

# FASTPOOL

A Carpooling Application designed especially for FASTians.

## Software Requirements 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

## Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Sauman Balkhi	Co-Lead Software Eng.	6-11-2016
	Akif Jawed	Co-Lead Software Eng.	6-11-2016
	Hammad Khan	Co-Lead Software Eng.	6-11-2016
	Ma'am Amber Khan	Instructor, CS 309	6-11-2016

## Contents

Revision History.....	ii
Document Approval .....	ii
1. Introduction.....	1
1.1 Purpose .....	1
1.2 Scope.....	1
1.3 Definitions, Acronyms, and Abbreviations.....	2
1.4 References.....	2
1.5 Overview .....	2
2. General Description.....	3
2.1 Product Perspective .....	3
2.2 Product Functions .....	3
2.3 General Constraints .....	4
2.5 Assumptions and Dependencies.....	4
3. Specific Requirements .....	5
3.1 External Interface Requirements.....	5
3.1.1 User Interfaces.....	5
3.1.2 Hardware Interfaces.....	5
3.1.3 Software Interfaces.....	5
3.1.4 Communications Interfaces.....	5
3.2 Functional Requirements .....	6
3.2.1. Sign Up .....	6
3.2.2. Sign In.....	7
3.2.3. Sign Out .....	8
3.2.4. Add Transportation Route .....	8
3.2.5. Delete Transportation Route .....	9
3.2.6. Request Ride.....	9
3.2.7. Search Ride .....	10
3.2.8. Block Rider .....	10
3.2.9. Rate User .....	10
3.3 Classes / Objects .....	11
3.4 Design Constraints .....	11
3.5 Non-Functional Requirements .....	12

3.5.1 Performance .....	12
3.5.2 Reliability .....	12
3.5.3 Availability .....	12
3.5.4 Security .....	12
NU mail will be used for security and verification.....	12
4. Analysis Models.....	13
4.1 Sequence Diagrams.....	13
4.1.1 Sign up.....	13
4.1.2 Sign in .....	14
4.1.3 Sign out.....	14
4.1.4 Add Route .....	15
4.1.5 Delete Route .....	15
4.1.6 Request Ride .....	16
4.1.7 Search Ride .....	17
4.1.8 Block User .....	18
4.1.9 Rate User .....	19
4.2 Collaboration Diagrams.....	20
4.2.1 Sign up.....	20
4.2.2 Sign in .....	20
4.2.3 Sign out.....	21
4.2.4 Add route .....	21
4.2.5 Request Ride.....	22
4.2.6 Search Ride .....	22
5. Change Management Process.....	23

# 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 Definitions, Acronyms, and Abbreviations

GUI	Graphical User Interface
DBMS	Database Management System
IEEE	Institute of Electrical and Electronics Engineers
SRS	System Requirements Specification
API	Application Programming Interface
PHP	Hypertext Preprocessor
SMS	Short Message Service
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart

### 1.4 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 Requirements Specifications
- [6] IEEE STD 830-1998, IEEE Recommended Practice for Software Requirements Specifications

### 1.5 Overview

The rest of the document contains overall description of the system which includes interface properties, product functions and dependencies. In addition, it contains system specific requirements which composed of functional and nonfunctional requirements. Moreover, there will be data and description models of the system and these models are specified with diagrams such as use cases. And finally at the end of the document, there is conclusion part which explains the overall description about the system. SRS is organized according to the table of content.

## 2. General Description

This section gives background information about specific requirements of the web based carpool environment to be developed in brief. Although we will not describe every requirement in detail, this section will describe the factors that affect the final product.

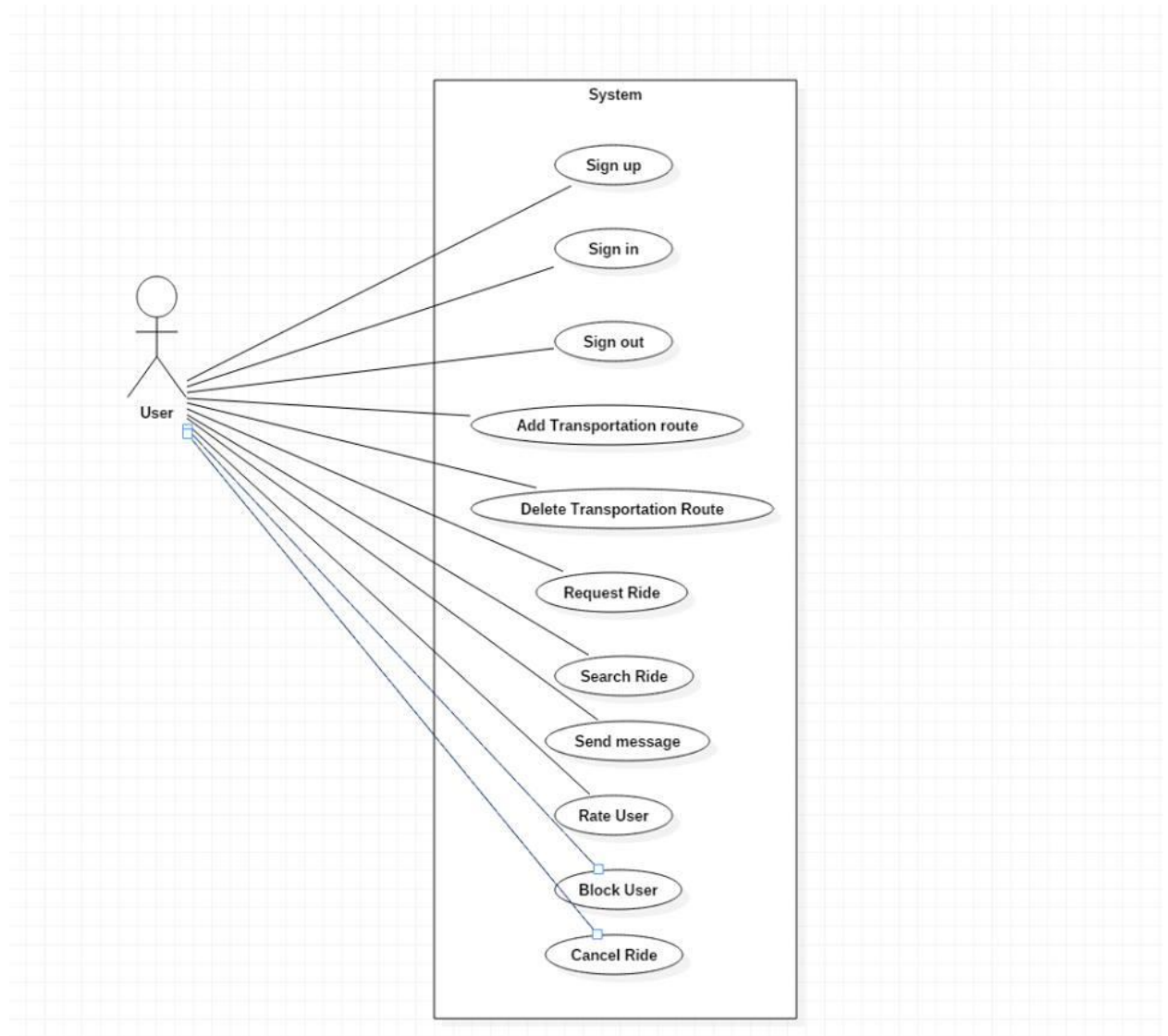
### 2.1 Product Perspective

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.

