**Design Document**

**For**

**Car Purchasing Web App**

**Prepared By: Team Project 1**

1. **Bosy Shafeey**
2. **Fatma Mohamed**
3. **Janaat Saeed**
4. **Hagar Mohamed**
5. **Mina Medhat**
6. **Sara Ibrahim**

**Table of Contents**

Contents

[1. High Level](#_30j0zll)

[1.1 High Level Definition](#_1fob9te)

[1.2 High Level Diagram](#_3znysh7)

[1.3 High Level Description](#_2et92p0)

[2. Data Flow Diagram](#_tyjcwt)

[2.1 Data Flow Diagram Definition](#_3dy6vkm)

[2.2 Data Flow Diagram](#_1t3h5sf)

[2.3 Data Flow Diagram Description](#_4d34og8)

[3. Class Diagram](#_2s8eyo1)

[3.1 Class Diagram Definition](#_17dp8vu)

[3.2 Class Diagram](#_3rdcrjn)

[3.3 Class Diagram Description](#_26in1rg)

[4. Use Case](#_lnxbz9)

[4.1 Use Case Definition](#_35nkun2)

[4.2 Use Case Diagram](#_1ksv4uv)

[4.3 Use Case Description](#_44sinio)

[5. Sequence Diagrams](#_2jxsxqh)

[6. Wire Frames](#_4i7ojhp)

.

# 

# 1. High Level

## 1.1 High Level Definition

Explains the architecture that would be used for developing a software product. The architecture diagram provides an overview of the entire system, identifying the main components that would be developed for the product and their interfaces.

## 

## 

## 1.2 High Level Diagram



## 

## 1.3 High Level Description

HLD diagram consists of actors, middleware modules, functions, and database.

There are three roles for the system.

First role is admin who can perform the following actions: add user(s), delete user(s), add car(s), delete car(s), and change car(s) status.

The second role is registered user who can perform the following actions: view free cars’ details, reserve free cars, view bought car(s) list and view reserved car(s) list.

The third role is passerby (guest) who can only view free cars without details, any unregistered guest can sign up.

Middleware modules are authentication, authorization, administrative control panel and cars view controller. These Modules are intermediate layers between the database and the end users functions to restrict and differentiate between each role for database accessibility.

# 2. Data Flow Diagram

# 2.1 Data Flow Diagram Definition

**Data-flow diagram** (DFD) is a way of representing a flow of data of a system.

The DFD also provides information about the outputs and inputs of each entity and the process itself. It provides a graphical representation of how information moves between processes in a system

# 2.2 Data Flow Diagram

# 

# 

# 

# 2.3 Data Flow Diagram Description

Car Purchasing Web Application is an innovative internet reservation software designed for cars.

There is only one database for all data system it contains user data, car data, and admin data.

There are two actors in our system [Admin and customer],

The first actor is Admin who is able to login to admin dashboard it will be navigated to login page, Admin set a login data and click to login button the request will be sent to database for verifying admin data if the data are correct database will response correct data so the user login successfully and finally admin enable access admin features

Such as

a) Admin can delete users from the database.

b) Admin can view users.

c) Admin can add users to the database.

d) Admin can change car status from reserved to bought.

The second actor is a customer who may be user in the system or guest and he able to make a registration.

When the customer opens web application the land page will be Home Page and customer enable to choose between make login or registration or browse the website.

If the user chooses login it will be navigated to the login page, user set a login data and click to login button the request will be sent to the database for verifying user data if the data are correct database will response correct data so the user login successfully.

If the guest chooses Registration page it will be navigated to the Registration page, a guest set a registration data and click to registration button, the request will be sent to the database for saving guest data if the data are correct database will response success saving so the Guest will be a user in our application.

If the user chooses to browse website is able to make a search based on the Search criteria filter for car color, price, and model. The request will be sent to the database to get car results based on these criteria so the database will send the car results information and user select from multiple results

Then the user selects a car from multiple results and clicks to it so the car description page will appear if he selects to reservation button it must be logged first.

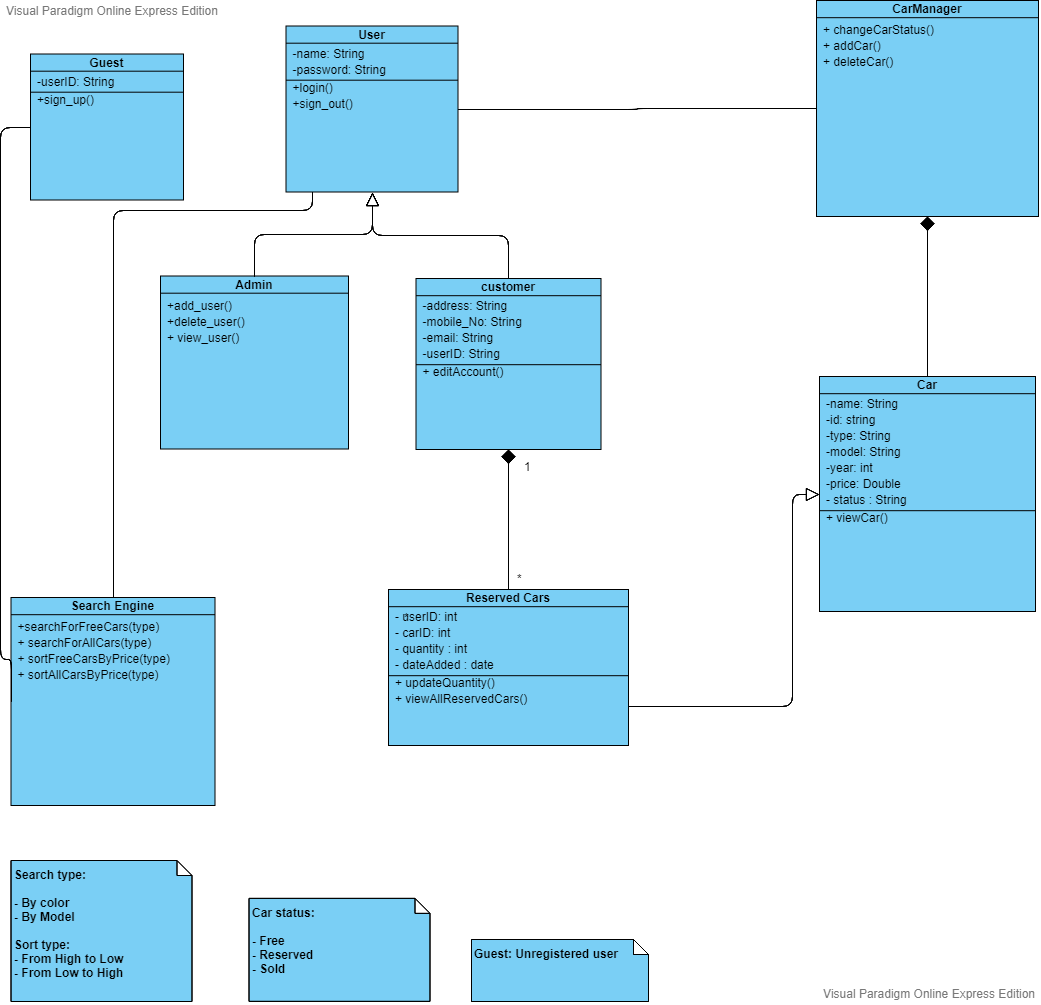
# 

# 3. Class Diagram

# 3.1 Class Diagram Definition

The following classes are used to design a car purchasing web application. A class diagram is used to represent the relationship between the classes in a Unified Modeling Language (UML).

# 3.2 Class Diagram



# 3.3 Class Diagram Description

3.3.1 Classes

* User Class:  
  User class represents any user has authority in the car purchasing web application.  
  There are two types of users: Admin and the registered customer.
* Admin Class:  
  Admin can add new customers, delete fake customers or view customer or search for cars (Free, reserved or sold cars) using search engine class
* Customer Class:  
  Customer class represents the registered user who has his own account with his information, he also can edit his/her information and can search for free cars using search engine class.
* Guest Class:  
  Guest Class represents the unregistered user, who can only search for free cars without more details and cannot reserve any cars without registration.
* Cars Class:  
  Cars Class represents our products’ details, this class will be affected if the admin add or delete any car and if the customer search for cars or reserve one.
* Reserved Cars Class:  
  This Class represents the cars reserved for each user and it’s quantity.
* Search Engine class:  
  The class that handles searching for cars in the web application, so the customer can search for free cars or sort free cars based on its price and admin can search for types of cars ( Free, reserved or sold) or sort them.
* Car Manager:  
  This Class manage any action related to cars including add cars, delete cars, change cars status from free to reserve (as a customer) or from reserve to sold/Free (as Admin).

