



Bahir-Dar Institute of Technology

Faculty of Computing

Department of computer science

Requirement Design Document (RDD)

For

Industrial Project on ***Android Based Taxi Booking and Ordering System***

*Submitted to the faculty of computing in partial fulfillment of the requirements for  
the degree of Bachelor of Science in Computer science*

**Group members**

Name	ID Number
1. Abraham Tilahun-----	1010667
2. Bahriw Amare -----	1010586
3. Abdurohman Abtew .....	1010699
4. Nabil Nuredin .....	1010954

Advisor : **Mrs. Mieraf K.**

**June, 2013 E.C**

## **Declaration**

The Project is our own and has not been presented for a degree in any other university and all the sources of material used for the project have been duly acknowledged.

<b>Name</b>	<b>Signature</b>
1. Abraham Tilahun	_____
2. Bahriw Amare	_____
3. Abdurohman Abtew	_____
4. Nabil Muhammed	_____

**Faculty:** Computing

**Program:** Computer science

**Project Title:** Android based taxi booking and ordering system

This is to certify that I have read this project and that in my supervision and the students' performance, it is fully adequate, in scope and quality, as a project for the degree of Bachelor of Science.

<b>Name of Advisor</b>	<b>Signature</b>	
<b>Mrs. Mieraf K.</b>	_____	
<b>Examining committee members name</b>	<b>Signature</b>	<b>Date</b>
1._____	_____	_____
2._____	_____	_____
3._____	_____	_____

It is approved that this project has been written in compliance with the formatting rules laid down by the faculty.

## **Acknowledgment**

First and Foremost we would like to thank creature of the universe, almighty God for giving us the wisdom and strength we need to fulfil our duty. In addition, we would like to thank our Advisor Mrs. **Mieraf K** and **Rahel M.** for their valuable comments, support, and guidance throughout this project work. Without their encouragement and constant guidance, we could not have finished this project.

We also wish to express our sincere gratitude to the Faculty of Computing, for providing as an opportunity to perform the project work on “android based taxi booking and ordering system.”

Furthermore, it is a great pleasure to get an opportunity to thank our families for supporting and teaching us what is important in life. Finally, we would like to thank our colleagues and friends for their encouragement and support throughout the project.

## **List of acronym**

**UI** =User Interface

**UML**=Unified Modeling Language

**UC**=Use Case

**BR**=Business Rule

## List of Figures

Figure 1 system use case .....	18
Figure 2 user interface prototype diagram.....	35
Figure 3 activity diagram for login.....	36
Figure 4 activity diagram for taxi registration.....	37
Figure 5 activity diagram for remove taxi.....	38
Figure 6 activity diagram for update taxi.....	39
Figure 7 activity diagram for make booking.....	40
Figure 8 activity diagram for update booking .....	41
Figure 9 activity diagram for cancel booking.....	42
Figure 10 activity diagram for driver registration .....	43
Figure 11 sequence diagram for login.....	44
Figure 12 sequence diagram for Taxi registration.....	45
Figure 13 sequence diagram for Make booking.....	46
Figure 14 sequence diagram for cancel booking.....	47
Figure 15 sequence diagram for update booking.....	48
Figure 16 sequence diagram for update taxi information.....	49
Figure 17 sequence diagram for remove taxi .....	50
Figure 18 sequence diagram for register driver.....	51
Figure 19 Sequence diagram for Remove Driver.....	52
Figure 20 Sequence diagram for Update Driver .....	53
Figure 21 conceptual modelling: class diagram .....	54
Figure 22 state chart diagram for login .....	55
Figure 23 state chart diagram for taxi registration .....	56
Figure 24 state chart diagram for Generate report.....	57
Figure 25 state chart diagram for show booking details .....	58
Figure 26 <i>state chart diagram for driver info</i> .....	59
Figure 27 state chart diagram for making a book .....	60
Figure 28 proposed software architecture.....	67
Figure 29 Component diagram for admin functionality .....	68
Figure 30 component diagram for passenger and driver.....	69
Figure 31 deployment diagram.....	70
Figure 32 Class modelling diagram.....	71
Figure 33 persistent model .....	72
Figure 34 Mapping.....	74

## List of Tables

<b>Table 1</b> schedule for doing the project.....	8
<b>Table 2.1</b> use case documentation for login.....	19
<b>Table 3.3</b> use case documentation for taxi registration.....	21
<b>Table 4.4</b> use case documentation for Make booking.....	22
<b>Table 5.5</b> use case documentation for update booking.....	23
<b>Table 6.6</b> use case documentation for Update taxi information.....	24
<b>Table 7.7</b> use case documentation for remove taxi.....	25
<b>Table 8.8</b> use case documentation for Update driver information.....	26
<b>Table 9.9</b> use case documentation for Remove driver.....	27
<b>Table 10.10</b> use case documentation for changing status of driver.....	28
<b>Table 11.11</b> use case documentation for generate report.....	29
<b>Table 12.12</b> use case documentation for show booking.....	30
<b>Table 13.13</b> use case documentation for accept booking.....	31
<b>Table 14.14</b> use case documentation for send confirmation.....	32
<b>Table 15.15</b> use case documentation for cancel booking.....	33
<b>Table 16</b> Collaboration modelling for login.....	61
<b>Table 17</b> Collaboration modelling for Taxi Registration .....	61
<b>Table 18</b> Collaboration modelling for make booking.....	62
<b>Table 19</b> Collaboration modelling for passengers.....	62
<b>Table 20</b> Collaboration modelling for passengers.....	62
<b>Table 21</b> collaboration modelling for taxi.....	63
<b>Table 22</b> Update driver user interface .....	74
<b>Table 23</b> Register driver user interface .....	75
<b>Table 24</b> Update taxi status user interface .....	76
<b>Table 25</b> Edit Booking user interface.....	77
<b>Table 26</b> Make Booking user interface .....	77

## Table of Contents

<b>Declaration.....</b>	i
<b>Acknowledgment.....</b>	ii
<b>List of acronym .....</b>	iii
<b>List of Figures.....</b>	iv
<b>List of Tables .....</b>	v
<b>Abstract.....</b>	3
<b>Chapter One:.....</b>	4
<b>1. Introduction.....</b>	4
<b>1.1 Background of the Study .....</b>	4
<b>1.2 Statement of the Problem .....</b>	5
<b>1.3 Objectives of the Study .....</b>	5
<b>1.4 Methodology .....</b>	6
<b>1.4.1 Requirement gathering methods .....</b>	6
<b>1.4.2 Analysis and design Methodology .....</b>	6
<b>1.4.3 Implementation Methodology .....</b>	7
<b>System Requirement tools used .....</b>	7
<b>1.4.4 Feasibility studies .....</b>	7
<b>1.5 Beneficiaries or significant of the project .....</b>	9
<b>1.6 Limitations of the project .....</b>	9
<b>1.7 Scope of the project.....</b>	10
<b>1.8 Organization of the project .....</b>	11
<b>Chapter Two: .....</b>	12
<b>System features.....</b>	12
<b>2.1 Introduction.....</b>	12
<b>2.2 Existing system .....</b>	12
<b>2.2.1 Existing system description .....</b>	12
<b>2.2.2 Major functions in the existing system.....</b>	12
<b>2.2.3 Bottlenecks of the existing system.....</b>	13
<b>2.3 Proposed System .....</b>	14
<b>2.4 Requirement Analysis.....</b>	14
<b>2.4.1 Functional Requirements .....</b>	14

<b>2.4.2 Nonfunctional requirements .....</b>	15
<b>2.5 Boundaries of the system.....</b>	16
<b>2.6 Users of the system.....</b>	17
<b>2.7 Interaction between the system and the external users .....</b>	17
<b>2.8 system use case .....</b>	18
<b>2.8.2 Use case documentation.....</b>	19
<b>2.9 Business Rules .....</b>	34
<b>2.10 User Interface prototype .....</b>	35
<b>2.11 Activity Diagram.....</b>	36
<b>2.12 Sequence diagram .....</b>	44
<b>2.13 Analysis level of class diagram.....</b>	54
<b>2.14 State chart diagram .....</b>	55
<b>2.15 Key abstraction with CRC analysis.....</b>	61
<b>2.16 Logic model and pseudo code .....</b>	64
<b>Chapter Three .....</b>	66
<b>System Design.....</b>	66
<b>3.1 Introduction.....</b>	66
<b>3.2 Purpose and Goals of design .....</b>	66
<b>3.3 Proposed software architecture .....</b>	67
<b>3.4 Component modelling.....</b>	68
<b>3.4.1 Component Diagram for Admin Functionality .....</b>	68
<b>Figure 29 Component diagram for admin functionality .....</b>	68
<b>3.4.2 Component diagram for Customers (passenger and driver) .....</b>	69
<b>3.4.3 Deployment Diagram .....</b>	70
<b>3.5 Detail Design.....</b>	71
<b>3.5.1 System Class Diagram .....</b>	71
<b>3.5.2 Persistent model .....</b>	72
<b>3.7 Access control and security .....</b>	78
<b>References.....</b>	79

## **Abstract**

Now a day, people are more inclined towards owning a smart phone. In such a scenario, mobile application development is one of most sought after platforms. Android is one of the largest platforms that run in most smart phones from manufacturers like Samsung, vivo, iPhone etc. Android based taxi booking is an Android phone taxi reservation application which targets the residents of Bahir-Dar city. Unfortunately, in city of Bahir-Dar, the passengers are just tired of waiting at different locations for a taxi that they have just reserved to pick them up and take them home. What we need is just a reliable mobile application which reserves a taxi for users from a specific place at a certain time and which tells its passengers the status of the taxi in order to keep them from waiting for long hours. Taxi booking serves just that purpose.

This android based taxi booking system allows the users to get a taxi from any location in the city of Bahir-Dar, even if they seem to be lost. One does not have to spend hours on phone with the taxi services to tell them where exactly he/she is located currently.

This application allows filling pick-up place and destination place to the passengers, and knows which taxi would take to reach to the destination. Once a user reserves a taxi, he/she is checked booking whether it is accepted or denied by the taxi driver.

The application provides the user both the facility to view available taxi or status of taxi and how much distance does she/he travels and the payment is done based on this distance calculation with money accordingly. In conclusion, the application helps both the driver and user to facilitate and accelerate the booking process and it will be effective in time and resource management.

# **Chapter One:**

## **1. Introduction**

### **1.1 Background of the Study**

Now a days, our life is a full of movements. Most of our movements are taxi service based everywhere including big cities, medium-sized cities and even small cities around the world.

Taxi systems try to meet service demands of native population and tourists that visit or work on these places. At most part of the city, it is possible to observe that taxi drivers and also users waste much time. Because of this, there are a large number of studies that intend to improve the efficiency of these services, without increasing the costs.

This project involves the study of current system in more detail to handle Taxi Booking at Bahir Dar city. Even if there is automated taxi booking system in large city like Addis Ababa, The existing system in Bahir Dar is still manual system, which means customers are stand long time on the street waiting for coming taxi or make a call for driver they already know. Call phones to taxi drivers or to the taxi service providers and make physical contact to the taxi drivers or taxi service providers.

This proposed project needs to make significant change on the existing system. In this project customers can book taxi from anywhere in the town at any time online. The drivers are only accepting order of system and serving customer. Then the central system make decision based on customer needs, send booked taxi numbers and arrival time with corresponding amount of the cost for the customer and send notification for selected taxi drivers. This range of features ensures booking a taxi online is a lot faster and easier than booking and dispatching a taxi by making call to taxi service providers or to the taxi drivers.

## **1.2 Statement of the Problem**

There are a lot of problems in the existing manual system. Passengers as well as taxi drivers are wasting their time and money, they are stand long time on the street waiting for coming taxi or make a call for driver they already know, so it is difficult to customers to find a taxi on their way.

In other way taxi drivers are also waste their time and resource by staying somewhere until the customer call them or come to them or they find customer by roaming . Although as there is no legally specified pick and drop fee for every place based on km or any distance measurement, customers are paying extra money for single route. Road side booking of passengers is inefficient and causes occasional disputes among bookers, passengers and drivers.

Therefore taxi booking project will came up with mobile based system to solve those problems mentioned above by enabling the customer to book taxi and drivers to receive booking order from the system anywhere in the city using GPS and Google map functionalities. This enables that each customer who made a booking is served within the shortest possible time, thus increases operational efficiency and enhancing customer satisfaction.

## **1.3 Objectives of the Study**

### ***1.3.1 General Objectives***

The aim of this work is to Design, Develop and implement an online Taxi Booking app to order and book taxi service online by using mobile phone.

### ***1.3.2 Specific Objectives***

- To plan the design and development of our system
- To develop procedures for appropriate recording of passengers
- To develop interactive and user friendly interface so that the user can easily book a taxi in few minutes by doing few clicks.
- To Program the functional units of the system

## **1.4 Methodology**

### **1.4.1 Requirement gathering methods**

We gather requirement from peoples who has direct relation to our system. For gathering requirements we use the following methods.

- **Interviews**

This is one of the methods used for the collection of data which is mostly used method. We have used the interview method to gather direct information from taxi users and the taxi drivers in Bahir Dar city.

- **Observation**

This is another type of method for collecting data and information in which we have witnessed the actual events which are happening in the city. In this method all we have done is observing and note down the events that we observed from the transportation roads, so here we have observed some events like, queue to wait for taxi, taxi driver waiting for passenger, disagreement of passengers and taxi drivers by payment amount.

### **1.4.2 Analysis and design Methodology**

We decide to use object oriented system analysis and design methodologies due to the following reason:

- Object-oriented techniques work well in situations where complicated systems are undergoing contentious maintenance, adaptation and design
- **Simplicity:** software objects model real world objects, so the complexity is reduced and the program structure is very clear.
- **Reusability:** the object oriented provides opportunities for reuse through the concepts of inheritance, polymorphism, encapsulation and modularity.
- **Increased Quality:** Increases in quality are largely a by-product of this program reuse.

- **Increased extensibility:** when we need to add new feature to the system we only need to make changes in one part of the applicable class.
- **Maintainable:** OOP methods make code more maintainable. Objects can be maintained separately, making locating and fixing problems easier. The principles of good OOP design contribute to an application's maintainability.
- **Modifiability:** It is easy to make minor changes in the data representation or the procedures in an OO program.

### **1.4.3 Implementation Methodology**

#### **System Requirement tools used**

The following materials must be fulfilled to defiantly run application.

#### **Hardware tools**

- Personal computer to run the application in administrator side.
- Any android device with compatible version
- And network devices

#### **Software tools**

- Visual paradigm: - to Draw User Interface Design
- Microsoft PowerPoint:-for presentation.
- Microsoft word: -to write our system documentation.
- Android studio to write a code

### **1.4.4 Feasibility studies**

#### **Economic feasibility**

The application is economically feasible as it only requires an android device with Android device and users should be able to connect to the internet either through cellular or Wi-Fi and should able to receive messages. This would be the only cost incurred on the project and this project is economically feasible.

## **Technical feasibility**

Our team can develop the application using the specified requirement within the techniques what we have now. To develop this application, an internet connection and a database server is required. The application is going to be deployed and will be tested on android mobile phone, thereby making it technically feasible.

## **Time feasibility**

Time feasibility is determine how the proposed system accomplished with the given time table. It implies effective time management for the system, and the project should finished within deadline. So the team decides to implement and configure the new system on time without any delay.

We have scheduled our time for our project as follow:

No	Task	May 10-20	May 21-30	June 01-20	June 30-August 20	August 21-30
1	Requirement Gathering					
2	Requirement analysis					
3	System design and documentation					
5	Implementation					
6	Testing					

*Table 1 schedule for doing the project*

## **1.5 Beneficiaries or significant of the project**

The significance of the study is to create android based taxi booking system that will provide services to customers and easy to use. Taxi Booking Application is the ideal taxi booking application for individuals on the go. This user-friendly application offers convenience by allowing users to pre-set their favorite locations and journeys for their taxi bookings. It even allows users to book a taxi at their current location. For ease of convenience, users can also access and book from a list of completed trips.

Instead of dialing to the service center, taxi booking helps users deliver their requests directly to the nearest taxi drivers through internet. It greatly saves the trouble calling to service center and waiting reply from it. In general it has benefits for:

- **Drivers** benefited from the proposed system by tracking the pickup and destination place of the passenger and decrease his tired of finding passengers on the road. They perform their work effectively and efficiently without loss time and resource.
- **Customers:** it totally remove wastage of customers time who are stand long time on the street waiting for coming taxi, gains fast access and save their time.
- **And taxi service center** can manage properly all taxi those are inside the service provider.
- **Government:** The development of this project includes sales report of drivers which is information source for the government and government can collect fair tax.

## **1.6 Limitations of the project**

This project is faced with the following limitations.

- This project is only limited to contract taxi service
- This project does not include online payment.
- Restricted only in Bahir-Dar city.

## **1.7 Scope of the project**

Proposed system focus only on contract taxi in Bahir Dar city on the following areas;

- **Booking Procedure:** passenger can book or order taxi by sending their pick up, drop off place, time and date. Therefore anyone who has an android device and the application can book taxi easily.
- **Driver Details to the Customer:** The driver can send his details to the customer including taxi current status.
- **Customer Details to the Driver:** The passengers can book with required details of them and send to the driver and they could check a confirmation message whether their request is accepted or denied by the driver. The request is then sent to be nearby registered taxi drivers. Those drivers can choose to accept or deny the request.
- **Location Tracking:** The driver can track the current location of passengers easily when the passenger send request by specifying starting location, destination and arrival time.
- If multiple drivers intend to accept the request, only the first one who taps the accept button will be able to take the order and lastly a confirmation message will be sent to the customer.
- Instead of dialing to the service center, taxi booking helps users deliver their request directly to the nearest drivers

## **1.8 Organization of the project**

Our project will consist of five chapters. The first chapter, requirement analysis, includes background, objective and methodology among others. After successfully complete requirement analysis part, Analysis of the system is going to be conducted. This will includes current system description, problem of the current system, alternative solution, proposed solution, functional and non-functional requirements and others tasks. Chapter three will discuss about design of proposed system using UML modeling techniques to realize the solution selected in previous chapter. In chapter four will implement the system based on identified solution. This chapter will include Writing code and testing. Finally in chapter five implemented system will checked whether it is successfully implemented to attain the problem of the organization (i.e. tasting and evaluation) and recommendations and conclusion will be discussed there.

## **Chapter Two:**

### **System features**

#### **2.1 Introduction**

After careful observing and analyzing the problem of existing manual taxi ordering system, our team decide to design and develop the android based Taxi Booking as a solution. The new mobile application, which we are going to develop, is a faster, accurate and a bit more personal application, which especially designed to make taxi reservation better and also provide a sustainable way to get taxi easily when we want to go everywhere in the town. The proposed application will use the major functionality of the prior related work application to advance the prior related work.

#### **2.2 Existing system**

##### **2.2.1 Existing system description**

Currently there many taxi owners' association in Bahir-Dar to give service for customers in the city. Taxis are assigned by their association on mentioned and specific area and the customers are waiting or roaming around this area until the costumer come or call them to reserve the journey.

The booking process is depending on the customer and driver agreement, which means the customer tell the destination or the place she or he wants to go , then the driver calculate the cost of journey then negotiate on it and if they agreed the customer reserve journey. The payment process depends on the way they negotiate, whether before or after the customer reaches their destination. If there is a change in a journey plan then they may negotiate new terms.

##### **2.2.2 Major functions in the existing system**

- Taxi booking
  - Input: - customer request.
  - Process: The customer waits taxi on the street or call for the taxi driver
  - Output: Make booking
- Modify booking
  - Input:-customer's request
  - Process: - customers and the drivers negotiation for new term
  - Output:-modify booking

- Cancel booking
  - Input: - customer request
  - Process: - if customer and driver are not agree in cost.
  - Output: - cancel booking

### **2.2.3 Bottlenecks of the existing system**

- Performance
  - Throughput: - the existing system is somehow not well organized, and even no has formal working system, it gives fewer throughputs like amount, quantity and output.
  - Response Time: - As the system have no formal way of communication customer and taxi drivers, getting taxi takes long time for customers.
- Information (and Data)
  - Outputs
    - The existing system has no formal of working rule, there is no any information captured from it.
  - Inputs
    - No data is given as an input to the existing system
  - Stored Data
    - As the system has no input and output and related information, there is no database for the system.
- Economics
  - Costs: Costs are unknown, ups and downs may happen.
  - Profits :The system gives less profit for customers and taxi drivers
  - Service: The system is awkward to use.

## **2.3 Proposed System**

The system that we are going to develop tries to solve problems which happens in unorganized existing system by making it formal, organized and well supported with the current taxi technologies. The proposed system will give significant importance for both drivers and customers. Customers will get good taxi service with fair price and the time needed to wait a taxi will also be decreased significantly. Drivers are fully aware of where to go, who to pick using the provided information. The system integrates a GPS and google technology with the application so that the customer knows which taxi is approaching to him/her and the driver knows if he/she is approaching to the customer or not. Therefore the system that we are going to develop will be effective in time management, have good driver and customer interaction, have good customer satisfaction.

## **2.4 Requirement Analysis**

### **2.4.1 Functional Requirements**

#### **➤ Taxi registration**

The application has taxi registration user interface and the administrator can register taxi. Lastly administrator submits taxi information to the system. Here if there is update on taxi information, the administrator update taxi information and cancel specific taxi from the system depending on the business rules specified.

#### **➤ Driver registration**

The proposed system provides application interface for driver's registration. The administrator fills the form and submits to the system. If there is update in the driver information, the administrator can update driver's information and remove drivers from the system.

