

In terms of user interface for website, HTML and Bootstrap library will be used to illustrate the system attractively. The client and server side attraction will be handled with Http Requests by JavaScript and PHP Languages.

For android app the front end will developed via XML and backend will be implemented via JAVA, Android studio will be used to develop the app.

2.1.4 Memory Constraints

Memory constraints cannot be defined till the completion of the project.

2.1.5 Operations

Riders can book rides and drivers can set routes and its details.

2.1.6 Site Adaptation Requirements

This system can run on any browser and on Android OS without any modifications.

2.2 Product Functions

The product have two types of users, drivers and riders. They have some common functions like sign up, sign in, sign out and some unique features like driver can add or delete route and rider can search and request a ride.

2.3 User Characteristics

Students of FAST NUCES.

2.4 Constraints

Internet connection is necessary when using the system. If the user does not have 3G service then problems will occur.

2.5 Assumptions and Dependencies

Everyone has a smartphone with internet connection.

3. DESIGN CONSIDERATIONS

3.1 Operating Environment

Any OS which can run a browser and Android Jellybean or above,

3.2 Fault Tolerant Design

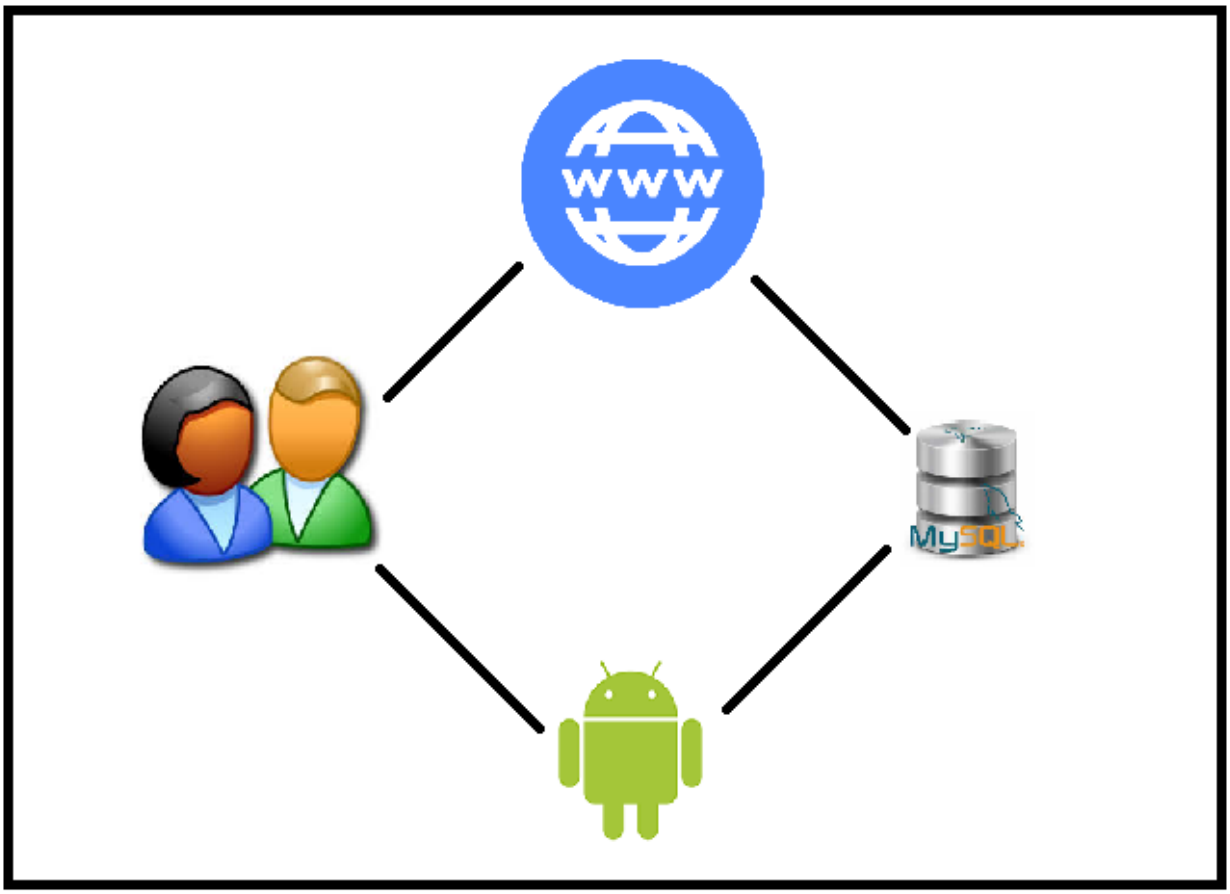
User can report any fault to the admin via a contact us link on the website.

3.3 Design Conventions

Object-oriented design is chosen for this project. This give us an advantage to integrate new features to our project and remove and replace the components which we want to.

