

**SOFTWARE ENGINEERING**

**ISOE**

**SOFTWARE DESIGN**

**SPECIFICATION**

**HOT-WHEELZ**

**GROUP MEMBERS-**

Hrutvik Nagrale (IIT2018088)

Vineet Kumar (IIT2018096)

Anshul Ahirwar (IIT2018099)

Ankit Kumar (IIT2018100)

## TABLE OF CONTENT

1.	Introduction.....	3
1.1.	Purpose of this document.....	3
1.2.	Scope of the development project.....	3
1.3.	Definitions, acronyms, and abbreviations.....	4
1.4.	References.....	4
1.5.	Overview of document.....	4
2.	Conceptual Architecture/Architecture Diagram.....	5
2.1.	Overview of modules / components.....	6
2.2.	Structure and relationships.....	6
2.2.1.	Admin side.....	6
2.2.2.	Buyer/Renter.....	7
2.2.3.	Seller side.....	8
2.3.	User interface issues.....	8
3.	Logical Architecture(Use Case,Class diagram,Sequence Diagram).....	10
3.1.	Logical Architecture Description.....	27
3.1.1.	Class Diagram Explanation.....	27
3.1.2.	Sequence Diagram Explanation.....	27
3.1.2.1.	FAQ.....	27
3.1.2.2.	About Section.....	28
3.1.2.3.	Register Page.....	28
3.1.2.4.	Login Page.....	28
3.1.2.5.	My Booking and book vehicle.....	28
3.1.3.	State diagram.....	29
3.2.	Class name:Login.....	31
3.2.1.	Method 1:postLogin().....	31
3.2.2.	Method 2:Admin_Login().....	32
3.2.3.	Method 3:Registration.....	32
4.	Pseudocode for components.....	32
4.1.	Class name:Login.....	32
4.2.	Class name:Admin Login.....	33
4.3.	Class name: Registration.....	34
4.4.	Class name:Logout.....	35
4.5.	Class name: Car Search.....	36
4.6.	Class name:Subscribe Newsletter.....	38
4.7.	Class name:Show Testimonial.....	39
4.8.	Class name:Admin Dashboard.....	39
4.9.	Class name:Admin Post Vehicles.....	41

# **The Software Design Specification**

## **1. Introduction**

The Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. This document includes case model, sequence diagrams, object behaviour models, and other supporting requirement information.

### **1.1 Purpose of this document**

This document will define the design of the website. It contains specific information about the classes and functions used. The interaction between classes to meet the desired specifications of the software.

### **1.2 Scope of the development project**

We describe what features are in the scope of the software and what are not in the scope of the software.

#### In Scope:

1. Browsing vehicles on the website.
2. One can buy or rent a vehicle directly from a dealer or from another owner, the gateway being our website.
3. Posting FAQs or queries on our website, which can be then answered by another individual, dealer or the website admin.
4. Providing a platform for an individual trying to sell his/her vehicle.
5. Dealers can display their vehicle and car modification accessories which could be then used to modify their vehicle taking it to the dealer.

#### Out of Scope:

1. Seller can't give critics on the profile of another seller.
2. Making the whole transaction online, from browsing to payment for the vehicle is not in the scope of this software. The concerned person must meet the seller for payment and vehicle verification.
3. Buyer can't just buy the modification parts they have to modify their vehicle along with it.

## 1.3 Definitions, acronyms, and abbreviations

### Acronyms and Abbreviations:-

- a. “Hot Wheelz<sup>®</sup>”: Copyrighted website name
- b. IEEE : Institute of Electrical and Electronics Engineering
- c. SDD: Software Design Document

### Definitions:

- a. Vehicle Customization : Altering a passenger vehicle substantially or to improve its Performance,often by replacing engine and transmission:mande into a personal styling Statement.
- b . Select : customers will select the option if they want to buy or rent that servies.
- c . Rent : pay someone to use something.
- d. Buy : does a person want to purchase the product.

## 1.4 References

### 1.4.1 IEEE SDS template

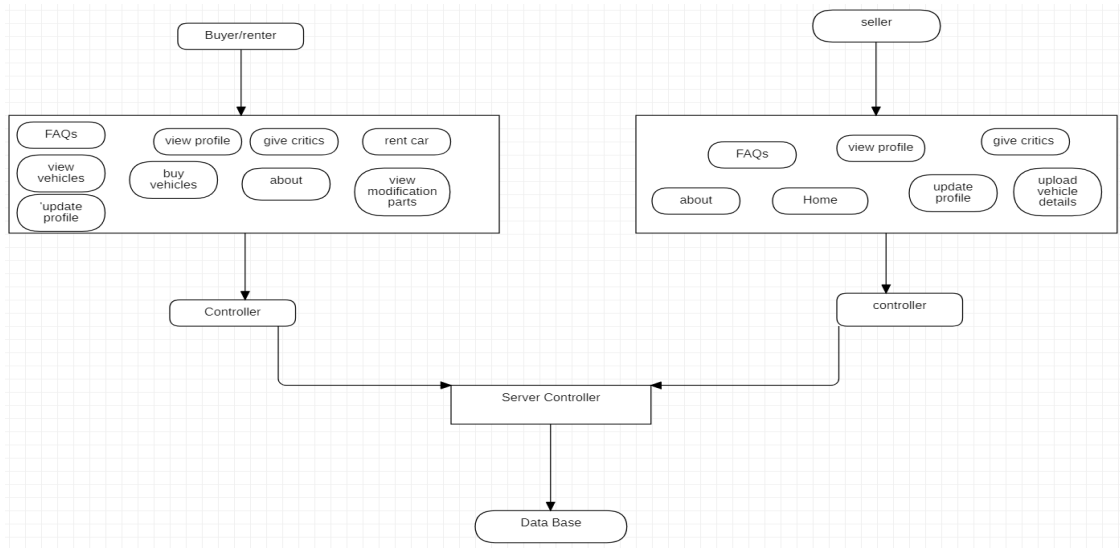
## 1.5 Overview of the document

This SDS is divided into 7 sections with various sub-sections. The sections are as follows:

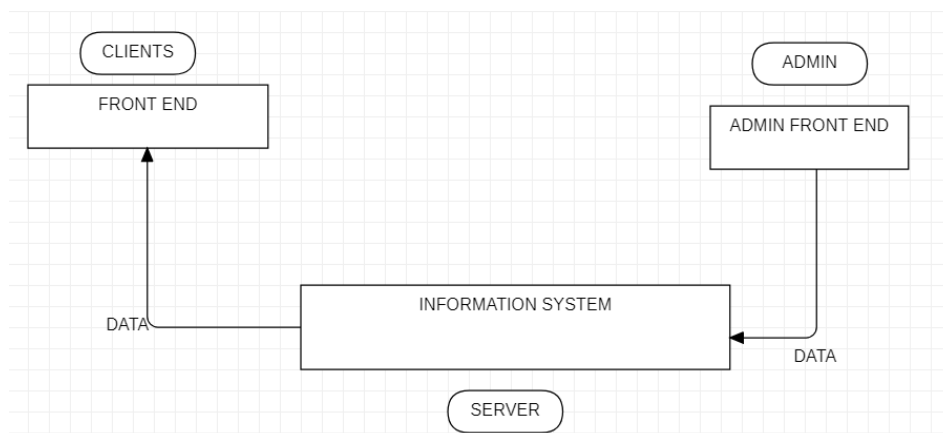
1. **Introduction:** describes about the purpose , scope of development project definitions and abbreviations used in this document.
2. **Conceptual Architecture/Architecture Diagram:** describes the overview of the components, modules, structure and relationships and the user interface issues.
3. **Logical Architecture:** describes the Logical Architecture Description and Components(Use Case diagram,Class diagram,Sequence diagrams)
4. **Execution Architecture** defines the number of processes,deployment views.
5. **Design Decision and Trade-offs :**describes the decisions taken along with the reason as to why they were chosen over other alternatives.
6. **Pseudocode for components:** describes pseudocode, as the name indicates.
7. **Appendices :**describes subsidiary matter if any.

## 2. Conceptual Architecture/Architecture Diagram

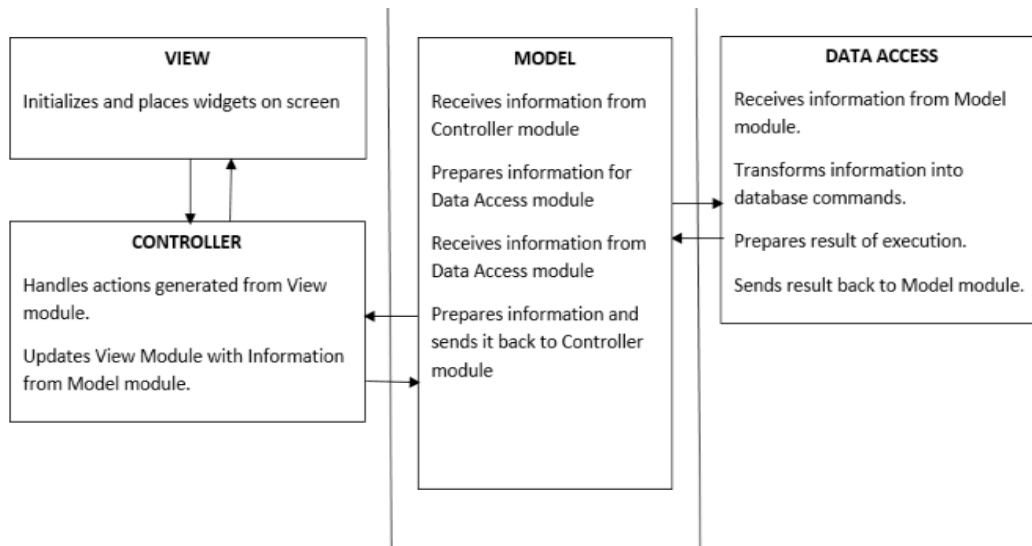
### Architecture Diagram 1:



### Architecture Diagram 2:



## 2.1 Overview of modules / components

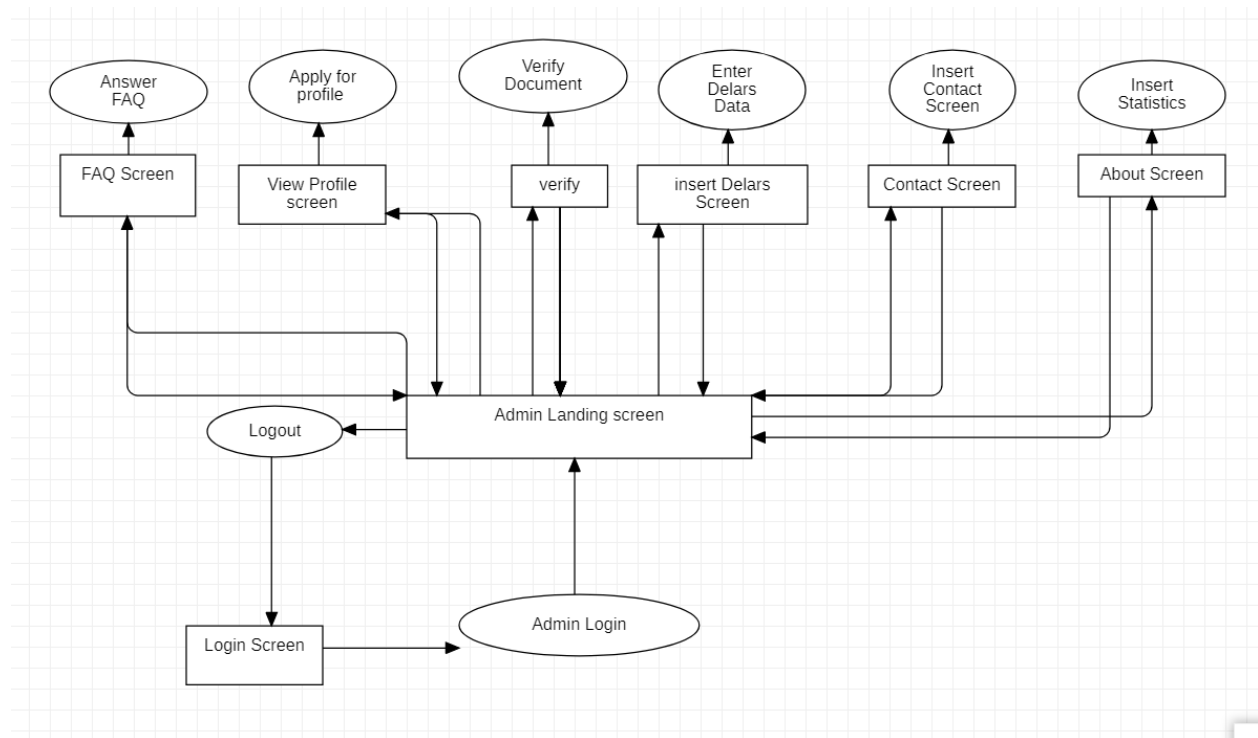


**NOTE:** The horizontal lines represent the separation of modules.

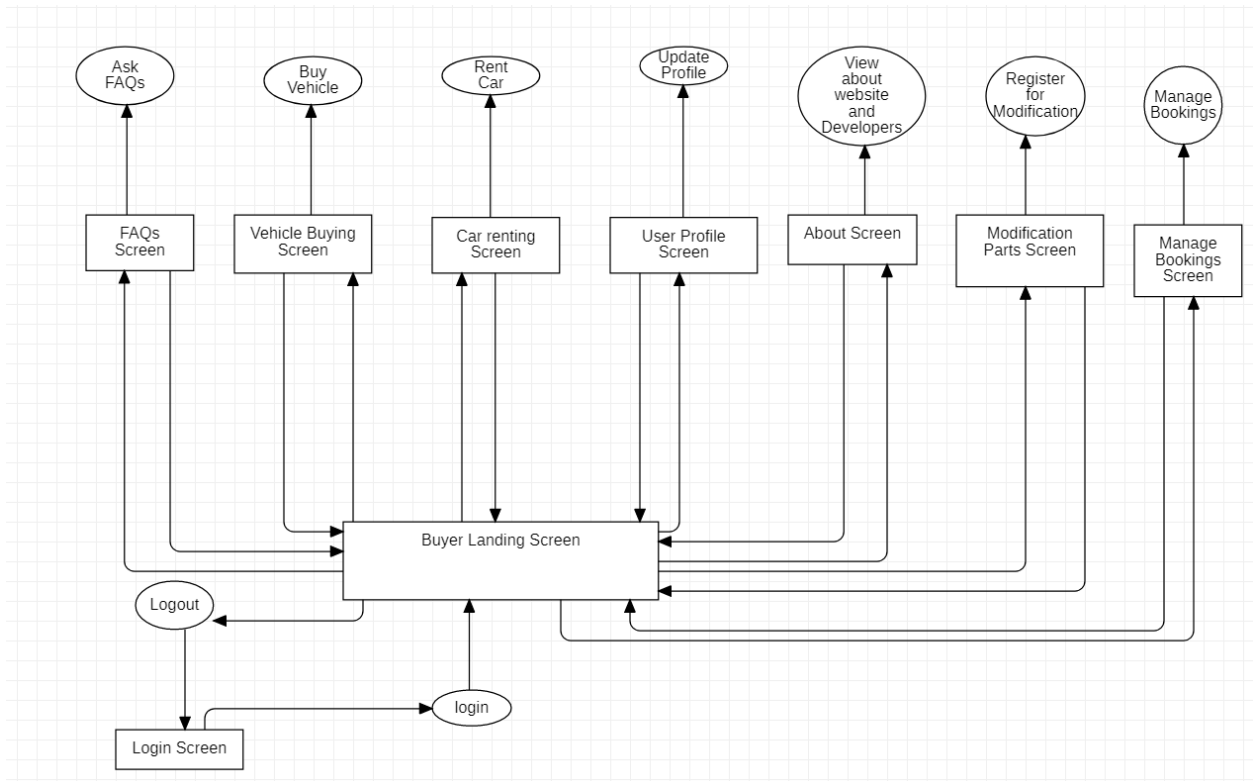
More than one box within the same section represents sub-modules.