#### **➤ Make booking**

To book taxi, Passengers fill booking information and submit to the system or they can send request to the server to order taxi and the server selects free available driver and dispatches passengers booking to the driver and replies to passenger, once the drivers accept booking the system sends successful notification or confirmation for passengers and passengers can receive confirmation that comes from the driver and accept confirmation as soon as they received.

Administrator and drivers can view booking. To modify booking Passenger requests the system. To cancel booking Passenger and Drivers request the system.

➤ **Show booking**

The administrator and driver can show or view all booking information that the system does daily,

➤ **Report generation**

The administrator generates reports of daily activities of the system. This includes:

- Time base booking information, means at what time the passengers mostly deserve taxi.
- Location base booking information means the place where the passengers mostly order taxi.

#### **2.4.2 Nonfunctional requirements**

➤ **Input Related Requirements**

There will be accurate and flexible input mechanisms. The input form must include name, date, time, starting and destination of the customer who is going to book taxi. The administrator must enter the password to access and view the details of the entire customers.

➤ **Process Requirements**

There will be efficient storage and easy traceability/giving an outline/ and customer must have his/her own temporary booking profile to cancel and modify booking. During booking the customer should fill the appropriate information in the specified places and booking code will generate used for securely accessing the system. If there is no code, anyone can delete others persons booking. Even for modifying booking the code is useful.

- The load time for customer interface shall take no longer than two seconds.
- The login information shall be verified within few seconds.
- Response time of the system will not take long time.
- The system is work 24 hours per day seven days in a week which means the system is always work.

- **Efficiency:** - The system will reduce resource consumption for both customers and drivers
- **Usability:** - The system can be easily used
- **Security:** - the system should store a backup database of all taxis and booking:-The system provides username and password to prevent the system from unauthorized access.
- **Availability:** - The system should always be available for access at 24 hours, 7 days a week. Also in the occurrence of any major system malfunctioning, the system should be available in 1 to 2 working days, so that business process is not severely affected.
- **User Interface:** - The android device format and the forms prepared for the information are easy and user friendly to the customer and can be easily understand. The system shall be design according to standards and automate existing system. Online taxi booking system shall provide an easy-to-use user interface; so that the user does not have to learn to use this new system to interact with. The best thing in the input design is to achieve all the objectives mentioned in the simplest manner possible and creating reports for displaying and storing information.

## **2.5 Boundaries of the system**

The new system addresses the problems of the existing system by supporting the online taxi booking system user's with Mobil based technology by providing well organized, flexible and effective means of taxi booking. This includes:-

- Develop online mobile based taxi booking system.
- Customer can book taxi anywhere at any time in the town.
- Developing easily accessible system that is clear to users (customer and driver).
- Develop the system that enables users to store their data easily and quickly.
- Develop the system that generates allover reports of the system's activity.
- Develop the system that sends booking information for drivers.

## **2.6 Users of the system**

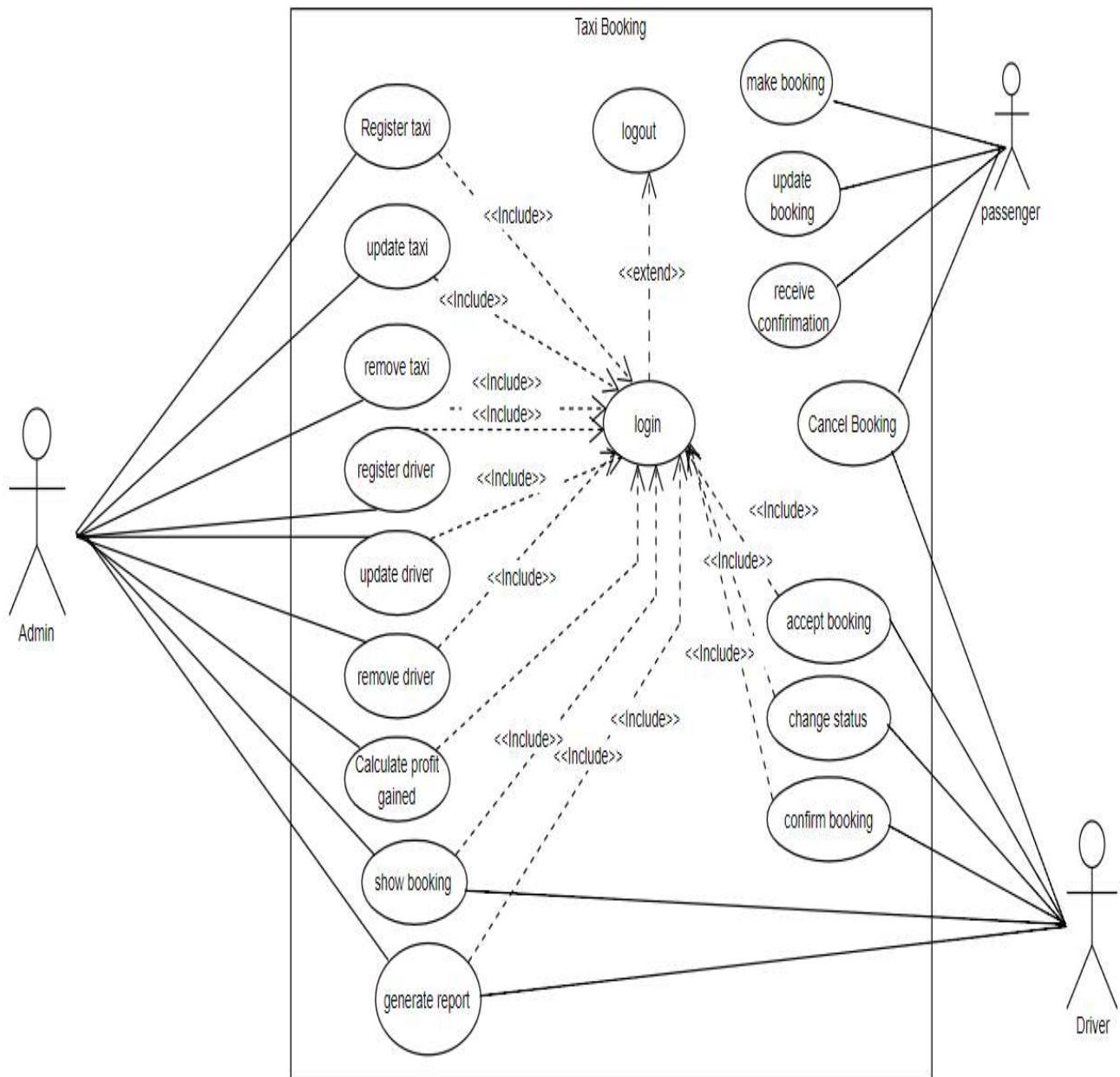
Current system encompasses different actors or users to carry out the jobs. The users are:-

- **Customer**:-the person who uses the taxi service.
- **Taxi drivers**:-the person who drive the taxi.
- **Taxi service providing center**: - the center which provides city taxi service.
- **Admin** a person who manages and controls all activities of taxi and drivers

## **2.7 Interaction between the system and the external users**

An external user (Customers, Drivers, and Admin) interacts with the system. Customers can book a taxi by registering and login into the system and they can cancel booking. And the administrator can manage or control all activities of the system after login into the system by registering taxi, updating taxi information, generating report and view report, drivers also receive booking or can decline booking.

## 2.8 system use case



**Figure 1 system use case**

## 2.8.2 Use case documentation

Section	Purpose	
Use case number	UI01	
Use case name	Log in	
Actor	administrator, driver	
Description	This use case describes how admin and driver login into the System.	
Goal	To login	
Precondition	The actor is on the homepage	
Post condition	If the use case was successful, the actor is now logged into the system. If not, the system state is unchanged.	
Basic course of action	User Action	System Response
	<p>1. The administrator and driver is on the home page to login to the system.</p> <p>3. The admin and driver enters username, password and click on Login Button.</p>	<p>2. The system promotes the admin and drivers to enter Username &amp; Password.</p> <p>4. The system verifies that all the filled have been filled out and valid.</p> <p>5. Then users logged in the system successfully.</p> <p>6 Use case Exit</p>
Alternate course of action	<p>6.1 If all fields are not filled out and not matched to the username and password the system notifies the actor a message to verify username or Password and then goes back or returns to step 4 of basic course of Action to enter again.</p>	

*Table 2.1 use case documentation for login.*

Section	Purpose	
Use case number	UI02	
Use case name	Driver registration	
Actor	Admin	
Description	System allows admin to register drivers	
Goal	To register drivers	
Precondition	Login	
Post condition	Registered	
Basic course of action	User Action	System response
	1. The administrator wants to register driver 3. The administrator open registration link and enter or fill necessary information about drivers.	2. The system allows admin to enter driver's information. 4. The system verifies if all the filled have been filled out and validate it. 5. The system notifies driver registration is successful 6. Use case end.
Alternate course of action	5.1 The system indicates the users that he/she entered invalid information or misses something 6.1 The use case continues to step 3 and the basic course of action ends.	

*Table 2.2 use case documentation for Driver registration*

Section	Purpose
Use case number	UI03
Use case name	Taxi registration
Actor	Administrator
Goal	To register new taxi
Description	Allow Administrator to register new taxi to the system
Precondition	UI01
Post condition	New taxi is registered to the system
Basic course of action	<ol style="list-style-type: none"> <li>1. The administrator wants to register new taxi.</li> <li>2. The administrator clicks “register new taxi button”.</li> <li>3. The system displays taxi registration form.</li> <li>4. The administrator enters full information about taxi to be registered and submit.</li> <li>5. The system verifies that if the basic fields have been filled.</li> <li>6. The system notifies that registration is successful.</li> <li>7. Use case end.</li> </ol>
Alternate course of action	<p>5.1 The system indicates that the fields have not filled correctly</p> <p>6.1 The use case goes back to step 4 and exit from use case.</p>

*Table 3.3 use case documentation for taxi registration.*

Section	Purpose	
Use case number	UI04	
Use case name	Make booking	
Actor	Passenger	
Description	The passenger can book taxi service on his/her mobile device.	
Goal	To order taxi	
Precondition	None	
Post condition	Passenger booked taxi	
Basic course of action	User Action	System Response
	<ol style="list-style-type: none"> <li>1. Passenger wants to make a book.</li> <li>3. Passenger fills up booking information.</li> <li>4. The passenger submits booking.</li> </ol>	<ol style="list-style-type: none"> <li>2. The system allows passengers to enter basic booking information.</li> <li>5. The system confirms booking.</li> <li>6. The use case end.</li> </ol>
Alternate course of action	<p>5.1 The system indicates that passenger enters invalid information or there is jumped fields.</p> <p>6.1 The system continues at step three and four. Finally exit from the use case.</p>	

*Table 4.4 use case documentation for Make booking.*

Section	Purpose	
Use case number	UI05	
Use case name	Update booking	
Actor	Passenger	
Description	Allow users to update the current location	
Goal	To change booking	
Pre-condition	Passenger must make booking first to update	
Post condition	Make booking	
Basic course of action	User Action	System Response
	1. Passenger wants to update booking  3. Passenger enters booking code	2. The system allows Passenger to enter booking code.  4. The system verifies booking code.  5. The system allows Passenger to edit basic booking information.  6. The system verifies basic booking information.  7. Use case ends.
Alternate course of action	4.1 The system notifies Passenger that he/she enter invalid booking code.  4.2 The system continues at step 2 and 3.  5.1 The system continues at step 5 and use case exits.	

*Table 5.5 use case documentation for update booking.*

Section	Purpose
Use case number	UI06
Use case name	Update taxi info
Actor	Administrator
Description	allows admin to update taxi information
Goal	To change the status of taxi
Precondition	Registered taxi to be updated and Login (UI01)
Post condition	Taxi information updated
Basic course of action	<ol style="list-style-type: none"> <li>1. Administrator wants to update taxi information and should be on the homepage.</li> <li>2. The system asks admin to enter taxi number.</li> <li>3. The administrator enter taxi number.</li> <li>4. The system verifies the taxi number.</li> <li>5. The system displays the taxi information.</li> <li>6. The system allows admin to modify the taxi information.</li> <li>7. Administrator modifies taxi information.</li> <li>8. Use case end.</li> </ol>
Alternate course of action	<p>4.1 The system notifies the admin that he/she entered invalid taxi number and the system continues back to step 3 to step 6.</p> <p>7.1 The system continues to modify the information and use case ends.</p>

*Table 6.6 use case documentation for Update taxi information.*

Section	Purpose
Use case number	UI07
Use case name	Remove taxi
Actor	Administrator
Description	Allows Administrator to delete taxi
Goal	To remove taxi
Precondition	There must be a registered taxi to remove
Post condition	Taxi will be removed
Basic course of action	<ol style="list-style-type: none"> <li>1. Administrator wants to remove taxi information.</li> <li>2. The system asks admin to enter taxi number.</li> <li>3. The administrator enters taxi number.</li> <li>4. The system verifies the taxi number.</li> <li>5. Administrator removes taxi information.</li> <li>6. Use case end.</li> </ol>
Alternate course of action	4.1 The system notifies that the entered taxi number is invalid and invites user to go back to step 3 and use case ends.

*Table 7.7 use case documentation for remove taxi.*

Section	Purpose
Use case number	UI08
Use case name	Update Driver
Actor	Administrator
Description	allows administrator to update driver information
Goal	To change driver info
Precondition	Registered driver must be found to be updated
Post condition	Driver information will be Updated
Basic course of action	<ol style="list-style-type: none"> <li>1. Administrator wants to update driver information.</li> <li>2. The system asks administrator to enter driver identification.</li> <li>3. The administrator enter driver ID.</li> <li>4. The system verifies driver ID.</li> <li>5. The system displays driver information.</li> <li>6. The system allows admin to modify driver information.</li> <li>7. Administrator modifies driver information.</li> <li>8 .Use case end.</li> </ol>
Alternate course of action	<p>2.1 The system notifies that the entered driver ID is invalid and invites the user to enter valid information.</p> <p>7.1 The system continues back to step 7 and use case ends.</p>

*Table 8.8 use case documentation for Update driver information.*

Section	Purpose
Use case number	UI09
Use case name	Remove Driver
Actor	Administrator
Description	Allows Administrator to Remove driver
Goal	To remove driver
Precondition	Driver registration
Post condition	Driver Removed
Basic course of action	<ol style="list-style-type: none"> <li>1. Administrator wants to remove driver information.</li> <li>2. The system asks admin to enter driver ID.</li> <li>3. The administrator enter driver ID.</li> <li>4. The system verifies the driver ID.</li> <li>5. Administrator removes driver information.</li> <li>6. Use case end.</li> </ol>
Alternate course of action	<ol style="list-style-type: none"> <li>4.1 The system notifies that the entered driver ID is invalid.</li> <li>4.2 The system continues back to step 3 and use case ends.</li> </ol>

*Table 9.9 use case documentation for Remove driver.*

Section	Purpose
Use case number	UI10
Use case name	Change status
Actor	Driver
Description	allows driver to change status from busy to free
Goal	To change status
Precondition	Busy
Post condition	Free and ready to give service
Basic course of action	<ol style="list-style-type: none"> <li>1. Driver wants to change status.</li> <li>2. The system asks driver to change status.</li> <li>3. The driver changes his/her status.</li> <li>4. Use case end.</li> </ol>
Alternate course of action	3.1 The system notifies that status of driver is free.

*Table 10.10 use case documentation for changing status of driver.*

Section	Purpose
Use case number	UI11
Use case name	Generate report
Actor	Administrator and Drivers
Description	To generate reports
Goal	Having report
Pre-condition	Login (UI01)
Post condition	Report generated
Basic course of action	<ol style="list-style-type: none"> <li>1. Administrator and drivers want to generate report.</li> <li>2. The system allows the users to generate report</li> <li>3. The system notifies the users that they generate report successfully.</li> <li>4. Use case ends.</li> </ol>
Alternate course of action	3.1 The system continues to step 2, 3 and use case end.

*Table 11.11 use case documentation for generate report.*

Section	Purpose
Use case number	UI12
Use case name	Show booking
Actor	Users (drivers and admin)
Description	Shows booking details.
Goal	To view booking
Precondition	Make booking
Post condition	View booking detail
Basic course of action	<p>1. Users want to show booking.</p> <p>2. The system allows users to show booking.</p> <p>3. Drivers and admin must log in but customers does not need to log in.</p> <p>4. The system allows drivers and admin to enter log in information. It allows customers to enter reservation code.</p> <p>5. The system verifies log in information and booking code.</p> <p>6. The system notifies the user that viewing is successful</p> <p>7. Use case ends.</p>
Alternate course of action	<p>5.1 The system notifies that log in information and reservation code is invalid or missed.</p> <p>5.2 continues to step 4 and use case end.</p>

*Table 12.12 use case documentation for show booking.*

Section	Purpose
Use case number	UI13
Use case name	Receive booking
Actor	Drivers
Description	Allows the drivers to receive booking
Goal	To accept booking
Precondition	Login
Post condition	Send confirmation
Basic course of action	<ul style="list-style-type: none"> <li>1. the drivers wants to receive booking</li> <li>2. the system allows drivers to receive booking</li> <li>3. the drivers can receive booking and can send confirmation to the passengers</li> </ul>
Alternate course of action	<ul style="list-style-type: none"> <li>4. If the driver cannot receive booking, the system notifies booking is rejected.</li> </ul>

*Table 13.13 use case documentation for accept booking.*

Section	Purpose
Use case number	UI14
Use case name	Send confirmation
Actor	Drivers
Description	Allows the drivers to reply passengers request
Goal	To send confirmation
Pre-condition	Receive booking
Post condition	Passengers receive confirmation
Basic course of action	<ol style="list-style-type: none"> <li>1. the drivers wants to send confirmations to the customers</li> <li>2. the system allows drivers to reply</li> </ol>
Alternate course of action	<ol style="list-style-type: none"> <li>3. The system notifies when the drivers send error confirmation</li> </ol>

*Table 14.14 use case documentation for send confirmation.*

Section	Purpose
Use case number	UI15
Use case name	Cancel booking
Actor	Drivers and Passengers
Description	Allows users to cancel booking
Goal	To cancel booking
Precondition	Make booking
Post condition	Booking canceled
Basic course of action	<ol style="list-style-type: none"> <li>1. Passengers and drivers want to cancel booking.</li> <li>2. The system allows drivers and passengers to enter (“cancel booking form”).</li> <li>3. Drivers and passengers enter booking code.</li> <li>4. The system verifies the booking code.</li> <li>5. Use case ends.</li> </ol>
Alternate course of action	<p>4.1 The system notifies that the user enters invalid booking code.</p> <p>4.2 The system continues to step 3 and 4 then use case ends.</p>

*Table 15.15 use case documentation for cancel booking.*

## **2.9 Business Rules**

ID: BR01

Name: Passenger Identification

Description: Passengers should have ID to make booking

ID: BR02

Name: Booking

Description: Passengers should fill appropriate personal information (Name, address, starting location, destination location, starting time, arrival time)

ID: BR03

Name: Driver registration

Description: Drivers should have valid and driving license to be registered in to the system

ID: BR04

Name: Cancelation

Description: Passenger and driver can cancel booking by giving their booking number for cancelation confirmation.

## 2.10 User Interface prototype

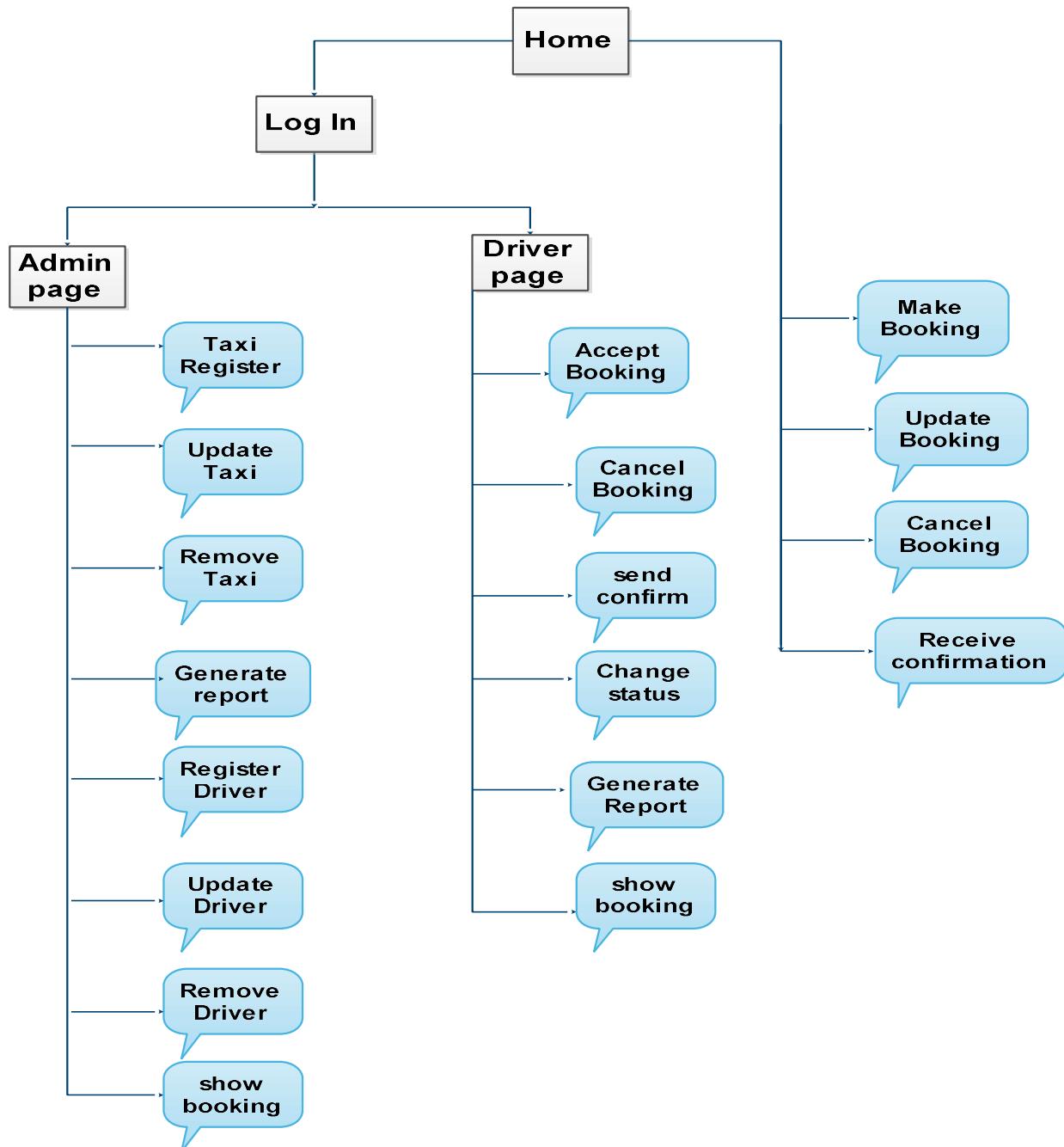


Figure 2 user interface prototype diagram

## 2.11 Activity Diagram

An Activity diagram is similar to a flowchart to represent the flow from one activity to another activity. Activity diagrams and State chart diagrams are related. While a State chart diagram focuses attention on an object undergoing a process (or on a process as an object), an Activity diagram focuses on the flow of activities involved in a single process. The Activity diagram shows how these single-process activities depend on one another.