### 2.2 Product Functions

All use cases are explained below:

- Sign Up: Users need to sign up to use the system. The users should have a username and password. After filling their name, surname, email, age, job, phone and gender information, they register the system.
- Sign In: If a user is signed up, s/he can sign in the system by filling username and password boxes.
- Sign Out: A user may need to sign out the system. S/he can do it by clicking the sign out button which is placed in every page.
- Add Transportation Route: Users may add transportations by specifying a route, time/time period and number of empty seats. The user can select the route by two different way. The first way is entering start and end locations. Thus, the route is drawn on the map. The other way is selecting start and end locations on the map.
- Delete Transportation Route: A user may delete his/her transportation route. After deleting route, other passengers in that transportation will be informed by the system.
- Request Ride: A user may use a transportation route by sending ride request to the driver.
- Search Ride: A user can search for rides that the user can see suitable routes to his/her route by specifying time and route.
- Cancel Ride: The rider or driver can cancel ride if they face any problem.
- Send Message: The users communicate each other by sending message.
- Block user: The driver can block the rider if he/she faces any inconvenience.
- Rate User: After a trip, the rider can rate the driver.



### 2.3 General 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

User interface and some functionalities can change during the development process of project. And also new functionalities can be added which is able to change the dependent system requirements.



### 3. Specific Requirements

This section is divided into External Interface Requirement and Functional Requirements.

#### 3.1 External Interface Requirements

##### 3.1.1 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.

##### 3.1.2 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.

##### 3.1.3 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.

##### 3.1.4 Communications Interfaces

The system shall send automatic verification e-mail to the user who wants to register to the system. For communication between users, system shall support SMS functionality and users can be able to send and receive SMS through the remote mobile devices.

## 3.2 Functional Requirements

### 3.2.1. Sign Up

Use Case ID	UC1
Actor(s)	User
Description	User Sign In (Register)
Preconditions	No precondition
Post conditions	User will be able to log in to the system
Precedence	Mandatory
Normal flow of event	<ol style="list-style-type: none"> <li>1. The user opens the browser and types the address of the system.</li> <li>2. User presses the sig up button.</li> <li>3. User enters her or his user name, surname, and password and e-mail information.</li> <li>4. User checks his or her e-mail account to verify his or her user information.</li> </ol>
Alternative Flow(s)	<p>Flow 1:</p> <ol style="list-style-type: none"> <li>1. If one of the required fields (user name, user surname, password, e-mail) in sign up page are not filled properly, the warning message will be shown by the system.</li> </ol> <p>Flow 2:</p> <ol style="list-style-type: none"> <li>1. If all the required fields are properly filled, the user will be redirected to the main page of the system.</li> </ol>

## 3.2.2. Sign In

Use Case ID	UC2
Actor(s)	User
Description	User Log In
Preconditions	The user shall be able to sign in to the system
Post conditions	User will be able to use the system
Precedence	Mandatory
Normal flow of event	<ol style="list-style-type: none"> <li>1. The user opens the browser and types the address of the system.</li> <li>2. User presses the log in button.</li> <li>3. User enters her or his user name and password.</li> <li>4. 4. If the user forget his or her account information, he or she get account information via the “forgot your password?” panel under the log in page.</li> </ol>
Alternative Flow(s)	<p>Flow 1:</p> <ol style="list-style-type: none"> <li>1. If the user enter wrong username or password information, the warning message for example “Wrong username and password information” will be shown to the user.</li> </ol> <p>Flow 2:</p> <ol style="list-style-type: none"> <li>2. 2. If the user enters his or her user name and password information correctly, user will be redirected to the application relevant page of the system.</li> </ol>

### 3.2.3. Sign Out

Use Case ID	UC3
Actor(s)	User
Description	User Log Out
Preconditions	The user shall be able to log out from the system.
Post conditions	User will be able to leave the system
Precedence	Not mandatory
Normal flow of event	<ol style="list-style-type: none"> <li>1. User presses the log out button.</li> <li>2. User leaves the system.</li> <li>3. The system's main page will be loaded.</li> </ol>

### 3.2.4. Add Transportation Route

Use Case ID	UC4
Actor(s)	Driver
Description	Driver shall be able to add route from the map.
Preconditions	The driver shall be able to sign in to the system.
Post conditions	Driver shall be retrieve ride requests from the other users.
Precedence	Mandatory
Normal flow of event	<ol style="list-style-type: none"> <li>1. Driver shall enter her or his profile page.</li> <li>2. User shall press add new transportation button.</li> <li>3. Driver route page will be loaded.</li> <li>4. Driver enters departure time, available seats and iteration of transportation like "one time" or "periodic".</li> <li>5. Driver draws a route on the map panel.</li> </ol>
Alternative Flow(s)	<p>Flow 1:</p> <ol style="list-style-type: none"> <li>1. The user clicks the radio button of "one time".</li> <li>2. User enters date information in terms of day, month and year.</li> </ol> <p>Flow 2:</p> <ol style="list-style-type: none"> <li>1. The user clicks the radio button of "periodic".</li> <li>2. User selects days from the week day check boxes.</li> </ol>

### 3.2.5. Delete Transportation Route

Use Case ID	UC5
Actor(s)	Driver
Description	Driver shall be able to delete route.
Preconditions	Route added previously.
Post conditions	Route dose not exists anymore
Precedence	Not mandatory
Normal flow of event	<ol style="list-style-type: none"> <li>1. Driver shall presses my transportations button.</li> <li>2. Driver route page will be loaded.</li> <li>3. Driver selects the route he or she wants to delete.</li> <li>4. Delete button is clicked.</li> <li>5. The Driver deletes the route.</li> </ol>

### 3.2.6. Request Ride

Use Case ID	UC6
Actor(s)	Rider
Description	Rider shall be able to request route.
Preconditions	The Rider should log in.
Post conditions	The Rider shall be able to book his ride.
Precedence	Not mandatory
Normal flow of event	<ol style="list-style-type: none"> <li>1. User presses the request route button.</li> <li>2. The driver receives the notification.</li> </ol>

### 3.2.7. Search Ride

Use Case ID	UC7
Actor(s)	Rider
Description	Rider can search route.
Preconditions	The driver should be logged in.
Post conditions	Rider will be able to select route from the available route list.
Precedence	Not mandatory
Normal flow of event	<ol style="list-style-type: none"> <li>1. Rider fills "from" input field.</li> <li>2. Rider fills "to" input field.</li> <li>3. Rider presses the search button.</li> </ol>