## 2.2 Structure and relationships

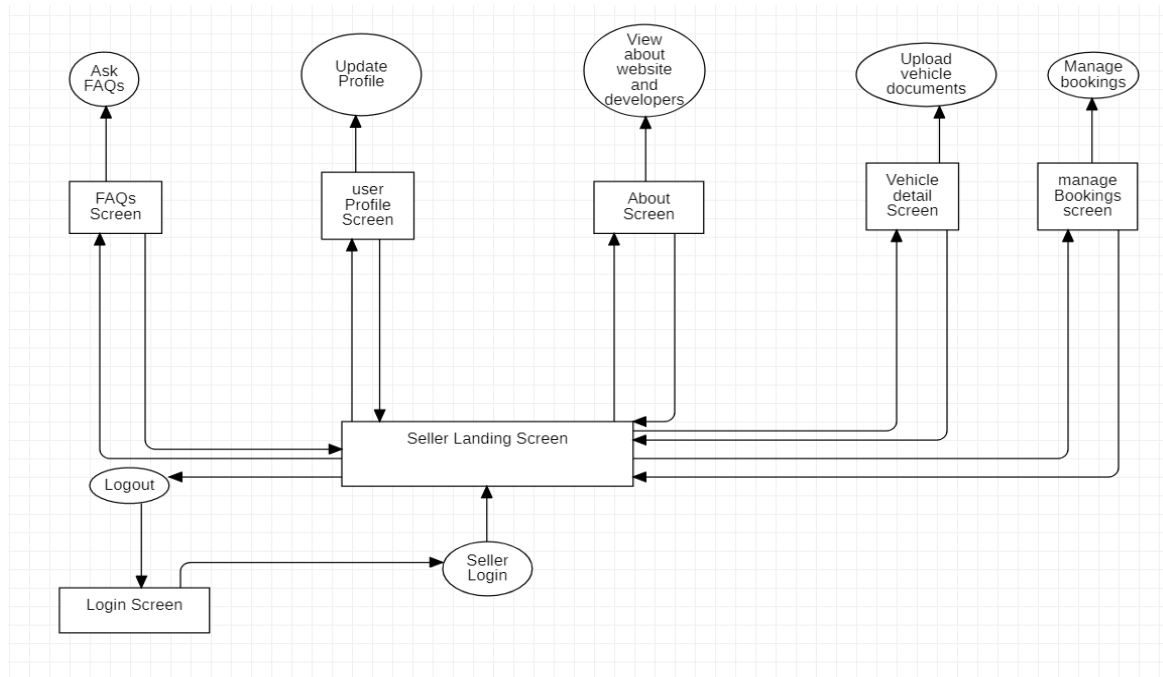
### 2.2.1 Admin's Side



## 2.2.2 Buyers/Renter Side



### 2.2.3 Seller Side



### 2.3 User interface issues

This section will address the User Interface issues as they apply to the following hypothetical users.

- User A is a 40-year-old male, who has decided to travel to a new place with his family. After reaching the destination he comes to know that the fares for taxis are very high and as his family has come to travel, explore and have fun they wouldn't prefer public transport. He is quite familiar with using online websites.

Since User A has come to travel with his family, he can rent a car for a particular time period from another individual who uses this website to provide vehicles for renting. He can search the nearest person who is willing to provide his car or he can even search for hubs maintained by dealers willing to provide cars for renting. He can then extract the contact number of the dealer or the person from the website after deciding which car to rent and then proceed with the rest of the processes like payment offline.

- User B is a 60 year - old woman which is that she has retired and want to travel the country with her kids so now they can take various package but thing is it will not a fun



for them at all as she is retired she have saving in her account so she can buy a car to travel as first of she is confused from where to buy so here comes our website where she can compare various car and see many dealers and also other second hand car price so She had good idea of the price car which can buy now she can reach the dealer herself

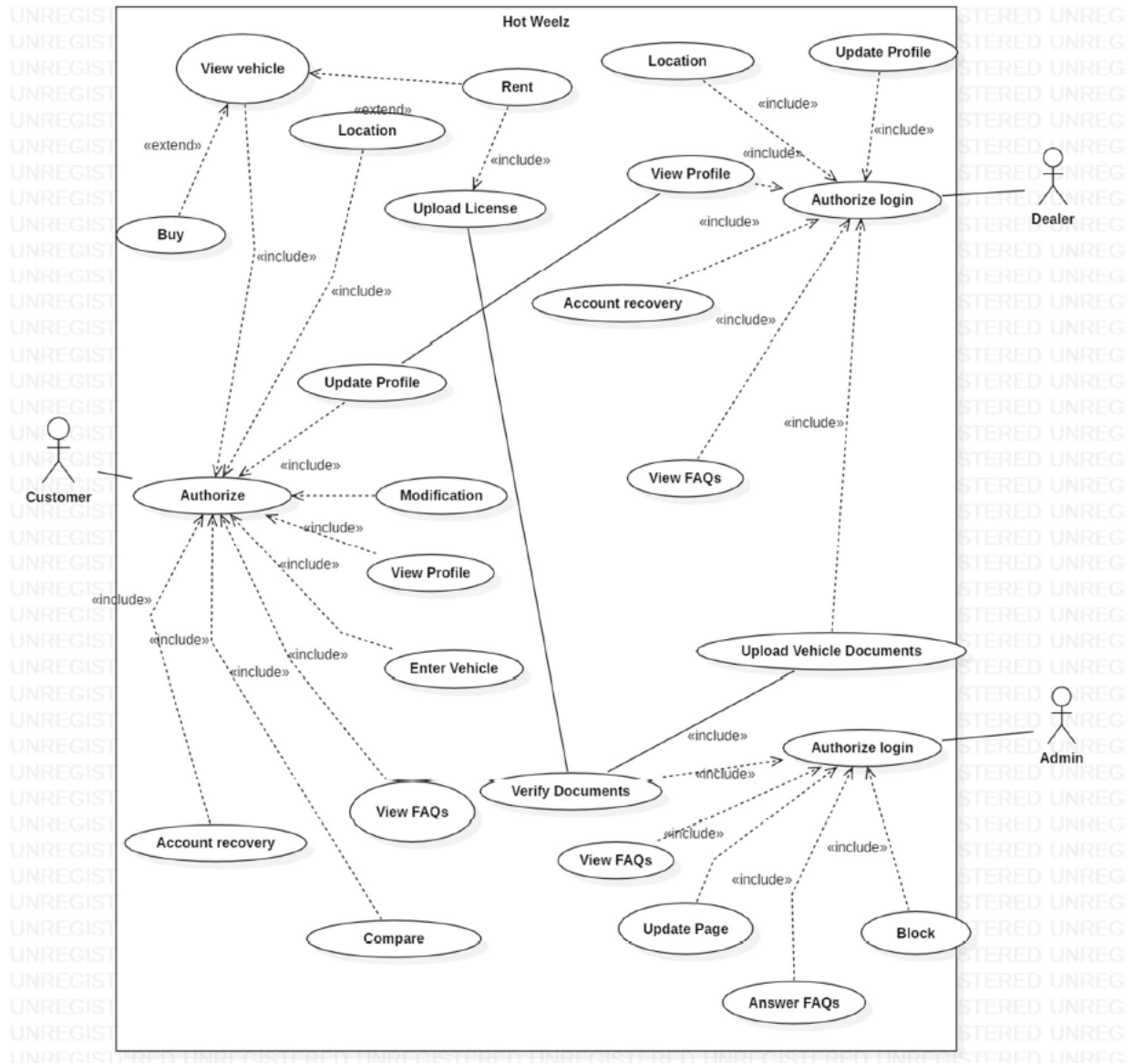
So now she buy her budget friendly car and which she can use to travel across the county After deciding to buy a car there are many processes involved in the middle until the car gets you so all the process can be taken care of on our site with the insurance company and all the others we provide the best.

- User C is a 20 year man who owns Bike and wants to modify silencer and seats like no automotive company does inbuilt manufacturing of such kind. So , he can look and compare on the website for different type of silencer and other specifications provided by different custom garage and can modify according to his likes considering his budget and other restrictions.
- User D is a 30-year-old man, who owns a car dealership and wishes to expand his business online.

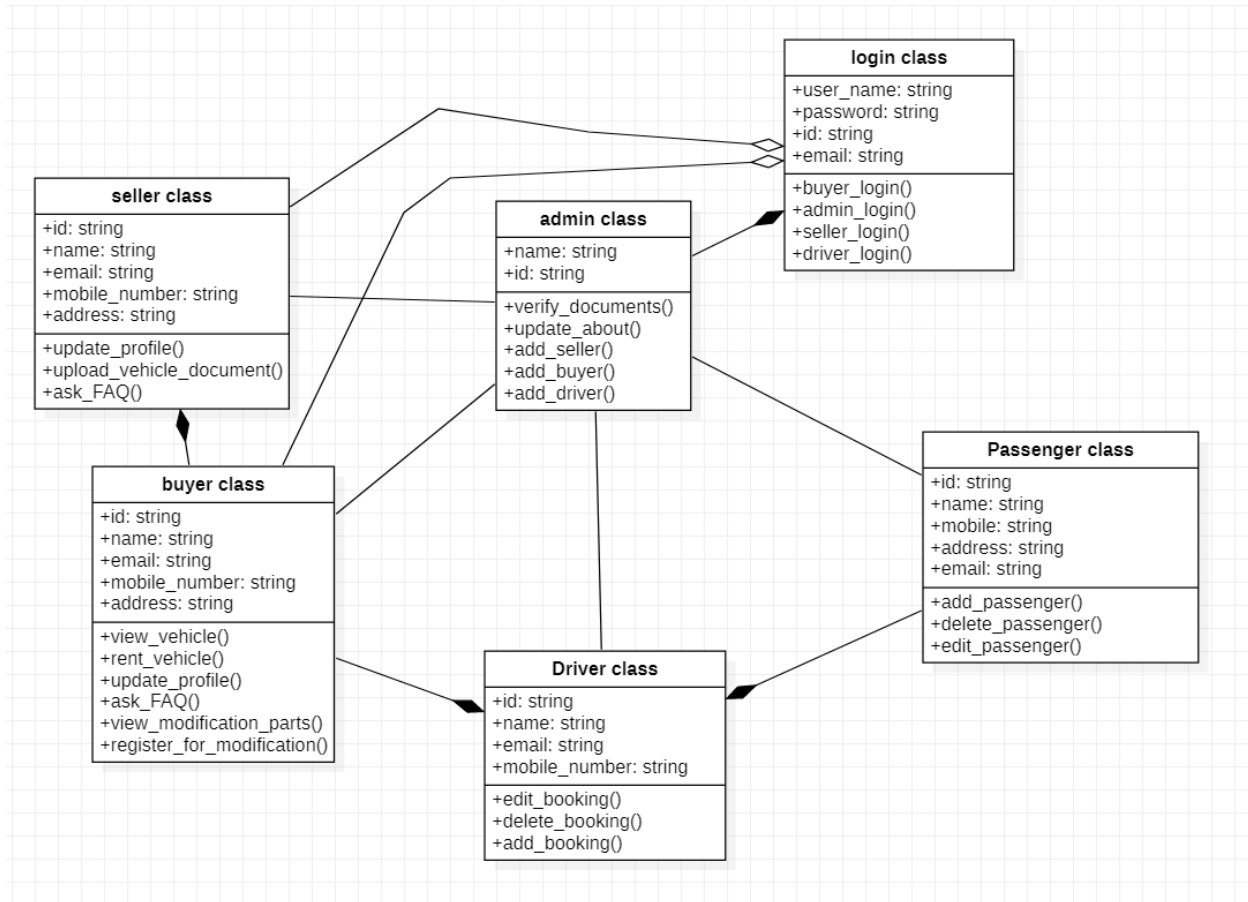
He can register onto Hot-Wheelz as a dealer and make his business online. He can upload the pictures of his vehicles, parts and other accessories online ultimately expanding his reach to potential customers. He can even give his cars for renting for some extra income. But the dealer has to get himself verified before he is able to proceed with the above things.