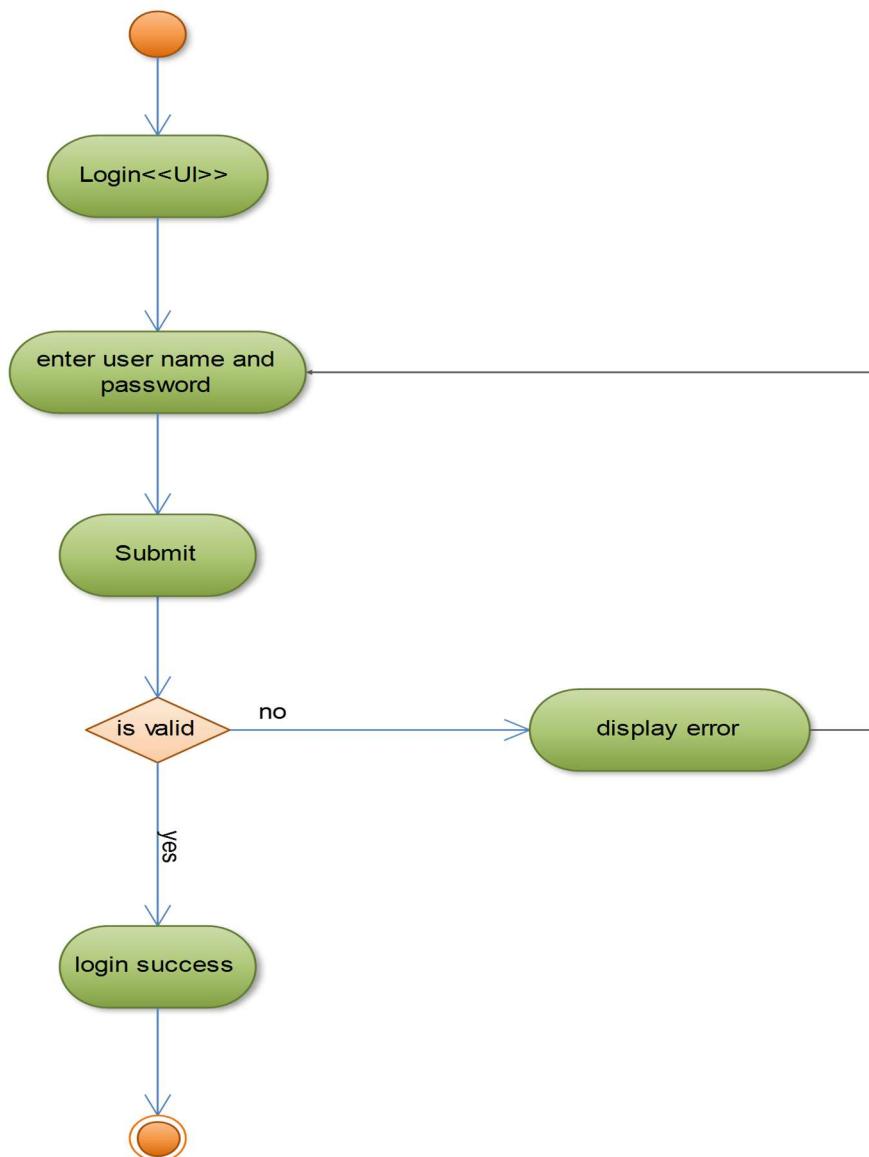
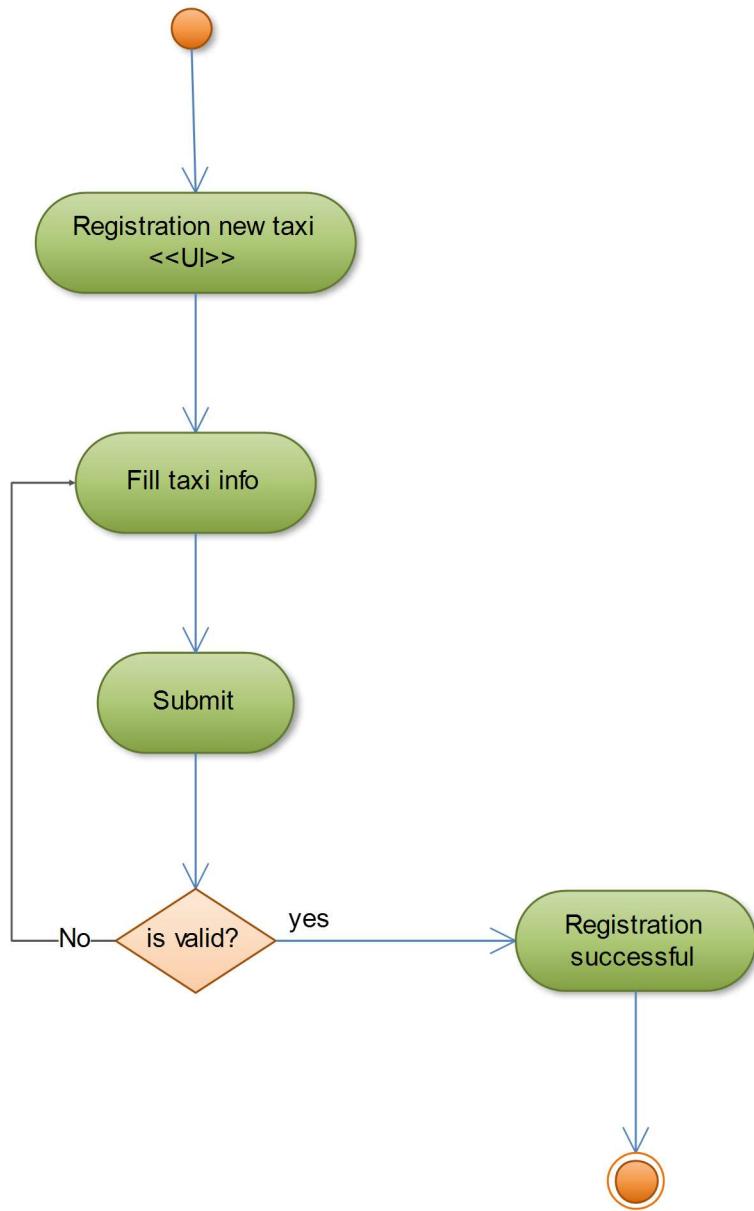
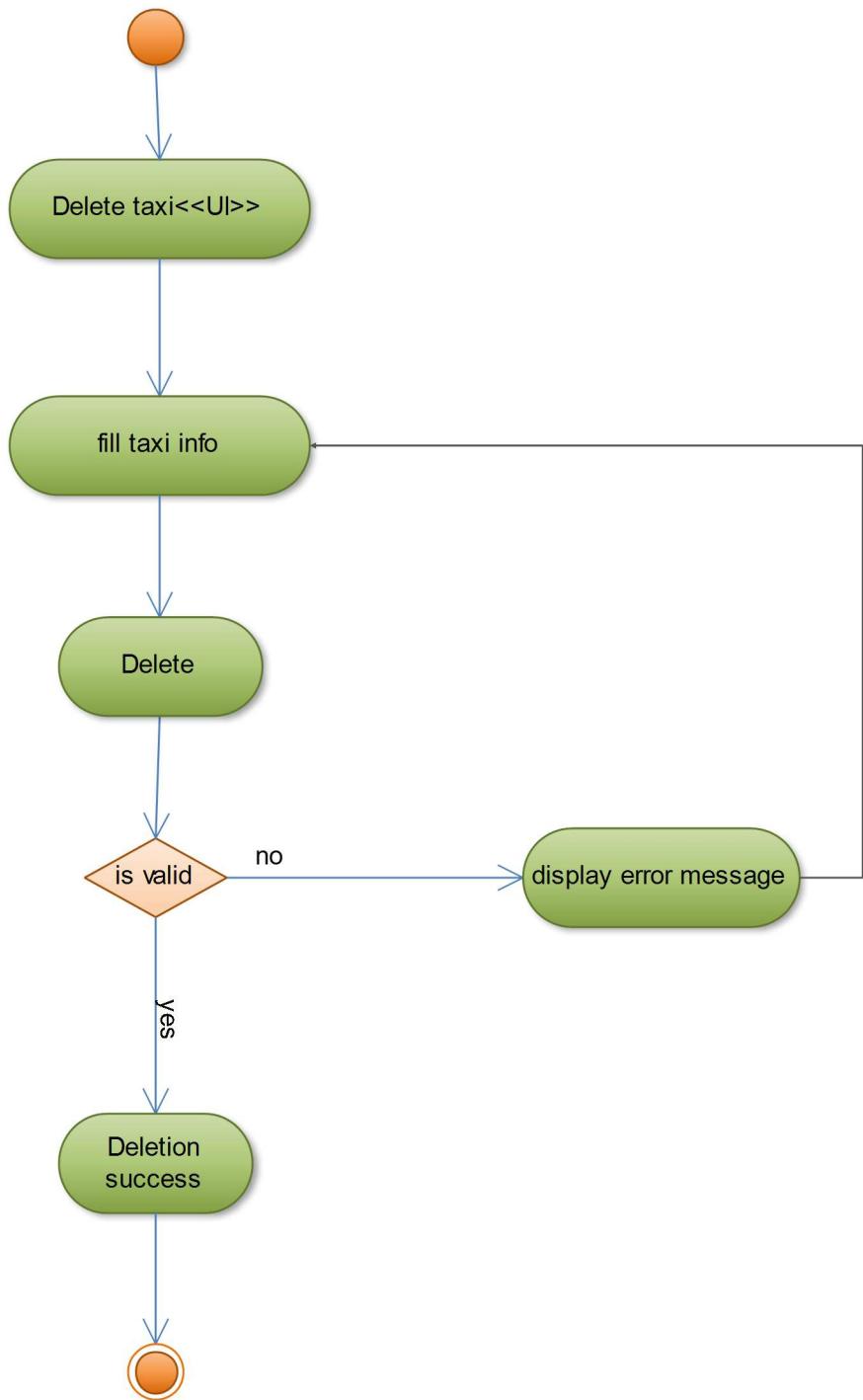


Figure 3 activity diagram for login



**Figure 4** activity diagram for taxi registration



**Figure 5** activity diagram for remove taxi

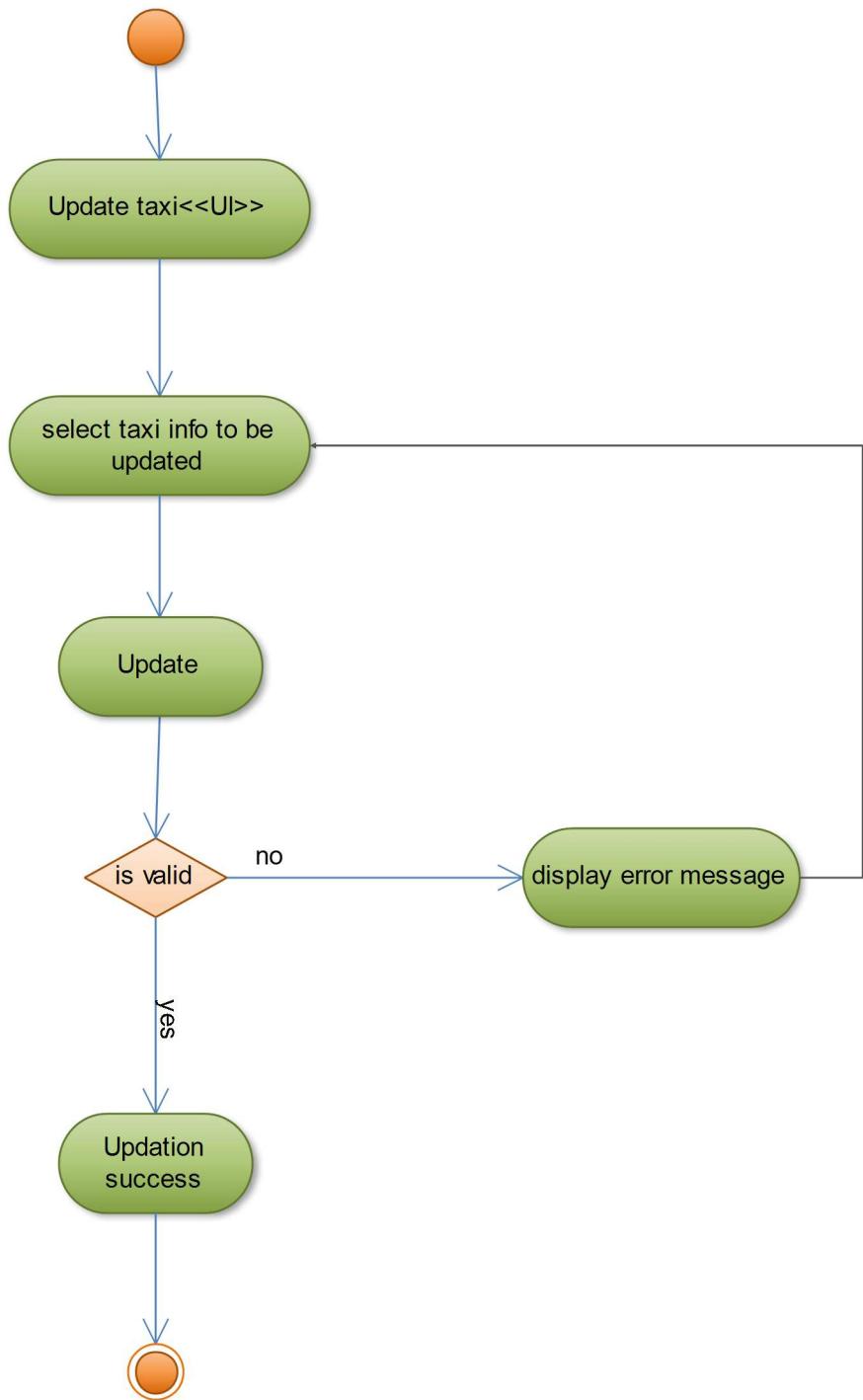
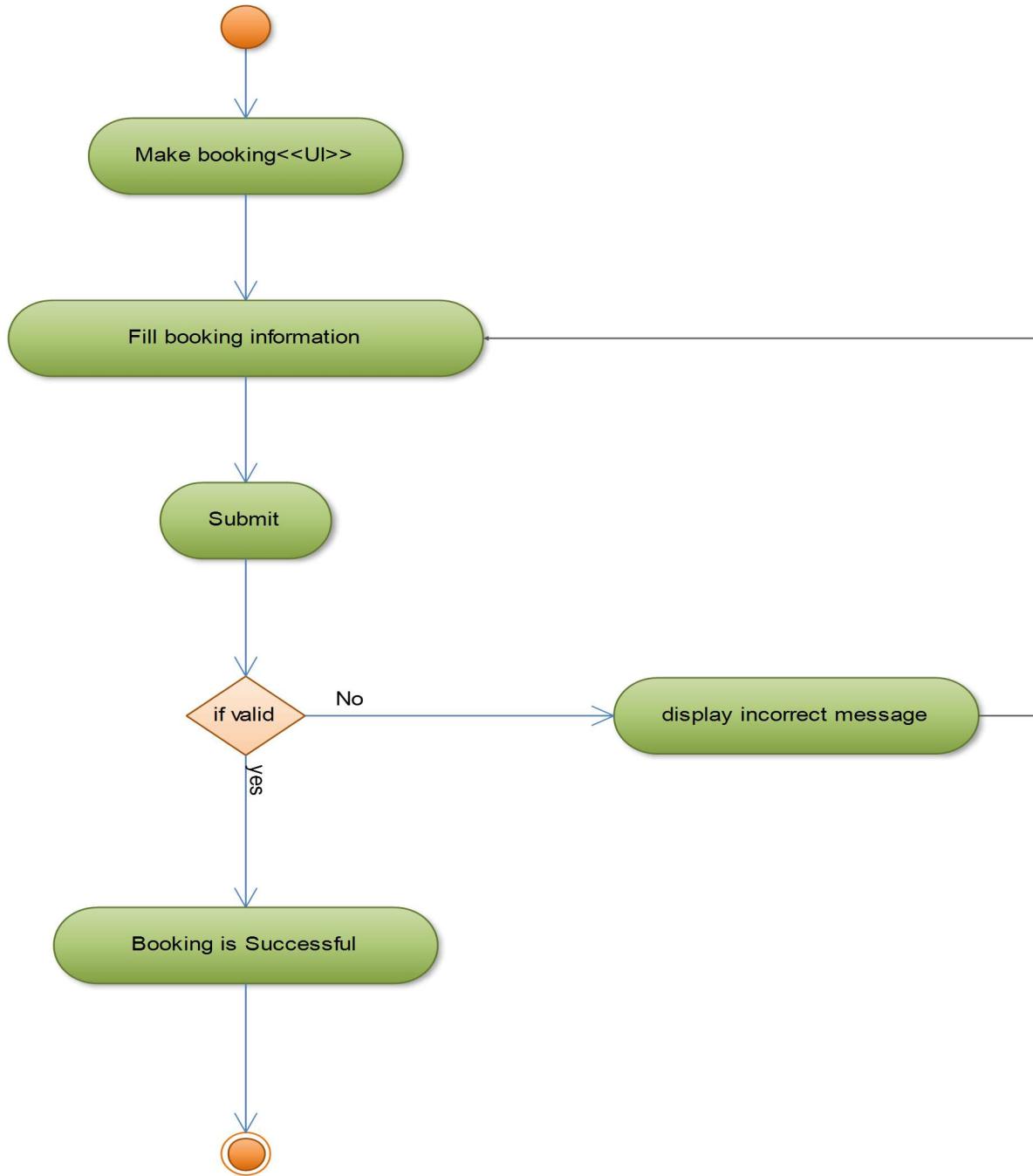
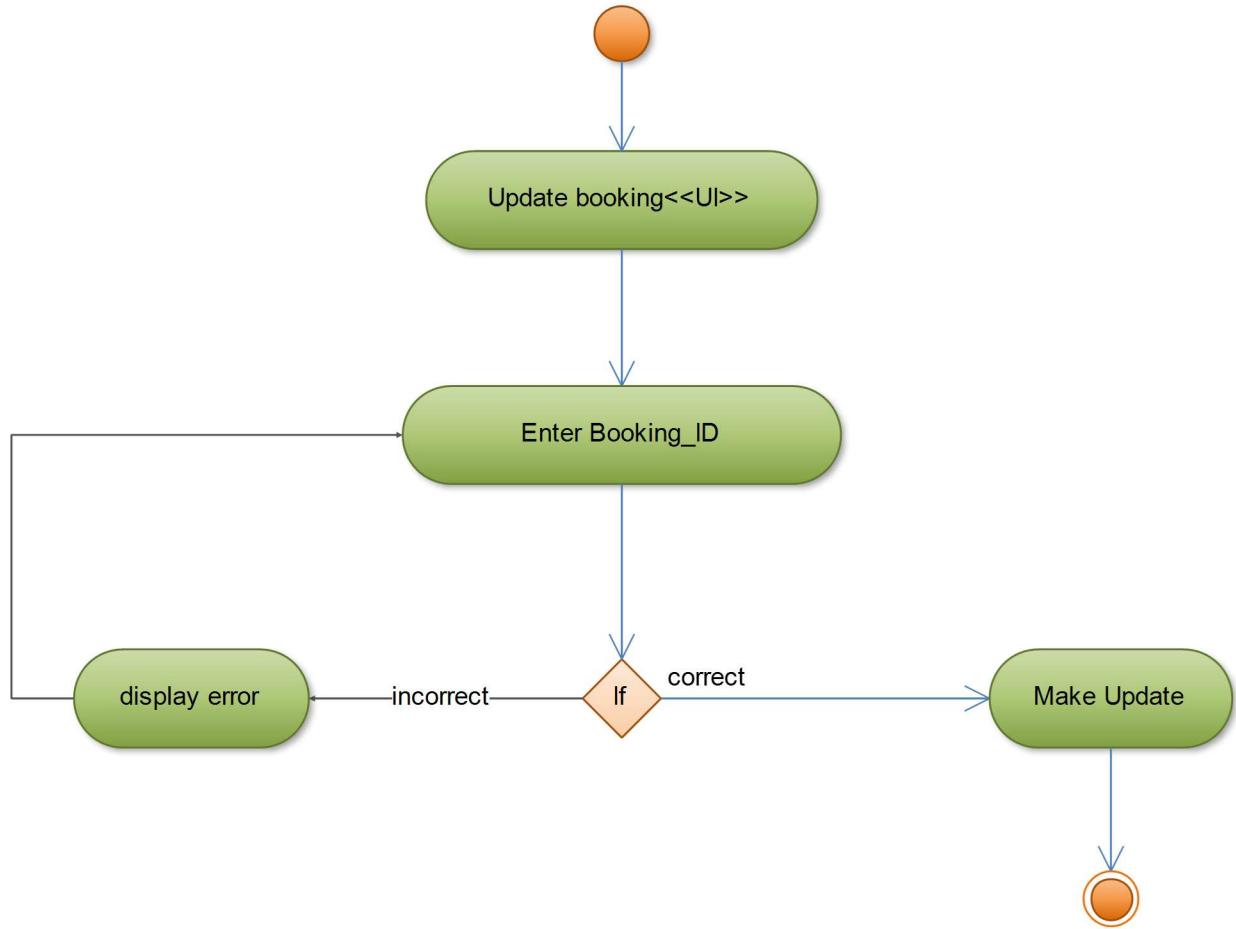