3.3.2 Internal Data Structure:

* User Class:  
   User name: it’s the name of either the admin or the customer that used to log in  
   Password: the password of the user ( customer or admin)
* Customer Class:  
   User ID: This is a unique identifier for any Customer  
   Address: the customer’s address  
   Email: the customer’s email.  
   Phone: the customer’s phone number
* Guest Class:  
  User ID: This is a unique identifier for any user
* Car Class:  
  Name: it’s a description  
  Id: each car has a unique identifier   
  Type: the brand of the vehicle  
  Model: the name used by a manufacturer to the market  
  Year: The car model year  
  Price: the car price on the website  
  Status: Free, Reserved or sold.
* Reserved Car Class:  
  User Id: the id of the user who reserved the car  
  Car id: identify the car reserved  
  Quantity: the quantity reserved for this car  
  Date Added: the date in which the user reserve the car

# 4. Use Case

# 4.1 Use Case Definition

The following use case is used to design a car purchasing web application. A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system.

# 4.2 Use Case Diagram



# 4.3 Use Case Description

**4.3.1 Admin Use Case**

* **Login** **(CAR\_UC\_01)**: Admin can login to admin home page using admin credential (user name and password) after clicking on login button from the car home page.
* **Change car status** **(CAR\_UC\_02)**: Admin can change car status from reserved to sold and from reserved to free from the status drop list found under each car in " all cars" page
* **Add users** **(CAR\_UC\_03)**: Admin can add users to the system after clicking on “add users” button from admin home page and it will redirect the admin to registration page.
* **View users** **(CAR\_UC\_04)**: Admin can view all information of registered users such as (username, phone, address, and email) except the password using "users" button in the admin home page.
* **Delete users** **(CAR\_UC\_05)**: In admin home page the admin can remove users from the system after clicking on "delete" button found under each user in "users" page.
* **Add cars** **(CAR\_UC\_06)**: Admin can add car(s) to the system using “Add car” tab in the admin home web page.
* **Remove cars** **(CAR\_UC\_07**): Admin can remove car(s) from the system using delete car button found under each car in the "All cars" page
* **Search for all cars** **(CAR\_UC\_08**): Admin home page shall contain "All Cars" button, once it pressed “All cars” (free, reserved and sold) shall be reviewed.
* **Search for reserved cars** **(CAR\_UC\_09)**: Admin can view all the cars the user had reserved using the admin home page after clicking on” Reserved “button.

**4.3.2 User Use Case**

* **Login** **(CAR\_UC\_10)**: The user can login using valid username and password after clicking on "login" button from the header of car home page and the user will be redirect to the user home page.
* **Logout** **(CAR\_UC\_11)**: The user can logout from user home page any time he/she wants after clicking on "logout" button from the header of user home page and it will redirect the user to car home page.
* **View cars details** **(CAR\_UC\_12)**: The user can view all cars details after clicking on see more button which redirects him/her to the car details page that may include( the model year, engine CC , gear transmission , condition(new or used) , kilometers(for used cars)
* **Search by price from high to** **low (CAR\_UC\_13)**: The user can sort cars by price by choosing "High to low" from the price drop down list found in the home page.
* **Search by price from low to high** (**CAR\_UC\_14)**: The user can sort cars by price by choosing "low to high" from the price drop down list found in the home page
* **Search by model (CAR\_UC\_15)**: The user can search for different cars models after writing the desired car model in the search bar found in home page of car web app
* **Search by Color** **(CAR\_UC\_16)**: The user can search for cars colors by choosing the desired color from the drop down list found in the home page.
* **Reserve (CAR\_UC\_17)**: The user can reserve a car by clicking on “Reserve" button found in the car details page so the car status will be converted from free to reserve.
* **Edit user information** **(CAR\_UC\_18)**: The user can edit his/her personal information (phone number-address) from "My profile" page after clicking on" my profile button" from user home page.

**4.3.3 Guest Use Case**

* **Register** **(CAR\_UC\_19)**: The guest can register to car web app after clicking on registration button from the header of home page and it will redirect him/her to registration page and it’s contain:

**Enter email (CAR\_UC\_25)**: Email field shall take the pattern as"email@example.com"

**Enter user name (CAR\_UC\_26)**: Username field shall accept all characters and numbers with maximum size 20 characters.

**Enter password (CAR\_UC\_27)**: Password field shall accept upper case, lowercase and special characters, size shall be more than 8 characters.

**Confirm Password (CAR\_UC\_28):** Shall match the same characteristics of the password field.

**Enter Address** (**CAR\_UC\_29):** Address field shall accept all characters and numbers with maximum size 100 characters.

**Enter phone number (CAR\_UC\_30):** Phone number field shall accept only 11 numbers.

* **Search by price from high to** **low (CAR\_UC\_20)**: The guest can sort cars by price by choosing
* "High to low" from the price drop down list found in the home page

**Search by price from low to high** (**CAR\_UC\_21):** The guest can sort cars by price by choosing "low to high" from the price drop down list found in the home page.

* **Search by model (CAR\_UC\_22)**: The user can search for different cars models after writing the desired car model in the search bar found in home page of car web app.
* **Search by color** **(CAR\_UC\_23)**: The guest can search for cars colors by choosing the desired color from the drop down list found in the home page
* **View cars with limited details (CAR\_UC\_24)**: The guest can only view list of cars with limited details (color, model, and price) from car web app home page.