### 3. Logical Architecture (Use Case Diagram, Class diagram, Sequence Diagram)

#### Use Case Diagram:

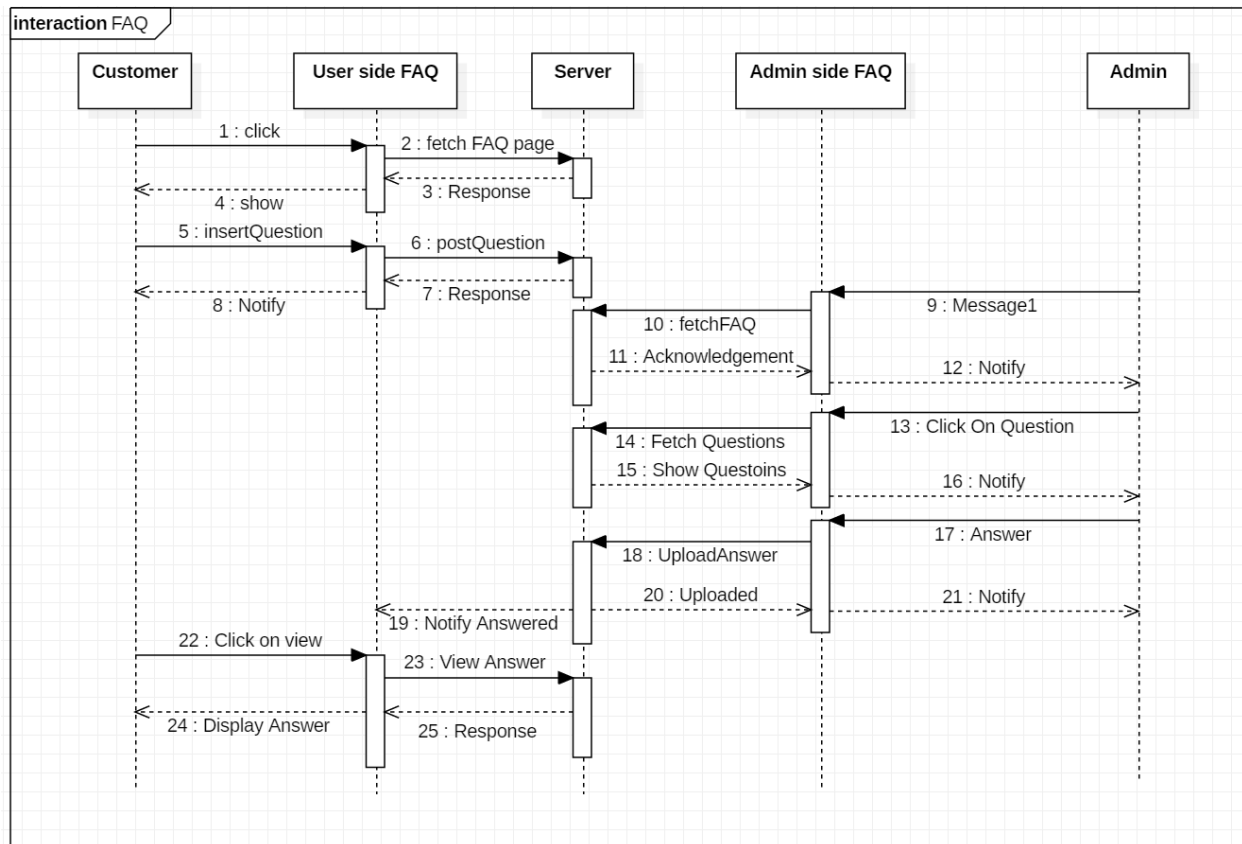


### Class Diagram:

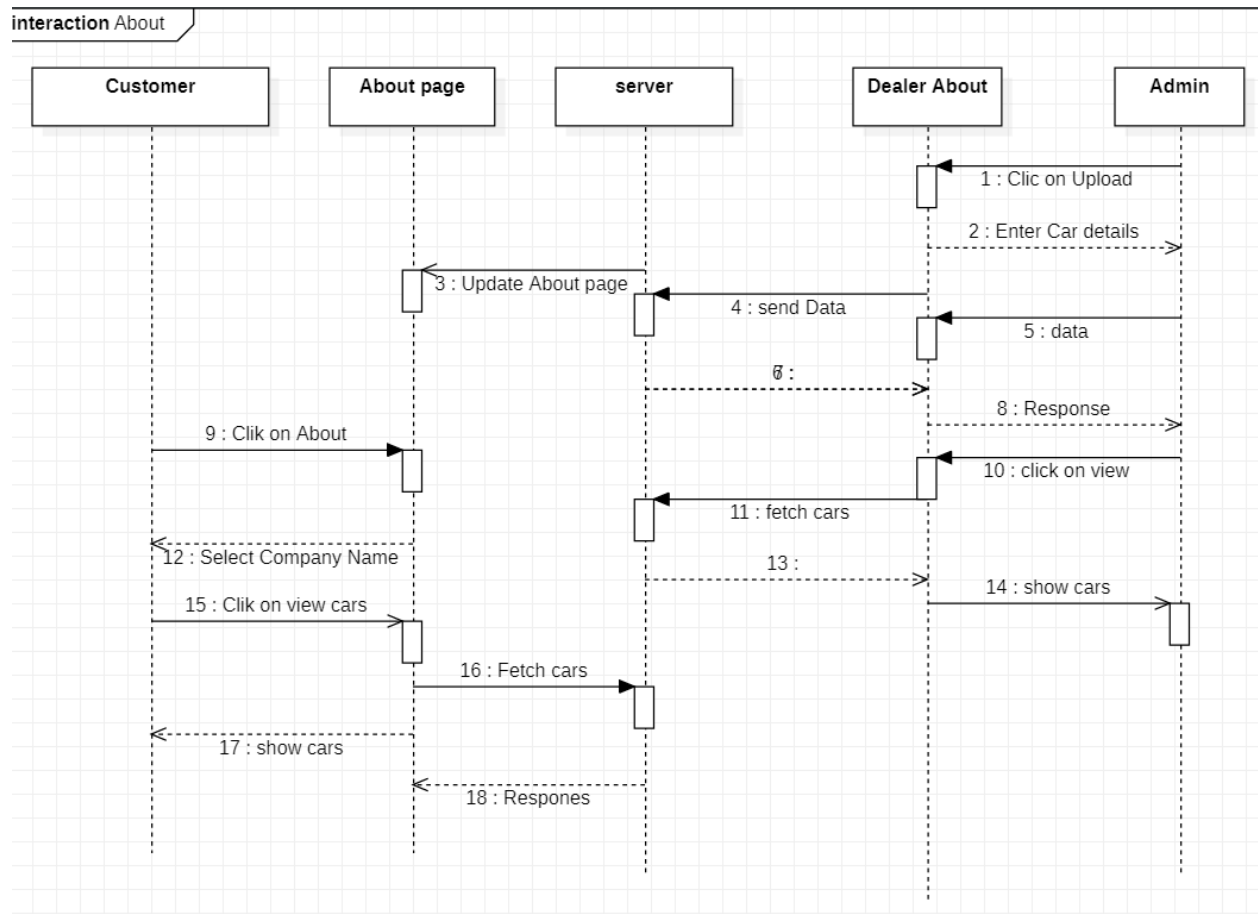


## Sequence Diagrams

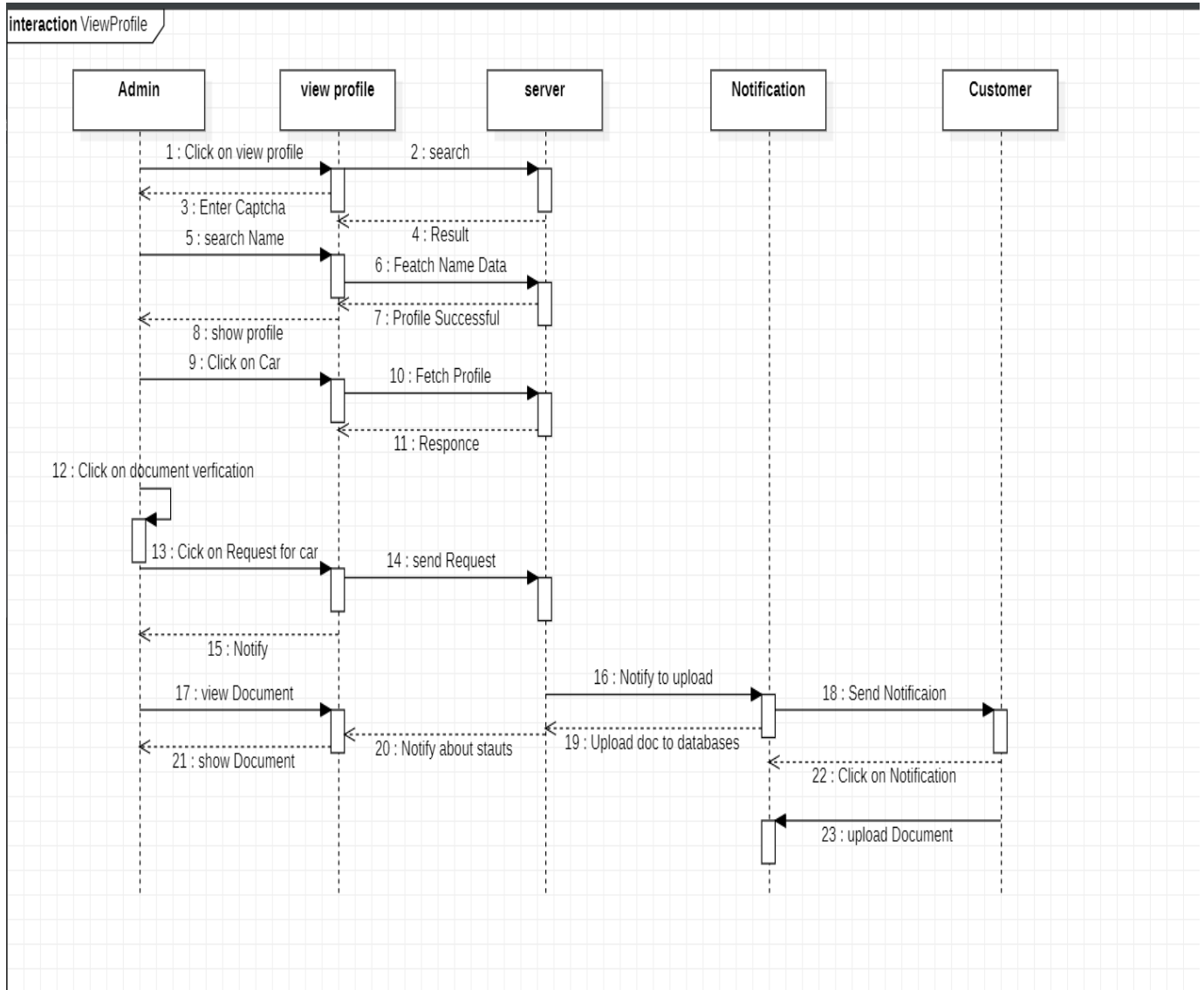
### Sequence Diagram: FAQ:



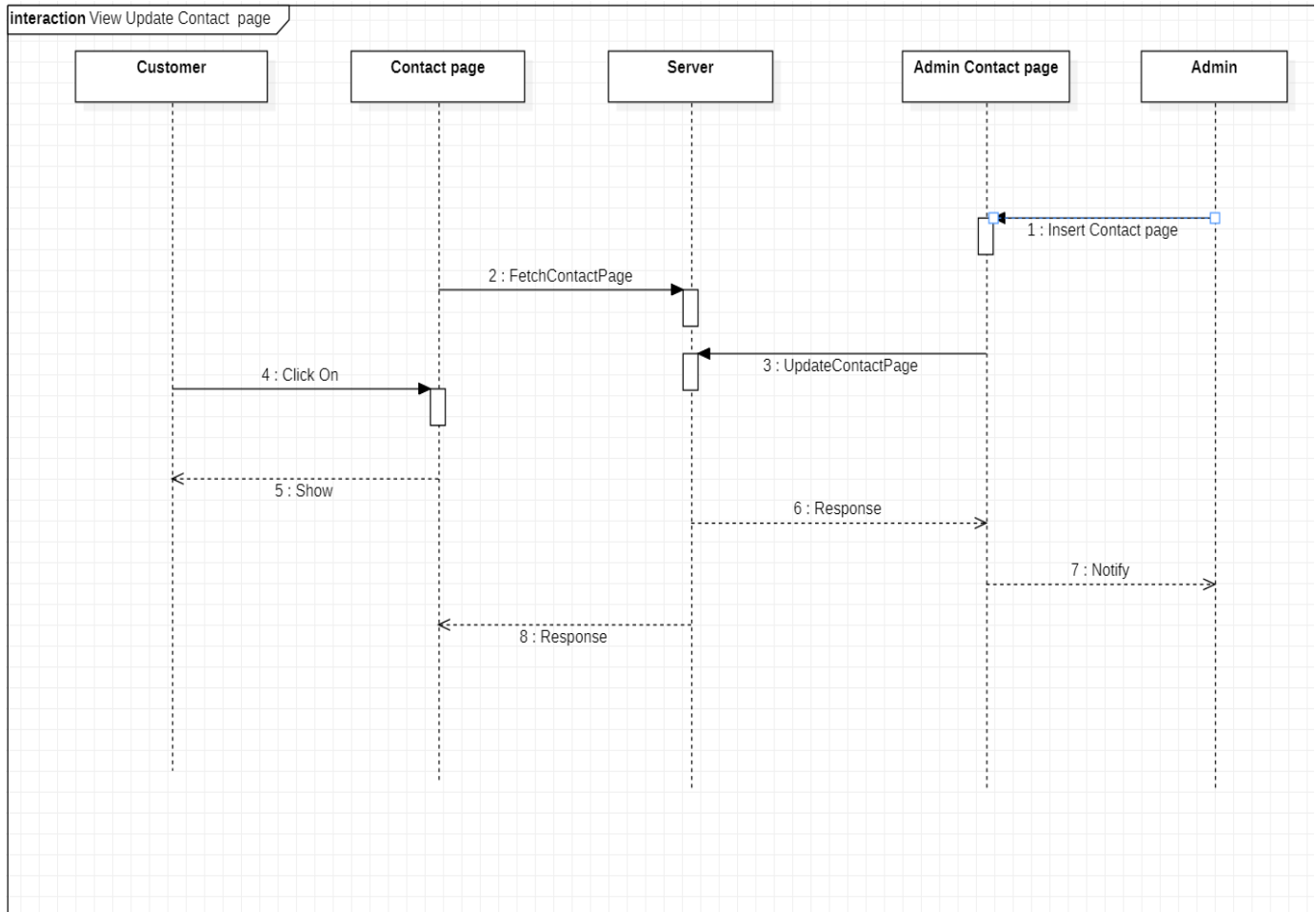
## Sequence Diagram: About Page



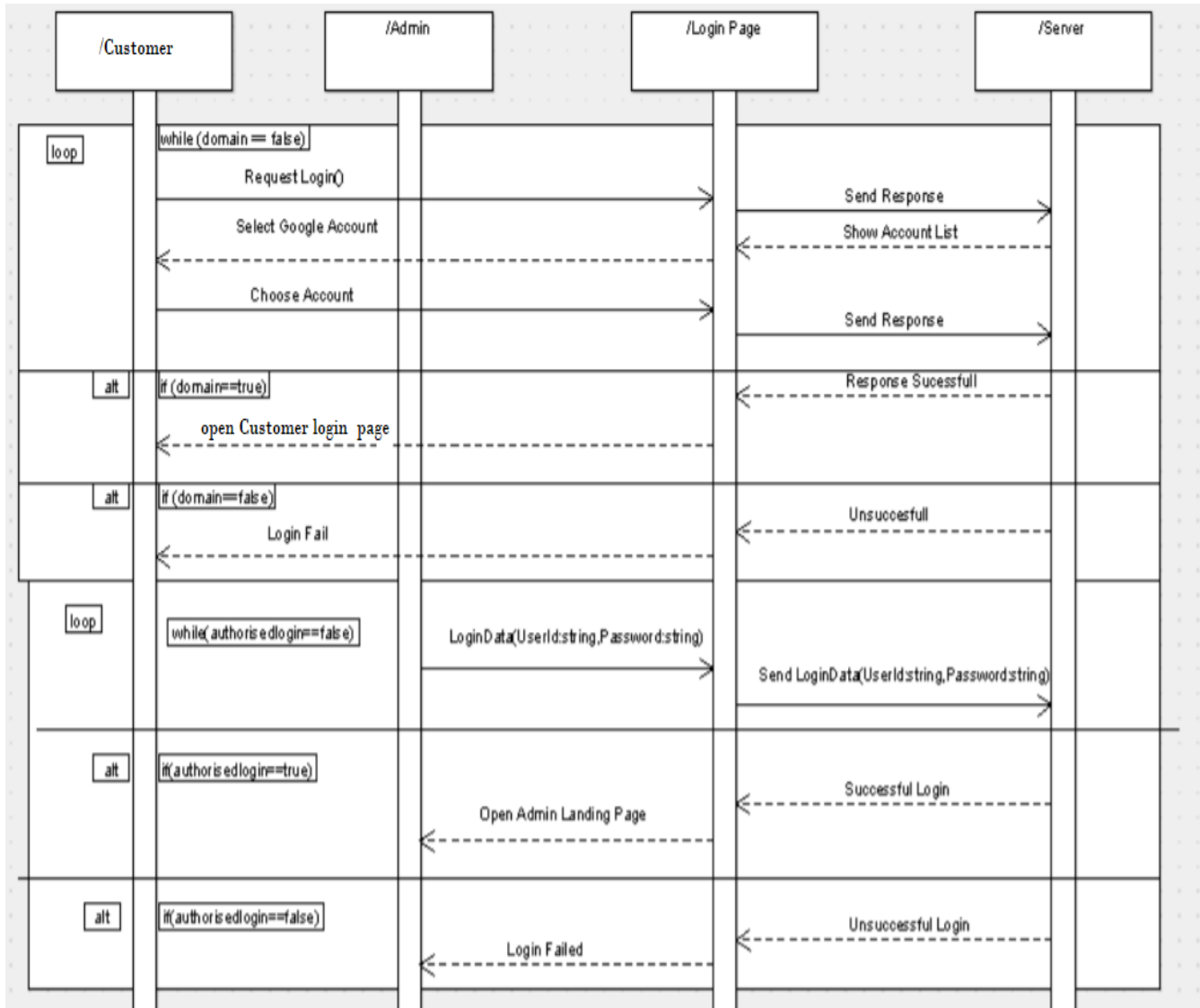
## Sequence Diagram: View Profile



## Sequence Diagram: Update Contact Page

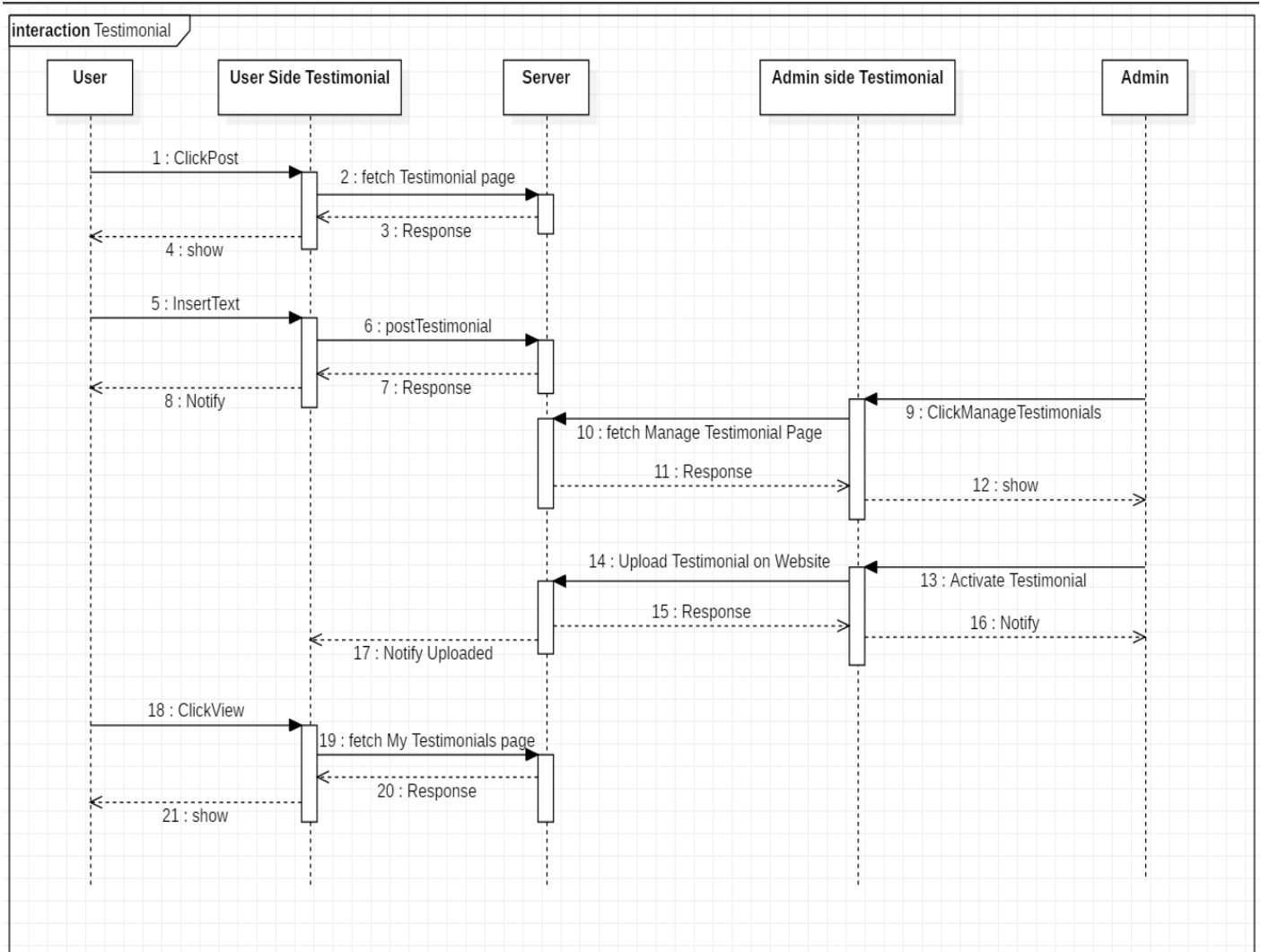


## Sequence Diagram: Login Page

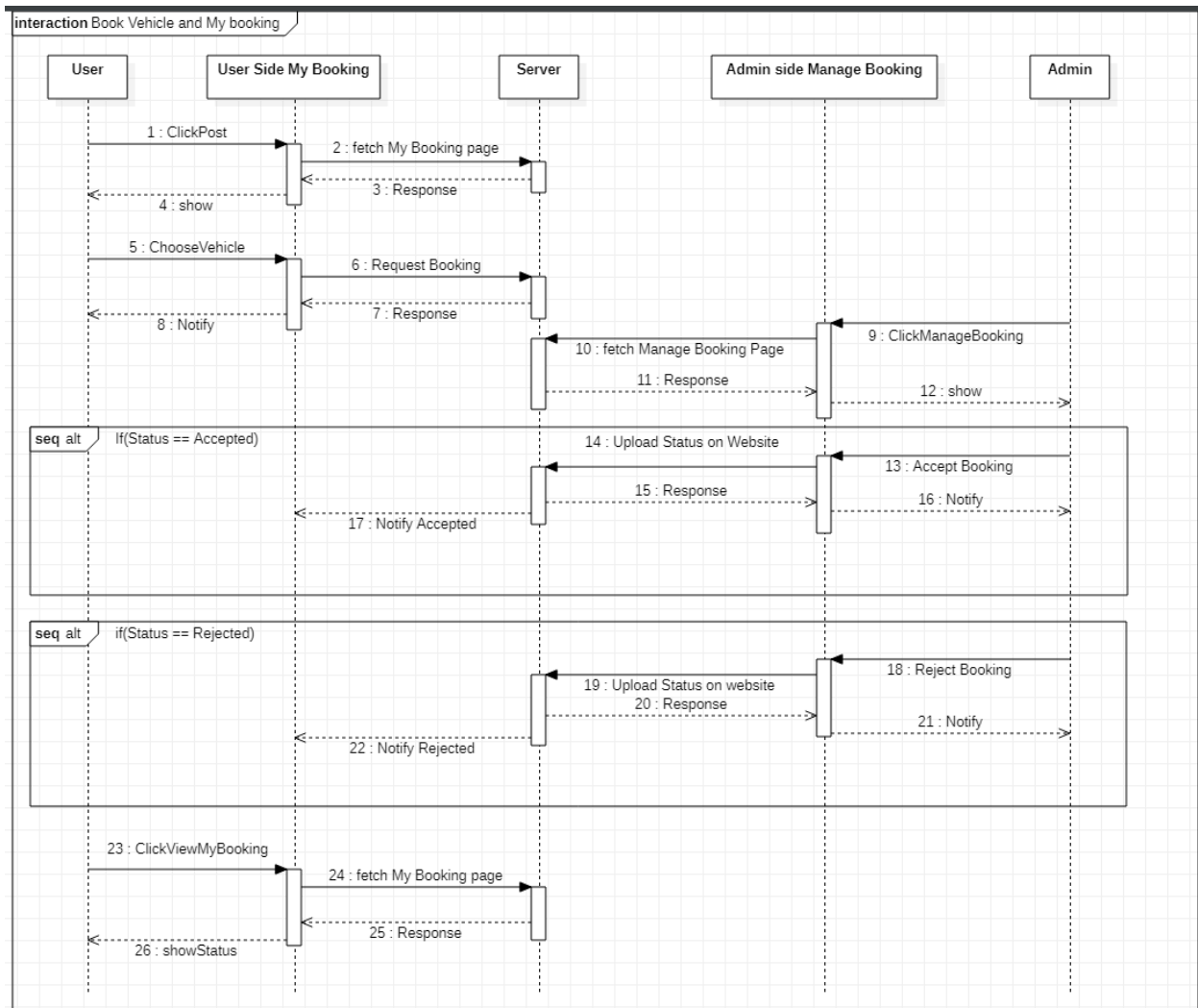




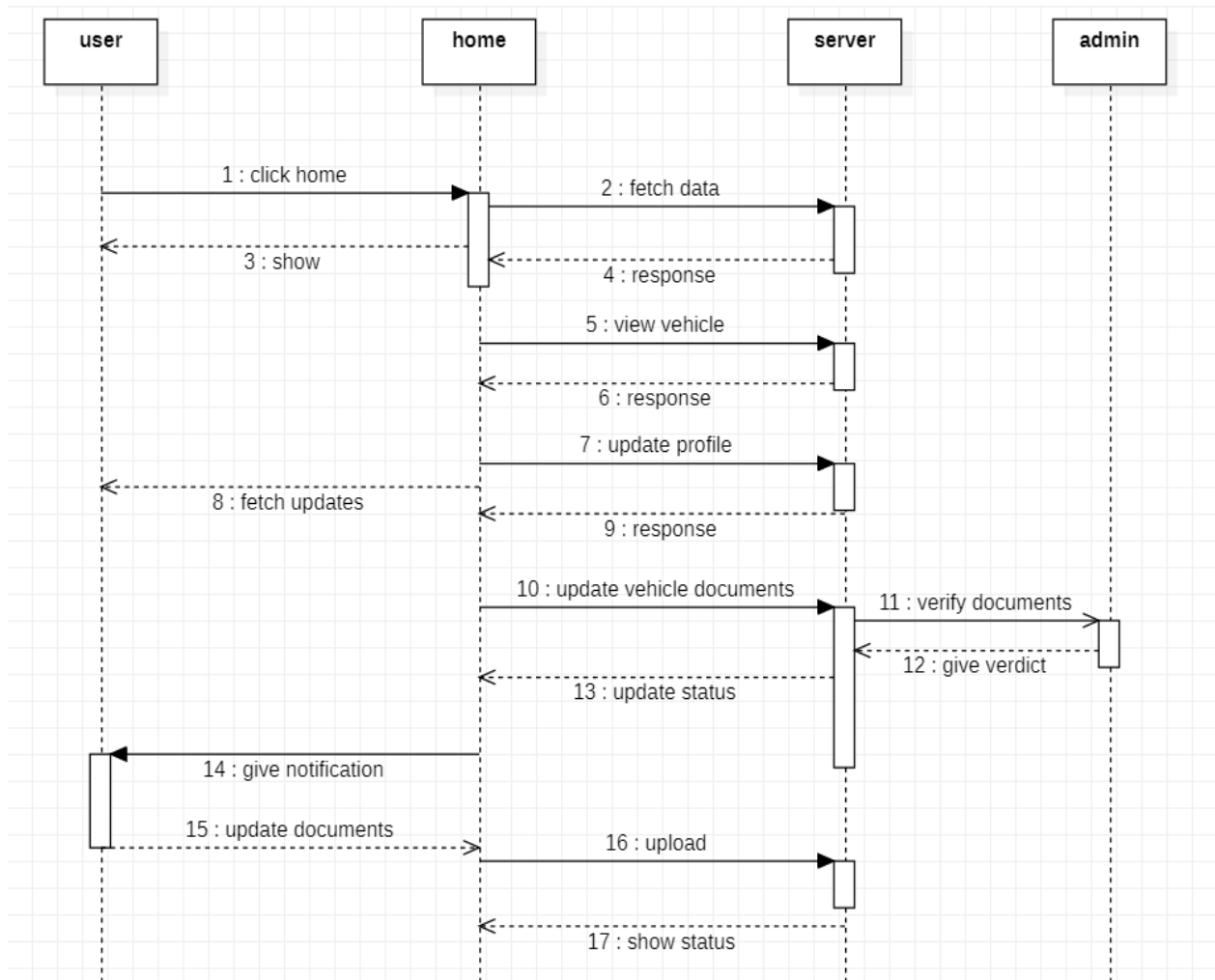
## Sequence Diagram: Testimonial



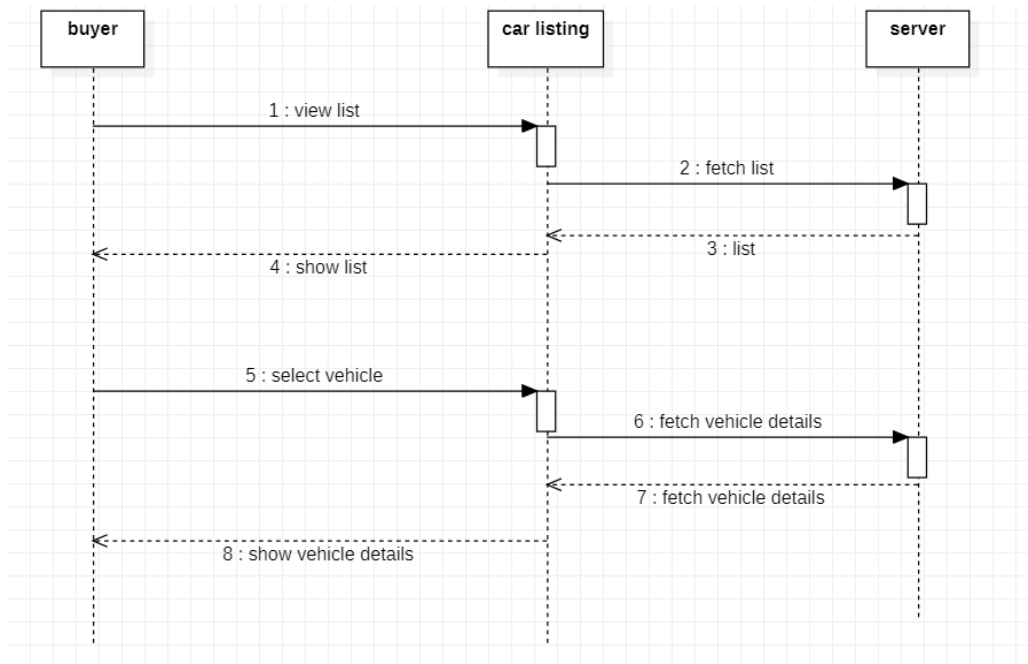
## Sequence Diagram: My Booking and Book Vehicle