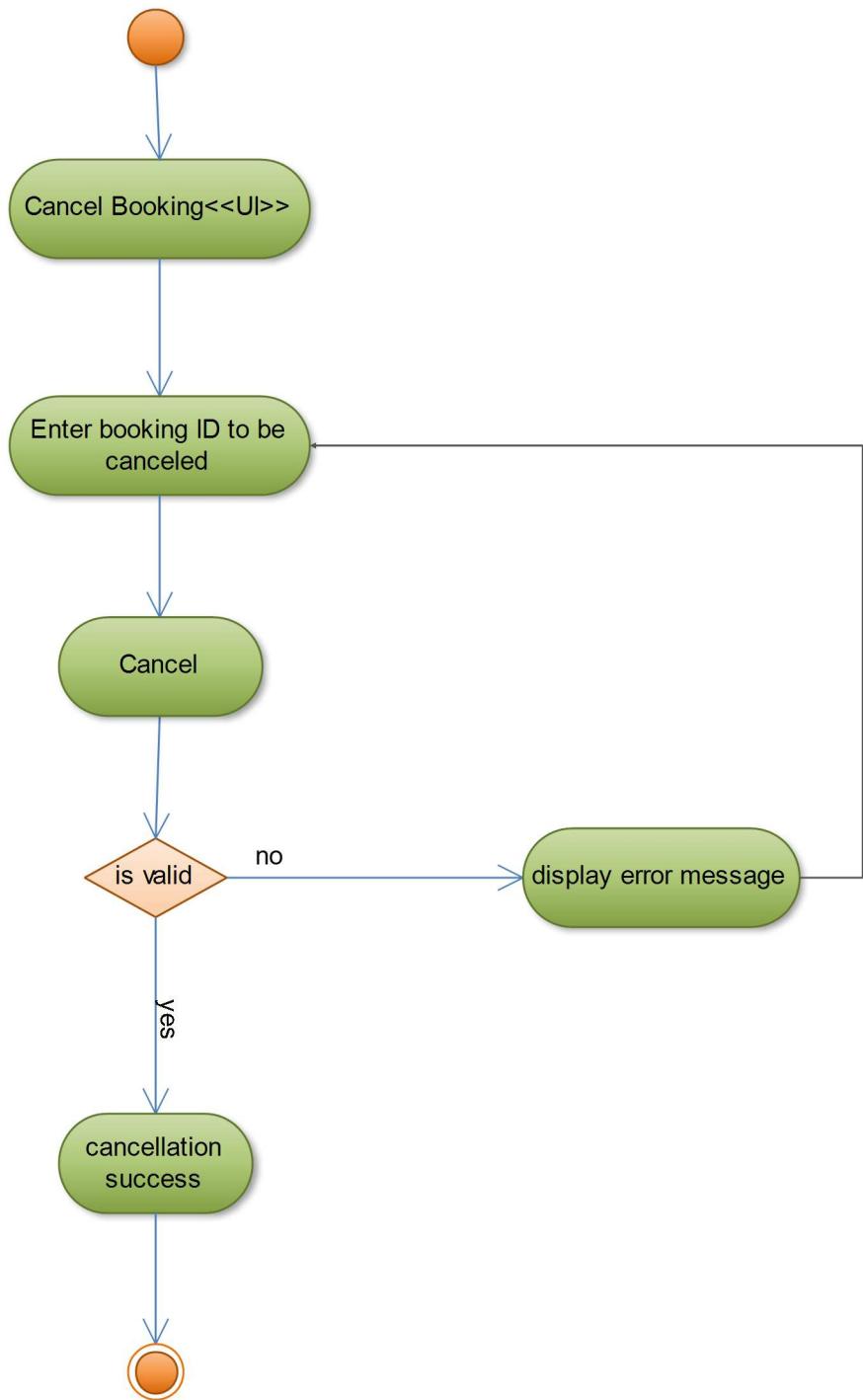
Figure 6 activity diagram for update taxi



**Figure 7 activity diagram for make booking**



**Figure 8** activity diagram for update booking



**Figure 9** activity diagram for cancel booking

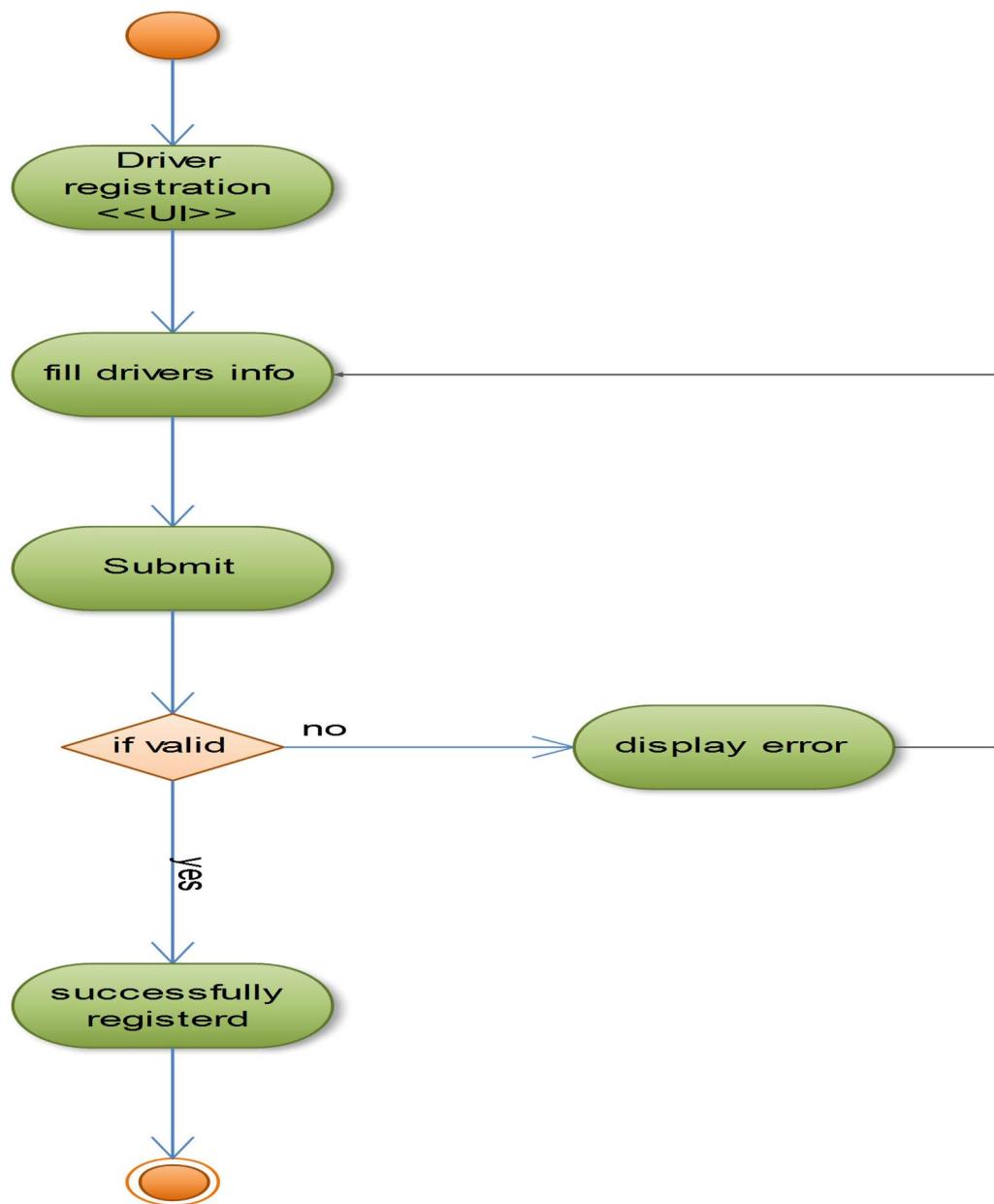


Figure 10 activity diagram for driver registration

## 2.12 Sequence diagram

A sequence diagram shows an interaction arranged in time sequence. In particular, it shows the instances participating in the interaction by their “lifelines” and the stimuli that they arranged in time sequence. It does not show the associations among the objects.

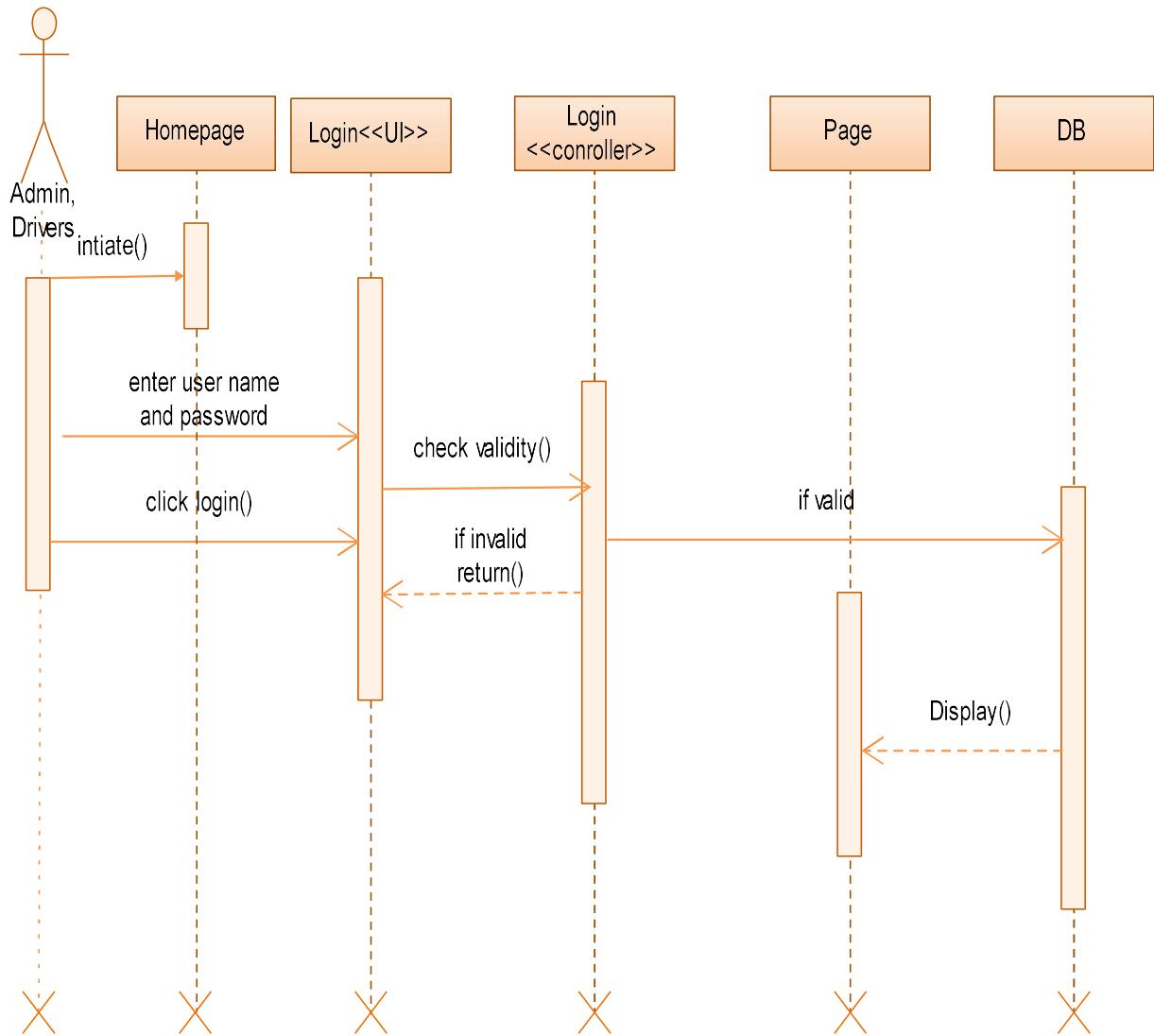
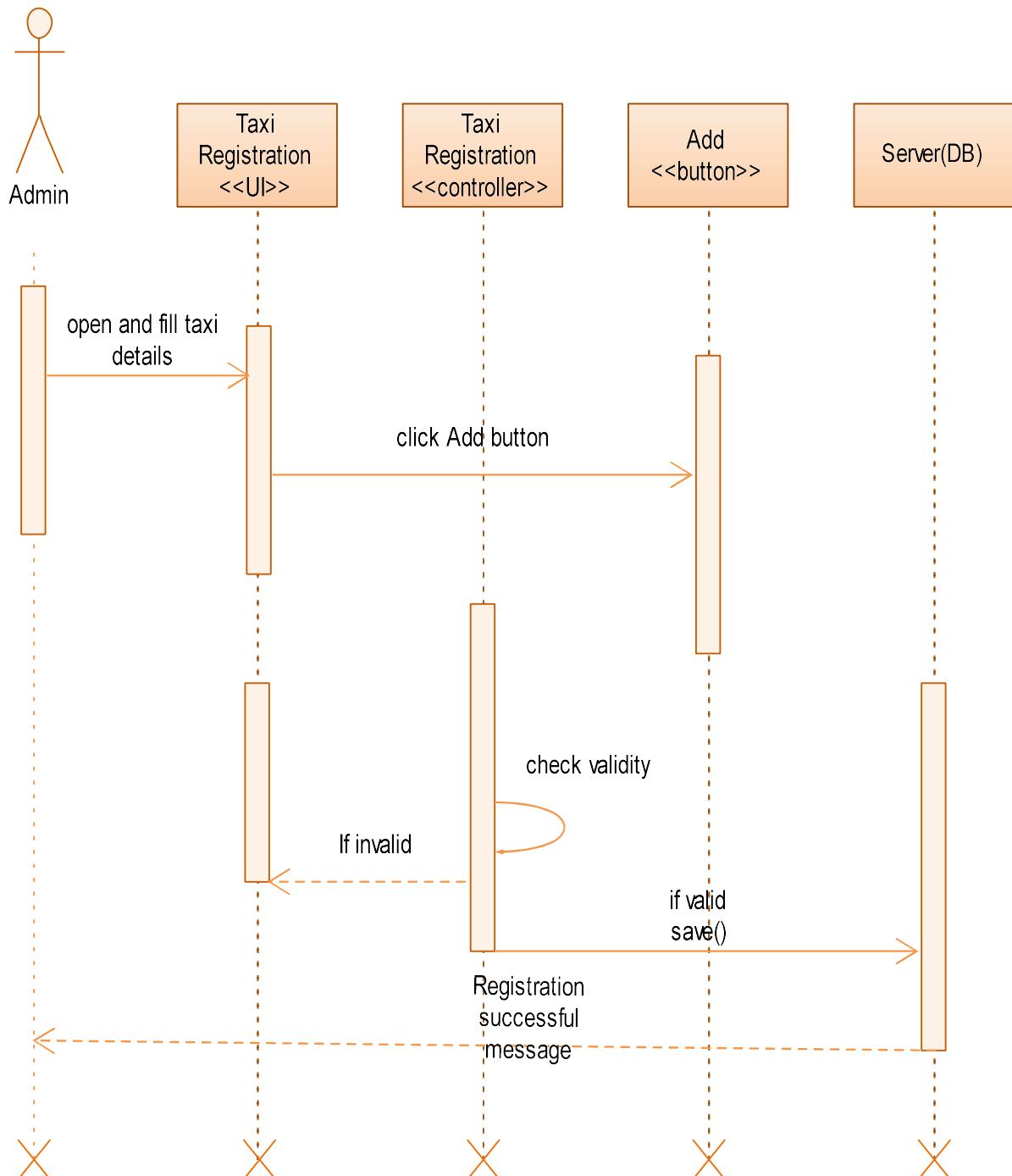
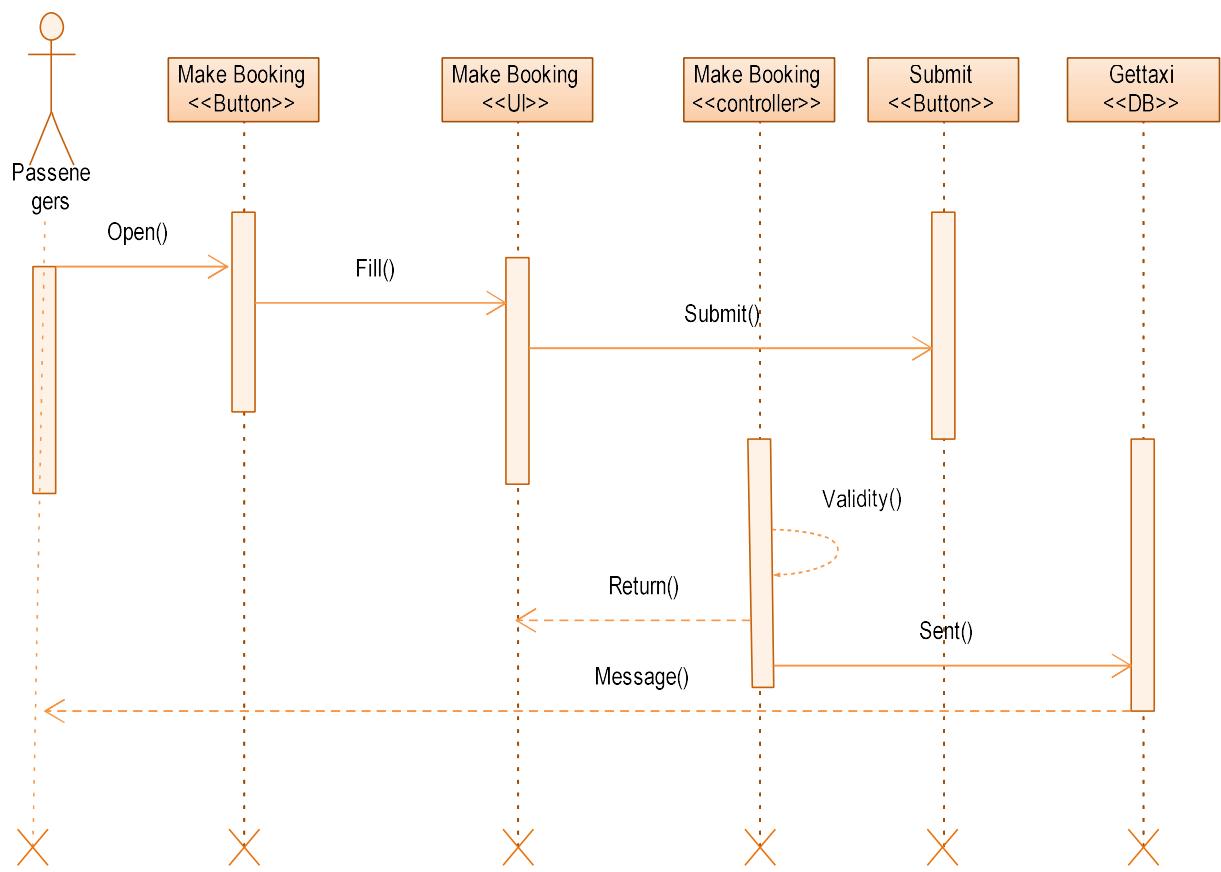