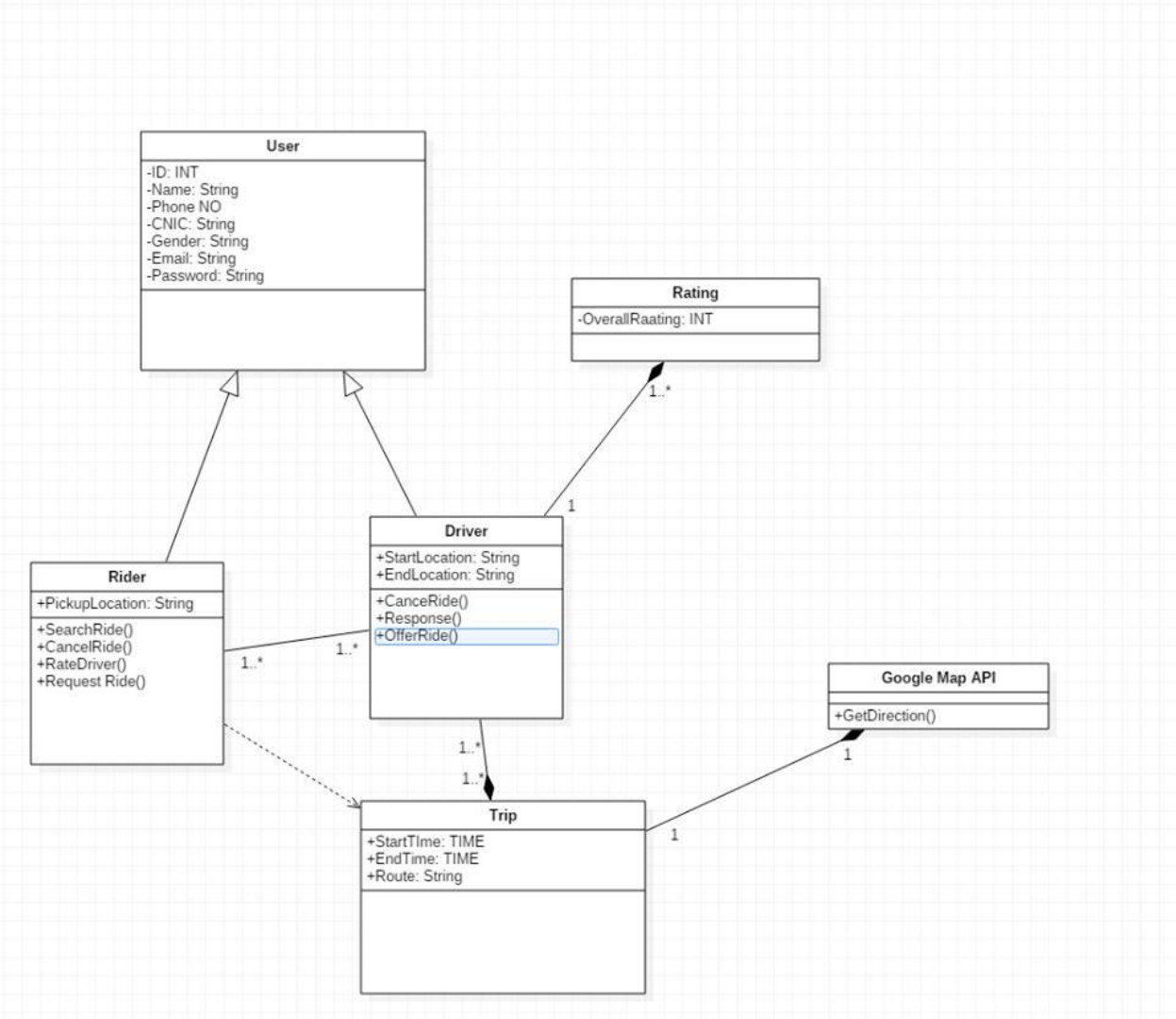
3.4 Architectural Design

The Architecture is explained in the diagram below.



4. SYSTEM ARCHITECTURE

4.1 View of Product Classes



4.2 Individual Classes of System

4.2.1 User

4.2.1.1 Attributes

- Id: This is an integer value that is unique for every user. This id is chosen by the system.
- Password: Every user has his/her own password that is selected by him/her.
- Name : This is first name of the user
- Email: This is e-mail of the user. Every user has to have unique e-mails.
- Phone: This is phone number of the user.
- Gender: This is sex of the user
- Cnic: String data type for national identification.

4.2.2 Rider

4.2.2.1 Attributes

- Pickup_Location: String input of the pickup point.

4.2.2.2 Functions

- serchRide: This function is called when user hits search button in search transportation page. And finds all transportations that includes the specified route.
- rateUser: This function is called when a user wants to rate a user that he/she transported with.
- cancelRide: This function is called when the rider wants to cancel a rider.
- requestRide: This fuction is called when the rider requests for a ride.

4.2.3 Driver

4.2.3.1 Attributes

- startLocation: String input for starting location.
- endLocation: String input for ending location.

4.2.3.2 Functions

- addRoute: This function is called when driver hits add route button.
- deleteRoute: This function is called when driver hits delete route button.

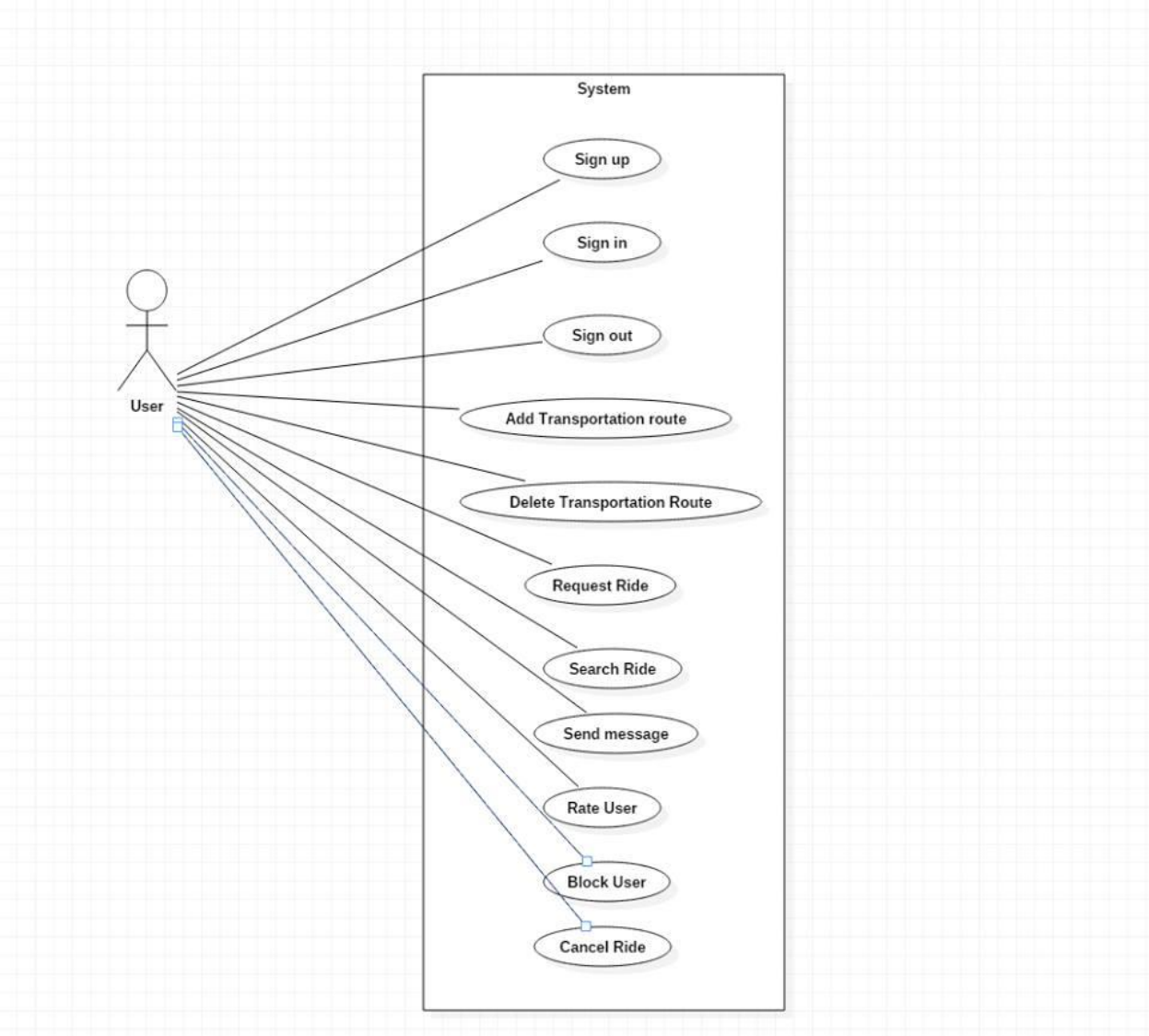
4.2.4 Trip

4.2.4.1 Attributes

- startTime: String input for starting time.
- endTime: String input for ending time.
- route: String input for route driven.