# 5 Sequence Diagrams

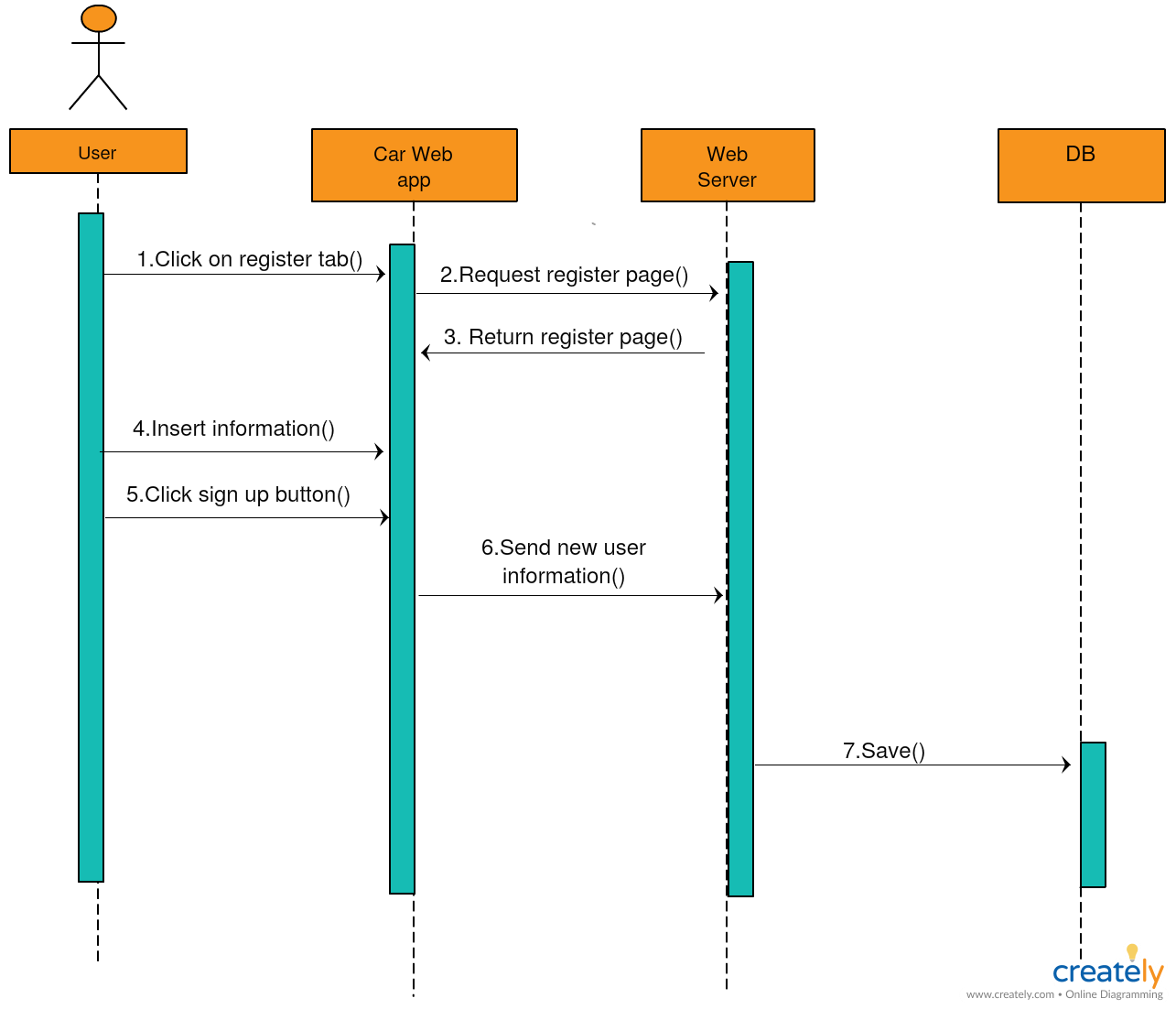
## Sequence Diagrams Definition

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction

## Car web application Sequence Diagrams

**ID:** **Car\_Seq\_01**

**Registration Sequence** **diagram**

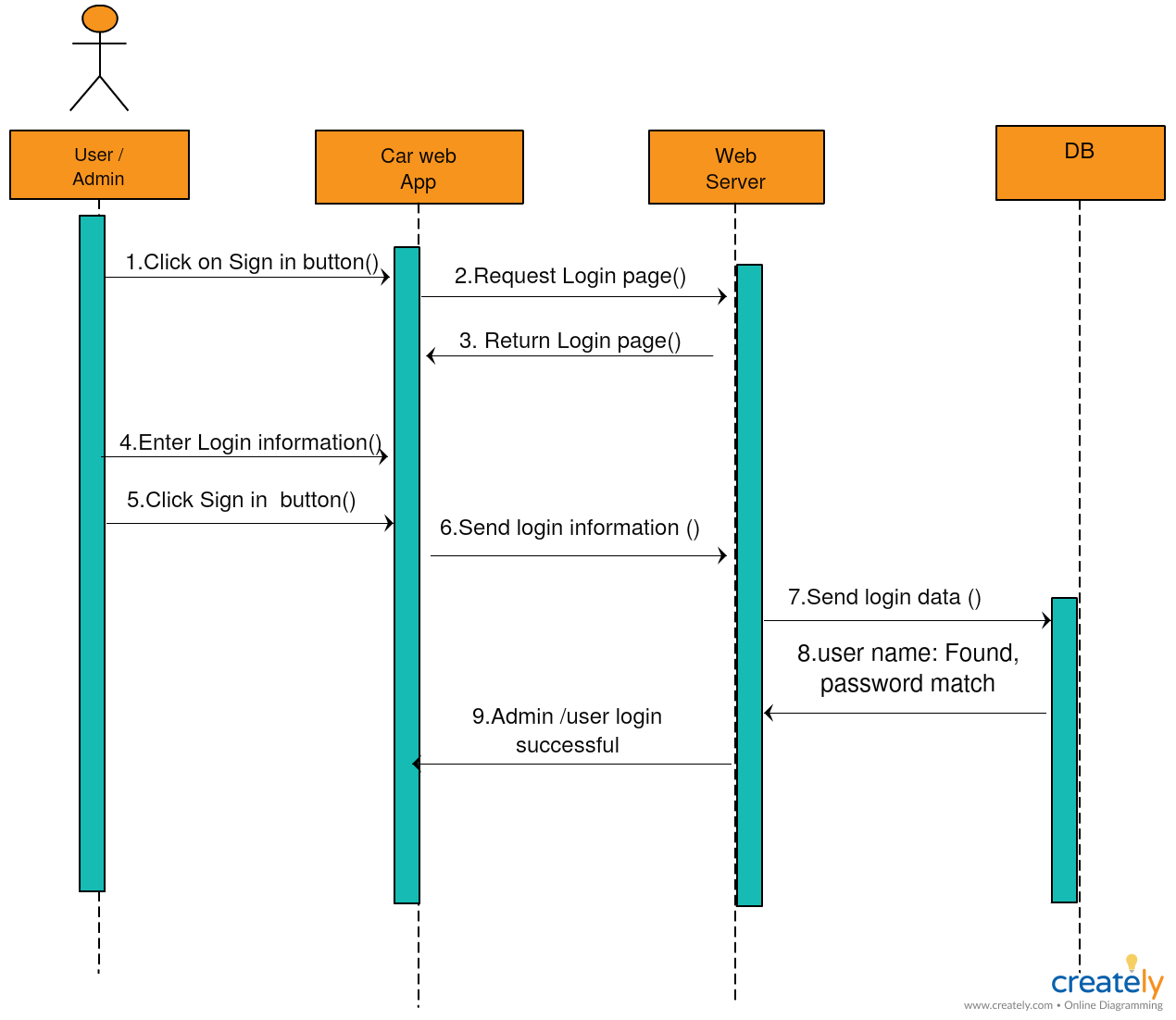
****

**Description:**

This diagram shows the sequence that the user shall follow when he/she want to register in the car purchasing application**,** the user shall enter those data for registration (username, address, email, phone number password, and confirmation password). Then press on sign up button, after that the data shall be saved on Database.

**ID:** **Car\_Seq\_02**

**Login Sequence diagram**

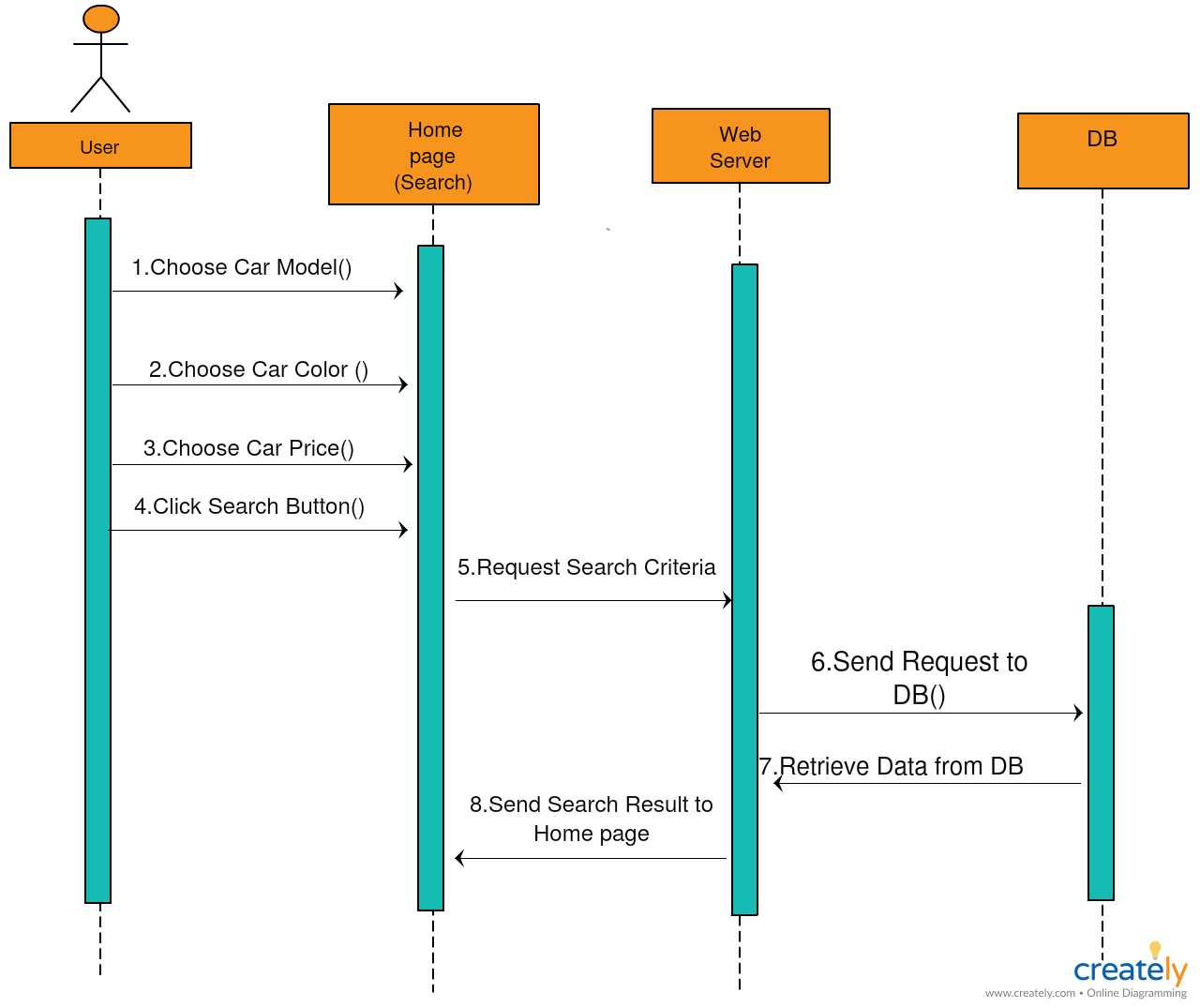
****

**Description:**

The user/admin can log in using his/her username and password then click on the login button if the username and password matched the login data on the database the user can log in