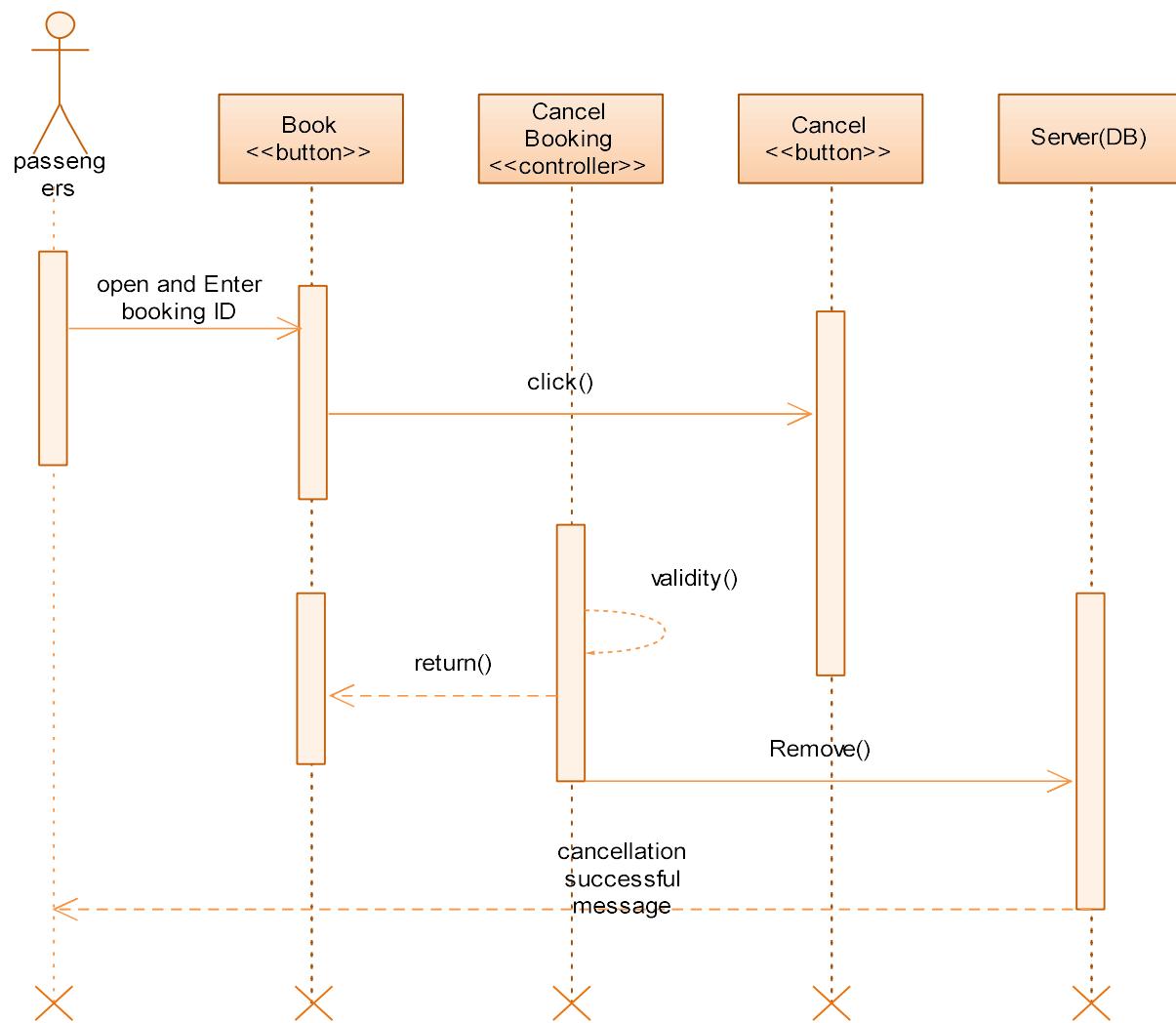
Figure 11 sequence diagram for login



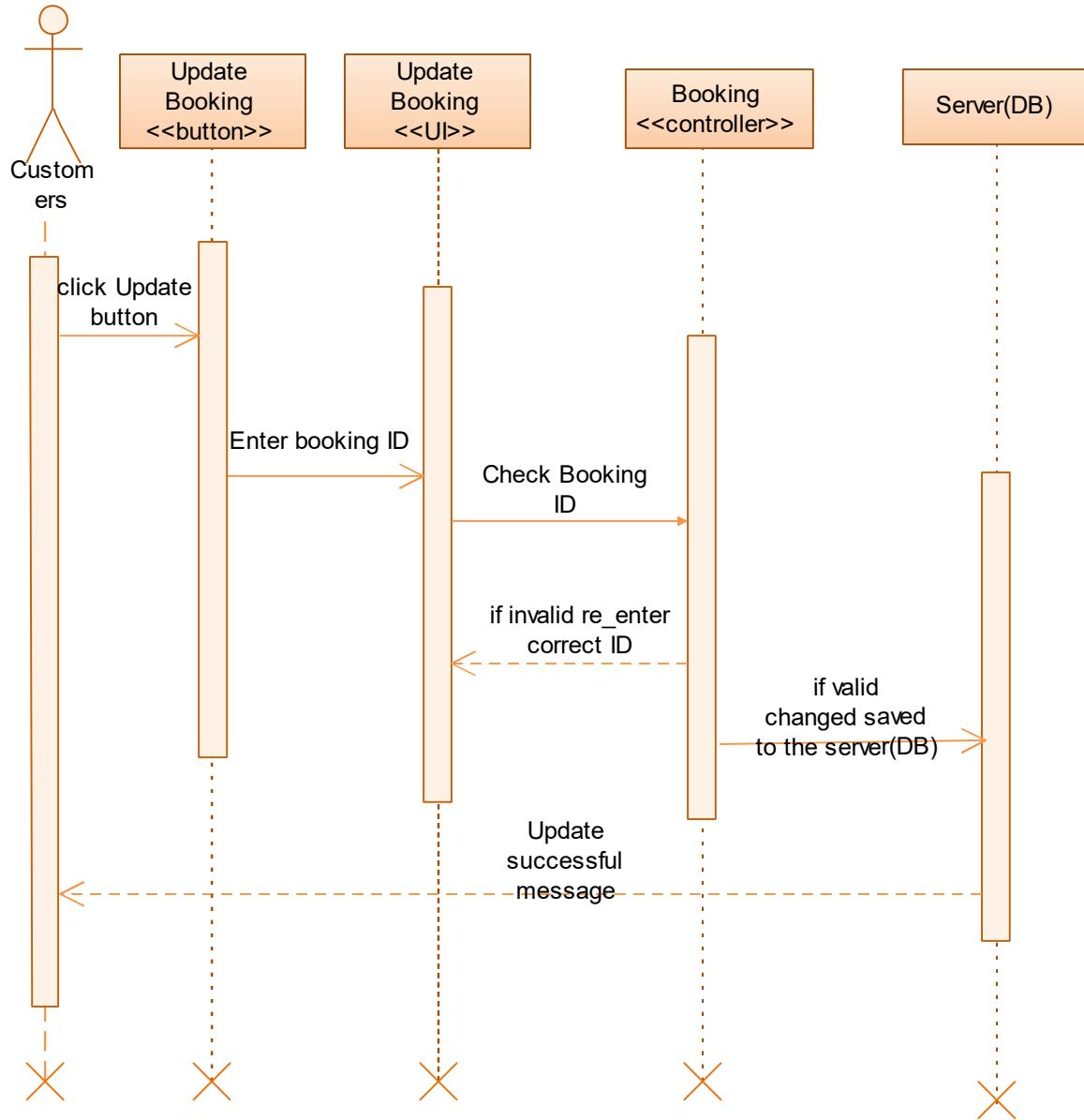
**Figure 12** sequence diagram for Taxi registration



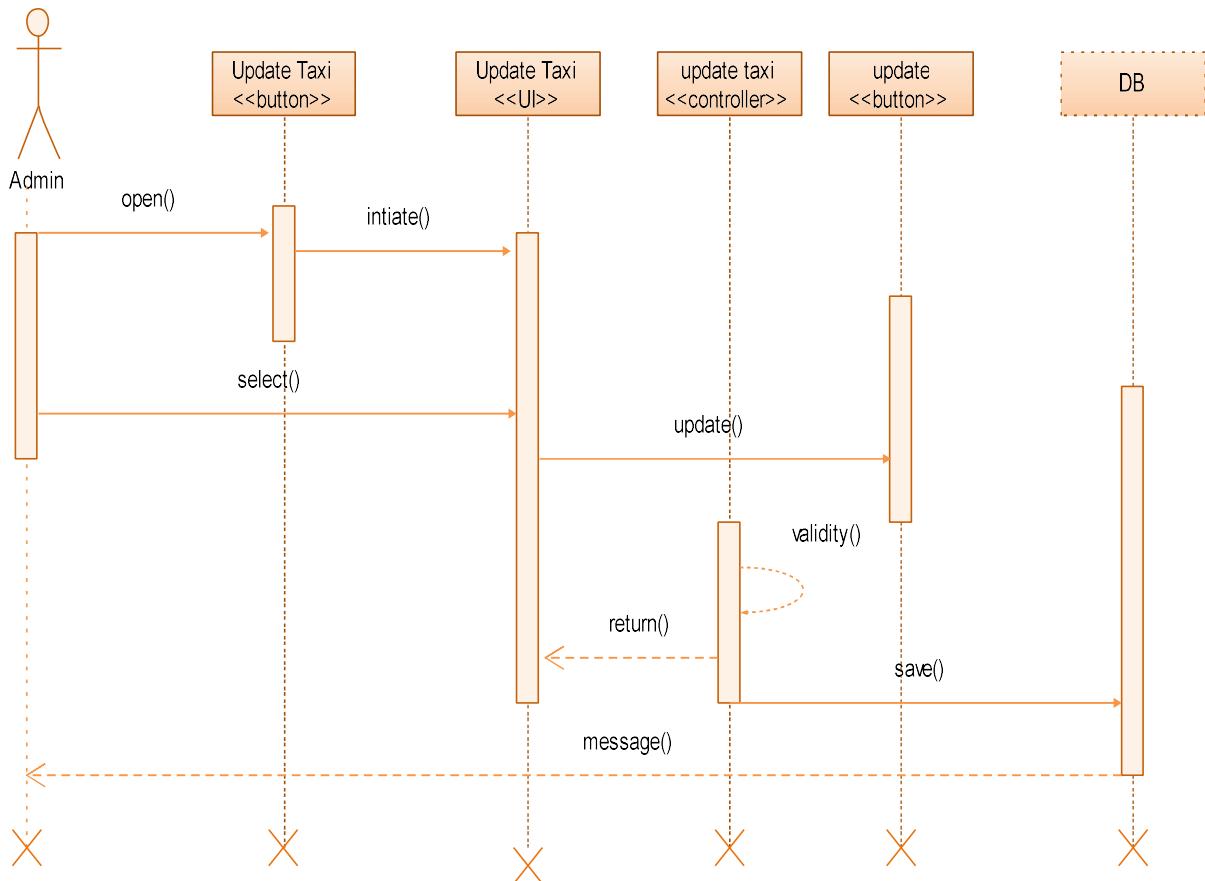
**Figure 13** sequence diagram for Make booking



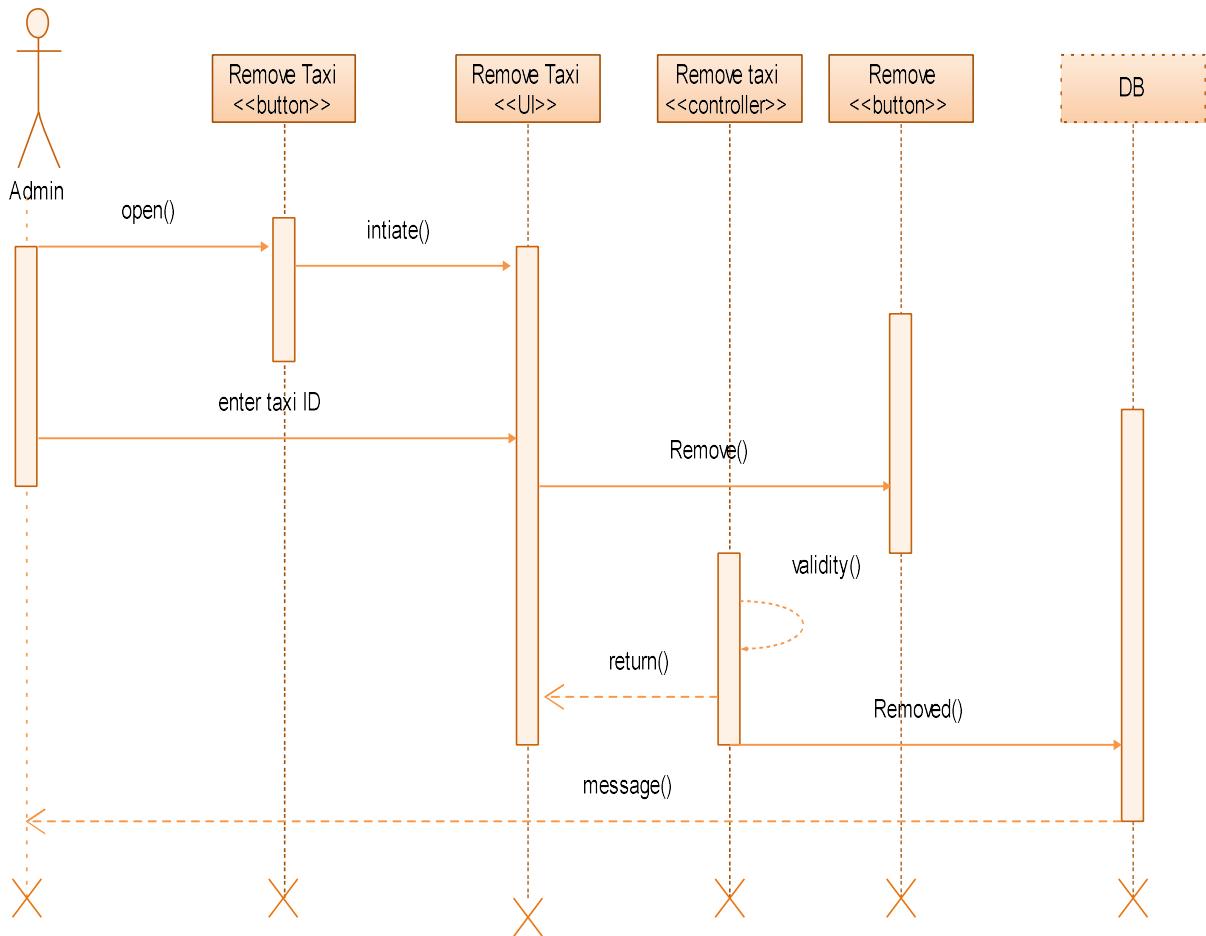
**Figure 14** sequence diagram for cancel booking



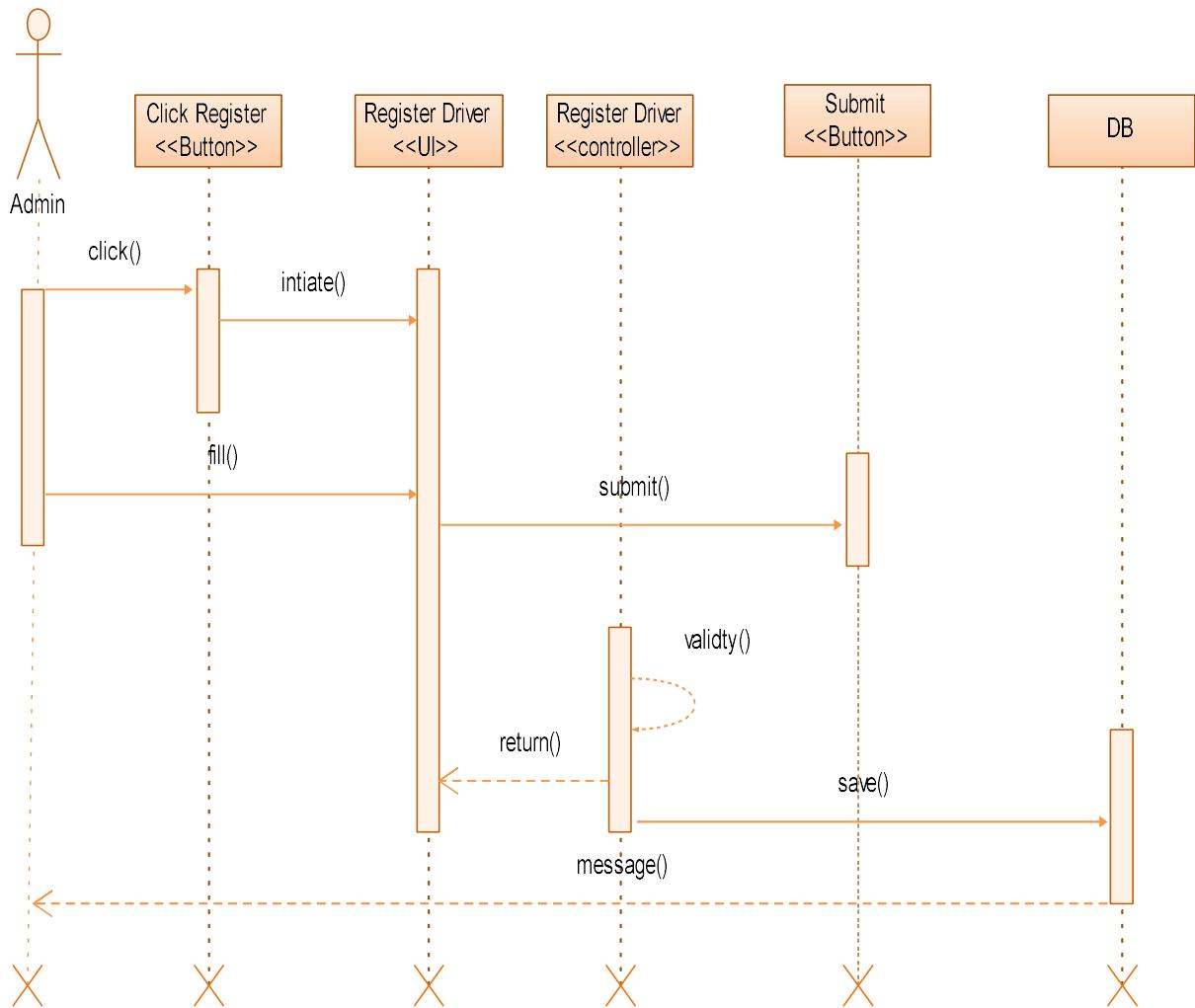
*Figure 15 sequence diagram for update booking*



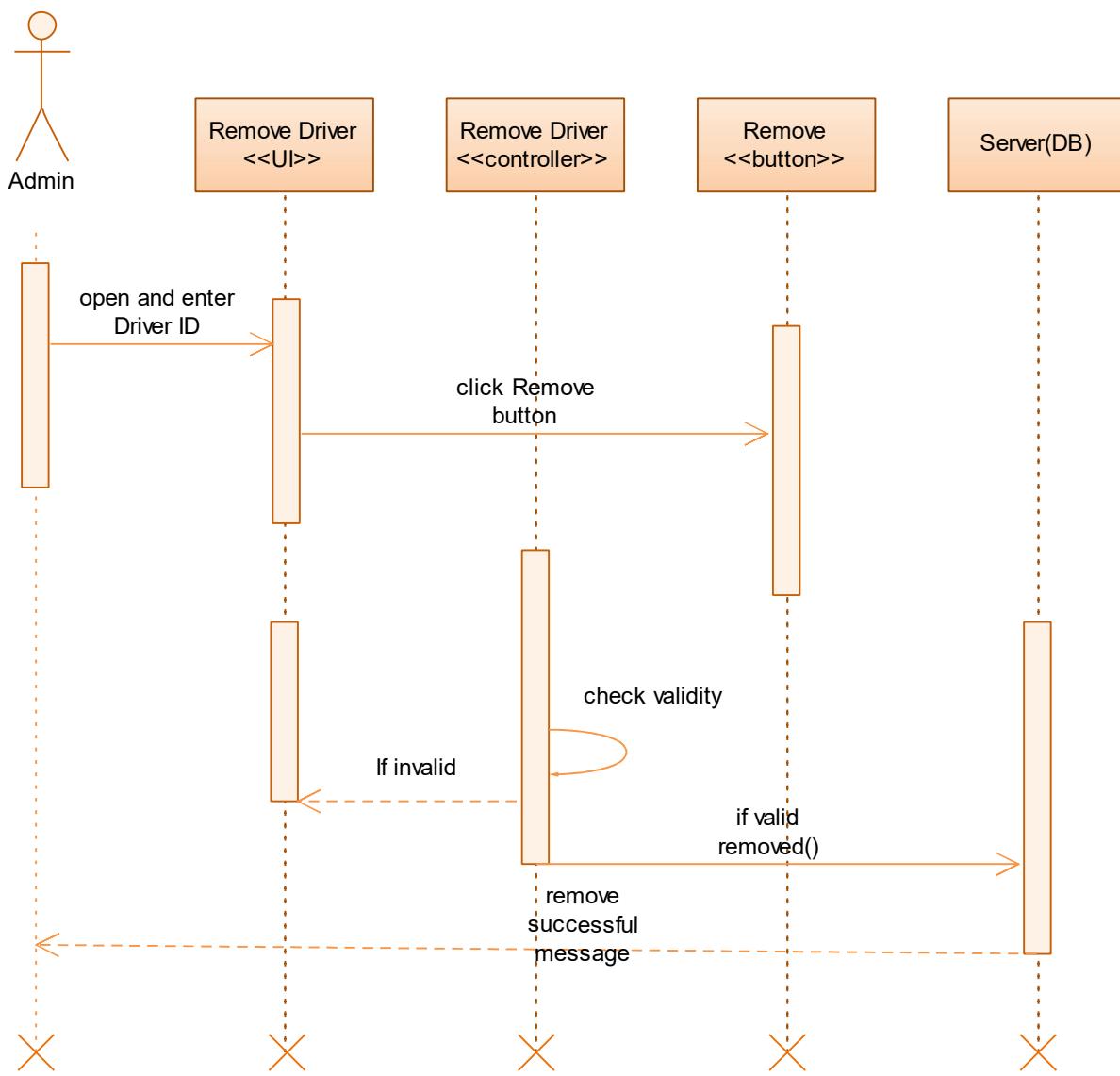
**Figure 16** sequence diagram for update taxi information