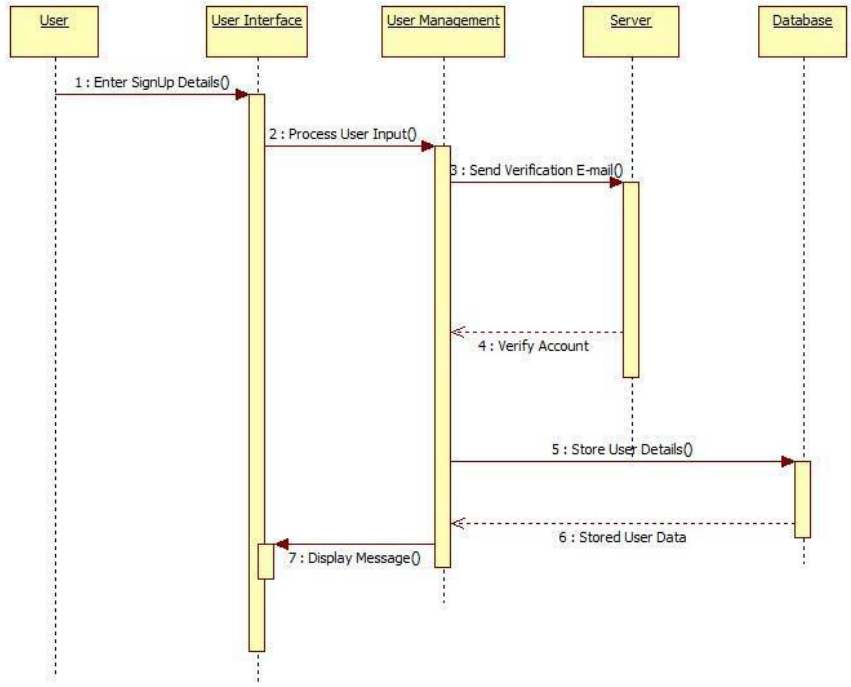
5. FIGURES

5.1 Use Cases

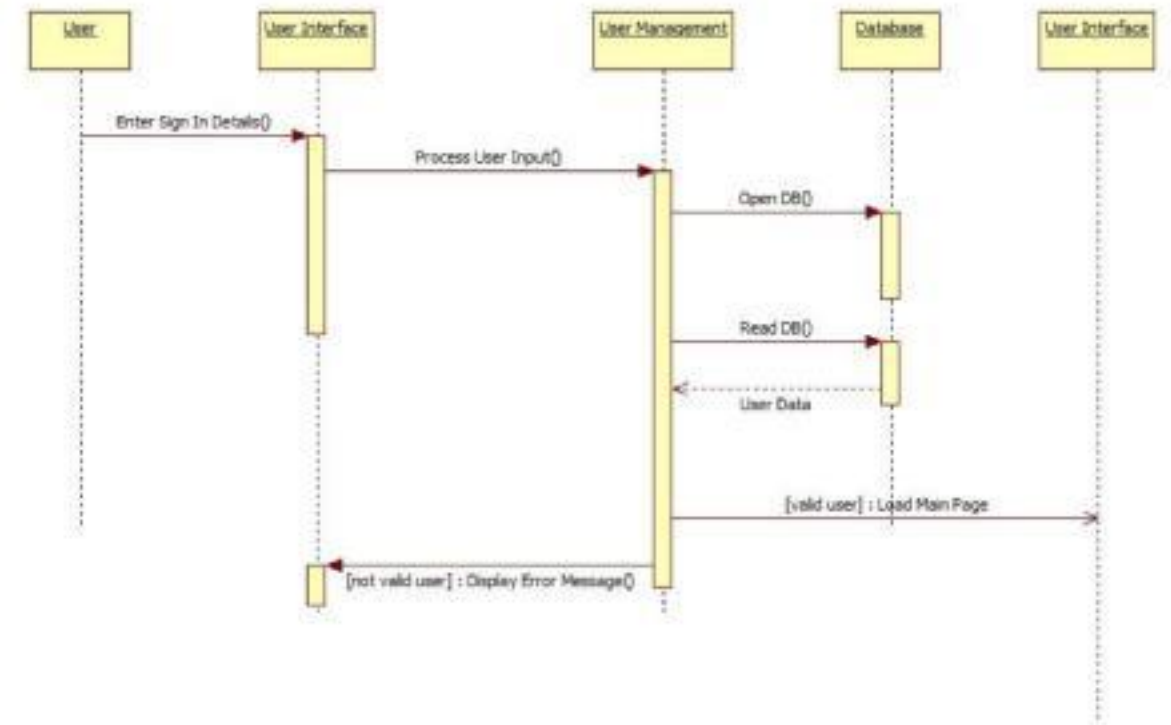


5.2 Sequence Diagrams

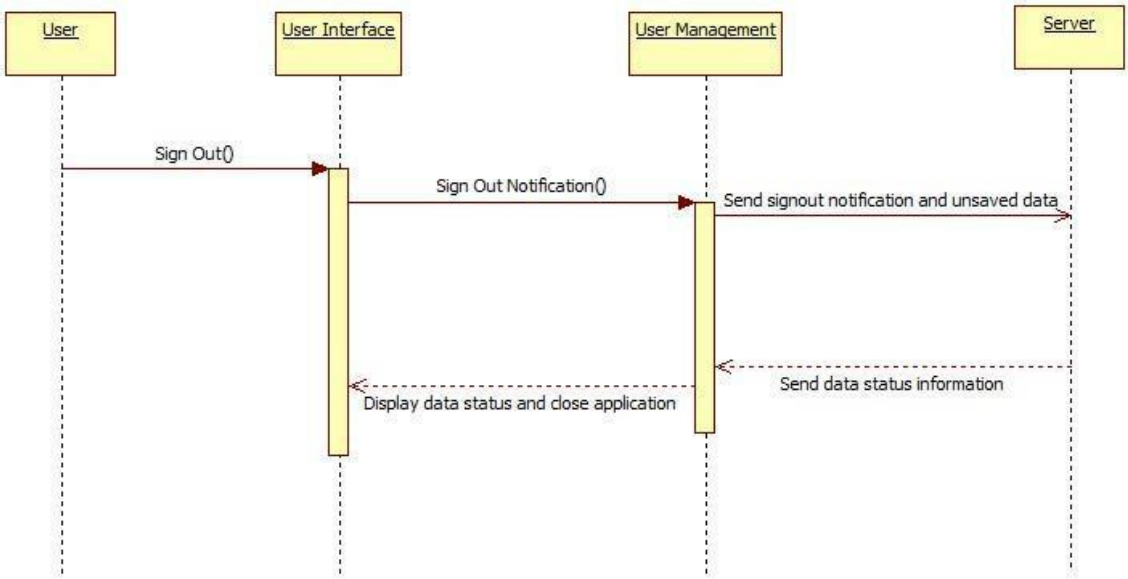
5.2.1 Sign up



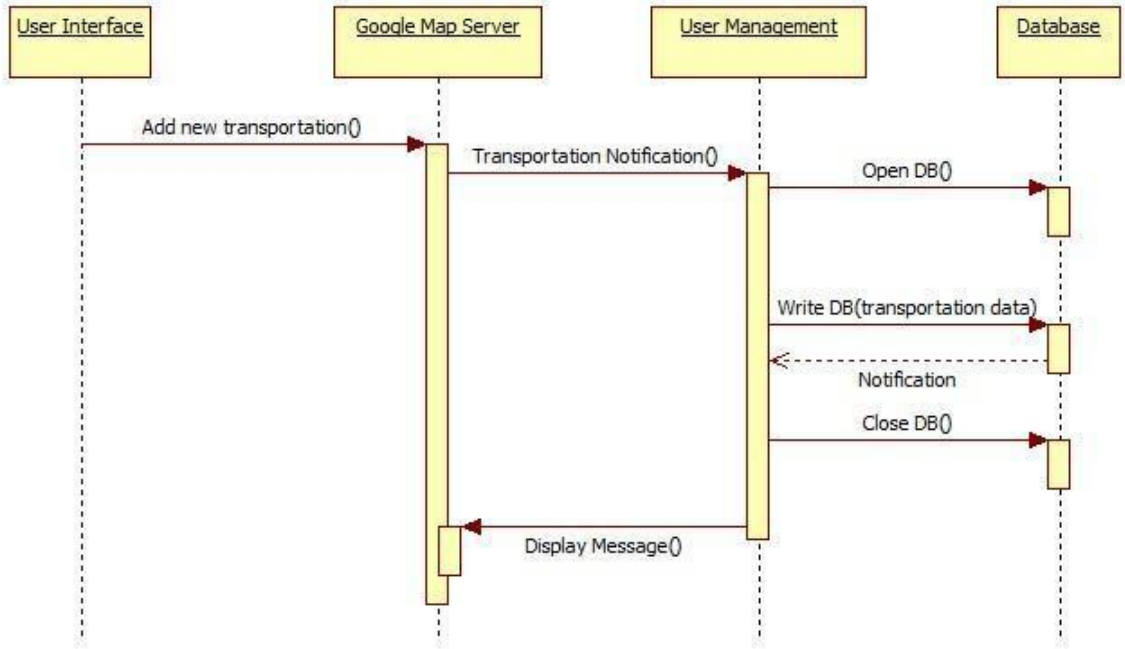