**ID:** **Car\_Seq\_03**

**Searching Sequence diagram**

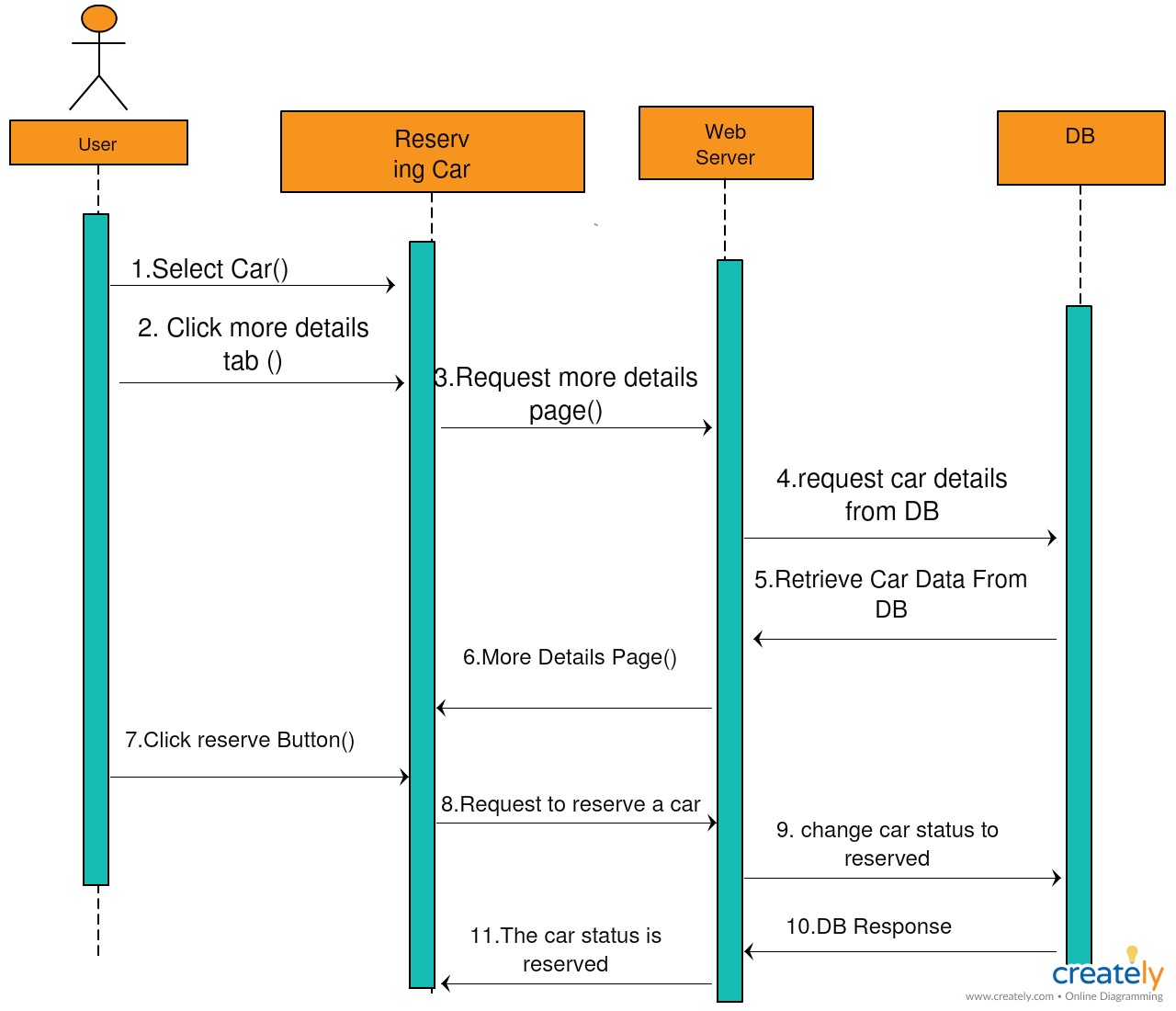
****

**Description:**

The user can search in the car purchasing web application using different criteria such as model, price or color, then click on the search button, after clicking the server send to the database and response shall appear in Homepage.

**ID:** **Car\_Seq\_04**

**Reserving car Sequence diagram**

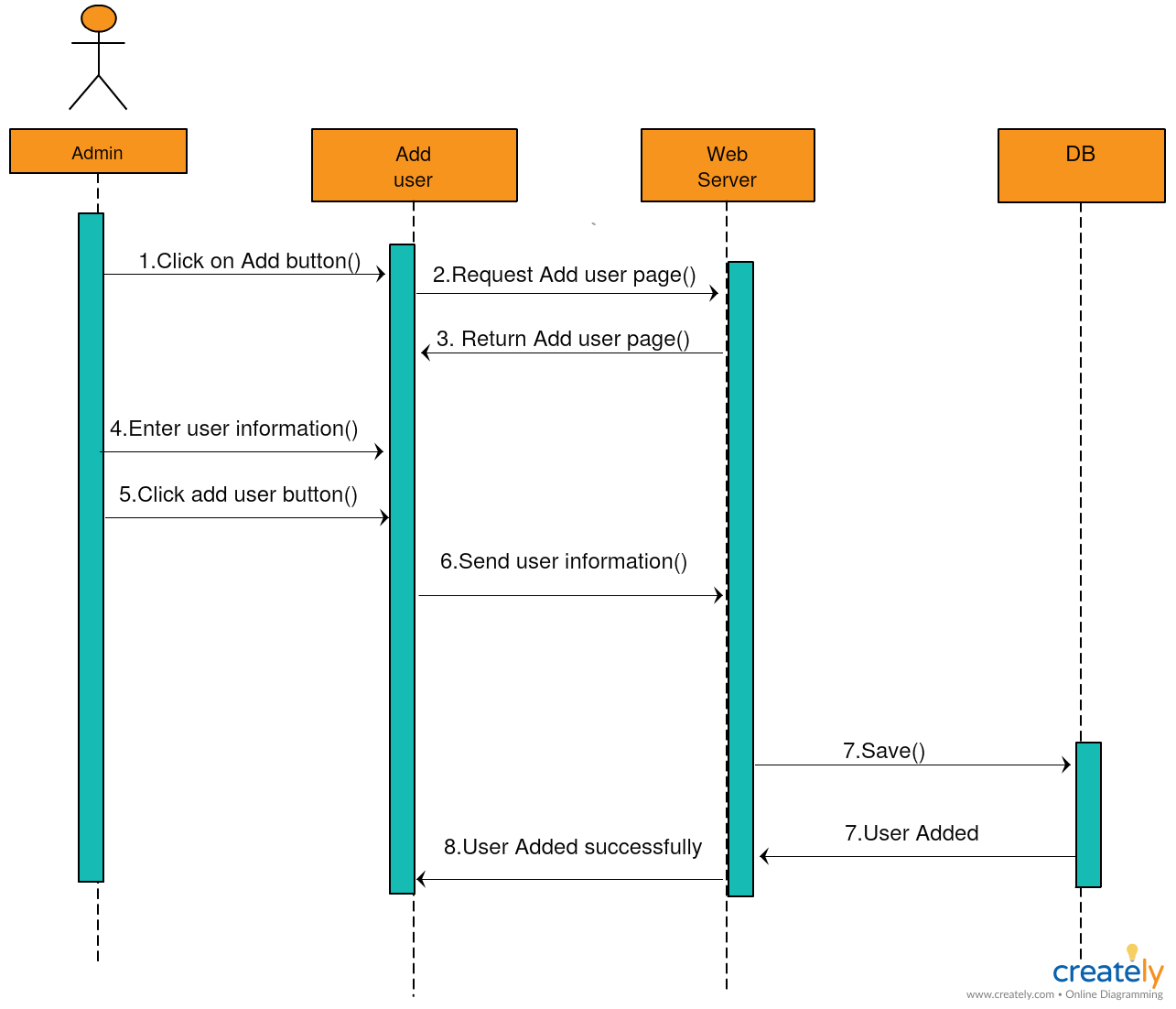
****

**Description:**

The user can select a car from the car purchasing web application, then click on more details tab, the database response will display on the more details tab then the user shall click on reserve button to reserve a car, then the car status changed to reserve then the admin can change car status to bought.