## Sequence Diagram: Home

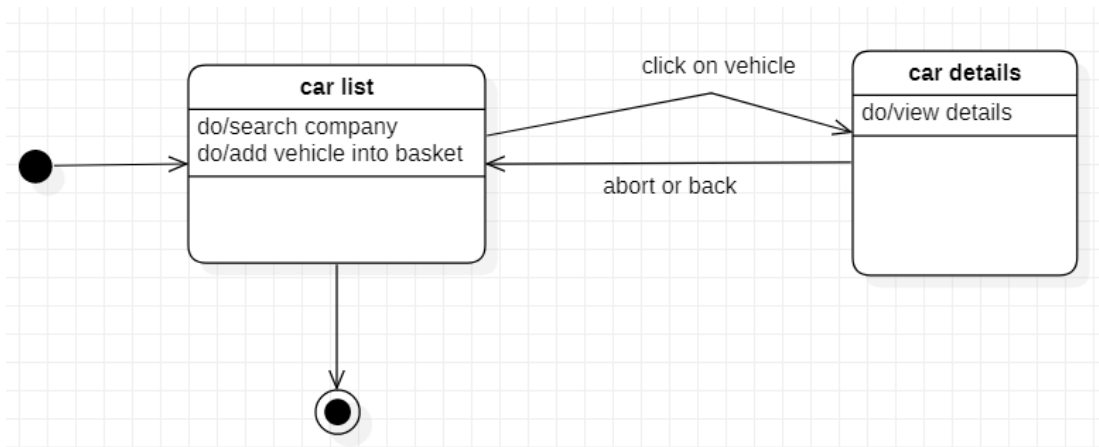


## Sequence Diagram: View Car List

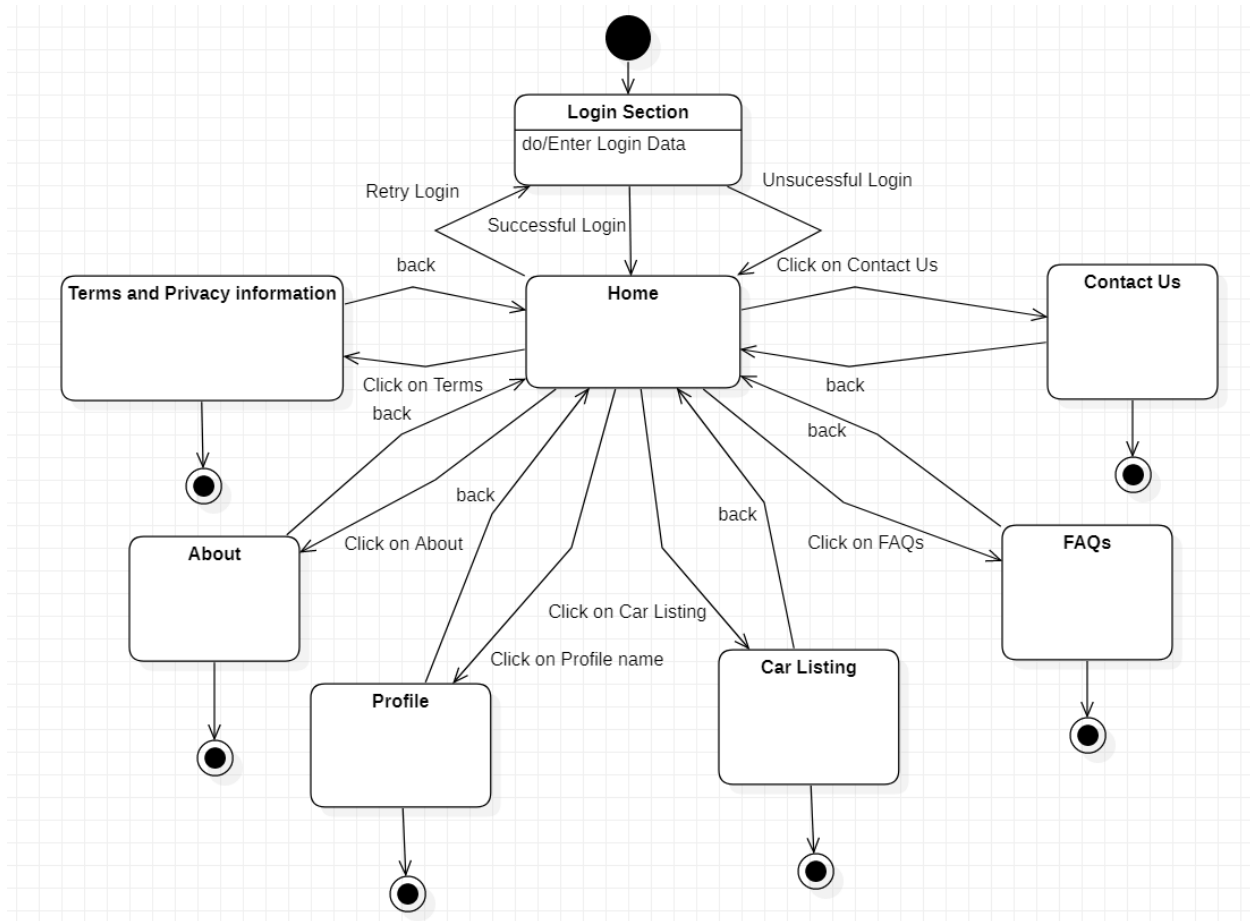


## State Diagrams

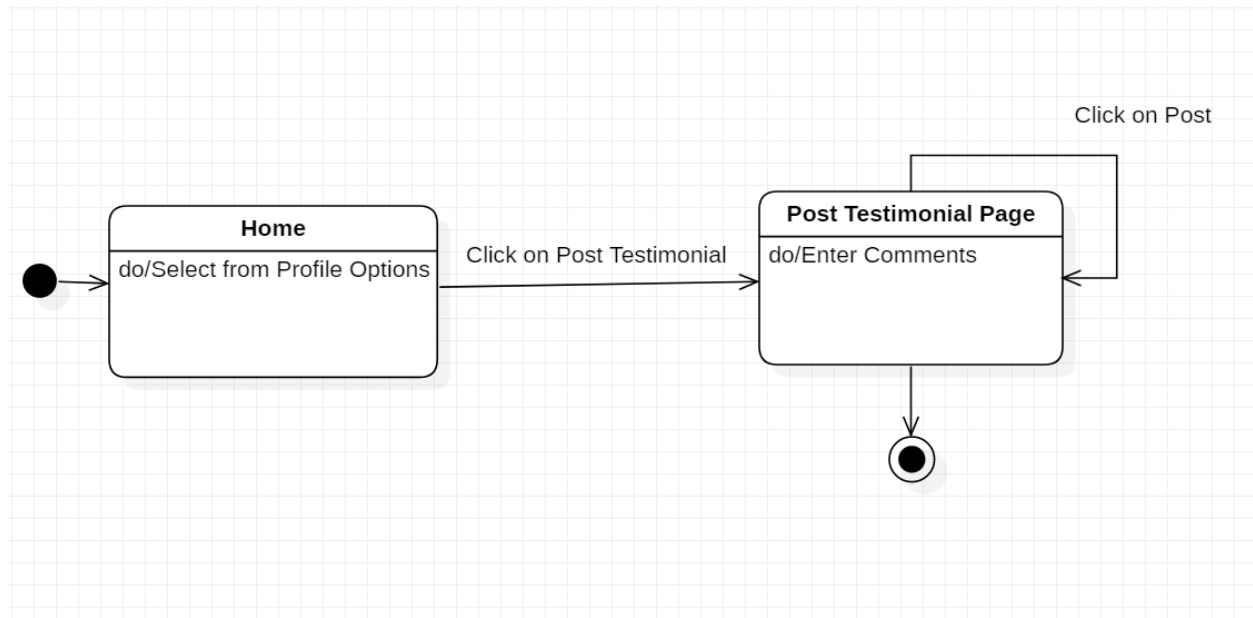
### State Diagram: View Car



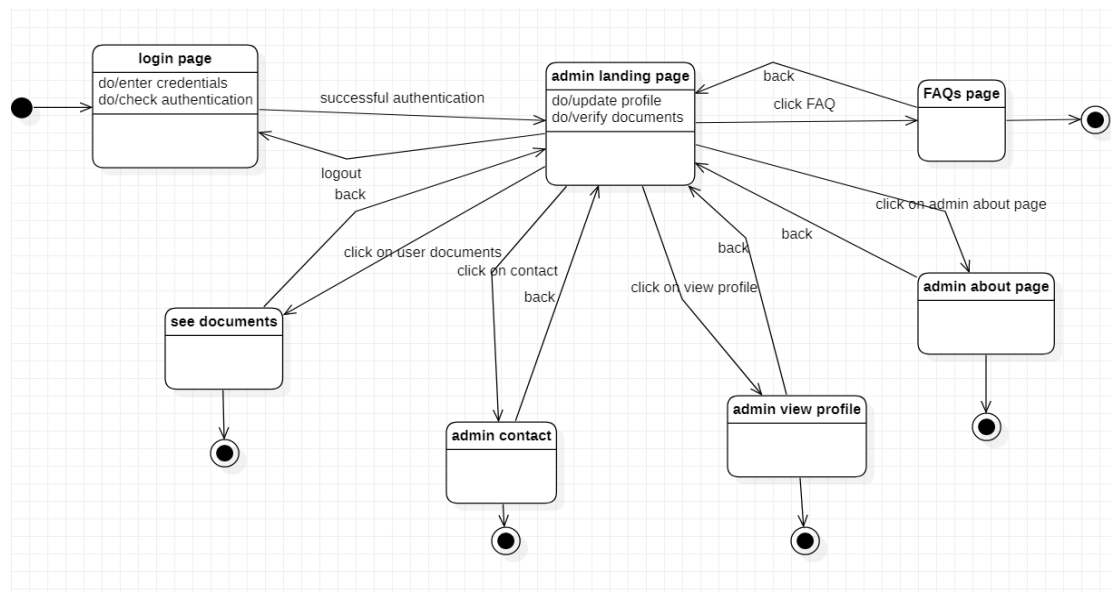
## State Diagram: User/Dealer Login



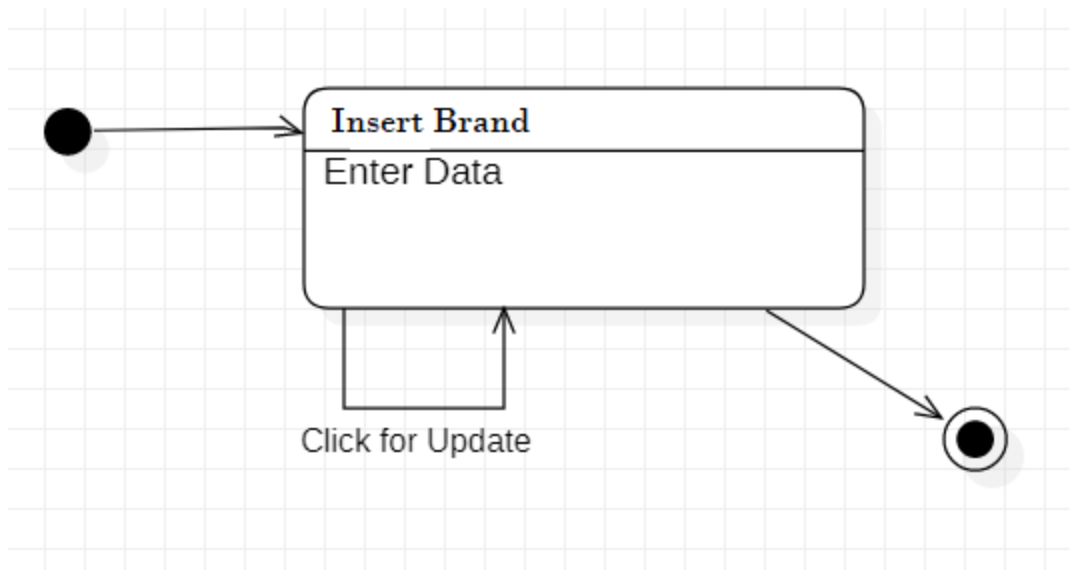
### State Diagram: Post Testimonial



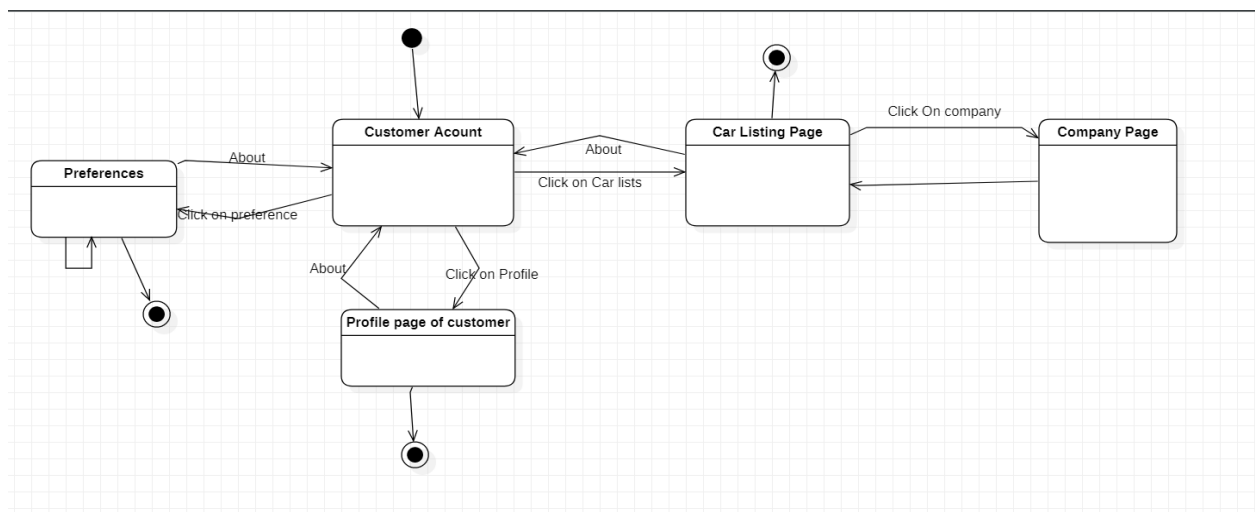
### State Diagram: Login Admin



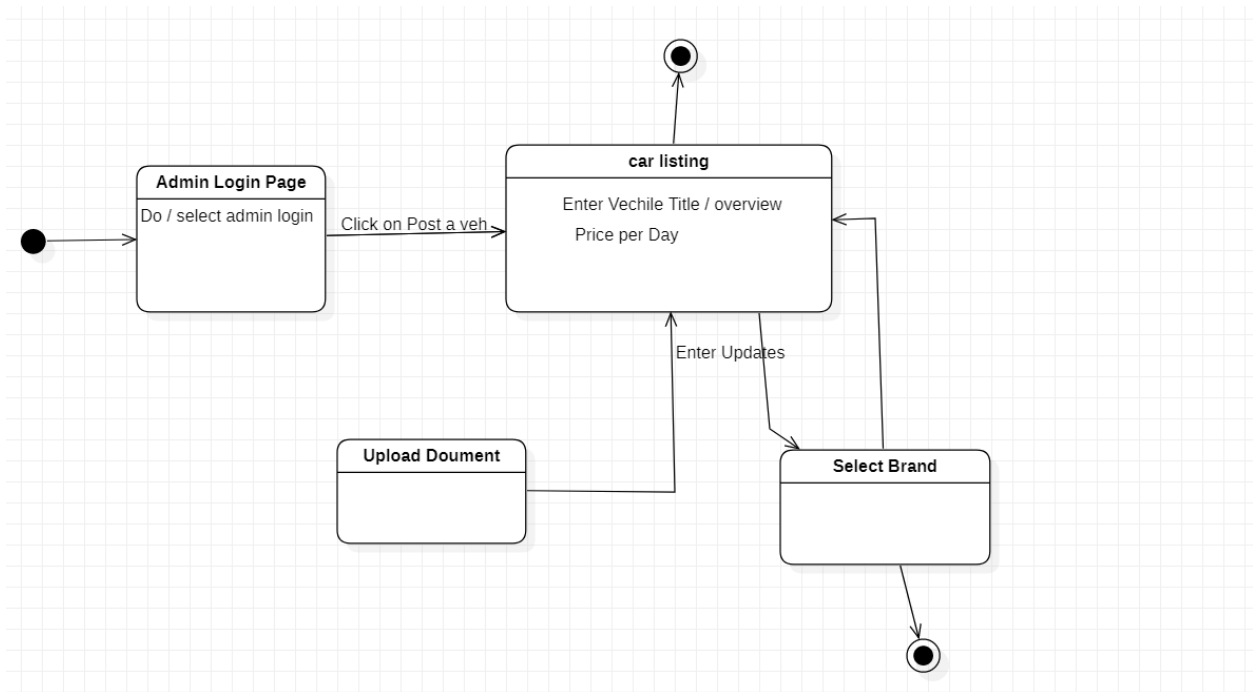
## State Diagram: Insert Brand



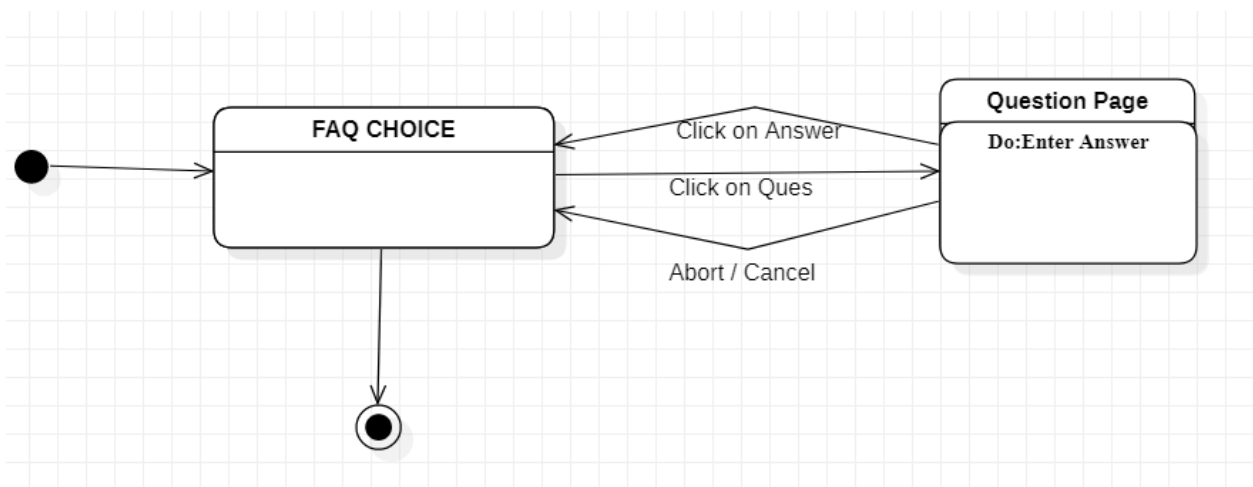
## State Diagram: Customer account page



## State Diagram: Insert cars

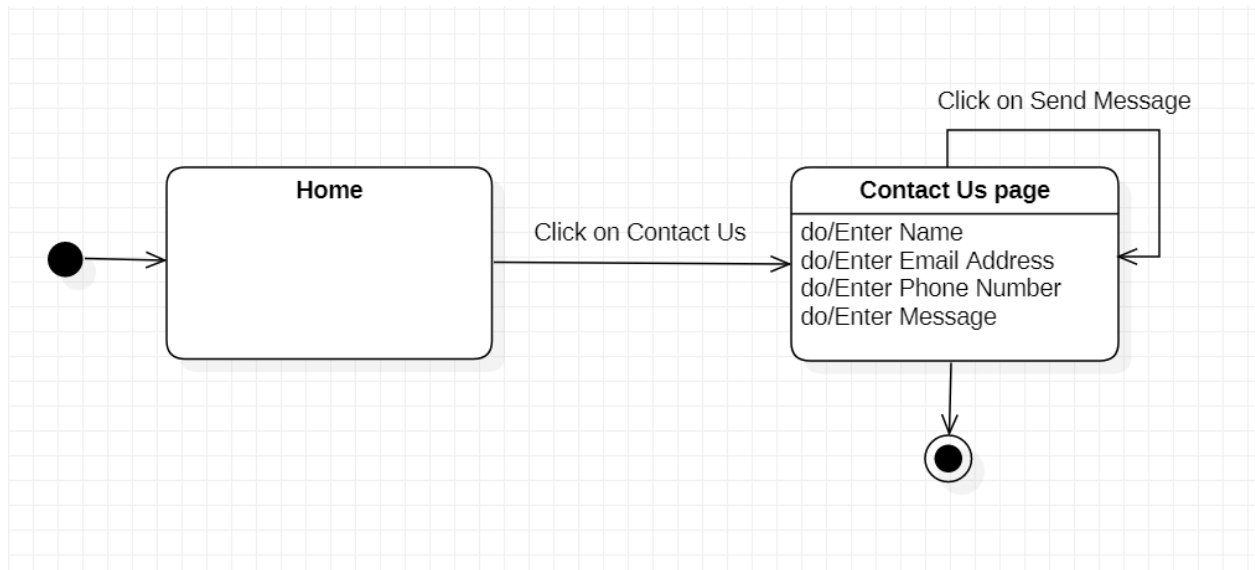


## State Diagram: FAQ Admin

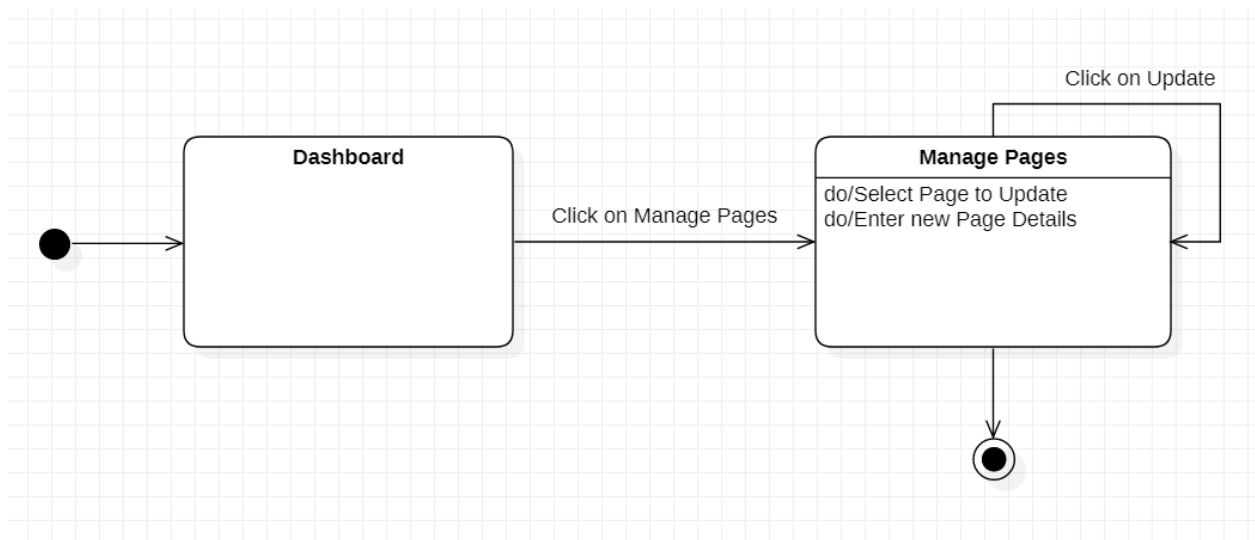




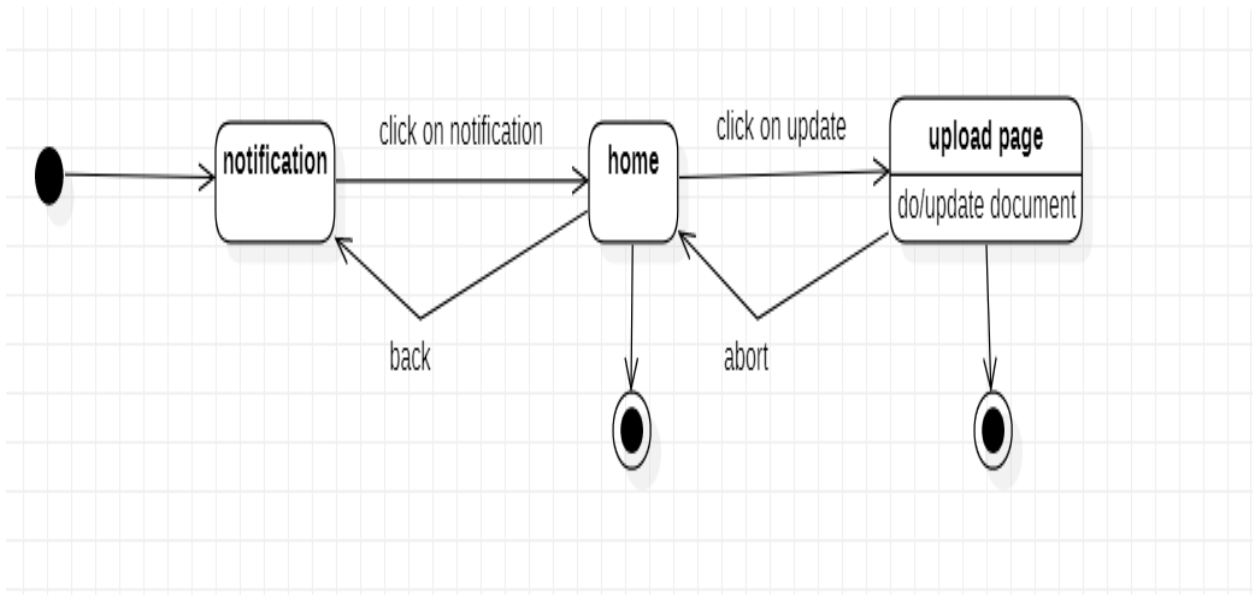
## State Diagram: Contact Us



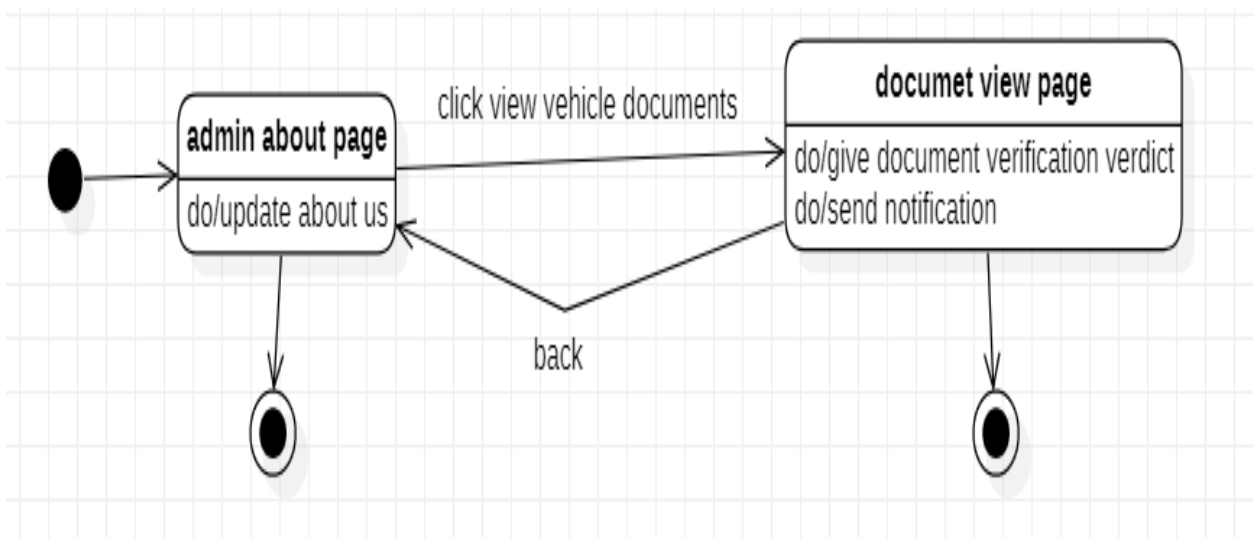
## State Diagram: Admin Manage Pages



## State Diagram: Notification and Home



## State Diagram: Admin About Page



## 3.1 Logical Architecture Description

### 3.1.1 Class Diagram explanation:

Most of the classes have a bi-directional connection with the admin class. The linkage is shown by a simple line joining the two classes. It means that one class just makes an instance of the other class, but does not depend upon each other in any way. These connections are for the following classes:

- Admin class and Seller class
- Admin class and Buyer class
- Admin class and Driver class
- Admin class and Passenger class

There is some compositional linkage between different pairs of classes, shown by the solid arrow. The tail of the arrow points towards the class which is composed of the class pointed by its head. These connections are for the following classes:

- Buyer class depends on Driver class
- Buyer class depends on seller class
- Passenger class depends on Driver class
- Admin class depends on Login Class

There are some classes linked under aggregation relation. This connection is represented by a bordered arrow. The arrow pointing toward the class from which another class has been produced. These are the connections representing aggregation:

- Seller class from Login class
- Buyer class from Login class

### 3.1.2 Sequence Diagram Explanation:

Solid arrow line signifies there is a message being sent. Response is shown by dotted line.