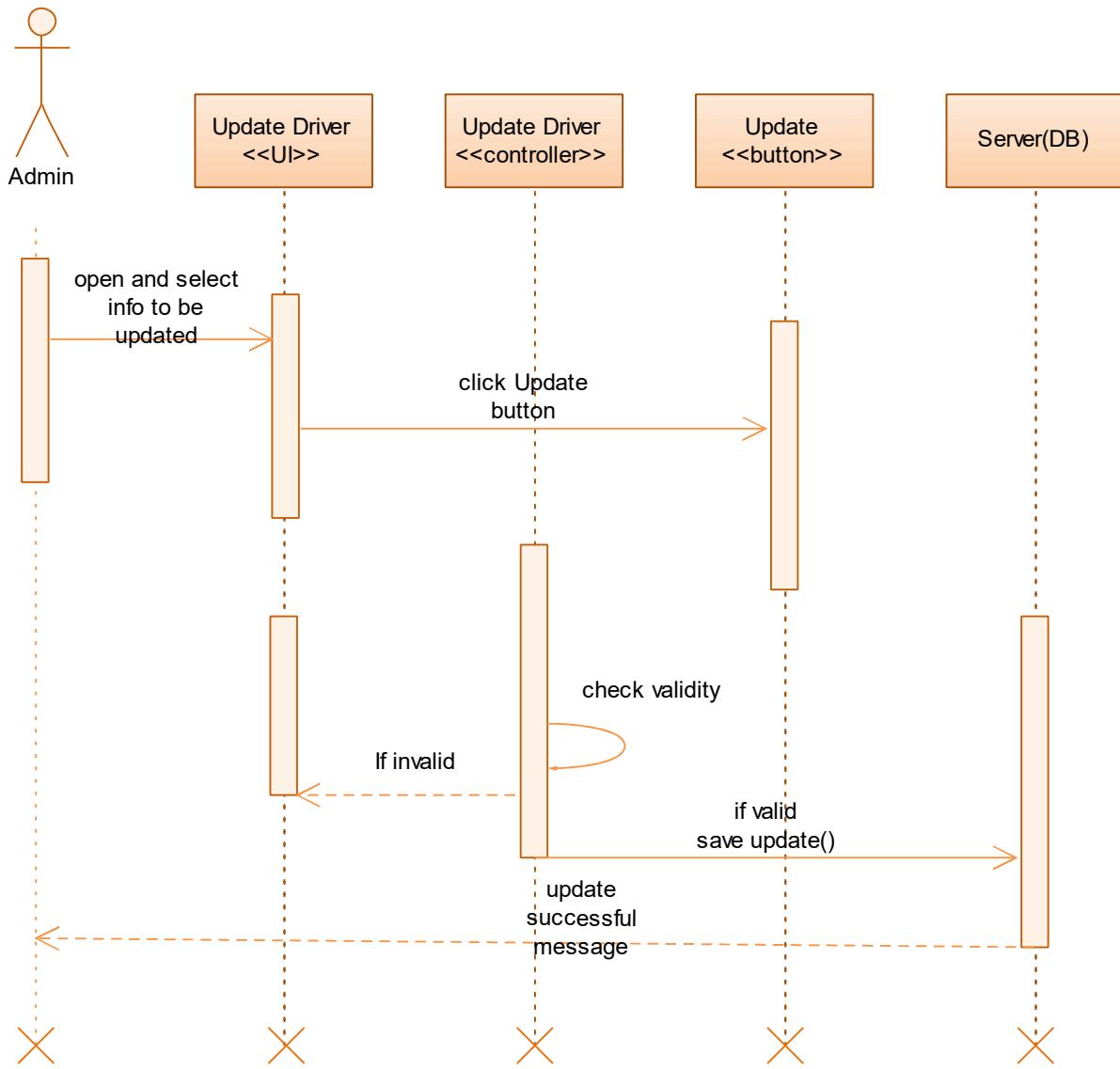
**Figure 17** sequence diagram for remove taxi



**Figure 18** sequence diagram for register driver



**Figure 19 Sequence diagram for Remove Driver**



**Figure 20 Sequence diagram for Update Driver**

## 2.13 Analysis level of class diagram

Class diagram is static model that shows the classes and the relationships among classes that remain constant over the time. Class is the main building block of class diagram, which stores and manages information in the system. Class diagram will be the building block of our system that we are going to develop. Class diagrams show the objects the system is comprised of and how they are interrelated.

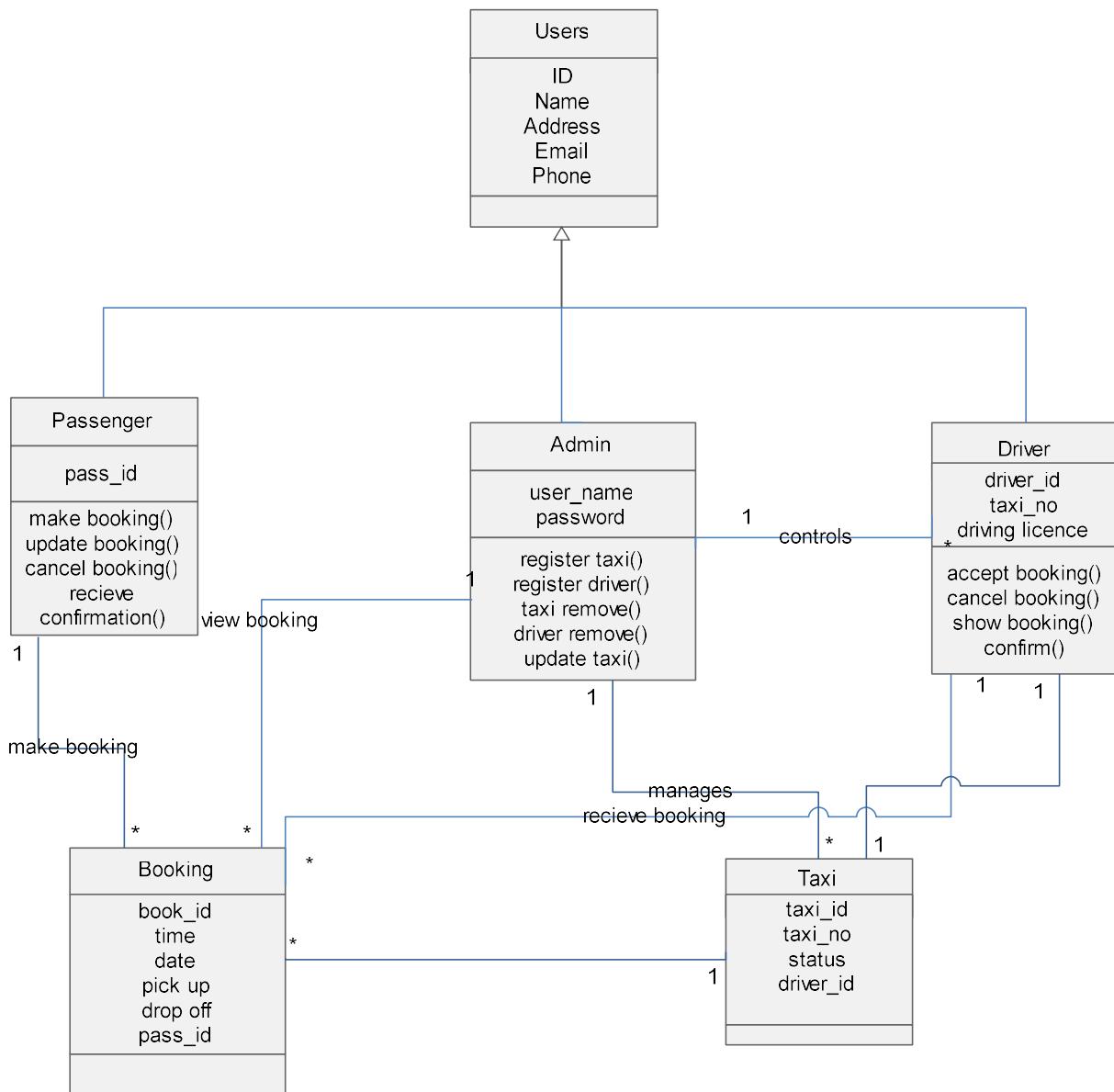


Figure 21 conceptual modelling: class diagram

## 2.14 State chart diagram

A state chart diagram is a view of a state machine that models the changing behavior of a state. State chart diagrams show the various states that an object goes through, as well as the events that cause a transition from one state to another. The common model elements that state chart diagrams contain are:

- States
- Start and end states
- Transitions

A state represents a condition during the life of an object during which it satisfies some condition or waits for some event. Start and end states represent the beginning or ending of a process. Some of the system state chart diagram that our system have seems like the following.

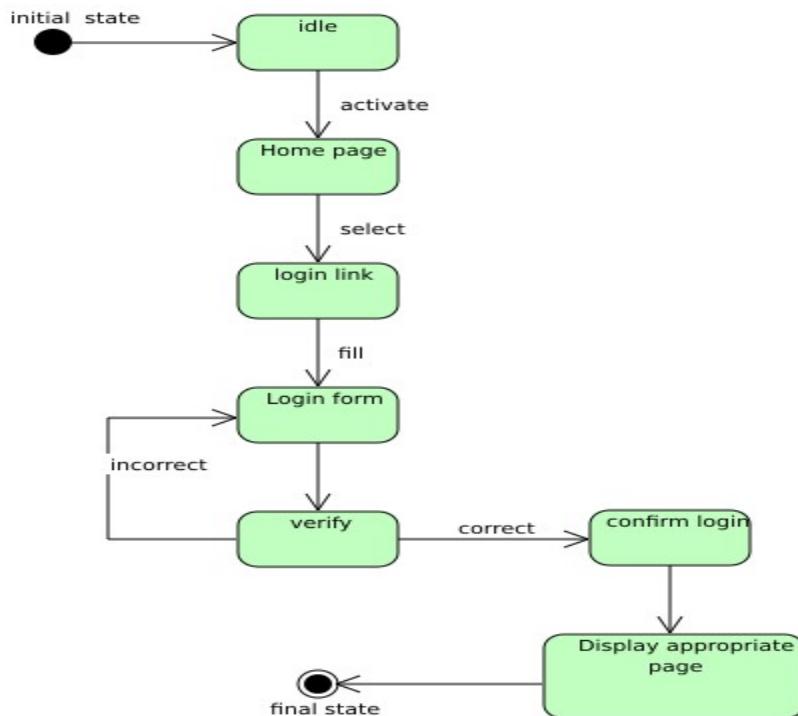


Figure 22 state chart diagram for login

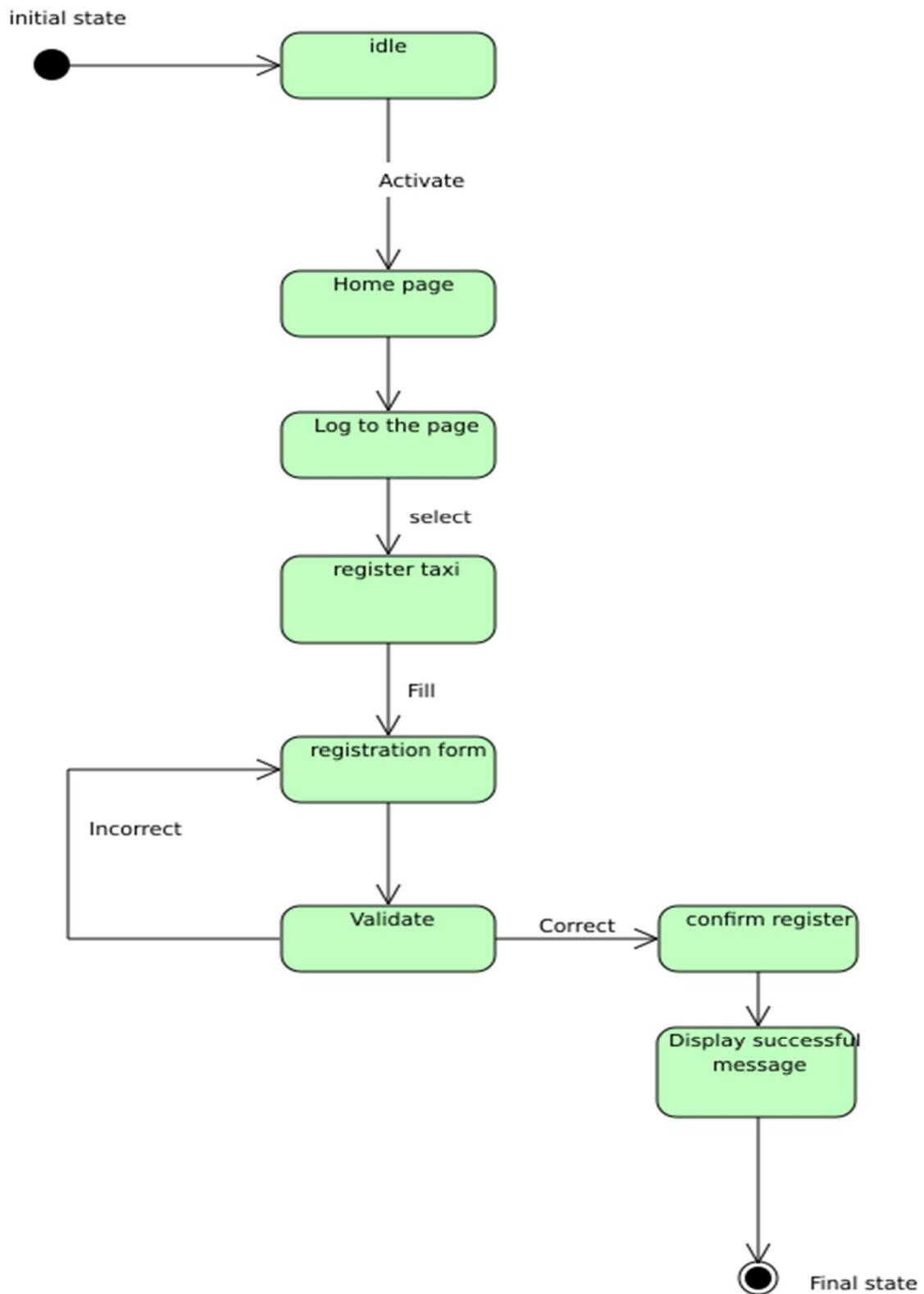


Figure 23 state chart diagram for taxi registration

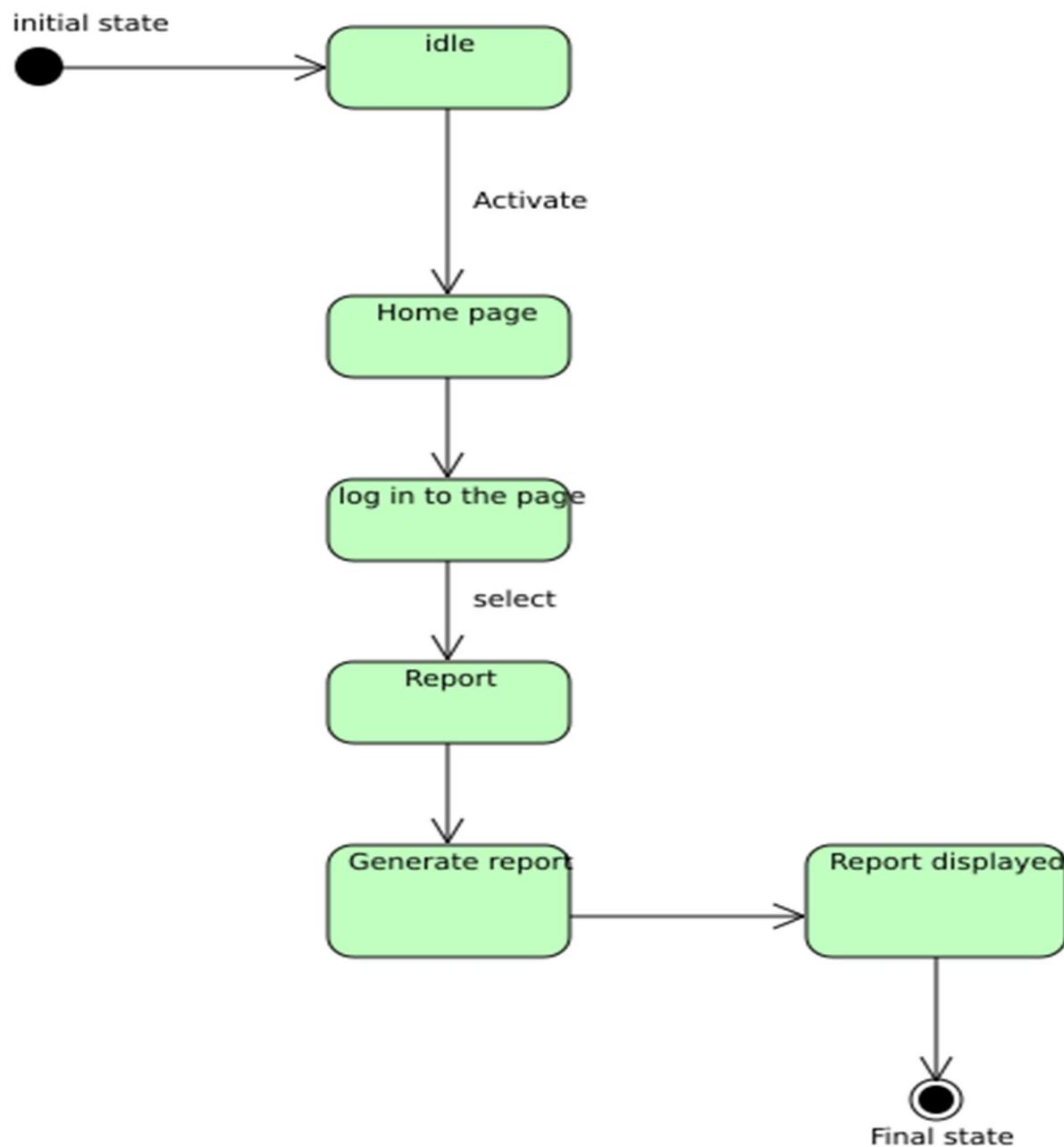
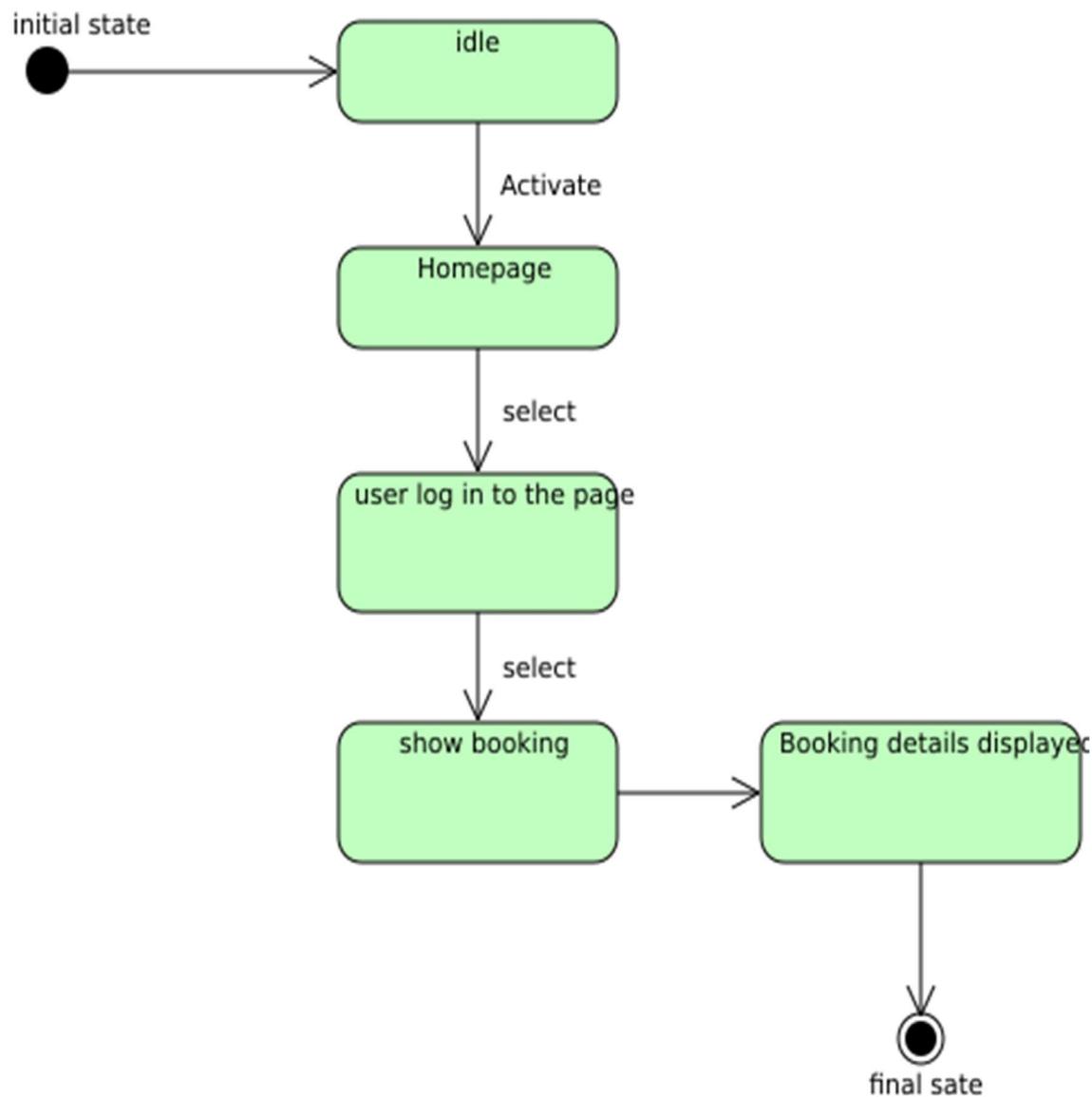