**ID:** **Car\_Seq\_05**

**Adding User Sequence diagram**

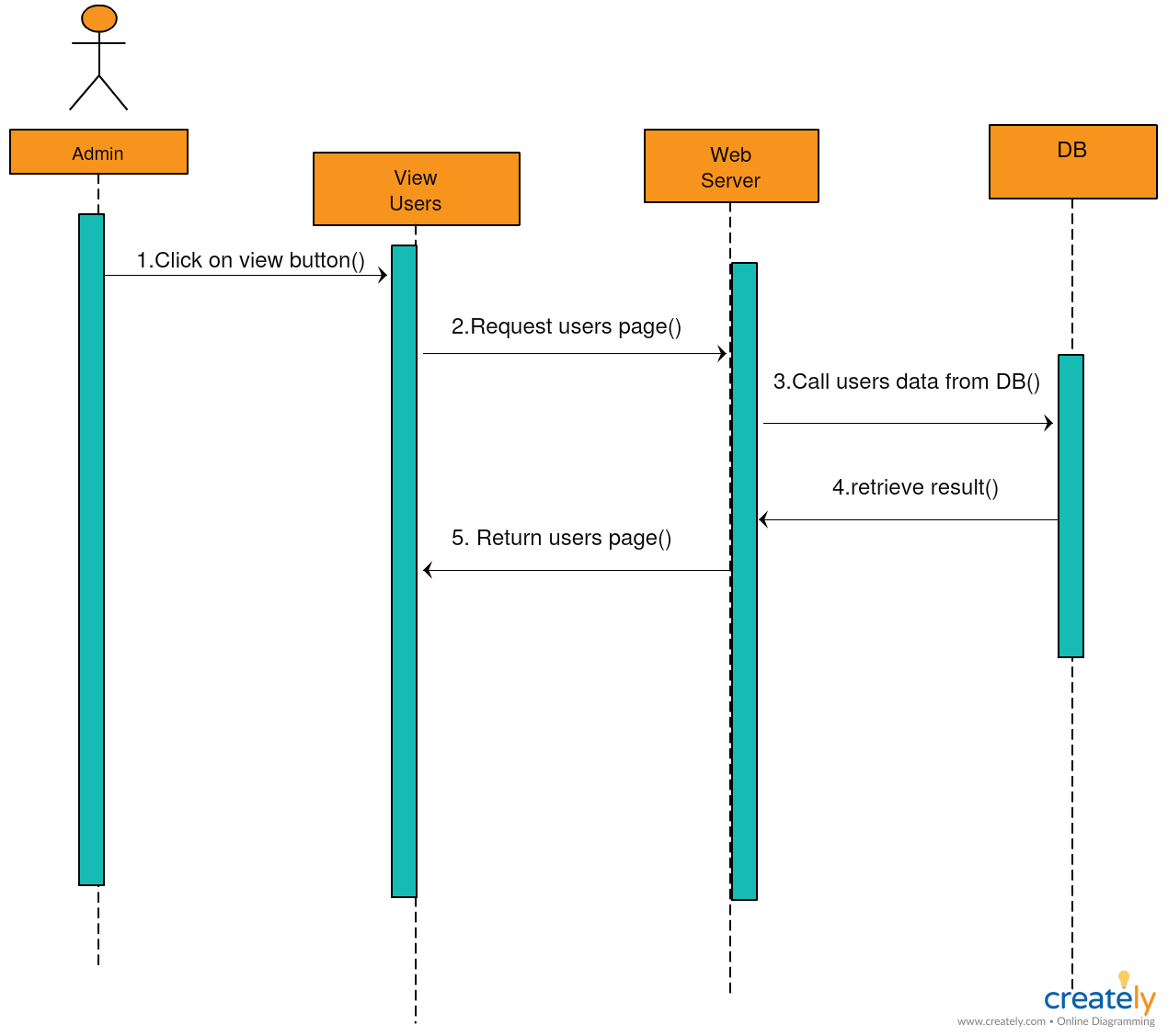
****

**Description:**

The admin can add a user on the application through clicking on the add button then enter the user data then he/she shall click on add user button.

**ID:** **Car\_Seq\_06**

**View Users Sequence diagram**

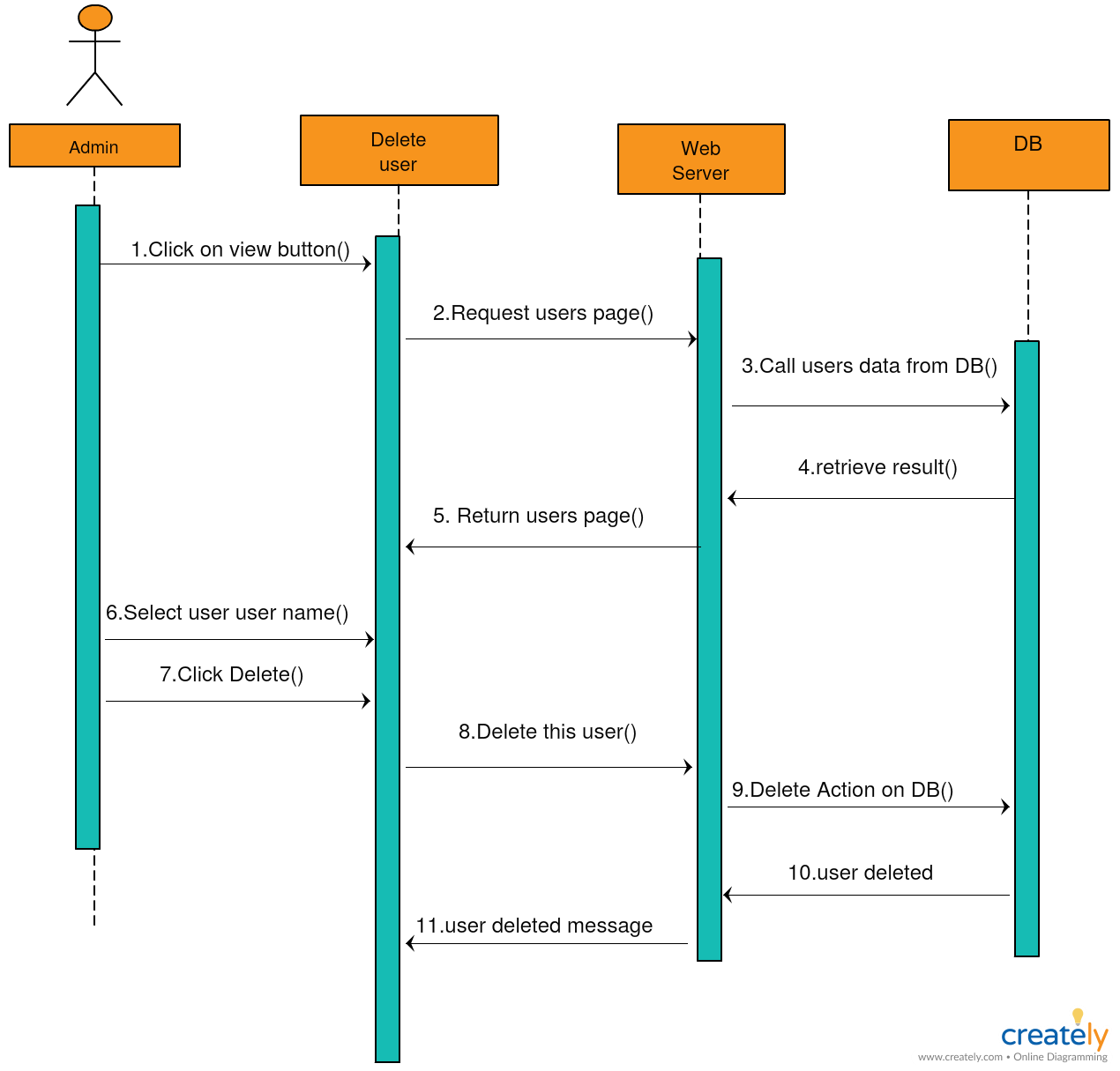
****

**Description:**

The admin can view users, admin shall click on the view users button to view registered users in the application, and then the server sends to database .the database shall response and return the users data.