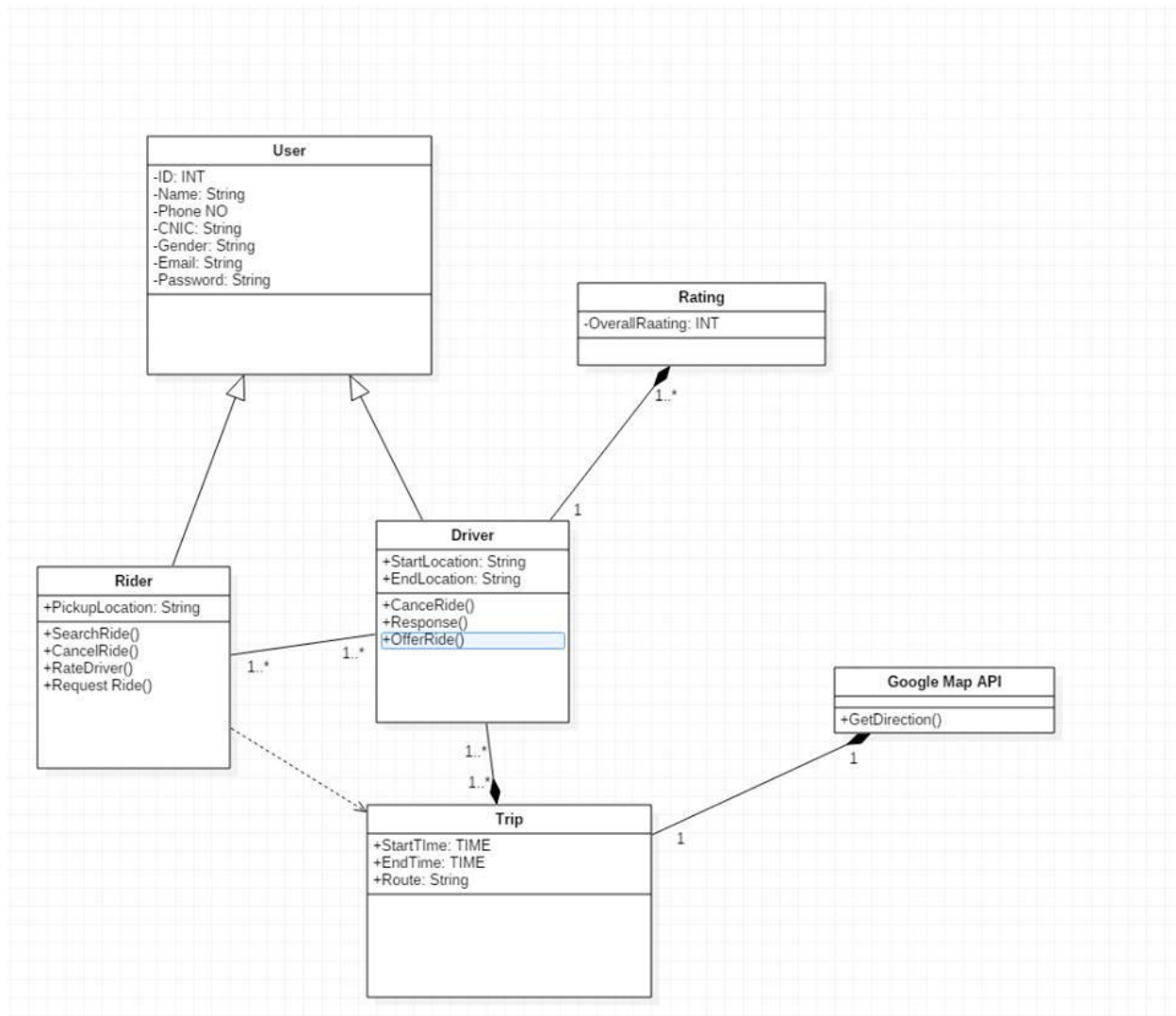
### 3.2.8. Block Rider

Use Case ID	UC8
Actor(s)	Driver
Description	Driver shall be able to block the rider through the system.
Preconditions	The driver should be logged in.
Post conditions	Rider will not be able contact the driver anymore.
Precedence	Not mandatory
Normal flow of event	<ol style="list-style-type: none"> <li>1. Driver enters the profile page of the rider who is intended to be blocked.</li> <li>2. Driver presses the block user button.</li> <li>3. The same page will be reloaded as marked as blocked.</li> </ol>

### 3.2.9. Rate User

Use Case ID	UC9
Actor(s)	Rider
Description	Rider should be able to rate the driver.
Preconditions	The ride should be completed with driver and rider.
Post conditions	The driver's rating will be updated.
Precedence	Not mandatory
Normal flow of event	<ol style="list-style-type: none"> <li>1. Rider clicks the star icon to rate driver's attitude.</li> </ol>

### 3.3 Classes / Objects



### 3.4 Design Constraints

Passwords of the user shall be encrypted in DBMS for security purposes. To prevent spam robots, the system has verification CAPTCHA module for security purposes. When the system crashes it will return back at most one hour in maintainability purposes.

The website will run on every browsers (Chrome, Safari, Mozilla Firefox etc.) and operating system (Windows, Linux, Mac Mavericks etc.). The mobile app will only run on Android.

### 3.5 Non-Functional Requirements

#### 3.5.1 Performance

The System requires normal speed internet.

#### 3.5.2 Reliability

The system will be very reliable.

#### 3.5.3 Availability

The availability will depend on the online database.

#### 3.5.4 Security

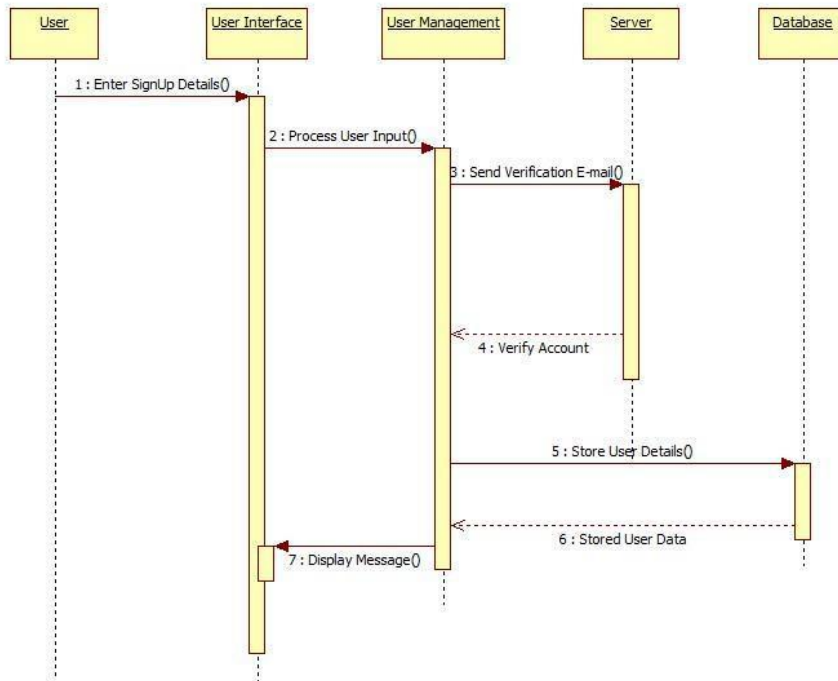
NU mail will be used for security and verification.