Figure 24 state chart diagram for Generate report



*Figure 25 state chart diagram for show booking details*

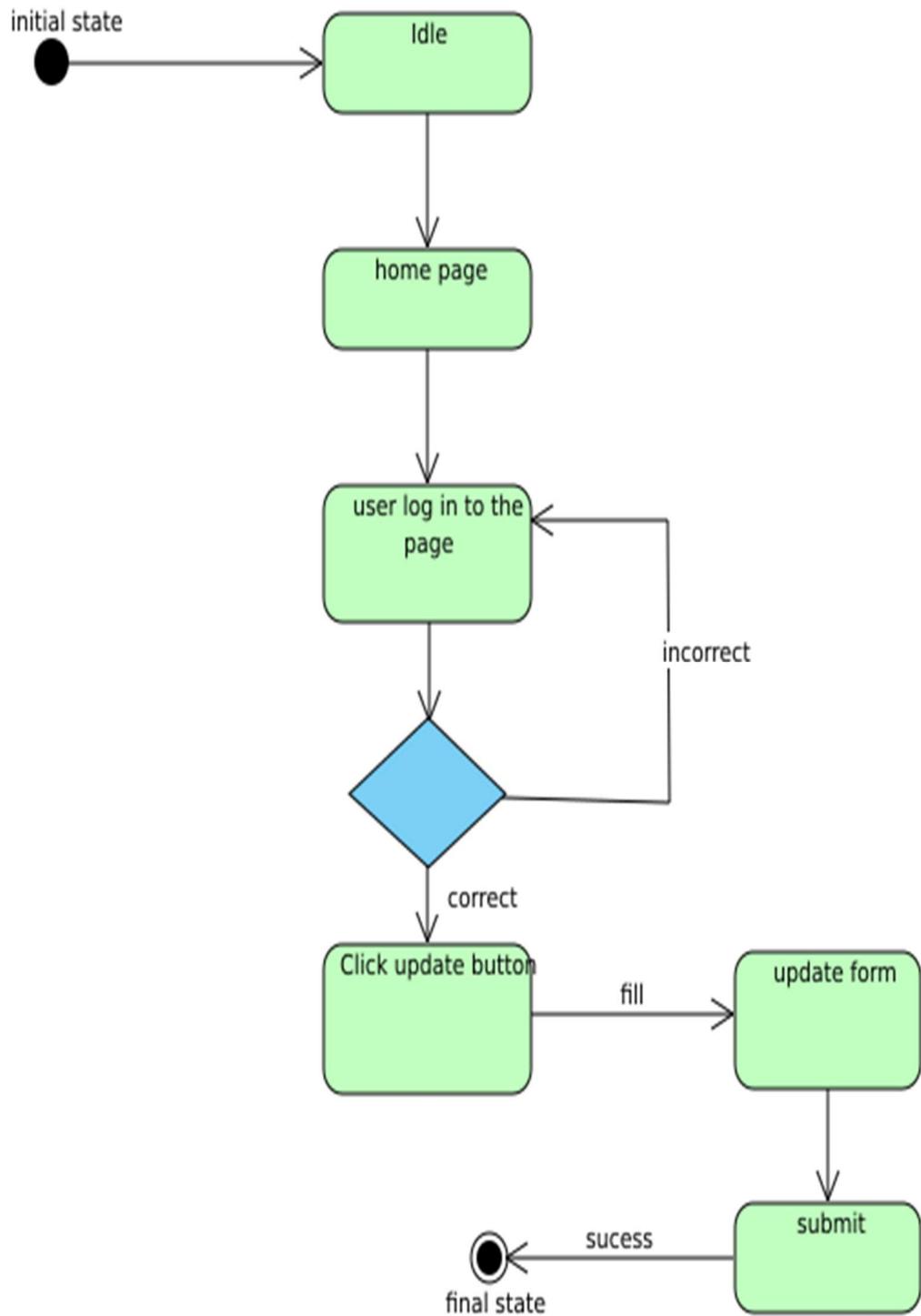
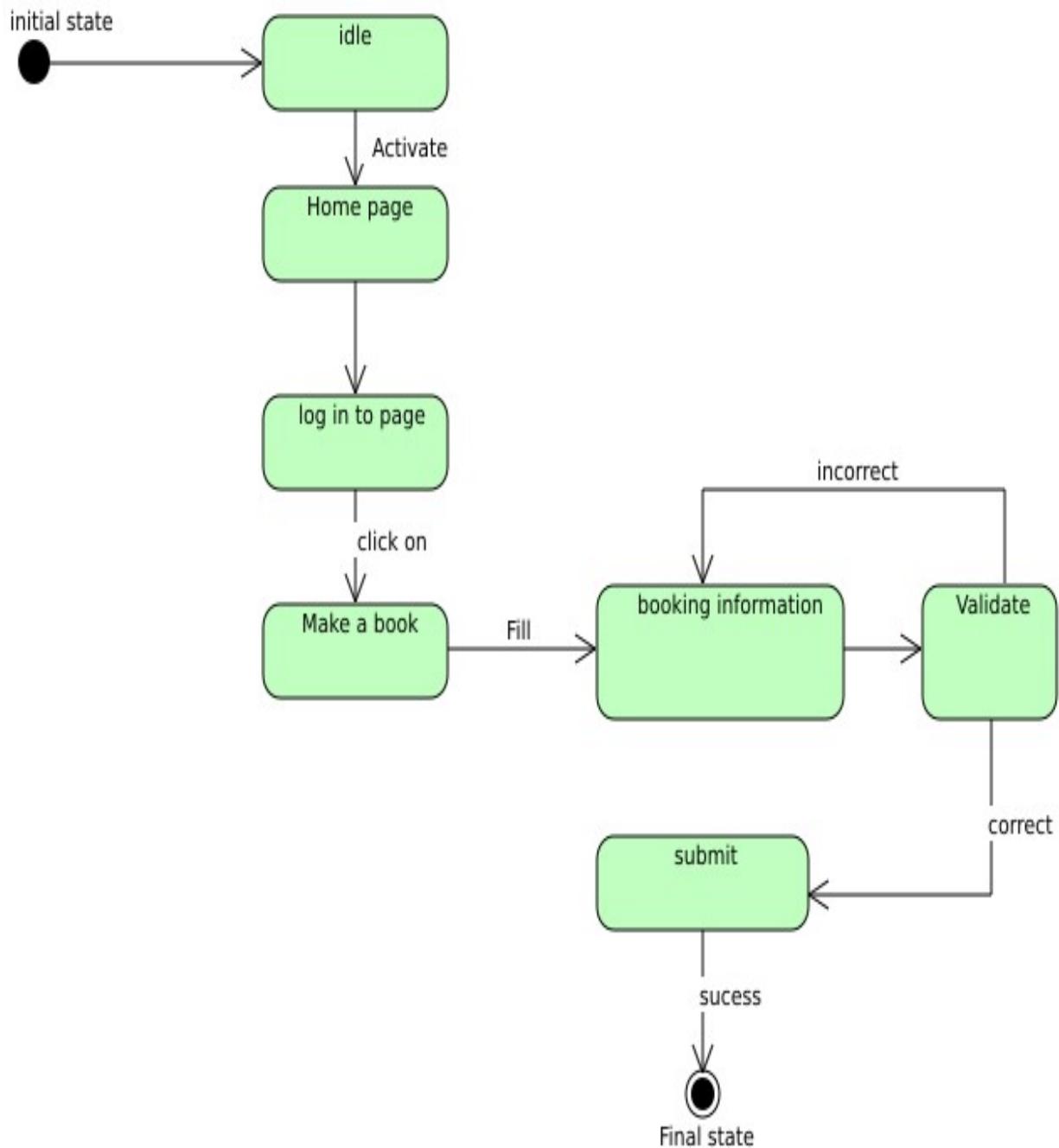


Figure 26 state chart diagram for driver info



**Figure 27 state chart diagram for making a book**

## 2.15 Key abstraction with CRC analysis

Collaboration diagram represent a combination of information taken from class, sequence, and use case diagrams describing both static structure and dynamic behavior of a system. And it also shows some data flows between objects and the interaction caused between them. Examples of the data flows among objects were show below.

**Table 16 Collaboration modelling for login**

Login	
user name password	Users (Administrator, Drivers and taxi) form
Enter pass Enter user Reset()	

**Table 17 Collaboration modelling for Taxi Registration**

Administrator	
User name User pass	Drivers
Register taxi() Delete taxi() View report() View reservation() View registered taxi()	Taxi

**Table 18 Collaboration modelling for make booking**

Booking	
Book id	
Date	Booking form
Pick up	Passengers
Drop off	Drivers
Fill info()	
Return error()	
Book taxi()	

**Table 19 Collaboration modelling for passengers**

Passenger	
pass_id	
F_name	
L_name	
Address	Book form
Email	Drivers Taxi
Fill info()	
Make booking()	
Cancel booking()	
Reset info()	
Generate error()	

**Table 20 Collaboration modelling for passengers**

Drivers	
Driver_id	
F_name	
L_name	
Address	Admin
Email	Passenger
Driving license	
Drive taxi() Cancel booking() View booking() Confirm message()	

**Table 21 collaboration modelling for taxi**

Taxi	
taxi_id	
taxi no	
status	
	admin, drivers and passengers

## **2.16 Logic model and pseudo code**

One common problem-solving technique includes analyzing a problem, outlining the Problem requirements and designing steps called an algorithm. Algorithms used to solve the problem. We also write algorithm for some basic method using human language (Pseudo code) in the following way.

Login in to the system

    Login ()

    Display Login Form

    Accept data from the Driver

    If "Login" button is clicked {

        Open database

        Open table

        Check if user type and password is matched

        If the if user type and password matched {

            Display user's home page}

        Else If {

            Display error message}

        Close the table

        Close the database}

    Else if "clear" button is clicked

    {

        Clear all the inputs and refresh the form and display login form

    End function

}

Reserve New Driver

Reserve New driver ()

Display Reserve driver Form

Accept input from the driver

If "reserve" button is clicked

{

Check for validity of data entered

If invalid data is entered prompt the user to correct the error by displaying error message }

If valid {

Open database

Open passenger table

Check the uniqueness of the passenger data entered

Else if the data is not unique

{

Prompt the passenger to enter another data

}

Else {

Reserve the information filled out on the form into the table}

Save change made to the table

Close the table

Close the database

Display message "drivers reserve successfully"

}

Else if "clear" button is clicked

{

Clear all the data and refresh the form and display the form

## **Chapter Three**

### **System Design**

#### **3.1 Introduction**

The purpose of designing is to show the direction how the system is built and to obtain clear and enough information needed to drive the actual implementation of the system. It is based on understanding of the model the software built on. The objectives of design is to model the system with high quality. Implementing of high quality system depend on the nature of design created by the designer. If one want changes to the system after it has been put in to operation depends on the quality of the system design. So if the system is design effectively, it will be easy to make changes to it.

#### **3.2 Purpose and Goals of design**

The design goals represent the desired qualities the system should have and provide a consistent set of criteria that would be taken into consideration when making design decisions. The following are mentioned as the design goals of “android based Taxi Booking system”.

**Security:** The system should authenticate it’s users by prompting them to enter user name and password in order to get access to the system.

**Extensibility:** the system should allow any additional services easily if needed, in other words it should not be difficult to extend the system if additions are necessary.

**Availability:** the system should be available every time the user needs to access it.

**Usability:** the system should have user friendly user interface to allow the user to interact with the system easily.

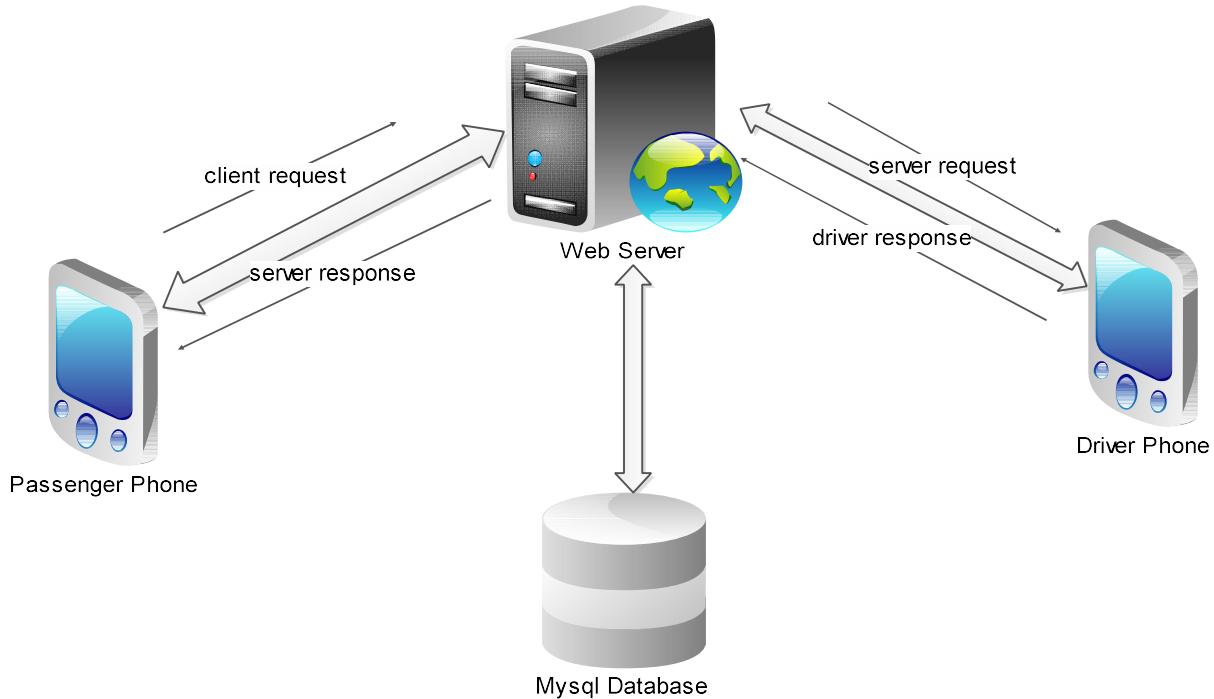
**Portability:** the system should be able to run on any mobile that supports mobile applications.

**Performance:** the main performance measure for a project is time, so the system should give fast responses for user requests.

### 3.3 Proposed software architecture

The architecture which will be used for the proposed system is a two tier architecture where the client or the user side is a mobile phone containing user interfaces like data entry interfaces, it is used to display information to the user. User directly interacts with the system through the interfaces on this layer.

The data layer or the database is responsible for storing all information needed for the system to function correctly.



*Figure 28 proposed software architecture*

### 3.4 Component modelling

Systems may be built from components in component based architecture. Component diagram shows how objects (classes) in our system are grouped together and form components. The components interact with each other either in giving service to other components or requesting service from other component.