**ID:** **Car\_Seq\_07**

**Delete User Sequence diagram**

****

Description:

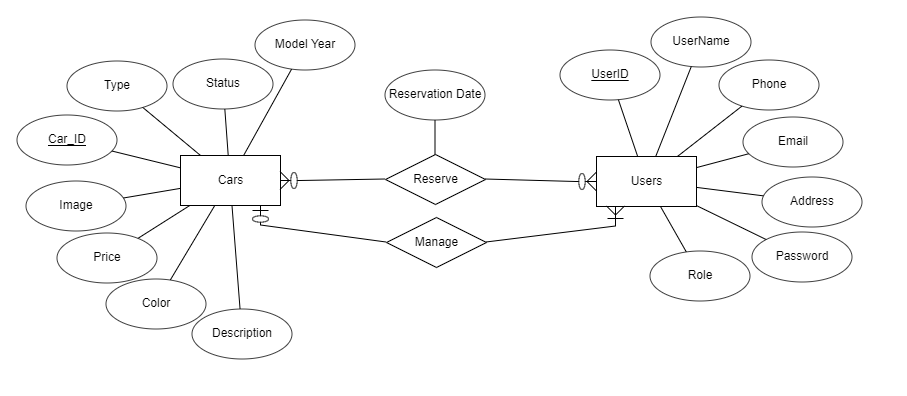
## The admin can delete user from the application, the admin shall click on view button first, then select the user he/she wants to delete. Then click on the delete button, the delete action shall send to the server, the server sends this action to the database. The database shall send a response to server “user deleted”, then user deleted message will send to the admin site.

# 6. Entity Relationship Diagram (ERD)

# 6.1 Entity Relationship Diagram

It’s a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities in the car purchasing web application.

# 6.2 Entity Relationship Diagram



# 

# 6.3 Entity Relationship Diagram Description

The system is designed to be simple, we will have just three entities

* User Entity:  
  User ID: a unique integer number for each user  
  User name: a unique name for each user  
  Password: a complex password   
  Email: a unique email for each user  
  Address: user address  
  Phone: user can add his/ her phone number

Role: the user role is (Customer or Admin)

* Cars:  
  Car ID: each car has a unique car ID  
  Description: each car has a detailed description with more attributes than it’s model, color and price

Type: represents the car type  
Model: represents the car model year  
Price: the car price  
Color: the car color  
Status: the car status can be free, reserved or sold

Image: the car image

* Reserve relationship between user and Cars  
  represents the relation between **Customer** and cars where any customer can change the status of car from free to reserved and the system will save the **Reservation date, user ID and Car ID**
* Manage relationship between user and Cars  
  represents the relationship between **Admin**  and cars where admin manage the cars because he car add cars, delete cars or change car status to free or sold

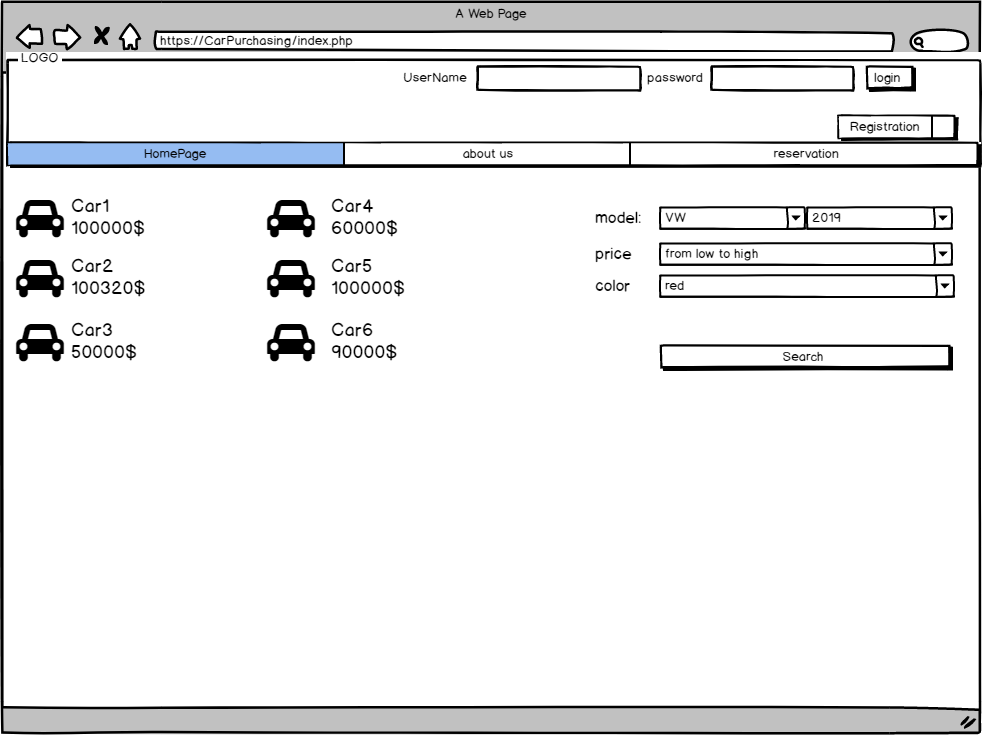
# Wire Frames

# 7.1 Wire Frames

# Wire framing is a way to design a website service at the structural level. A wireframe is commonly used to layout content and functionality on a page which takes into account user needs and user journeys. Wireframes are used early in the development process to establish the basic structure of a page before visual design and content are added.

**ID:** **Car\_Wireframe\_01**

**Homepage wireframe**

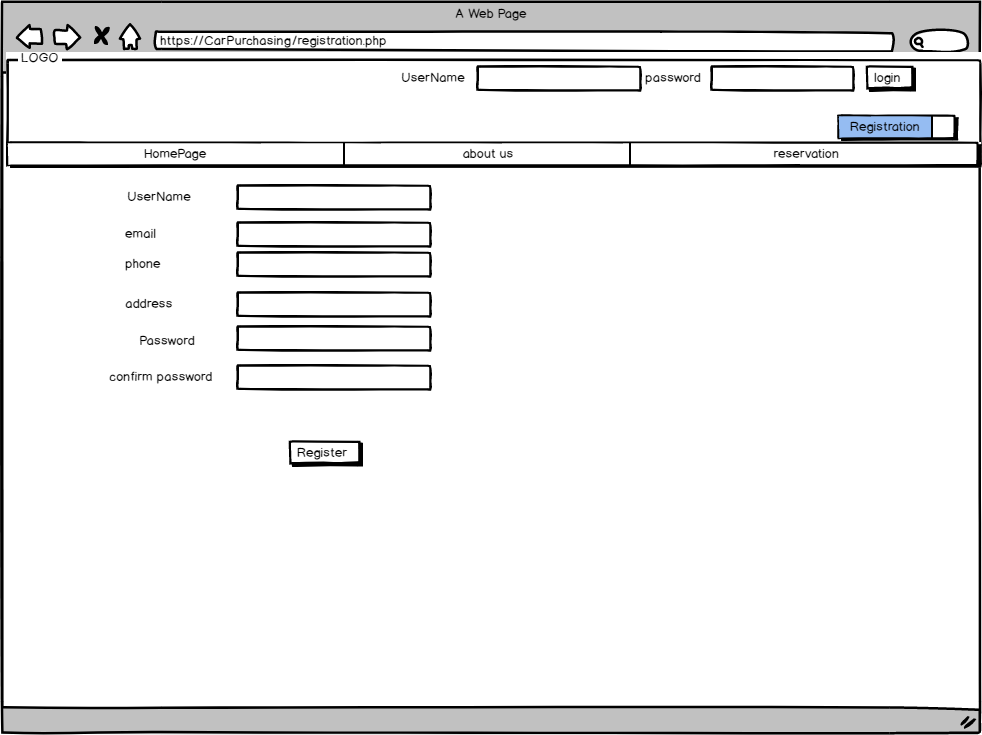
****

**Description:**

The homepage contains the most popular cars and any user can search on the application of cars.