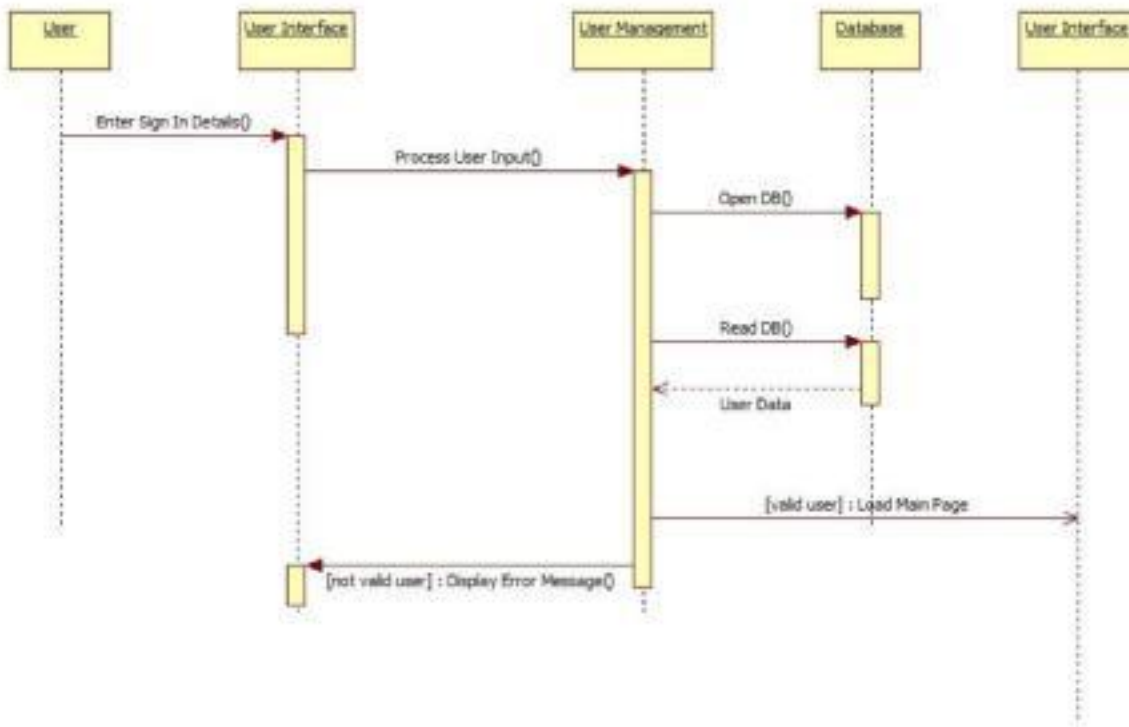
## 4. Analysis Models

### 4.1 Sequence Diagrams

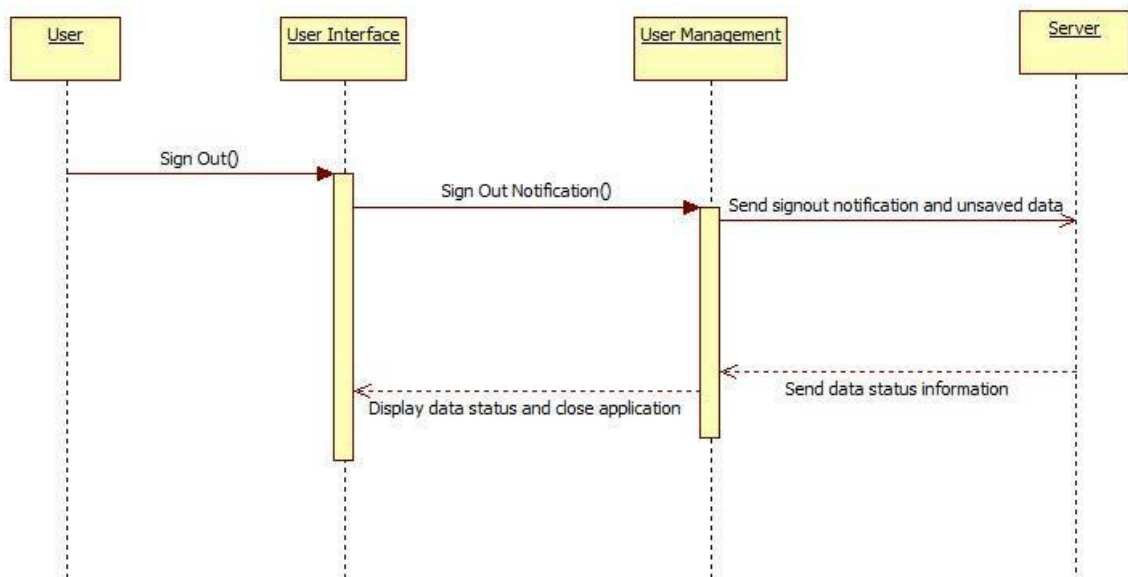
#### 4.1.1 Sign up



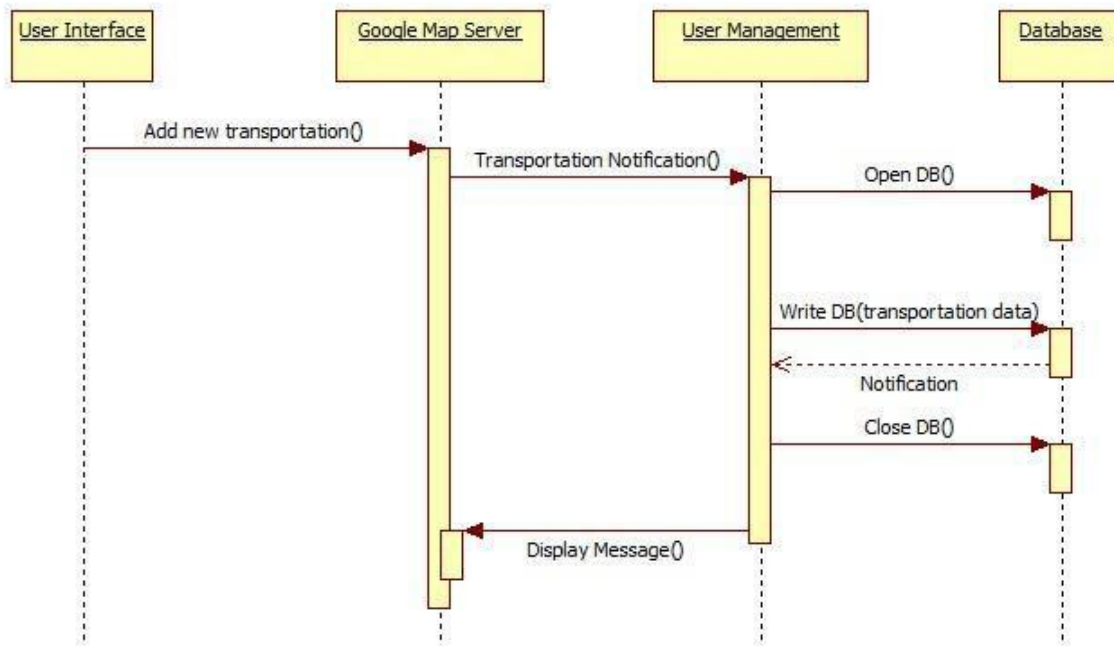
## 4.1.2 Sign in



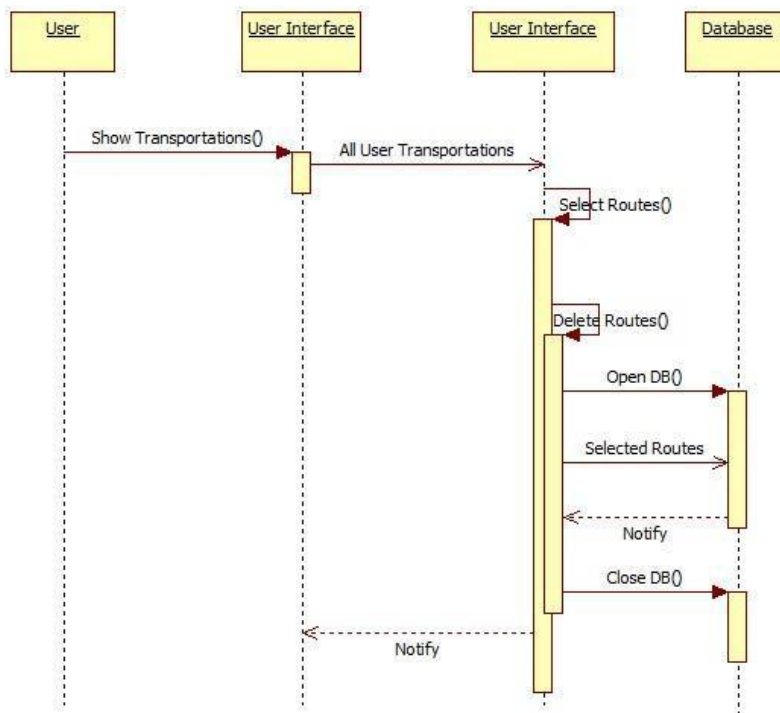