5.2.2 Sign in



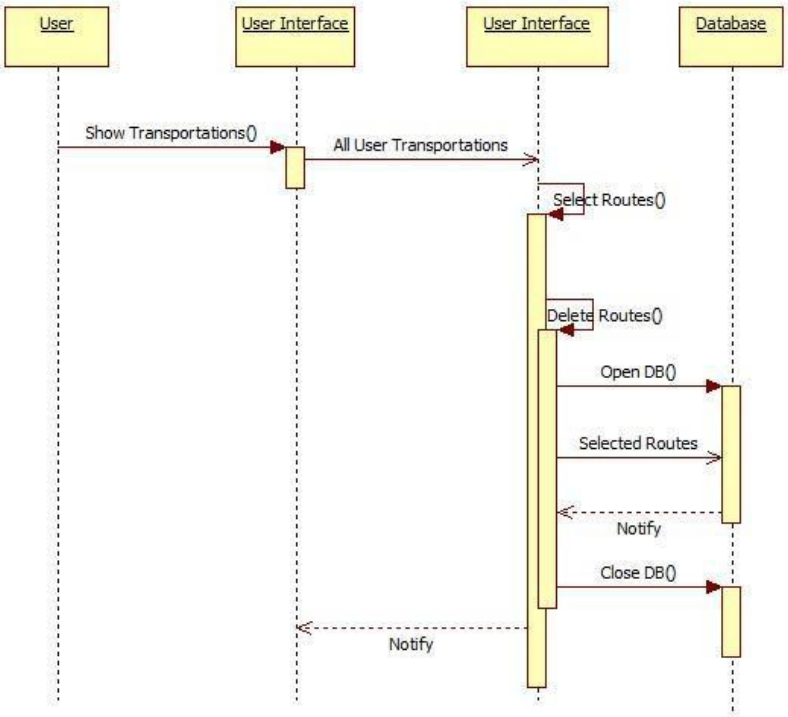
5.2.3 Sign out



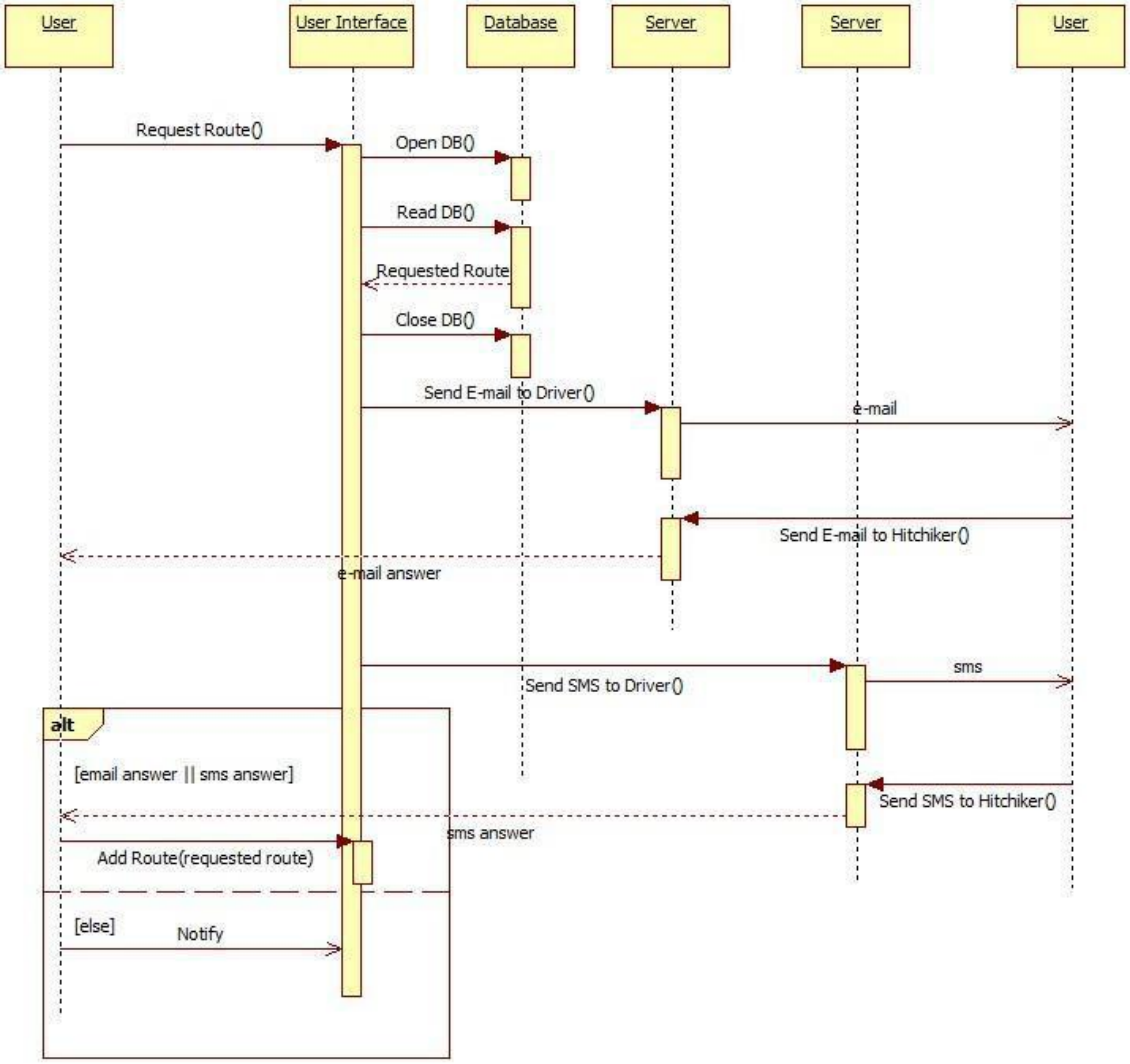
5.2.4 Add route



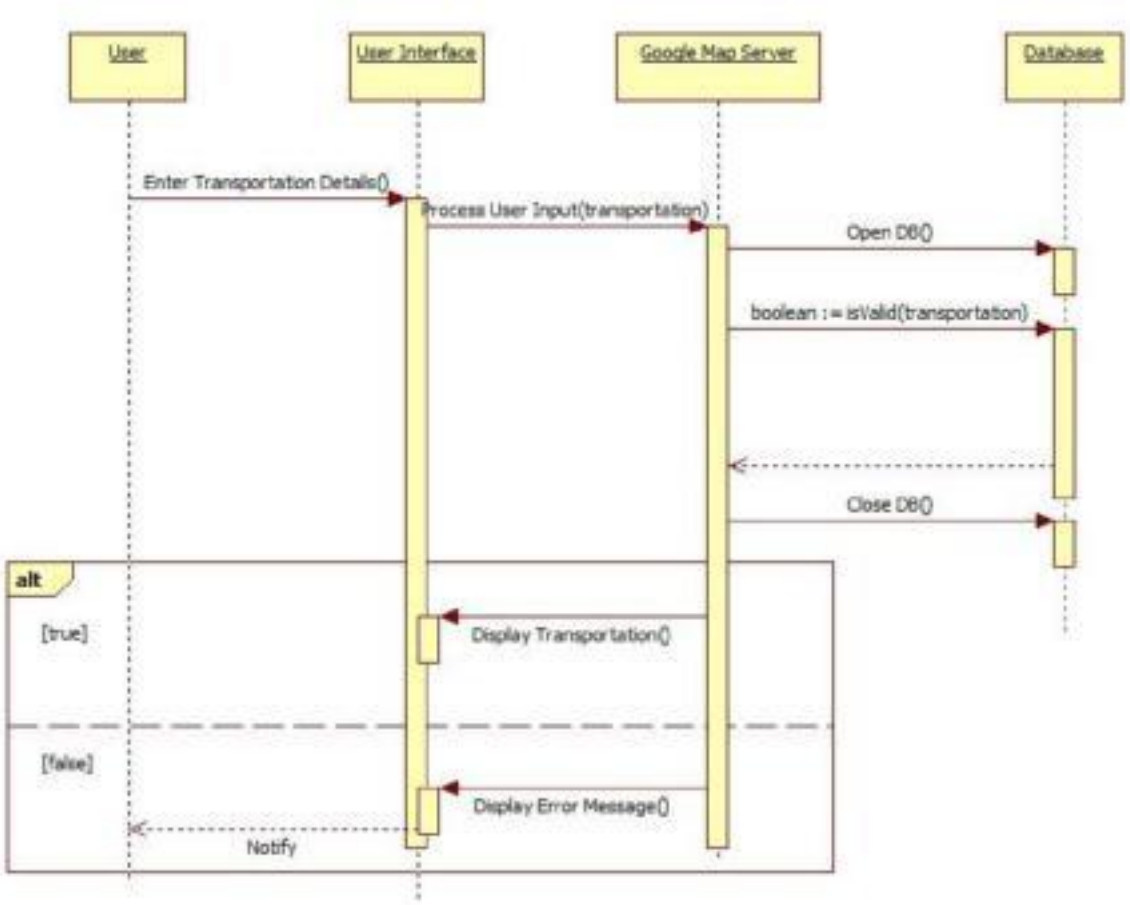
5.2.5 Delete Route