**ID:** **Car\_Wireframe\_02**

**Registration wireframe**

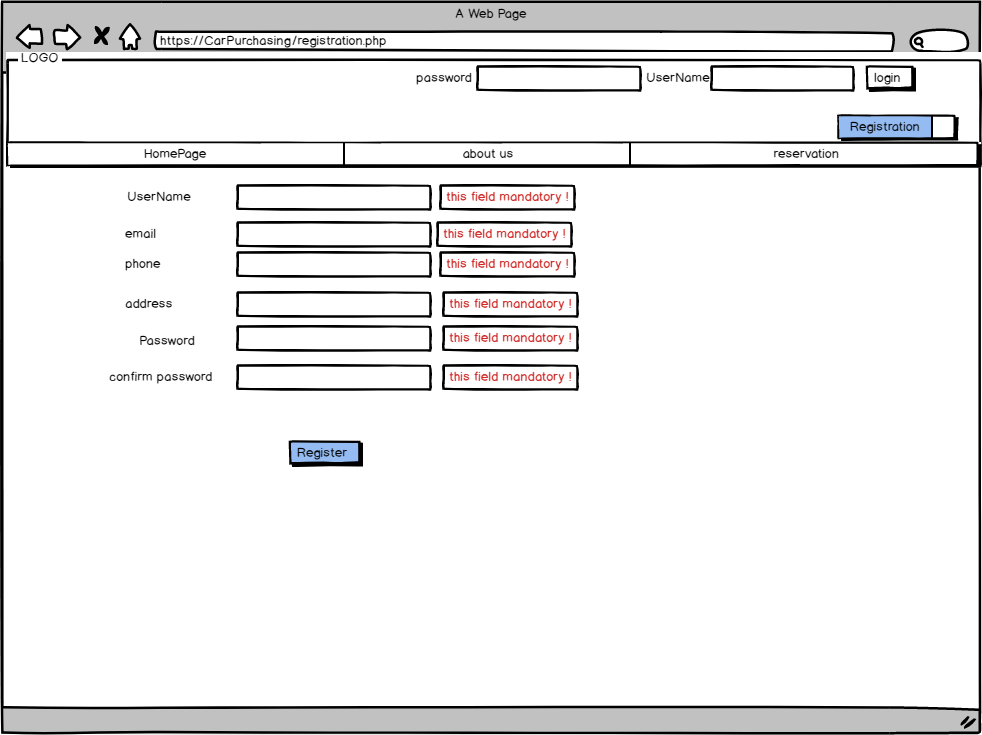


**Description**:

The user can register on the application by click on the register tab and enter his/her Name, Email, Password, Address, Confirm Password, and Phone)

**ID:** **Car\_Wireframe\_03**

**The Registration error**

****

**Description:**

When the user tries to click on the register button without entering the data this error messages shall appear.

**ID:** **Car\_Wireframe\_04**

**The Registration constraints**

****

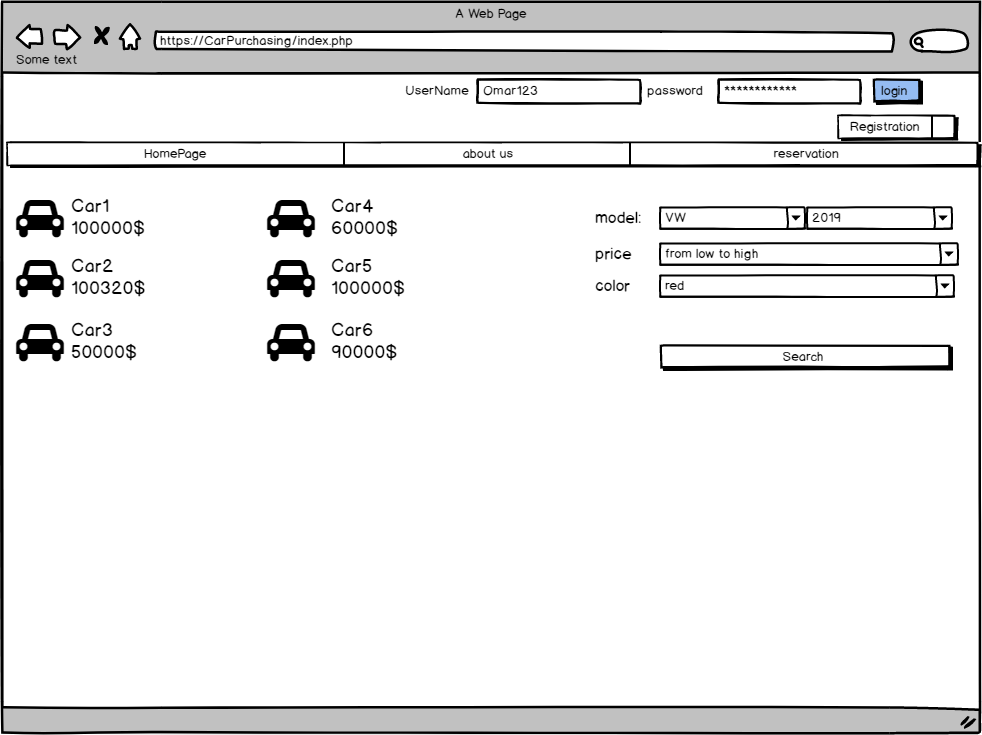
**Description:**

When the user enters invalid email an error message will appear “The email shall be example@domain.com”

When the user enters invalid password an error message will appear “The password shall contain at least one special character, upper letter, lower letter, number, and at least 8 characters.

**ID:** **Car\_Wireframe\_05**

**Login wireframe**

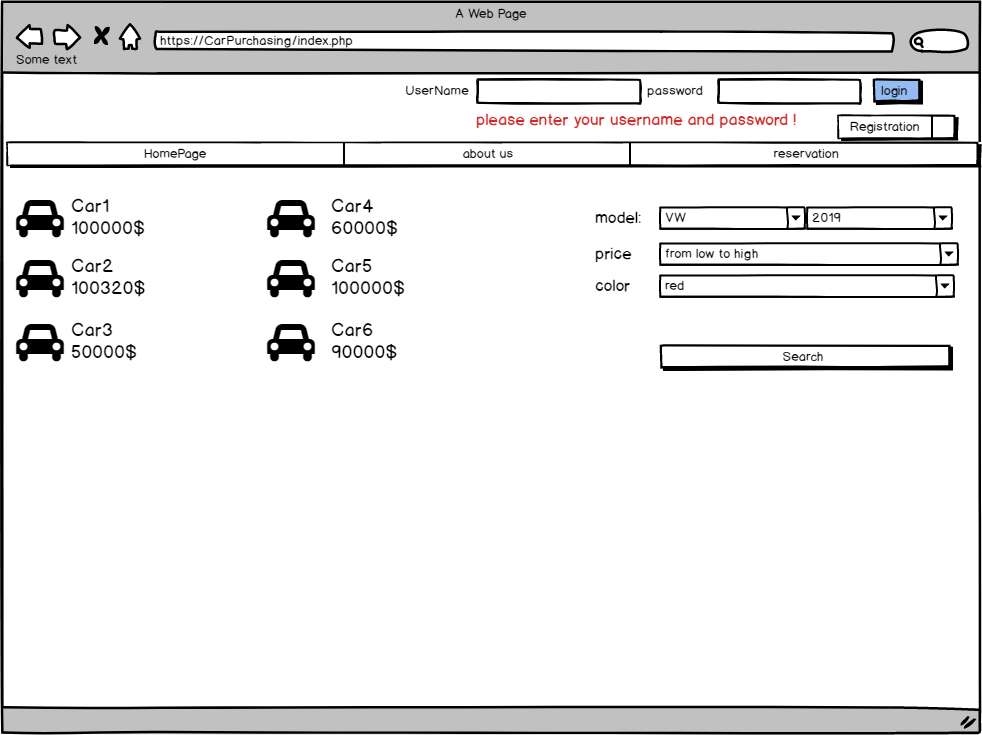
****

**Description:**

When the user enters his/her correct username and correct password then click on the login button, the login function will be with no error message.

**ID:** **Car\_Wireframe\_06**

**The Login error**

****

**Description:**

When the user clicks on login button without entering login data an error message will appear. “Please enter username and password”

**ID:** **Car\_Wireframe\_07**

**About us page**

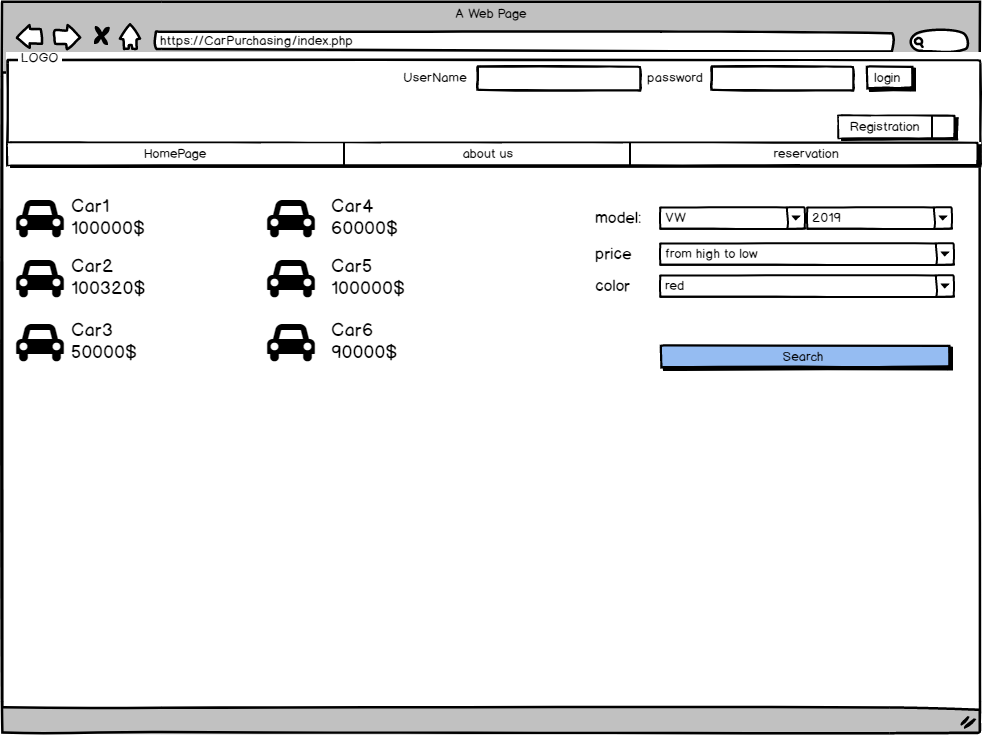
****

**Description:**

About us page contains information about the car purchasing application history.