**3.1.2.1 FAQ:** Customers or sellers post questions on this page, which can be answered by Admin. The website notifies the person who has asked a question. Anyone can access answered FAQs on the website.

**3.1.2.2 About Section:** Admin provide the information regarding the website, terms and condition, social media handles where people can reach him/her or know more about the website.

**3.1.2.3 Register Page:** This page allows one to register onto the website. By registering oneself enables the ability to rent/buy/sell vehicles on this website. For selling vehicles another stage of document verification is needed. There are only a few functionalities on the website which doesn't need one to register like browsing a vehicle. A user can register oneself as a regular customer or as a dealer. Registering as a dealer gives users to provide multiple vehicles for rent and manage his/her bookings on **Manage Bookings Page**.

**3.1.2.4 Login Page:** This login page serves two purposes- a customer/dealer can log into their account after Registering and it serves as an Admin login. An Admin can login and tweek the website, answer FAQs, block users (along with some reason) and view statistics of the performance of the website.

**3.1.2.5 My Booking and Book Vehicle:** User can rent vehicle or can buy vehicle on car listing page. The user needs to register and then login to the website before he/she can request a booking. One can choose which car he/she needs to buy from the car listing page, by clicking onto view details the website is redirected onto the page which provides details of the vehicle and provides a section to book the vehicle. The user needs to fill in details like the time period for which the user requires the vehicle and the user can also provide a message with the request to book. Admin is then responsible to confirm the booking or cancel the request. The concerned user can know the status of the booking by going onto my booking section which can be accessed by clicking onto the arrow beside the profile name, then navigating to My Bookings. The admin can confirm or reject the booking on the Manage bookings page which can be accessed from the dashboard.

*Note: The rest of the explanations of the sequence diagrams are given in the state diagram explanation section as the functionalities in the rest sequence diagrams are similar to the functionalities in the state diagram.*

### **3.1.3 State Diagram Explanation**

Arrow line signifies there is a send message taken place. Response is being shown by dotted arrows.

#### **3.1.3.1 View Car**

On clicking the view car button we land on a page from where we can browse cars or add some car into the wishlist. On clicking on some car we can see the details of that particular car along with the seller.

#### **3.1.3.2 Login Admin**

Allows users to enter credentials, which are being checked for authentication in the back-end. After being authenticated successfully, it lands up on the Admin Landing page. It has various buttons like answer FAQ, ViewProfile, See documents, and About to land up on their respective pages, and click on Back to return to the Admin Landing page. On clicking on Logout, it returns to the parent page.

#### **3.1.3.3 Notification And Home**

On clicking notification, user lands up on Home Page, on which user can click on update notification to view page, or click on back to return to the parent page. From the parent page, on clicking UploadDoc notification, it lands up on the Upload page, which allows the user to select file from device and upload the file as notified by the admin. On clicking Back, it returns to Home Page, and on further clicking on Back, it returns to the parent page.

#### **3.1.3.4 User/Dealer Login**

This login page has two purposes- a customer or a dealer can log into their registered account. Users here open up functionalities like booking a slot of a particular vehicle, post FAQs, post Testimonials, update profile and dealer can provide vehicle for renting only after logging into the website.

#### **3.1.3.5 Contact Us Page**

This page allows potential customers to get in touch with us. Any user of the website can use this page to contact us, even if they aren't registered or logged in. To contact us, the user needs to provide their name, email address, phone number, and a short description about the query. A

notification notifies if the query was successfully sent. The admin of the website is responsible for answering the query by an email or by a phone call and marking it as resolved.

### **3.1.3.6 Admin Manage Pages**

This page can be only accessed by the admin of the website. The admin first needs to login to the website from a link given at the bottom of the homepage. After logging in the website is redirected to the admin landing page, and when Manage pages is selected the admin can access the page. Manage Pages is a functionality of the website which allows the admin to manage the text of the pages like terms, policy, FAQs and about us. The admin can update the page on a text editor which is provided on the website with basic functionalities.

### **3.2.3.7 Post Testimonial Page**

This page allows users to convey their experience with the website onto the website. One needs to login to the website to post on this page. The button to access the post testimonial page can be found by clicking onto the arrow beside the profile name. Users are only allowed to enter text into the text box. The testimonial needs to be approved by the admin before it shows upon the website. The stats of the visibility of the testimonial can be seen on My Testimonials page, which can also be accessed by the arrow beside the profile name. The page shows whether the testimonial is active or inactive.

### **3.1.3.8 FAQ Admin**

On clicking on any question, it lands up in the Question page, on which the answer can be typed for it. On clicking on the answer button, it would again land up the user FAQ page, on simply stating that it answers the question requested by the user FAQ page.

### **3.1.3.9 Insert Brand**

Allows admin to enter new brand data by clicking on the update button. After initially inserting some brand names we can use the update to update the data and store it in the databases.

### 3.1.3.10 Customer Account Page

There are three buttons available to be clicked: car Listing.Preference, and view Profile,clicking on which it lands up the user on their respective pages.The Car listing page allows the user to click on the brand to land up on brand page and return to car listing on Clicking Back. On further clicking back,moves the user to the parent page.Preference page allows the user to choose the preferable vehicles. On clicking Back, the user can return to the parent page.

### 3.1.3.11 Insert cars

On clicking Insert cars page there contain the three main part where on first basic info ,upload images,accessories.In the Basic Info for the Inserting of cars enter the vehicle Title, from the Insert brand select the Brand for the vehicle then describe the overview of the uploaded vehicle.similarly upload the detail of Price per day ,model year and seating capacity.On coming to next part Upload the images of the same vehicle that we upload the details.then we have the remaining part to choose for the accessories from the allotted part.

### 3.1.2.12 Admin About Page

On clicking the document view button, it lands up the user uploaded vehicle document Data page, from where he/she can verify the documents or send a notification. On clicking Back, it returns to the parent page.

## 3.2 Class name:Login

**Description :** This class allows the user to enter the system by authenticating entered credentials.

### 3.2.1 Method 1: postLogin()

**Input:** email\_id,password,login

**Output:** logged in (index.php)

Method Description:

When an Activity first calls or launches a user click on the login button.then get the value email input from the user.Then the \$password check for MD5sum .*Md5 function encrypt user input*

*password into Md5 sum.* So, this method takes input as email id and password as a result opens the landing page if login successful.

### 3.2.2 Method 2: Admin\_Login()

**Input:** username, password, sign in

**Output:** logged in (change-password.php)

#### Method Description:

This method takes input email and password from the customer and checks whether it is authorized login or not and if the result is successful it leads to login the account.

### 3.2.3 Method 3: Registration

**Input:** email\_id, password, login

**Output:** logged in (index.php)

**Description :** This method will get the registered user .And get it login id and password.

## 4.0 Pseudocode for components

### 4.1 Class Name: Login

#### Pseudo-code :

**Input:** email\_id, password, login

**Output:** logged in (index.php)

```
1. if(isset($_POST['login'])) // if user click on login button
2. $email = get value from email input
3. $password = md5(get value from password input)
   //Md5 function encrypt user input password into Md5 sum
4. $sql = "SELECT EmailId, Password, FullName FROM tblusers WHERE      EmailId=:email
   and Password=:password";
   //sql query
5. $query = $dbh -> prepare($sql);
6. $query-> bindParam(':email', $email, PDO::PARAM_STR); // binding email parameter
7. $query-> bindParam(':password', $password, PDO::PARAM_STR); // binding password
parameter
```



```

8.$query-> execute(); //executing sql query
9.$results=$query->fetchAll(PDO::FETCH_OBJ); //fetching all records from database
10.if(email id and password found in database)
11.    $_SESSION['login']=$_POST['email'];
12.    $_SESSION['fname']=$results->FullName;
13.    $currentpage= active page url
14.    Open current page on successfully logged in
15.endif;
16. else
    Print "Invalid Email id or Password"
17.endelse;
18.endif;

```

## 4.2 Class name:Admin Login

**Pseudo-code :**

**Input:**username,password,sign in

**Output:** logged in (change-password.php) //admin password change page

```

1.if(isset($_POST['login'])) // if user click on login button
2.$email=get value from username input
3.$password=md5(get value from password input)
    //Md5 function encrypt user input password into Md5 sum
4.$sql ="SELECT UserName,Password FROM admin WHERE UserName=:email and
Password=:password"; //sql query
5.$query= $dbh -> prepare($sql);
6.$query-> bindParam(':email', $email, PDO::PARAM_STR); // binding email parameter
7.$query-> bindParam(':password', $password, PDO::PARAM_STR); // binding password
parameter

8.$query-> execute(); //executing sql query
9.$results=$query->fetchAll(PDO::FETCH_OBJ); //fetching all records from database
10.if(email id and password found in database)
11.    $_SESSION['login']=$_POST['email'];

```

```

12. $_SESSION['fname']=$results->FullName;
13. Redirect to change-password.php on successfully logged in
14.endif;
15. else
    Print "Invalid Details"
16.endif;
17.endif;

```

### 4.3 Class Name: Registration

**Pseudo-code :**

**Input:** email\_id,password,login

**Output:** logged in (index.php)

```

1.if(isset($_POST['signup'])) //if user click on sign up button
2.$fname=get full name from user;
3.$email= get email address from user
4.$mobile=get mobile number from user
5.$password=md5(password obtained from user);
6.$type = dealer or customer;
7.$sql="INSERT INTO tblusers(FullName,EmailId,ContactNo>Password,Type)
VALUES(:fname,:email,:mobile,:password,:type)";
    // SQL query to insert into database

8.$query = $dbh->prepare($sql); // preparing sql query
9.$query->bindParam(':fname',$fname,PDO::PARAM_STR); //binding fname variable
10.$query->bindParam(':email',$email,PDO::PARAM_STR); //binding email variable

11.$query->bindParam(':mobile',$mobile,PDO::PARAM_STR); //binding mobile variable

12.$query->bindParam(':password',$password,PDO::PARAM_STR); //binding password variable

13.$query->bindParam(':type',$type,PDO::PARAM_STR); //binding type variable

14.$query->execute(); //executing sql statement
15.$lastInsertId = $dbh->lastInsertId();
16.if($lastInsertId)
17.    if(registered user is dealer type)

```

```

        $sql="INSERT INTO
tbldealers(id,FullName,EmailId,ContactNo,Password,Type)
VALUES(:lastInsertId,:fname,:email,:mobile,:password,:type)"; //insert into dealer table
18.     $query = $dbh->prepare($sql);
19.     $query->bindParam(':lastInsertId',$lastInsertId,PDO::PARAM_STR);
20.     $query->bindParam(':fname',$fname,PDO::PARAM_STR);
21.     $query->bindParam(':email',$email,PDO::PARAM_STR);
22.     $query->bindParam(':mobile',$mobile,PDO::PARAM_STR);
23.     $query->bindParam(':password',$password,PDO::PARAM_STR);
24.     $query->bindParam(':type',$type,PDO::PARAM_STR);
25.     $query->execute();
26.     Print "Registration successful. Now you can login"
27.Endif
28. else
29.     Print "Something went wrong. Please try again";
30. Endelse
31.Endif
32.Endif

```

#### 4.4 Class Name:Logout

**Pseudo-code :**

**Input:** //user should be logged in

**Output:** log out (Index.php) //Redirect to index.php page

```

1.session_start(); //this function will start session,to access all session variables values
2.$_SESSION = array(); //array() will return stored variables (array)
3.if (data is stored in cookies) //if any of the data is stored in cookies like cart
4.  $params = session_get_cookie_params(); // get all cookies
5.  setcookie(session_name(), "", time() - 60*60,
    $params["path"], $params["domain"],
    $params["secure"], $params["httponly"]
    );      // setting time limit for saving data in cookies

6.unset(); // unset all variable (i.e, change their value to null)
7.session_destroy(); // destroy session
8.header("location:index.php"); // redirect to index.php

```

## 4.5 Class Name:FAQ'S page

**Pseudo-code :**

**Input:**

**Output:** page.php?type=faqs      //redirect to page.php and display faqs

```
1.$pagetype= get page type      // in our case its faq type
2.$sql = "SELECT type,detail,PageName from tblpages where type=:pagetype";
    // SQL query to get details from database regarding faq
3.$query = $dbh -> prepare($sql);   //preparing SQL statement
4.$query->bindParam(':pagetype',$pagetype,PDO::PARAM_STR); //binding page type
parameter
5.$query->execute(); //running SQL query
6.$results=$query->fetchAll(PDO::FETCH_OBJ); //fetching data from database and storing
in $result variable
7.$cnt=1;    // count number of results(rows in database table)
8.if(any result found)
9.foreach($results as $result)
10. //display data (in our case FAQS regarding data)
<section class="page-header aboutus_page">
  <div class="container">
    <div class="page-header_wrap">
      <div class="page-heading">
        <h1><?php echo htmlentities($result->PageName); ?></h1>
      </div>
      <ul class="coustom-breadcrumb">
        <li><a href="#">Home</a></li>
        <li><?php echo htmlentities($result->PageName); ?></li>
      </ul>
    </div>
  </div>
  <!-- Dark Overlay-->
  <div class="dark-overlay"></div>
</section>
<section class="about_us section-padding">
  <div class="container">
    <div class="section-header text-center">
```

```

        <h2><?php echo htmlentities($result->PageName); ?></h2>
        <p><?php echo $result->detail; ?> </p>
    </div>
</div>
</section>
11.EndForeach loop
12.Endif