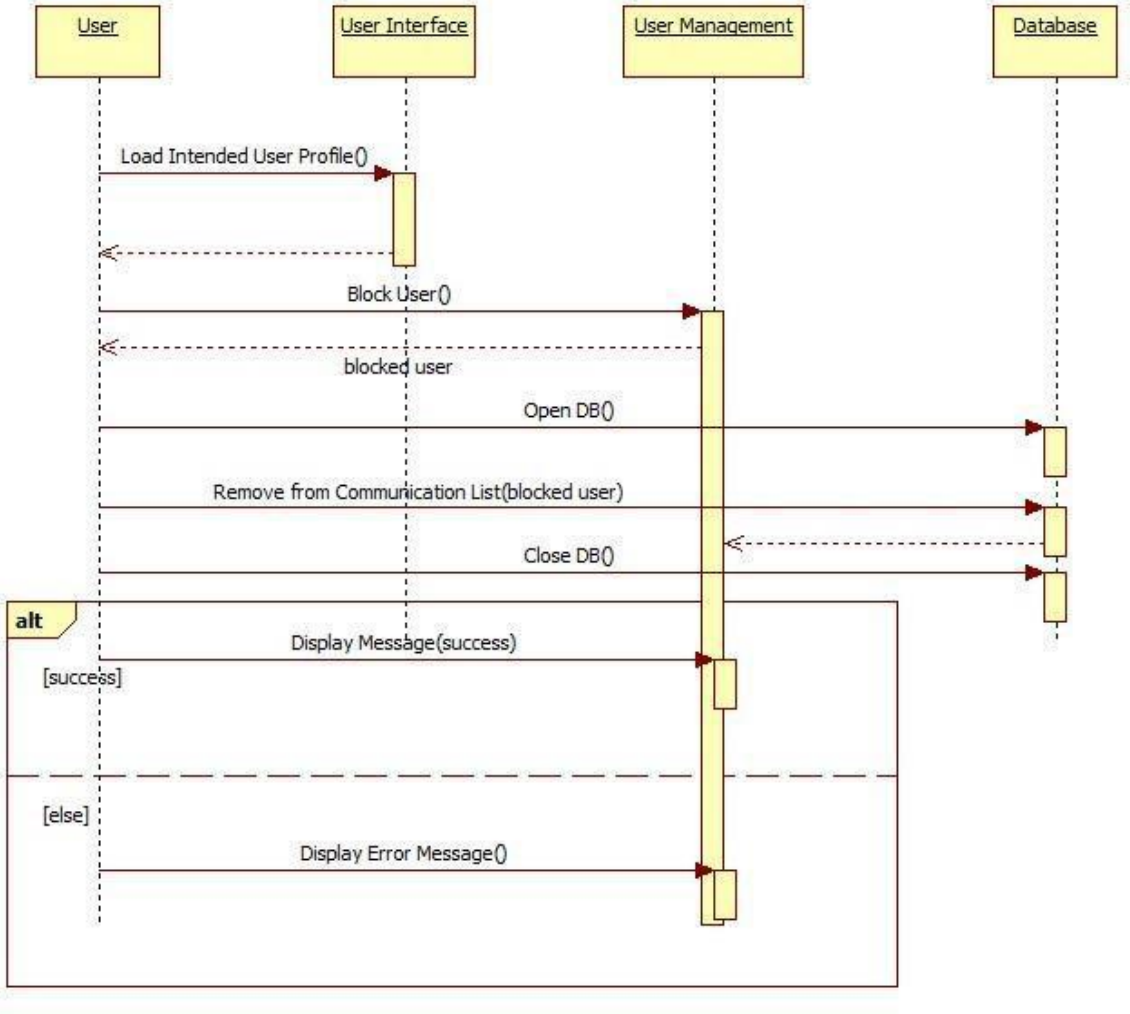
5.2.6 Request Ride



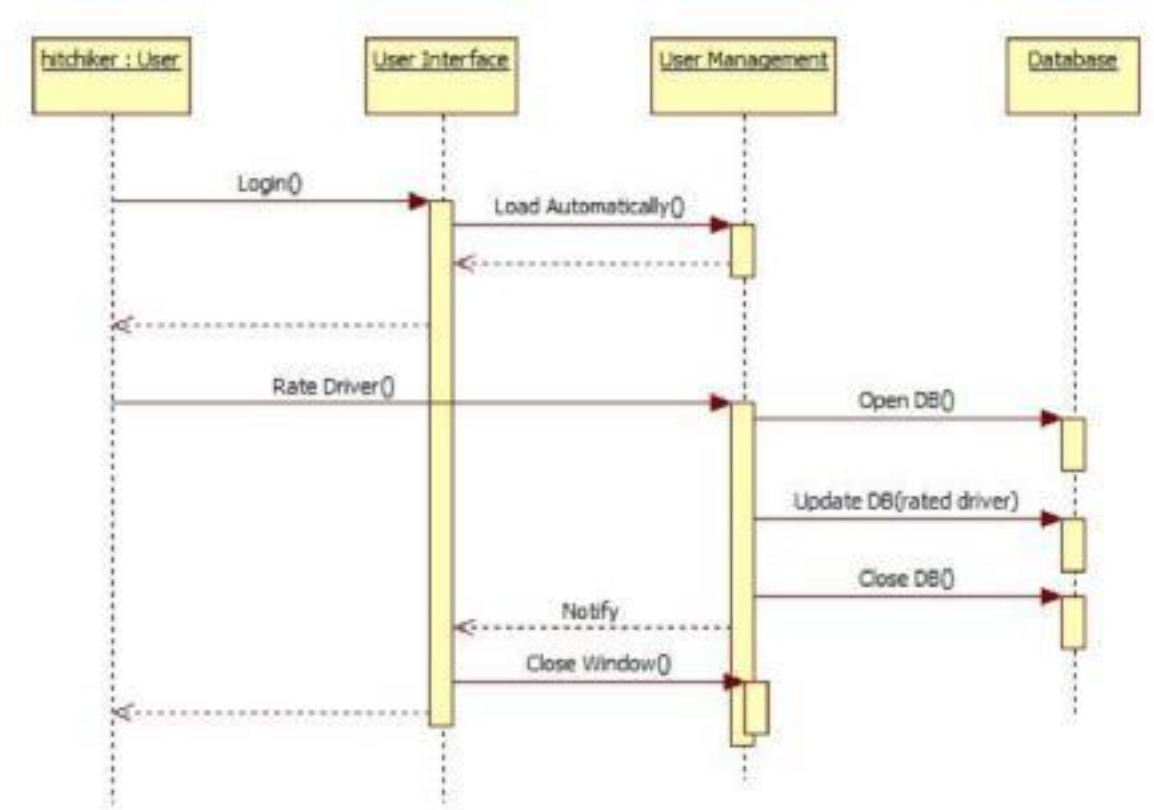
5.2.7 Search Ride



5.2.8 Block User

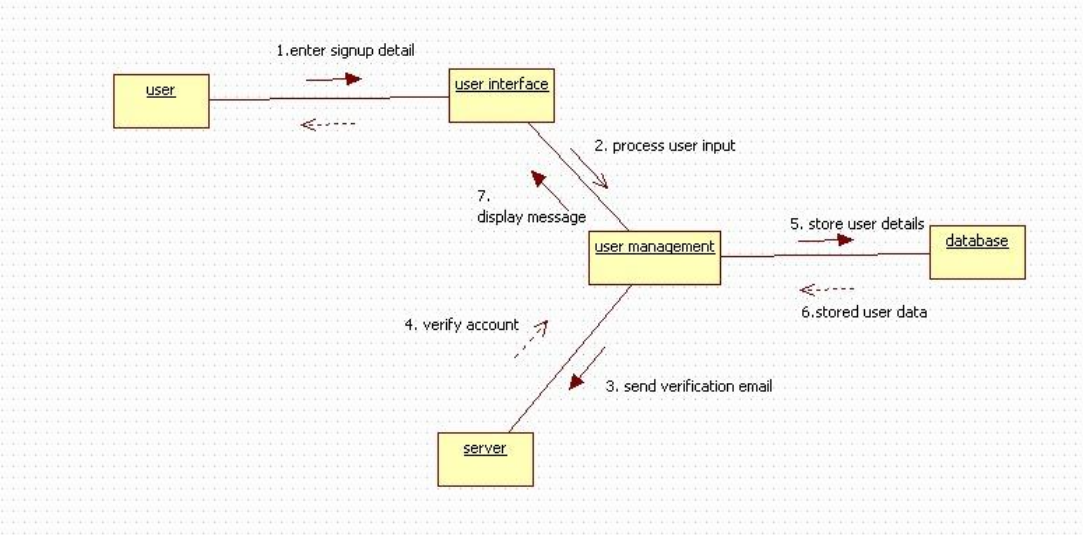


5.2.9 Rate User