## 4.1.3 Sign out



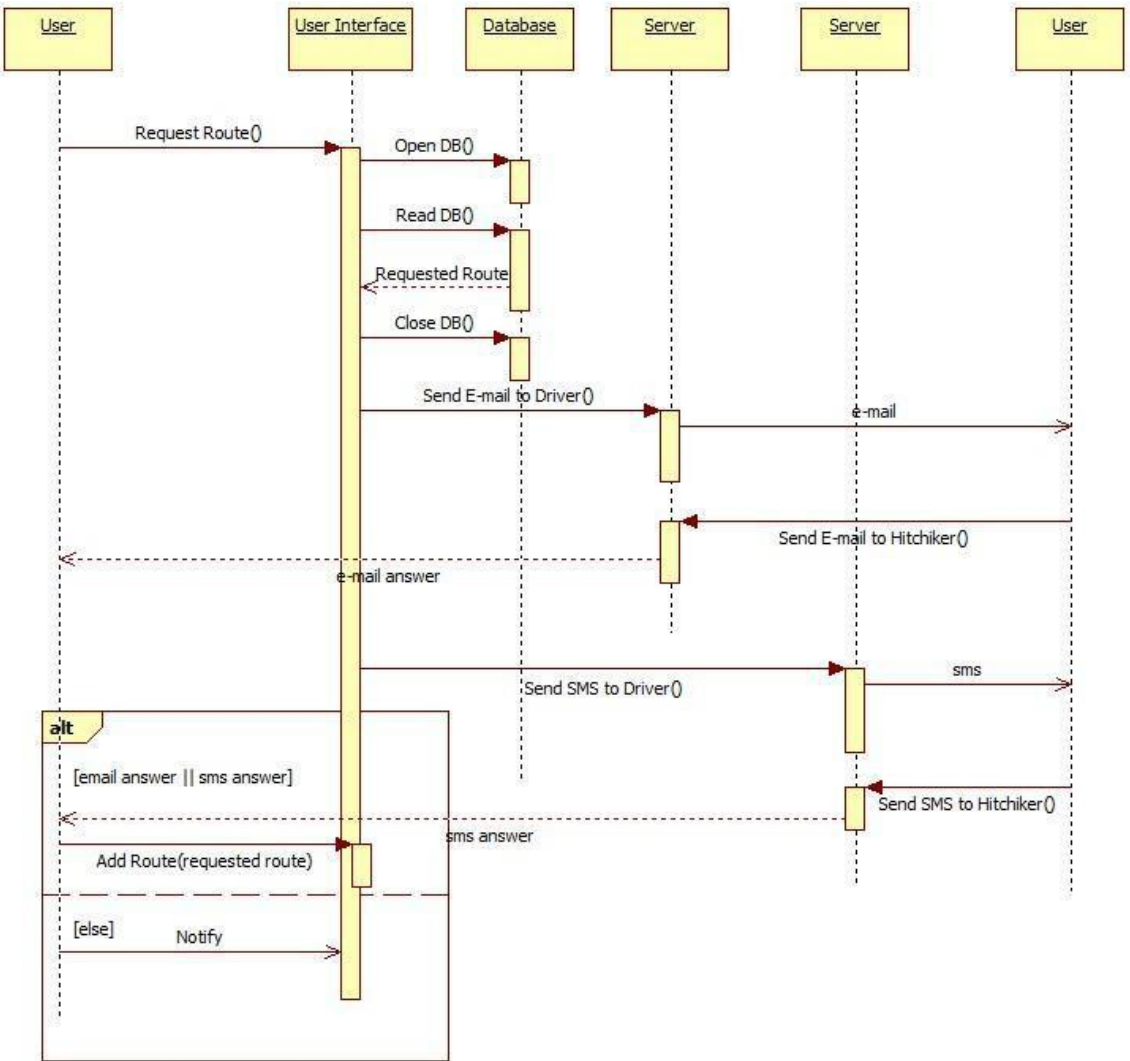
## 4.1.4 Add Route



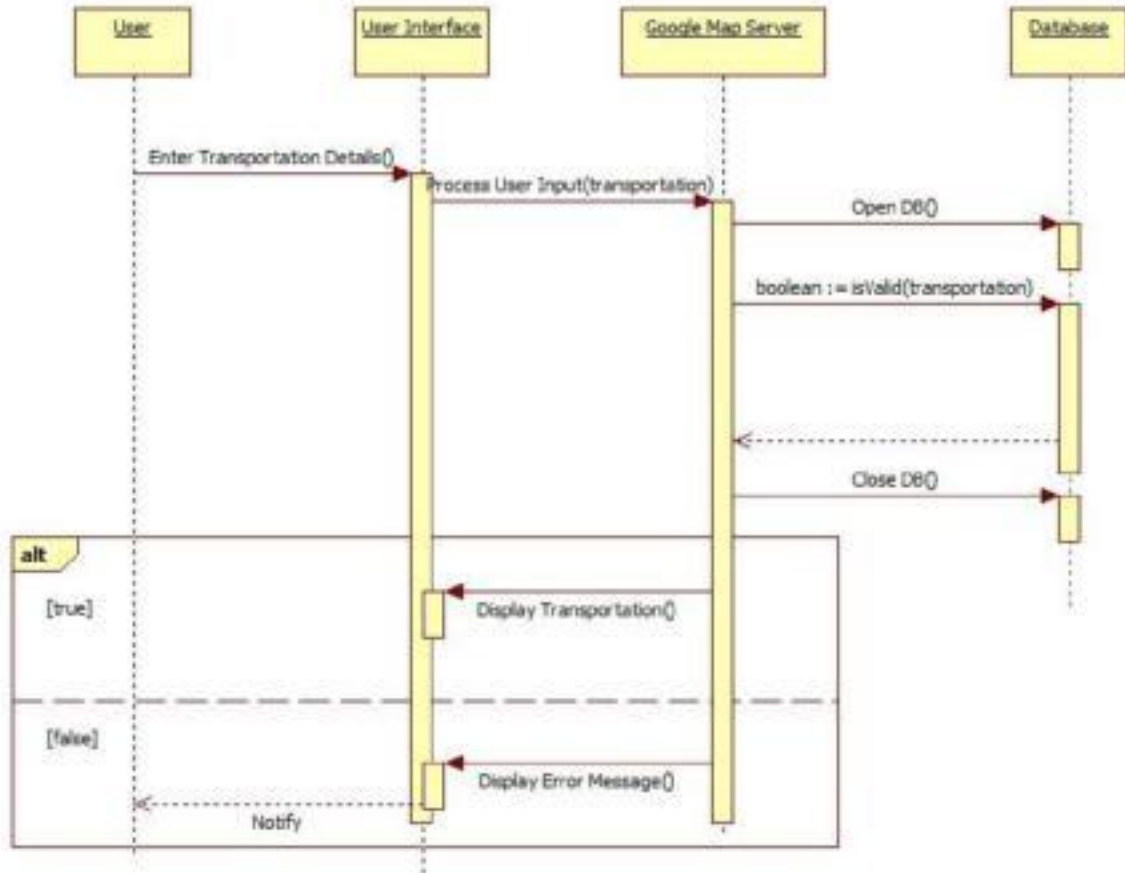
## 4.1.5 Delete Route



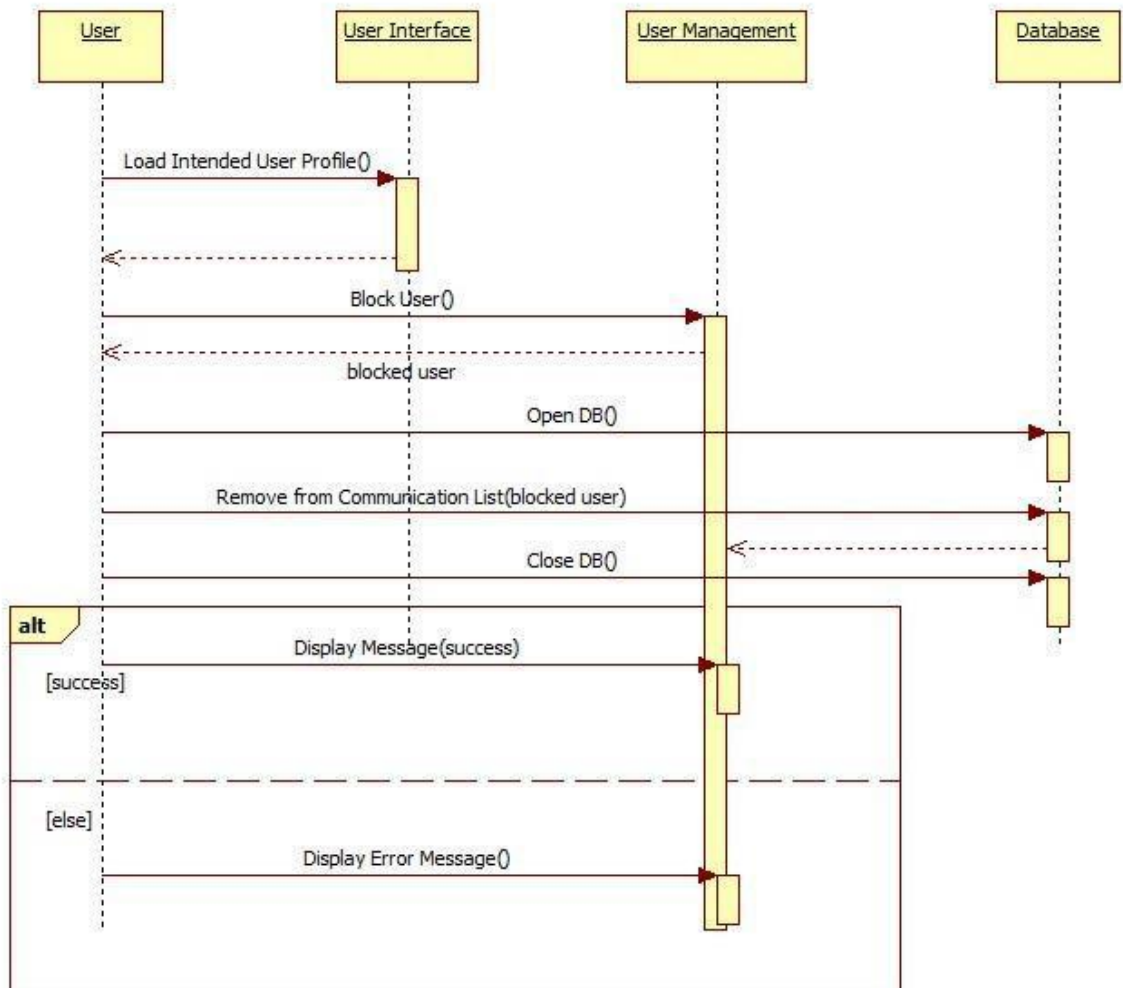