#### 3.4.1 Component Diagram for Admin Functionality

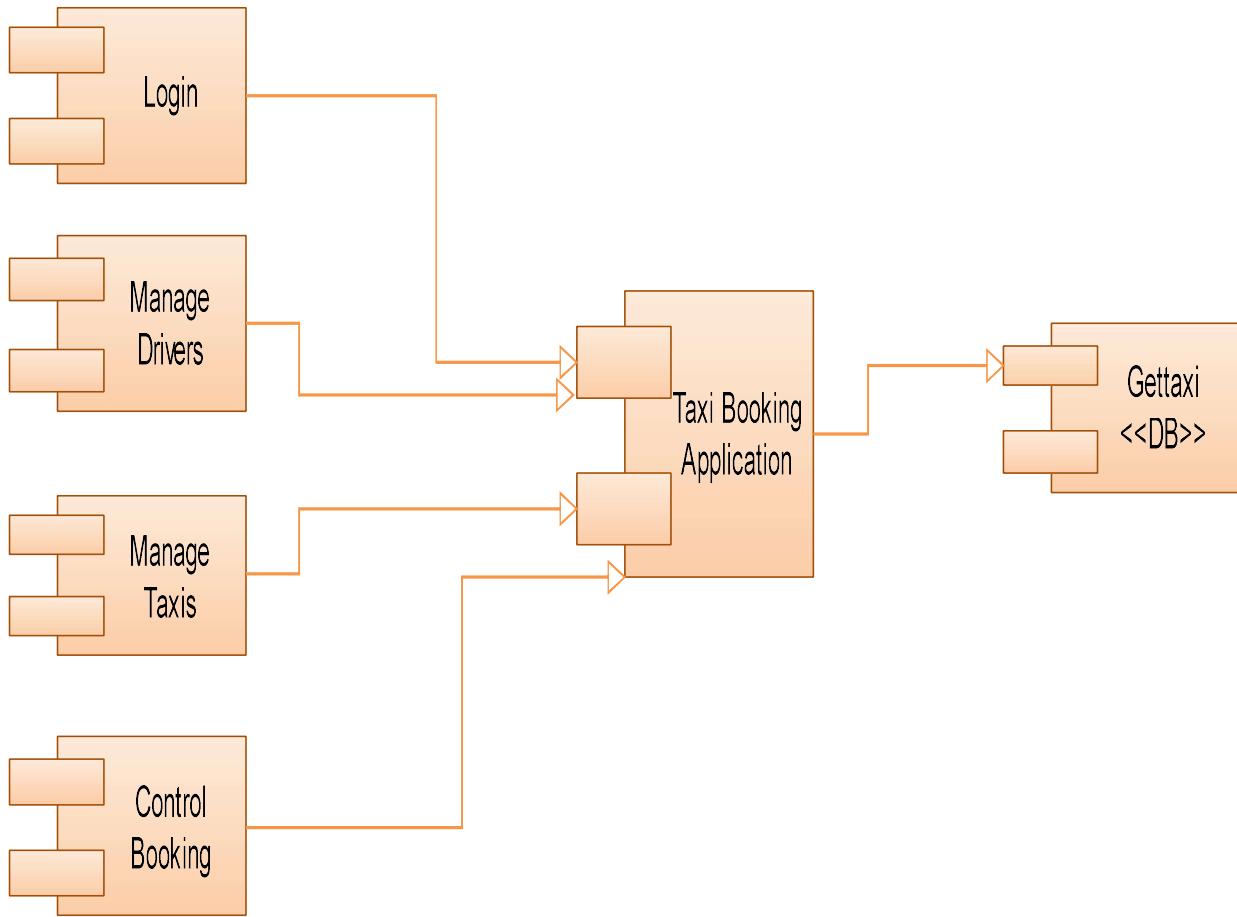


Figure 29 Component diagram for admin functionality

### 3.4.2 Component diagram for Customers (passenger and driver)

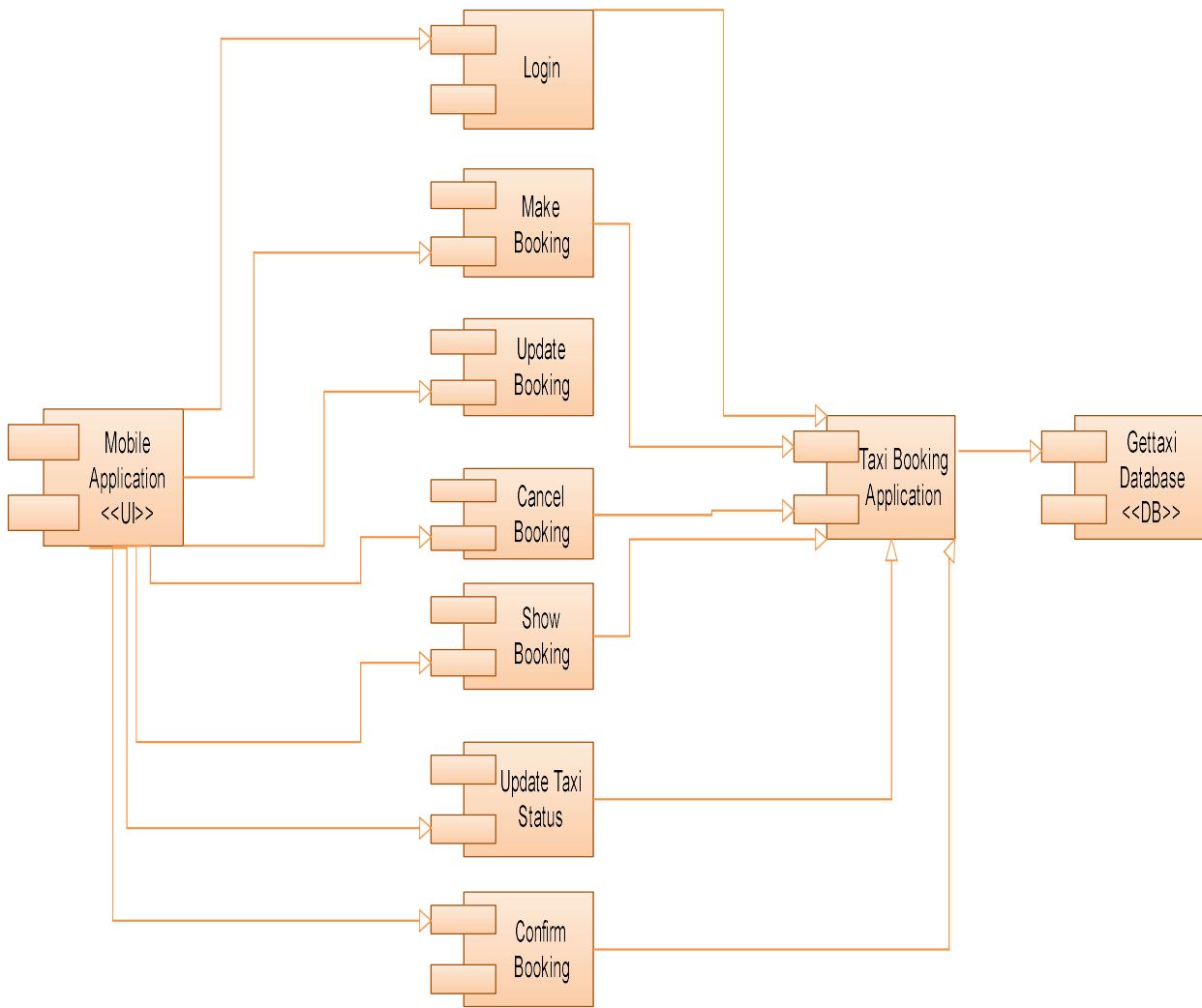
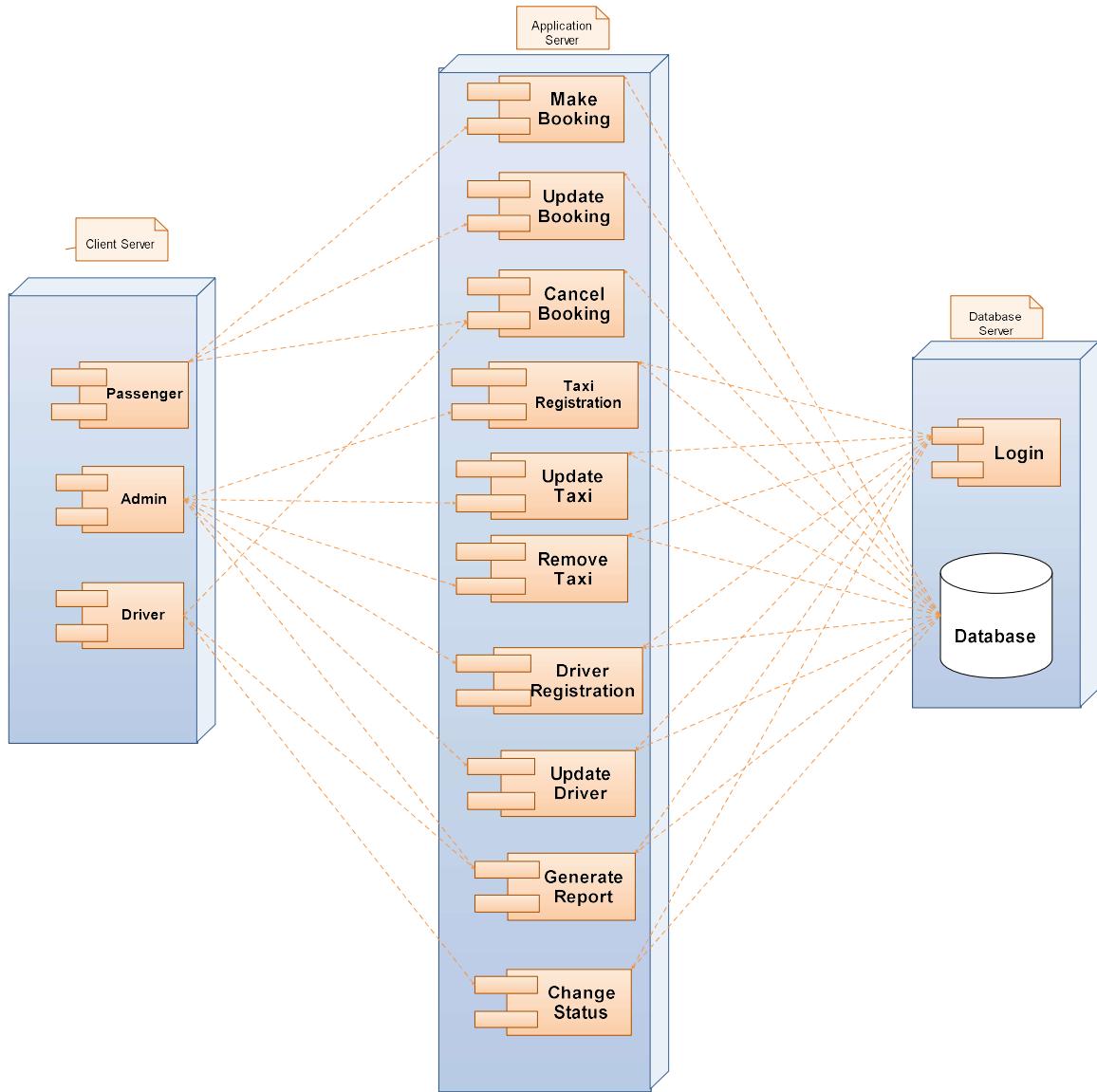


Figure 30 component diagram for passenger and driver

### 3.4.3 Deployment Diagram

Deployment diagram is a static view of the run-time configuration of hardware nodes and the software components that run on those nodes. It shows the hardware of system, the software that is installed on that hardware.



**Figure 31 deployment diagram**

## 3.5 Detail Design

### 3.5.1 System Class Diagram

The class diagram is a static diagram that represents the static view of an application. It describes the attributes and operations of a class and also the constraints imposed on the system. So class diagram is the diagram that used to express the class in our system. This class diagram contains conceptual class or main class. It contains one class has relation to another class, attribute and operation of the class.

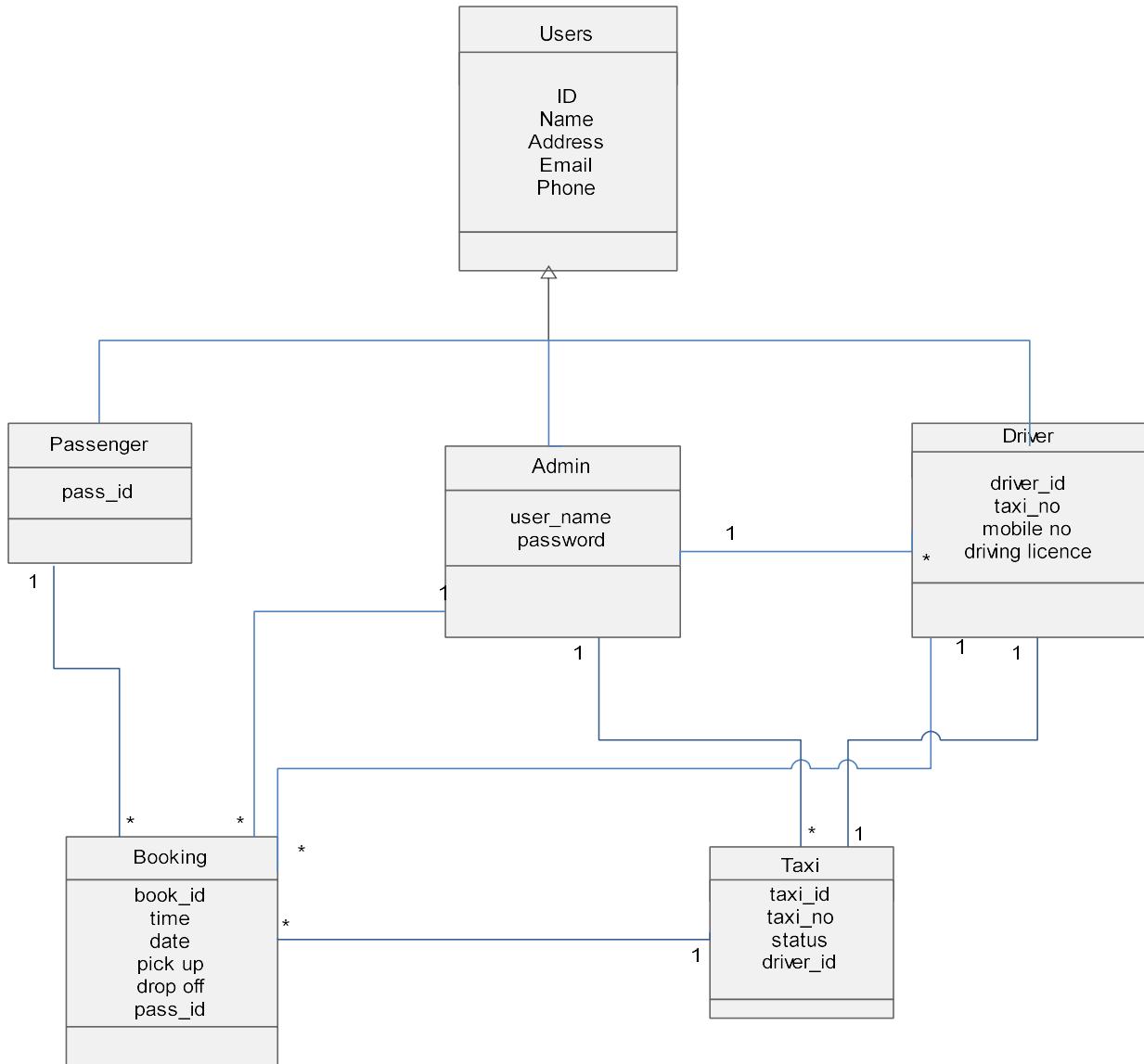
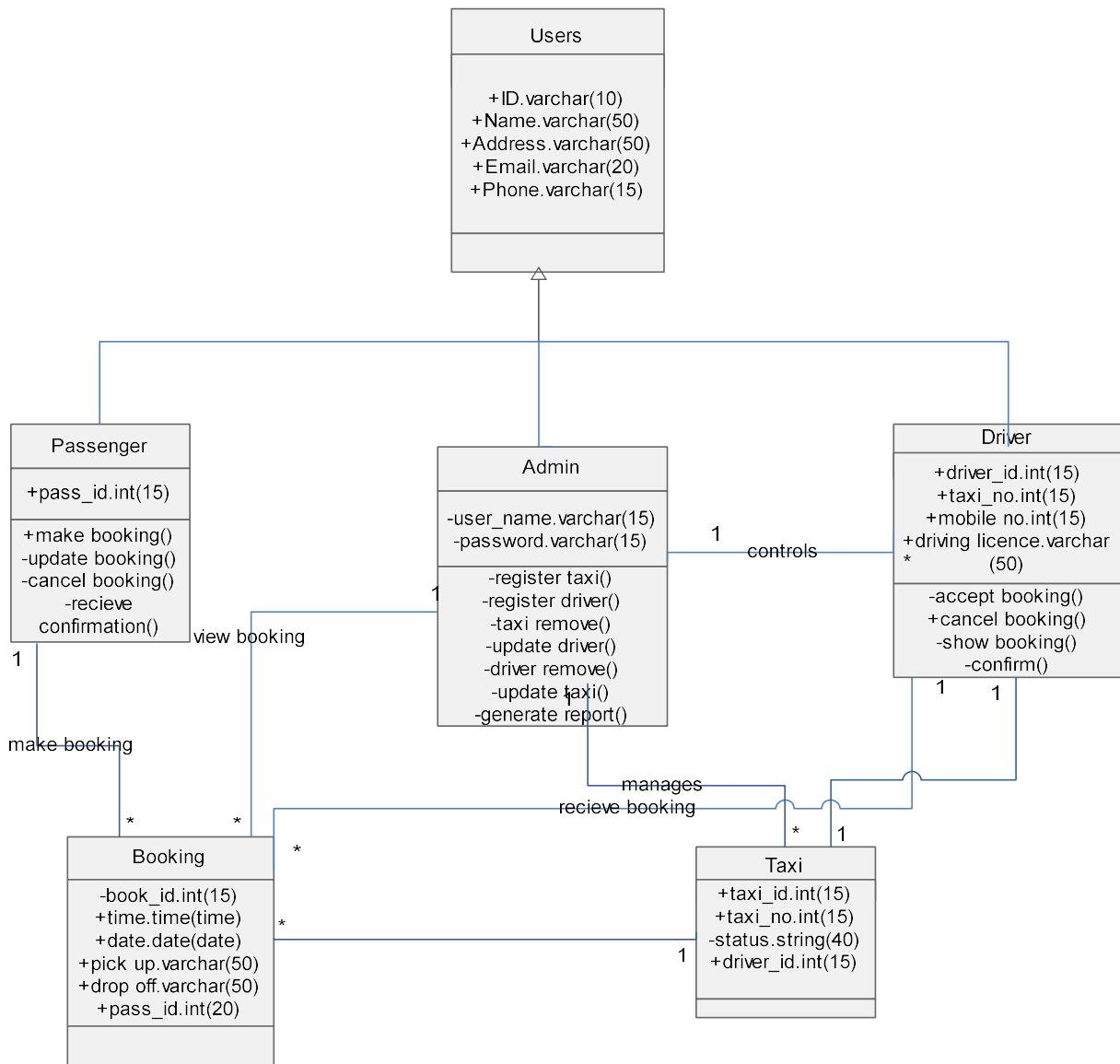


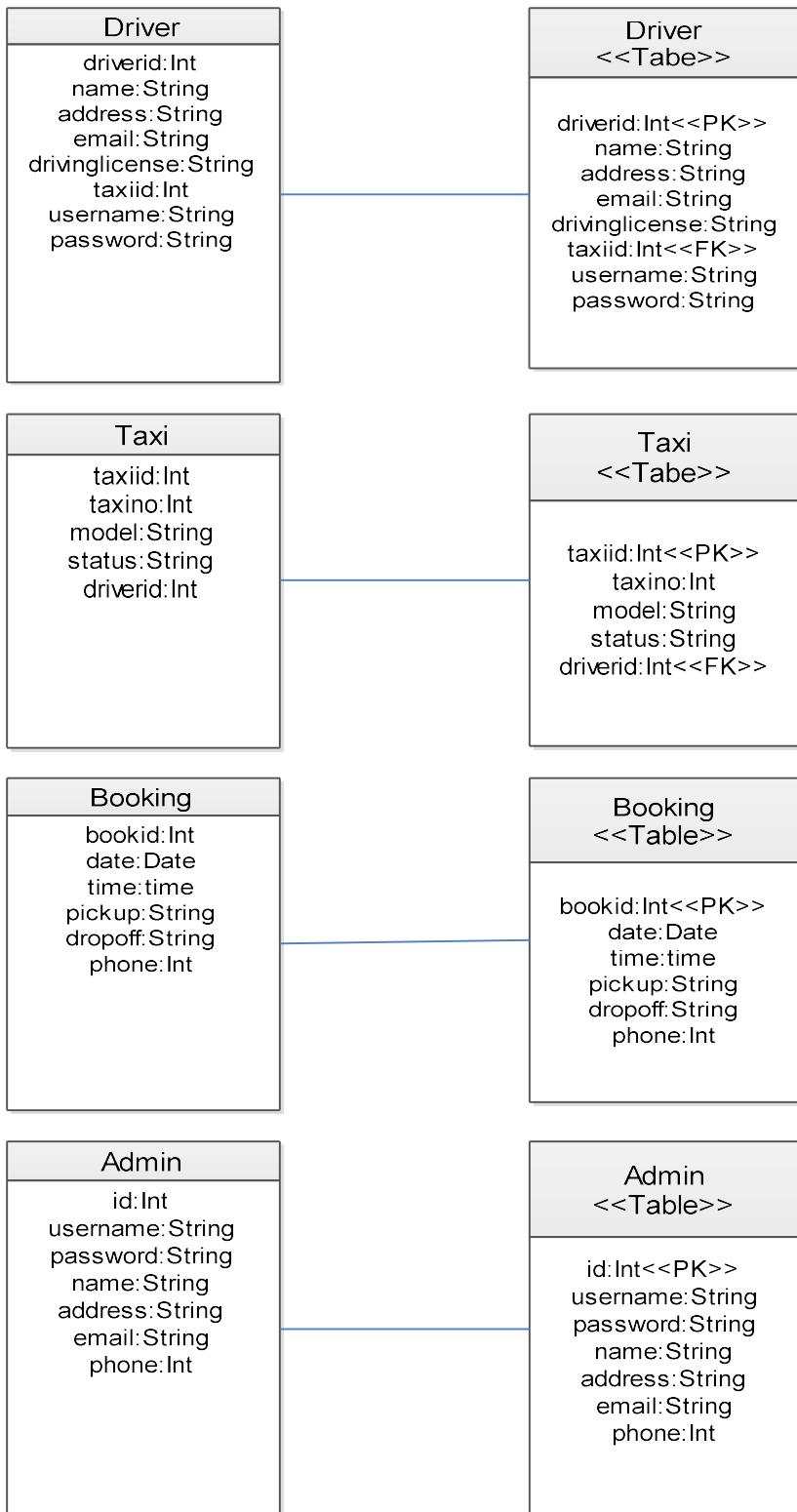
Figure 32 Class modelling diagram

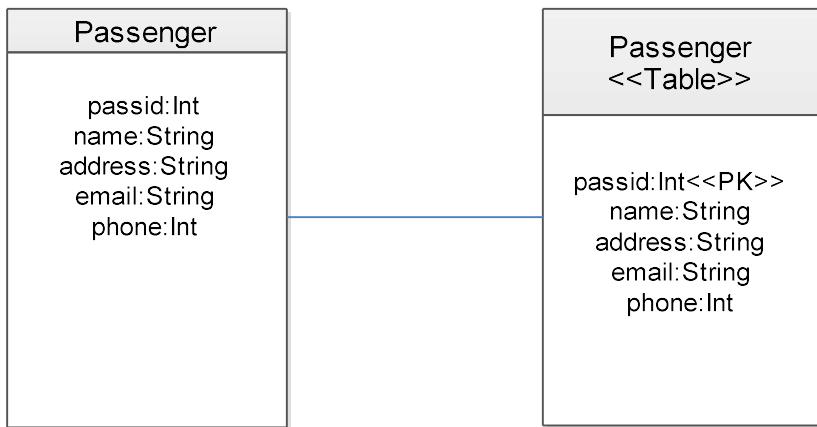
### 3.5.2 Persistent model

*Figure 33 persistent model*



## Mapping





**Figure 34 Mapping**

### 3.6 User Interface Design

The screenshot shows the 'Online taxi ordering and booking system' user interface. At the top, there is a header bar with links for Home, About us, Login, Help, and Contact. Below the header, a sub-header reads 'System administrator page'. On the left side, there is a sidebar with a yellow taxi icon and three menu sections: 'Manage Driver' (with options: Register driver, update driver, View Driver detail), 'Manage Taxi' (with options: Register Taxi, update Taxi, View Taxi details, Remove Taxi), and 'Control Booking' (with options: View Booking tool, Time based booking, Location based booking, Other Booking). To the right of the sidebar, there is a table titled 'System administrator page' with the following data:

ID	Name	Address	Email	Driver license	Taxi ID	User Name	Password	Action
1234	Abraham	Bahir dar	abrihs...	Renewd	2131	abrihs...	12345678	update
1235	Bahirew	Gondar	abrihs...	Renewd	2132	bahirew	12345679	update

**Table 22 Update driver user interface**

Online taxi ordering and booking system

---

[Home](#) [About us](#) [Login](#) [Help](#) [Contact](#)

---

System administrator page

---



**Manage Driver:**  
Register driver  
update driver  
View Driver detail

**Manage Taxi:**  
Register Taxi  
update Taxi  
View Taxi details  
Remove Taxi

**Control Booking:**  
View Booking tool  
Time based booking  
Location based booking  
Other Booking

Name :

user Name :

Password:

Address:

Email

**Table 23 Register driver user interface**



**Table 24** Update taxi status user interface



Table 25 Edit Booking user interface



Table 26 Make Booking user interface

### **3.7 Access control and security**

In the system, different actors have access to different functionality and data. The system has different users such as passengers, administrator and driver.

**The passengers:** Represents unauthenticated user. It is used by without Login subsystems to book taxi. And also they responsible to fill booking form, select, search booking.

**Driver:** Is an authenticated user who is responsible to accept and reject booking, change taxi current status and cancel booking.

**Administrator:** Is an authenticated user. It is used by login subsystem. He or she is responsible for managing drivers, taxi and control booking.

	admin	Driver	passenger
login	✓	✓	
Receive Notification	✓	✓	✓
Send Notification	✓	✓	✓
Verify User	✓		
Accept Request	✓	✓	
Reject Request	✓	✓	
Report	✓	✓	
Alert user	✓	✓	
Control booking	✓		
Manage driver	✓		
Manage taxi	✓		
Accept booking		✓	
Reject booking		✓	
Change status		✓	
Book taxi			✓
Search booking			✓

## **References**

- S. Cheng and X. Qu, X. A Service Choice Model for Optimizing Taxi Service Delivery, Research Collection School of Information Systems, v. 209, 2009.
- Spatio-temporal Efficiency in a Taxi Dispatch System. Santani, D., Balan, R. K. and Woodard, C. J. Research Collection School of Information Systems (SMU Access Only). Paper 9. 2008.
- Virrantaus, K., Markkula, J., Garmash, A., Terziyan, V., Veijalainen, J., Katanosov, A., and Tirri, H. Developing gissupported location-based services. InWeb Information Systems Engineering (2001), IEEE, pp. 66\_75.
- Consortium, O. G. Open location services 1.1, 2005.
- TOGO CABS - AN ANDROID PHONE CAB RESERVATION APPLICATION by NIRUPAMA MRINALINI MEESALA B.E., Muffakham Jah College of Engineering and Technology, Hyderabad, India, 2008
-