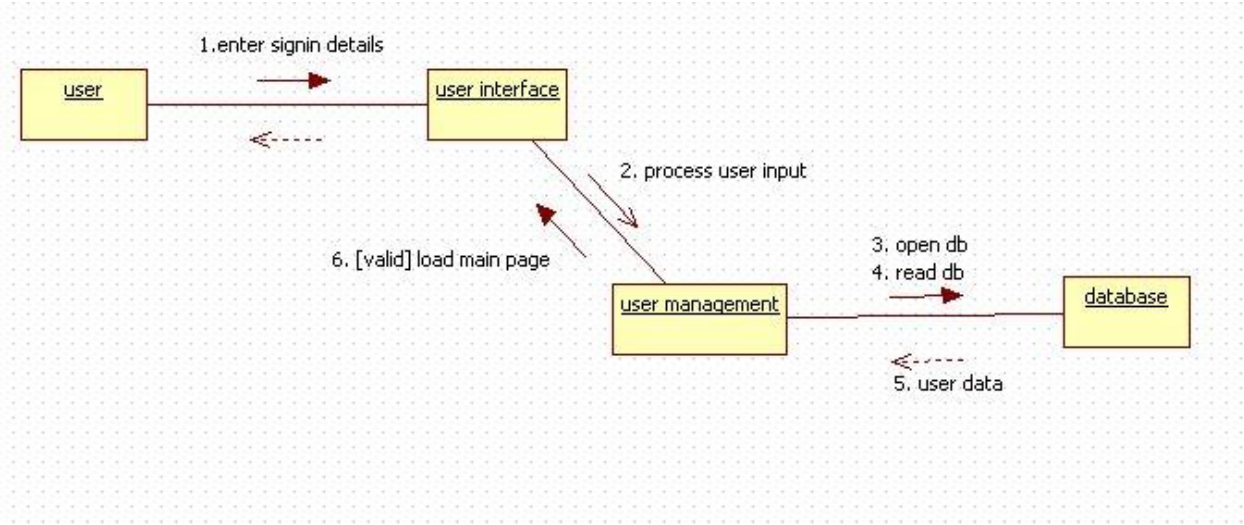


5.3 Collaboration Diagrams

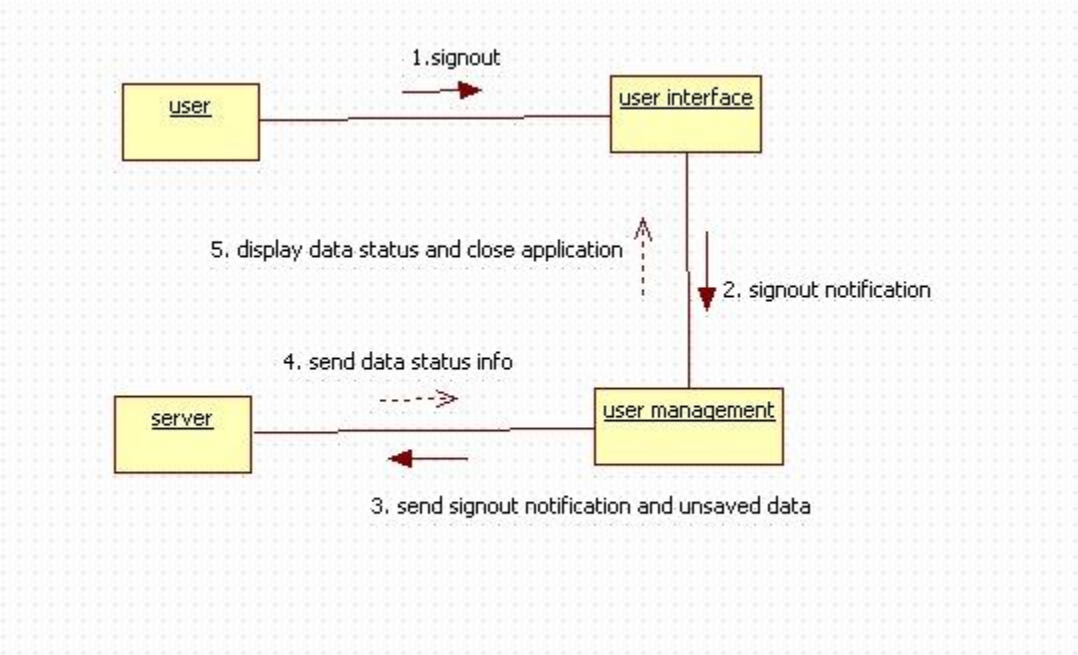
5.3.1 Sign up



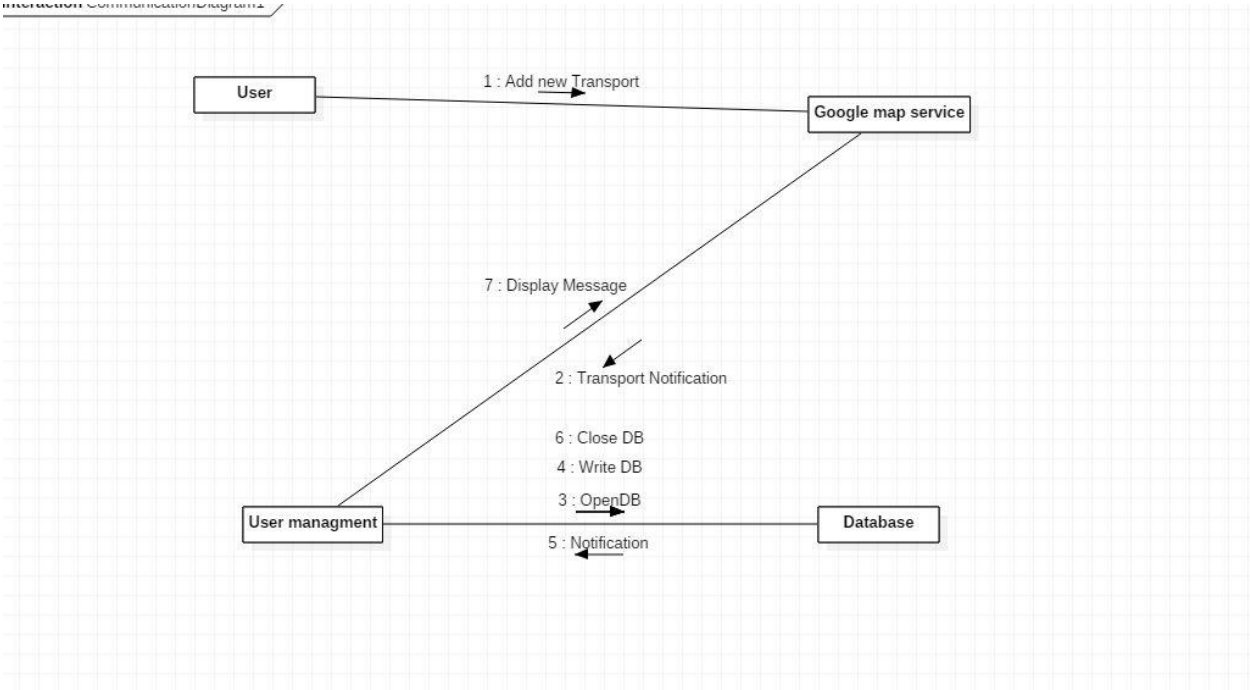
5.3.2 Sign in



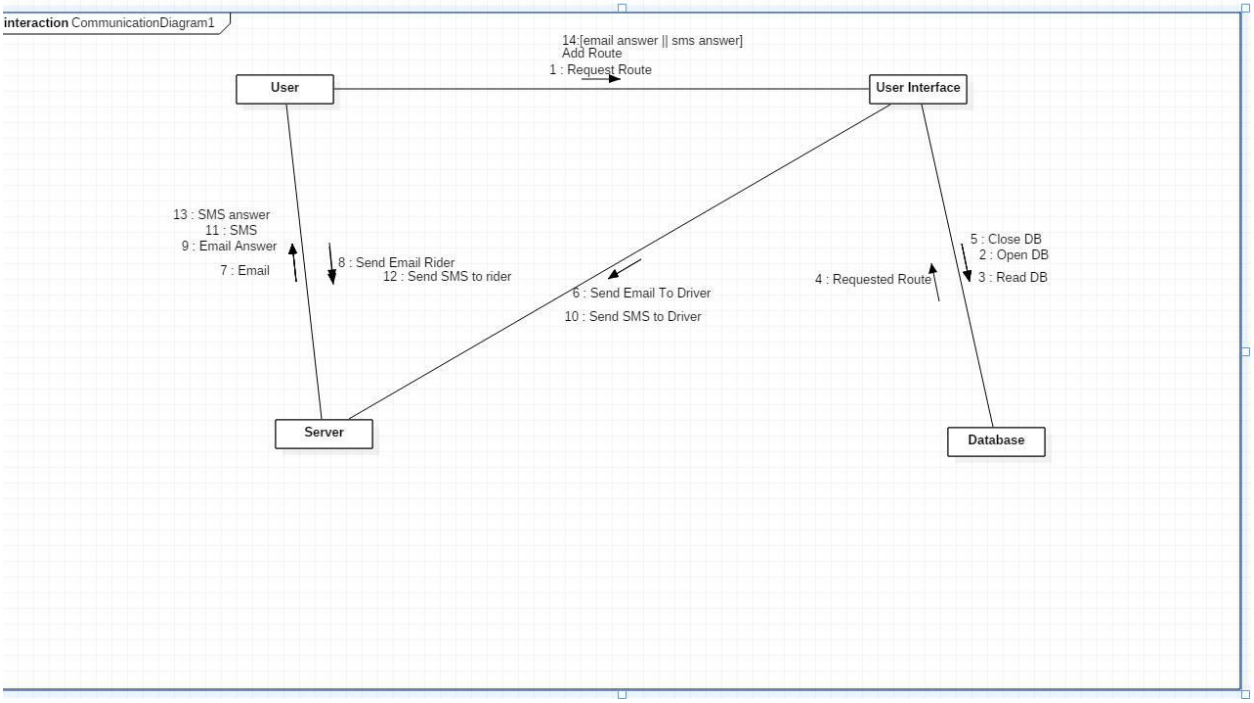