4.1.6 Request Ride



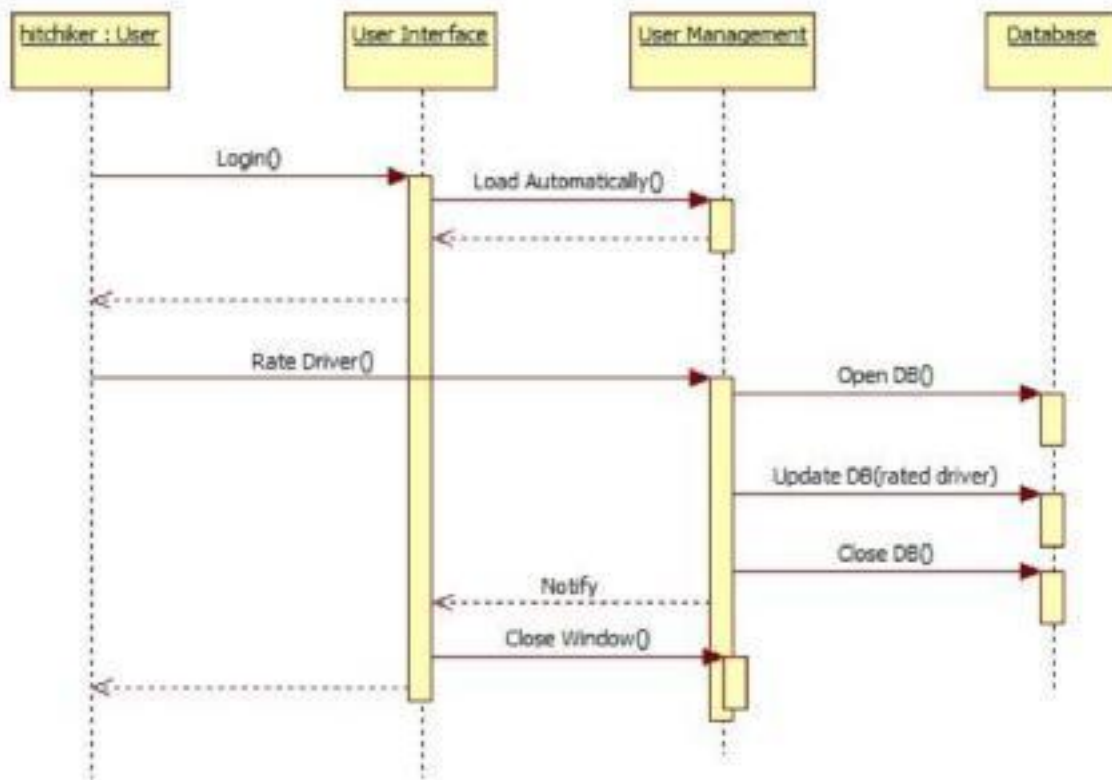
## 4.1.7 Search Ride



## 4.1.8 Block User

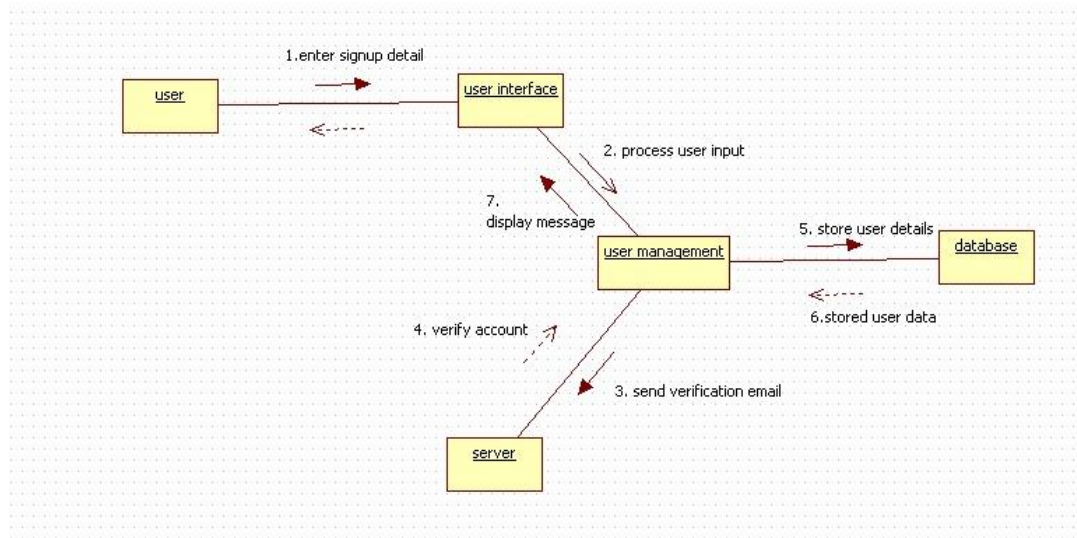


## 4.1.9 Rate User

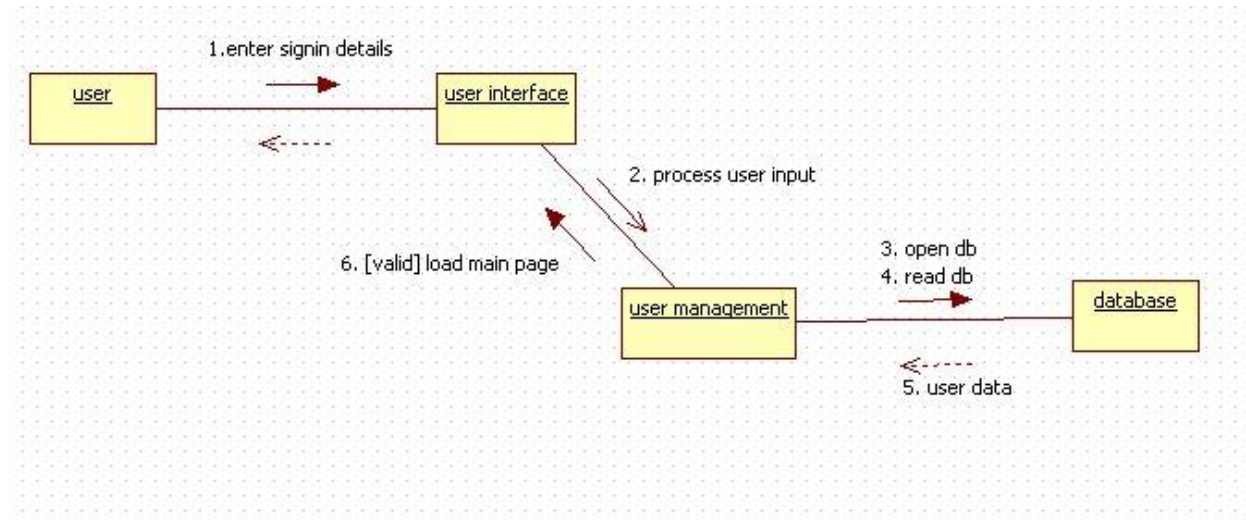