```

## 4.5 Class Name:Car Search

**Pseudo-code :**

**Input:** brand,fuel type

**Output:** car-listing.php      **//redirect to car-listing.php and display cars**

1. \$sql = "SELECT tblvehicles.\*,tblbrands.BrandName,tblbrands.id as bid from tblvehicles join tblbrands on tblbrands.id=tblvehicles.VehiclesBrand order by id desc limit 5"; **//SQL query to select all columns from tblvehicles table where vehicle brand = brand (input) and vehicle fuel = fuel type(input)**

2.\$query = \$dbh -> prepare(\$sql); **//preparing SQL statement**

3.\$query->execute(); **//running SQL query**

4.\$results=\$query->fetchAll(PDO::FETCH\_OBJ); **//fetching data from database and storing in \$result variable**

5.\$cnt=1; **// count number of results(rows in database table)**

6.IF(any result found)

7.    FOREACH(\$results as \$result) ?>

8.            <div class="recent\_post\_img"> <a href="vehical-details.php?vhid=<?php echo htmlentities(\$result->id);?>"></a> </div>      **// print vehicle image**

9.            <div class="recent\_post\_title"> <a href="vehical-details.php?vhid=<?php echo htmlentities(\$result->id);?>"><?php echo htmlentities(\$result->BrandName);?> , <?php echo htmlentities(\$result->VehiclesTitle);?></a>      **// print brand name and vehicle title**

```

10. <p class="widget_price">${<?php echo htmlentities($result->PricePerDay);?> Per Day</p>
</div>    // print car price per day
11.    END FOREACH;
12.END IF;

```

## 4.6 Class Name: Subscribe Newsletter

**Pseudo-code :**

**Input:** Email address

**Output:** index.php //redirect to index.php

```

1.IF( user provides email address)
2.    $subscriberemail=get user provided email adress
3.    $sql ="SELECT SubscriberEmail FROM tblsubscribers WHERE
        SubscriberEmail=:subscriberemail"; //SQL query to select subscriber email from
tblsubscribers table

4.    $query= $dbh -> prepare($sql); //preparing SQL statement
5.    $query-> bindParam(':subscriberemail', $subscriberemail, PDO::PARAM_STR); //
binding variables parameter
6.    $query-> execute(); // running sql query

7.    $results = $query -> fetchAll(PDO::FETCH_OBJ); // fetch all data from table
8.    $cnt=1; // number of results found
9.    IF(any result found)
10       print user already subscribed
        ENDIF;
11.    ELSE
        $sql="INSERT INTO tblsubscribers(SubscriberEmail) VALUES(:subscriberemail)"; //
if user have not subscribed earlier then insert provided email address into tblsubscribers
table.
12.    $query = $dbh->prepare($sql); //preparing sql statement
13.    $query->bindParam(':subscriberemail',$subscriberemail,PDO::PARAM_STR); // binding
sql parameters
14.    $query->execute(); //running sql query
15.    $lastInsertId = $dbh->lastInsertId(); //get the last inserted data id in table
16.    IF( any data get inserted )
17.        print user subscribed successfully

```

```

18.      EndIF;
19.      ELSE
20.          echo "<script>alert('Something went wrong. Please try again');</script>"; // print
21.      ENDELSE
22. ENDIF

```

## 4.7 Class Name: Show Testimonials

**Pseudo-code :**

**Input:**

**Output:** index.php      //redirect to index.php

```

1. $sql = "SELECT tbltestimonial.Testimonial ,tblusers.FullName from tbltestimonial join
tblusers on tbltestimonial.UserEmail=tblusers.EmailId where tbltestimonial.status=:tid"; // SQL
queryt to select testimonial,user FullName from tbltestimonial table.

```

```

2.$query = $dbh -> prepare($sql); //preparing sql statemnet
3.$query->bindParam(':tid',$tid, PDO::PARAM_STR); // binding sql parameters
4.$query->execute(); //running sql query
5.$results=$query->fetchAll(PDO::FETCH_OBJ); //fetching results
6.$cnt=1; //number of results
7.IF(any result found)
8.FOREACH($results as $result)
9.    print Fullname of testimonial user
10.    print testimonial
11.END FOREACH;
12.ENDIF;

```

## 4.8 Class Name: Admin Dashboard

**Pseudo-code :**

**Input:**

**Output:** dashboard.php      //redirect to dashboard.php

```

1. $sql ="SELECT id from tblusers "; // SQL query to select all id from tblusers table.
3.  $query = $dbh -> prepare($sql); //preparing sql statemnet
4.  $query->execute(); //running sql query
5.  $results=$query->fetchAll(PDO::FETCH_OBJ); //fetching all result from database

```

```

6. $regusers=$query->rowCount(); ?> // number of registered users
7. print total numbr of registered users
   $sql1 ="SELECT id from tblvehicles "; //SQL query to select all vehicles id from
tblvehicles table.
8. $query1 = $dbh -> prepare($sql1); // preparing SQL statement
9. $query1->execute(); //running SQL query
10. $results1=$query1->fetchAll(PDO::FETCH_OBJ); //fetching all results
11. $totalvehicle=$query1->rowCount(); ?> // get total number of total vehicles.
    printing total number of vehicles
12. $sql2 ="SELECT id from tblbooking "; // select all id from tblbooking table.
13. $query2= $dbh -> prepare($sql2); //preparing SQL query.
14. $query2->execute(); //running SQL query
15. $results2=$query2->fetchAll(PDO::FETCH_OBJ); //fetching all result from databse
16. $bookings=$query2->rowCount(); ?> // get total number of bookings print
    total number of bookings
17. $sql3 ="SELECT id from tblbrands "; //SQL query to select all id from table tblbrands.
18. $query3= $dbh -> prepare($sql3); //preparing SQL query
19. $query3->execute(); //running SQL query
20. $results3=$query3->fetchAll(PDO::FETCH_OBJ); //fetch all results from database
21. $brands=$query3->rowCount(); ?> //total numbers of brands available

22. $sql4 ="SELECT id from tblsubscribers "; //SQL statement to select all id from
tblscrubscibers table
23. $query4 = $dbh -> prepare($sql4); //preparing SQL statement
24. $query4->execute(); //running SQL statement
25. $results4=$query4->fetchAll(PDO::FETCH_OBJ); //fetching all result from databse
26. $subscribers=$query4->rowCount(); // get total number of subscribres
27. print total number of subscribers
28. $sql6 ="SELECT id from tblcontactusquery "; //SQL query to select all id from
tblcontactquey table
29. $query6 = $dbh -> prepare($sql6); //preparing SQL query
30. $query6->execute(); //running SQL query
31. $results6=$query6->fetchAll(PDO::FETCH_OBJ); //fetching all id from database
32. $query=$query6->rowCount(); // get total number of result

33. print total number of result found
34. $sql5 ="SELECT id from tbltestimonial "; //SQL query to select all id from
tbltestimonial table
35. $query5= $dbh -> prepare($sql5); //preparing SQL query

```



36. `$query5->execute();` **//running SQL query.**
37. `$results5=$query5->fetchAll(PDO::FETCH_OBJ);` **//fetching all results from database**
38. `$testimonials=$query5->rowCount();` **//total number of results found**
39. `print` total number of testimonials

## **4.9 Class Name: Admin Post Vehicles**

**Pseudo-code :**

**Input:**

**Output:** post-vehicle.php      **//redirect to post-vehicle.php**

1. `session_start();` **// start server SESSION which stores presently logged in users details**
2. `error_reporting(0);`
3. `include('includes/config.php');` **//including config.php page**
4. `IF`(any user is logged in )
5. `header('location:index.php');` **//redirect to index.php page**
6. `ENDIF;`
7. `ELSE;`
8. `IF`(admin click on submit button)
9. `$vehicletitle=$_POST['vehicletitle'];` **//get vehicletitle input by admin**
10. `$brand=$_POST['brandname'];` **//get vehiclebrand input by admin**
11. `$vehicleoverview=$_POST['vehicalorview'];` **//get vehicleoverview input by admin**
12. `$priceperday=$_POST['priceperday'];` **//get vehiclepriceperday input by admin**
13. `$fueltype=$_POST['fueltype'];` **//get vehiclefueltype input by admin**
14. `$modelyear=$_POST['modelyear'];` **//get vehiclemodelyear input by admin**
15. `$seatingcapacity=$_POST['seatingcapacity'];` **//get vehicleseating capacity input by admin**
16. `$vimage1=$_FILES["img1"]["name"];` **//get vehicle image provided by admin**
17. `$vimage2=$_FILES["img2"]["name"];` **//get vehicle image provided by admin**

18. \$vimage3=\$\_FILES["img3"]["name"]; **//get vehicle image provided by admin**
19. \$vimage4=\$\_FILES["img4"]["name"]; **//get vehicle image provided by admin**
20. \$vimage5=\$\_FILES["img5"]["name"]; **//get vehicle image provided by admin**
21. \$airconditioner=\$\_POST['airconditioner']; **//airconditioner available or not**
22. \$powerdoorlocks=\$\_POST['powerdoorlocks']; **//powerdoorlocks available or not**
23. \$antilockbrakingsys=\$\_POST['antilockbrakingsys']; **//antibracking system available or not**
24. \$brakeassist=\$\_POST['brakeassist']; **//brake assist available or not**
25. \$powersteering=\$\_POST['powersteering']; **//power steering available or not**
26. \$driverairbag=\$\_POST['driverairbag']; **//driver air bag available or not**
27. \$passengerairbag=\$\_POST['passengerairbag']; **//passenger air bag available or not**
28. \$powerwindow=\$\_POST['powerwindow']; **//power window available or not**
29. \$cdplayer=\$\_POST['cdplayer']; **//cd player available or not**
30. \$centrallocking=\$\_POST['centrallocking']; **//centrallocking available or not**
31. \$crashcensor=\$\_POST['crashcensor']; **//crash censor available or not**
32. \$leatherseats=\$\_POST['leatherseats']; **//leather seats available or not**
- 33.move\_uploaded\_file(\$\_FILES["img1"]["tmp\_name"],"img/vehicleimages/".\$\_FILES["img1"]  
["name"]);
- 34.move\_uploaded\_file(\$\_FILES["img2"]["tmp\_name"],"img/vehicleimages/".\$\_FILES["img2"]  
["name"]);
- 35.move\_uploaded\_file(\$\_FILES["img3"]["tmp\_name"],"img/vehicleimages/".\$\_FILES["img3"]  
["name"]);
- 36.move\_uploaded\_file(\$\_FILES["img4"]["tmp\_name"],"img/vehicleimages/".\$\_FILES["img4"]  
["name"]);

```
37.move_uploaded_file($_FILES["img5"]["tmp_name"],"img/vehicleimages/".$_FILES["img5"]
["name"]);
```

```
38. $sql="INSERT INTO
tblvehicles(VehiclesTitle,VehiclesBrand,VehiclesOverview,PricePerDay,FuelType,ModelYear,Se
atingCapacity,Vimage1,Vimage2,Vimage3,Vimage4,Vimage5,AirConditioner,PowerDoorLocks,
AntiLockBrakingSystem,BrakeAssist,PowerSteering,DriverAirbag,PassengerAirbag,PowerWind
ows,CDPlayer,CentralLocking,CrashSensor,LeatherSeats)
VALUES(:vehicletitle,:brand,:vehicleoverview,:priceperday,:fueltype,:modelyear,:seatingcapacit
y,:vimage1,:vimage2,:vimage3,:vimage4,:vimage5,:airconditioner,:powerdoorlocks,:antilockbrak
ingsys,:brakeassist,:powersteering,:driverairbag,:passengerairbag,:powerwindow,:cdplayer,:centr
allocking,:crashcensor,:leatherseats)";
```

**// SQL query to insert all filled details by admin into tblvehicles table**

```
39. $query = $dbh->prepare($sql); // preparing SQL query
```

```
40. $query->bindParam(':vehicletitle',$vehicletitle,PDO::PARAM_STR); //binding SQL
parameter
```

```
41. $query->bindParam(':brand',$brand,PDO::PARAM_STR); //binding SQL parameter
```

```
42.$query->bindParam(':vehicleoverview',$vehicleoverview,PDO::PARAM_STR); //binding
SQL parameter
```

```
43. $query->bindParam(':priceperday',$priceperday,PDO::PARAM_STR); //binding SQL
parameter
```

```
44. $query->bindParam(':fueltype',$fueltype,PDO::PARAM_STR); //binding SQL parameter
```

```
45. $query->bindParam(':modelyear',$modelyear,PDO::PARAM_STR); //binding SQL
parameter
```

```
46. $query->bindParam(':seatingcapacity',$seatingcapacity,PDO::PARAM_STR); //binding
SQL parameter
```

```
47. $query->bindParam(':vimage1',$vimage1,PDO::PARAM_STR); //binding SQL parameter
```

```
48. $query->bindParam(':vimage2',$vimage2,PDO::PARAM_STR); //binding SQL parameter
```

```
49. $query->bindParam(':vimage3',$vimage3,PDO::PARAM_STR); //binding SQL parameter
```

50. \$query->bindParam(':vimage4',\$vimage4,PDO::PARAM\_STR); **//binding SQL parameter**

51. \$query->bindParam(':vimage5',\$vimage5,PDO::PARAM\_STR); **//binding SQL parameter**

52. \$query->bindParam(':airconditioner',\$airconditioner,PDO::PARAM\_STR); **//binding SQL parameter**

53.\$query->bindParam(':powerdoorlocks',\$powerdoorlocks,PDO::PARAM\_STR); **//binding SQL parameter**

54.\$query->bindParam(':antilockbrakingsys',\$antilockbrakingsys,PDO::PARAM\_STR); **//binding SQL parameter**

55. \$query->bindParam(':brakeassist',\$brakeassist,PDO::PARAM\_STR); **//binding SQL parameter**

56. \$query->bindParam(':powersteering',\$powersteering,PDO::PARAM\_STR); **//binding SQL parameter**

57. \$query->bindParam(':driverairbag',\$driverairbag,PDO::PARAM\_STR); **//binding SQL parameter**

58.\$query->bindParam(':passengerairbag',\$passengerairbag,PDO::PARAM\_STR); **//binding SQL parameter**

59. \$query->bindParam(':powerwindow',\$powerwindow,PDO::PARAM\_STR); **//binding SQL parameter**

60. \$query->bindParam(':cdplayer',\$cdplayer,PDO::PARAM\_STR); **//binding SQL parameter**

61. \$query->bindParam(':centrallocking',\$centrallocking,PDO::PARAM\_STR); **//binding SQL parameter**

62. \$query->bindParam(':crashcensor',\$crashcensor,PDO::PARAM\_STR); **//binding SQL parameter**

63. \$query->bindParam(':leatherseats',\$leatherseats,PDO::PARAM\_STR); **//binding SQL parameter**

```
64. $query->execute(); //running SQL query
65. $lastInsertId = $dbh->lastInsertId(); //get last inserted id
66. IF(any data inserted into table)
67.     print  Vehicle posted successfully
68. ENDIF;
69. ELSE
70.     print  Something went wrong. Please try again
71. ENDELSE;
```