5.3.3 Sign out



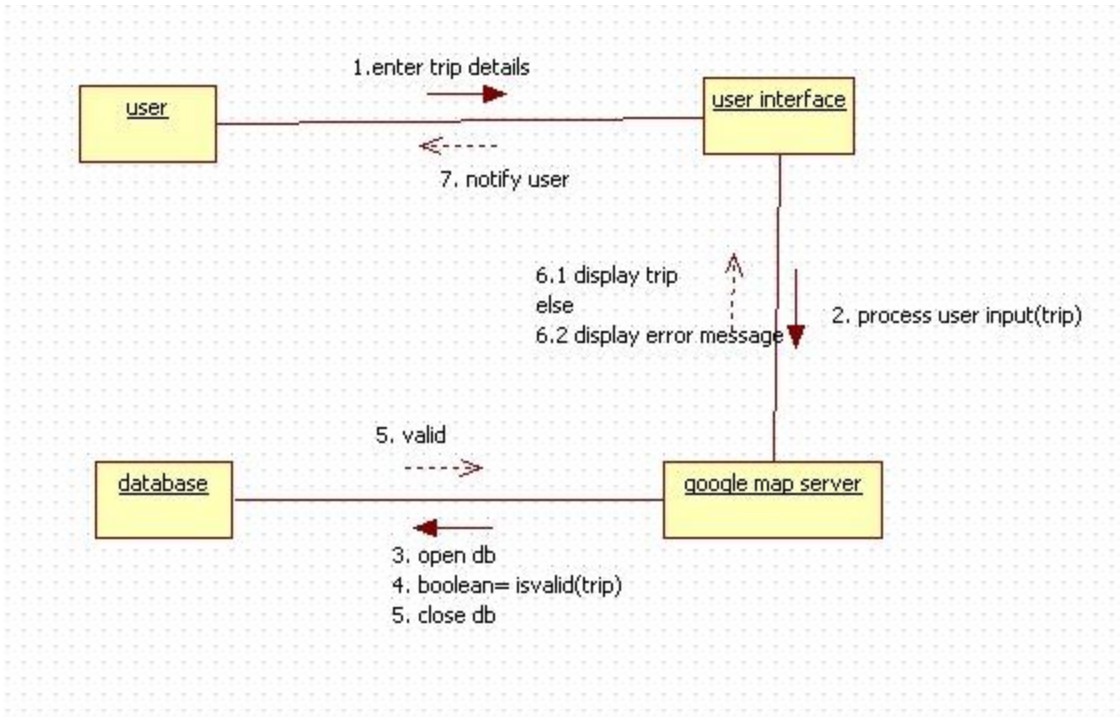
5.3.4 Add route



5.3.5 Request Ride



5.3.6 Search Ride



6. REFERENCES

6.1 References

[1] https://www.uber.com/
[2] https://www.careem.com/karachi/node
[3] https://tripda.com/
[4] https://www.rdvouz.com/
[5] IEEE STD 1233-1998, IEEE Guide for Developing System Design Specifications
[6] IEEE STD 830-1998, IEEE Recommended Practice for Software Design Specifications