## 4.2 Collaboration Diagrams

### 4.2.1 Sign up

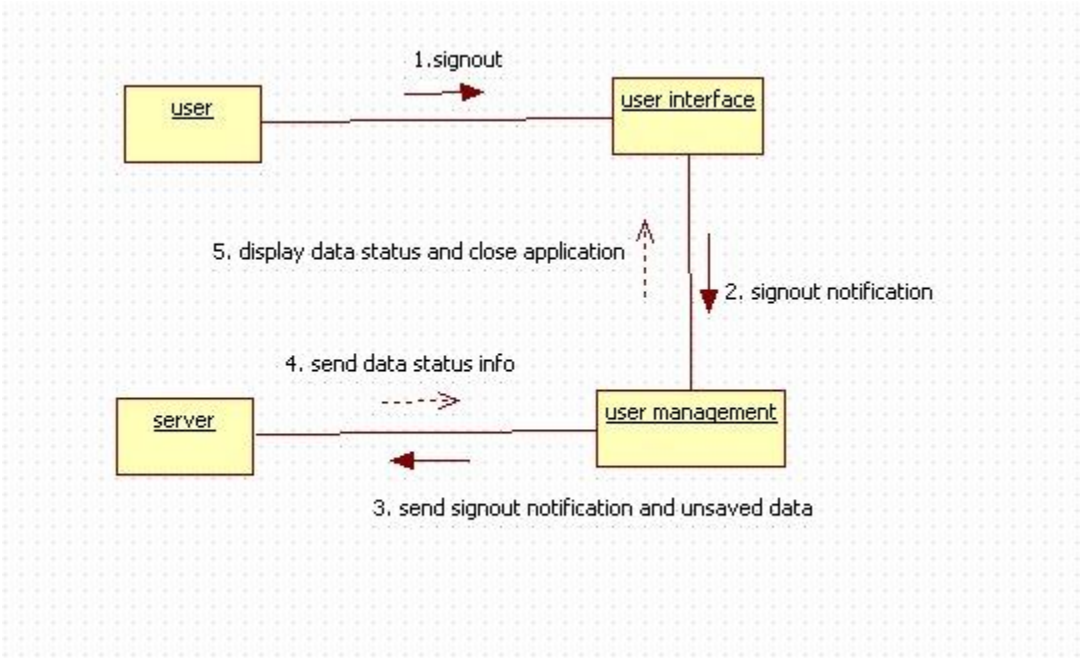


### 4.2.2 Sign in

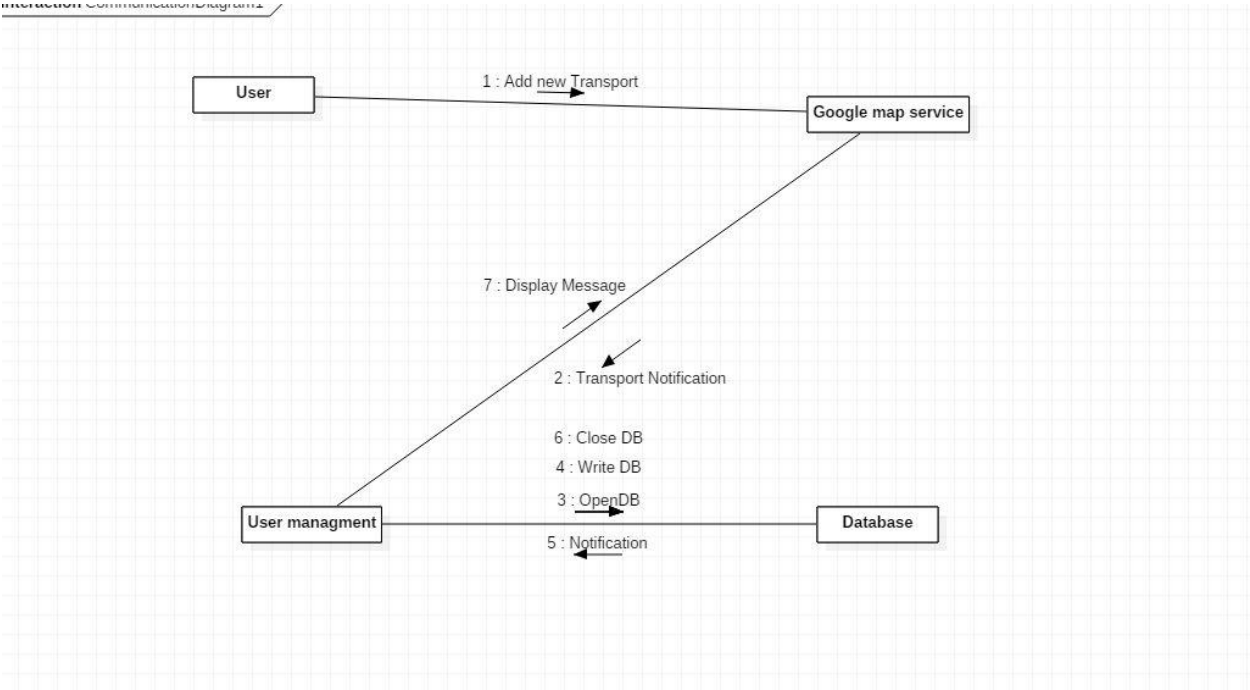




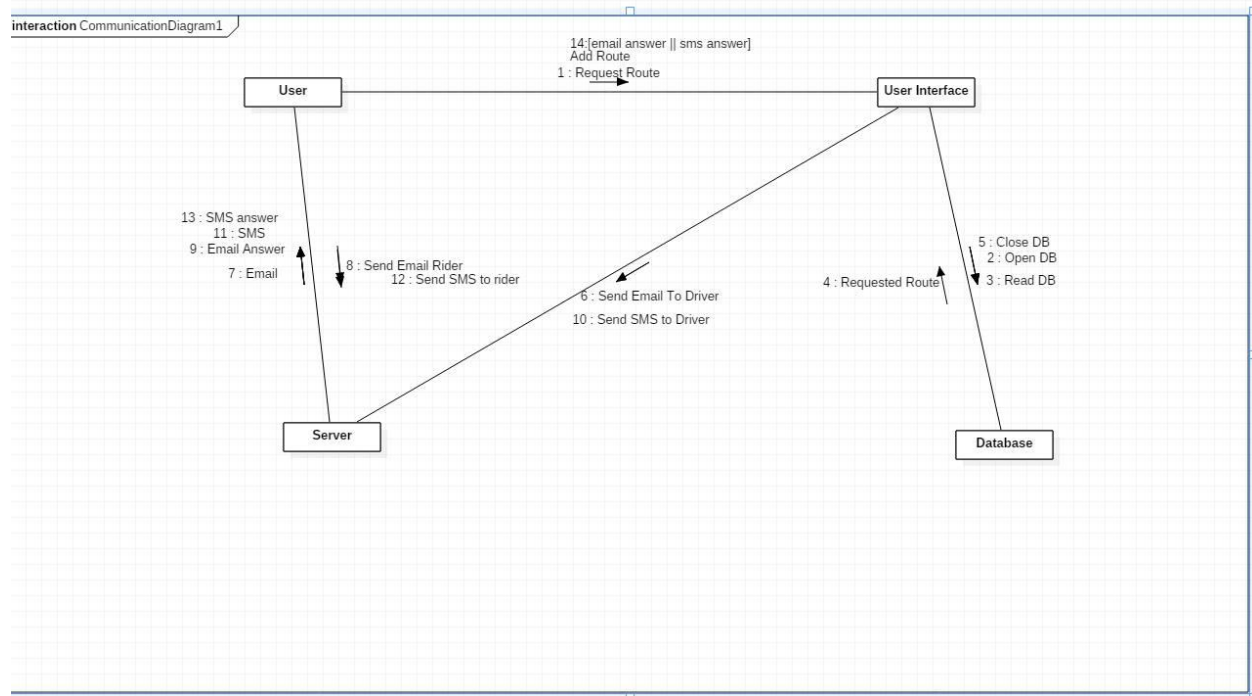
4.2.3 Sign out



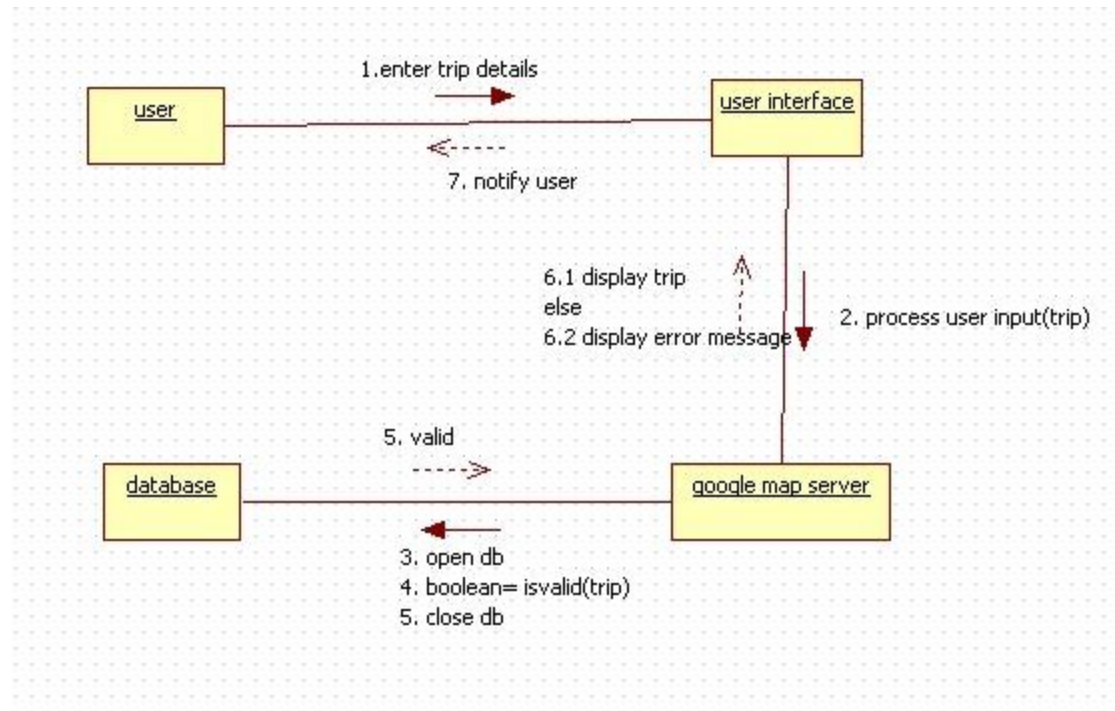
4.2.4 Add route



### 4.2.5 Request Ride



### 4.2.6 Search Ride



## 5. Change Management Process

Any update to current document will be notified to the concerned person as soon as possible through any mean possible.