**ID:** **Car\_Wireframe\_08**

**Searching in the application**

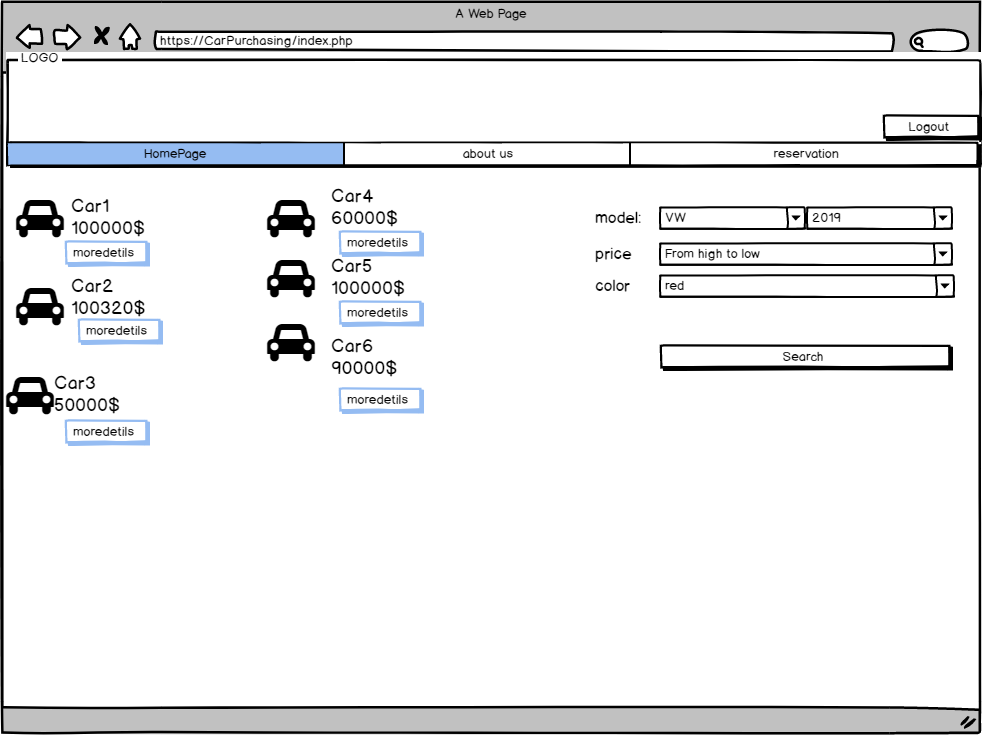
****

**Description:**

The user can search on the application by car model, color and price.

**ID:** **Car\_Wireframe\_09**

**More details tab**

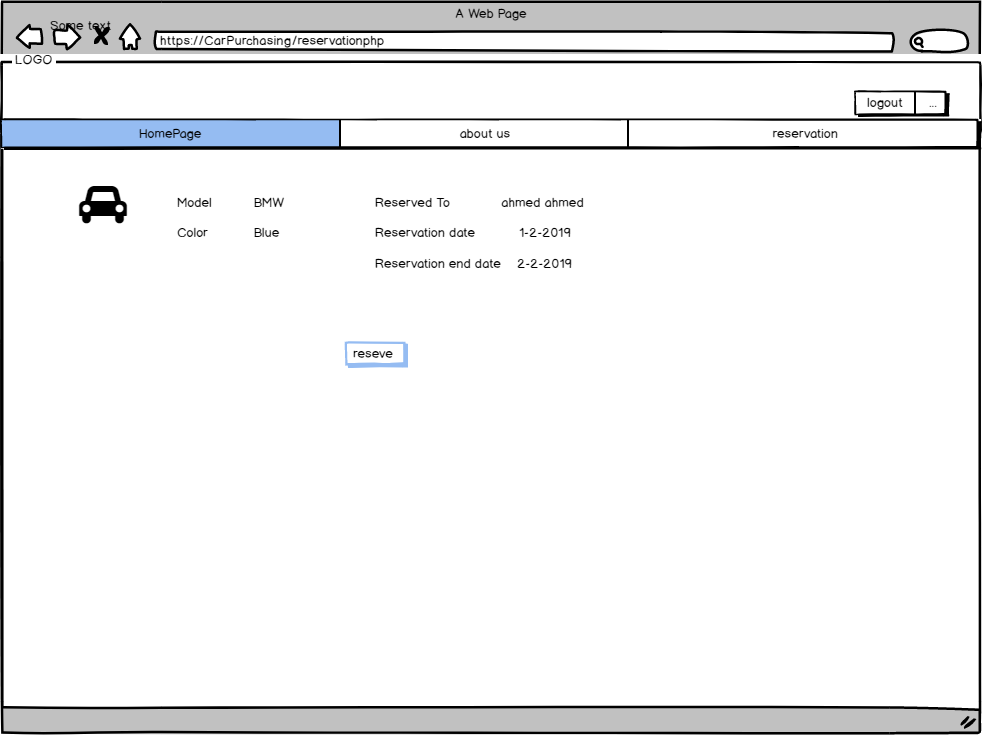
****

**Description:**

The more details tab shows more information about the selected car, this page displays only for logged users.

**ID:** **Car\_Wireframe\_10**

**Reserving a car**

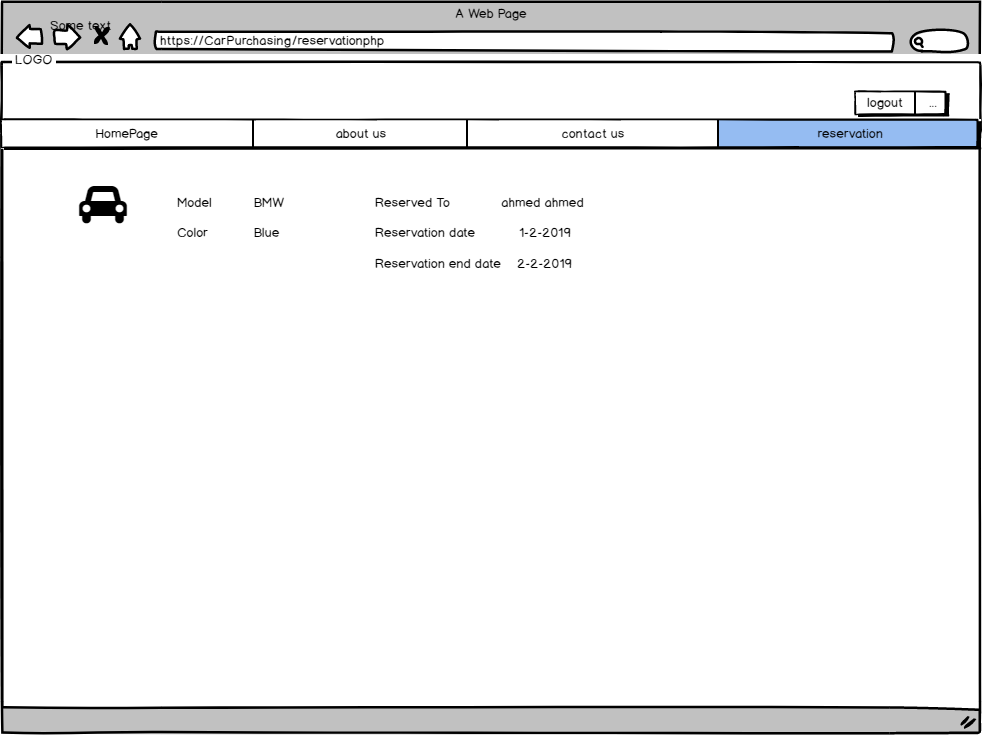
****

**Description:**

The user can click on the reserve button if he wants to buy this car, then this car status will change to be reserved.

**ID:** **Car\_Wireframe\_11**

**Reservation page**

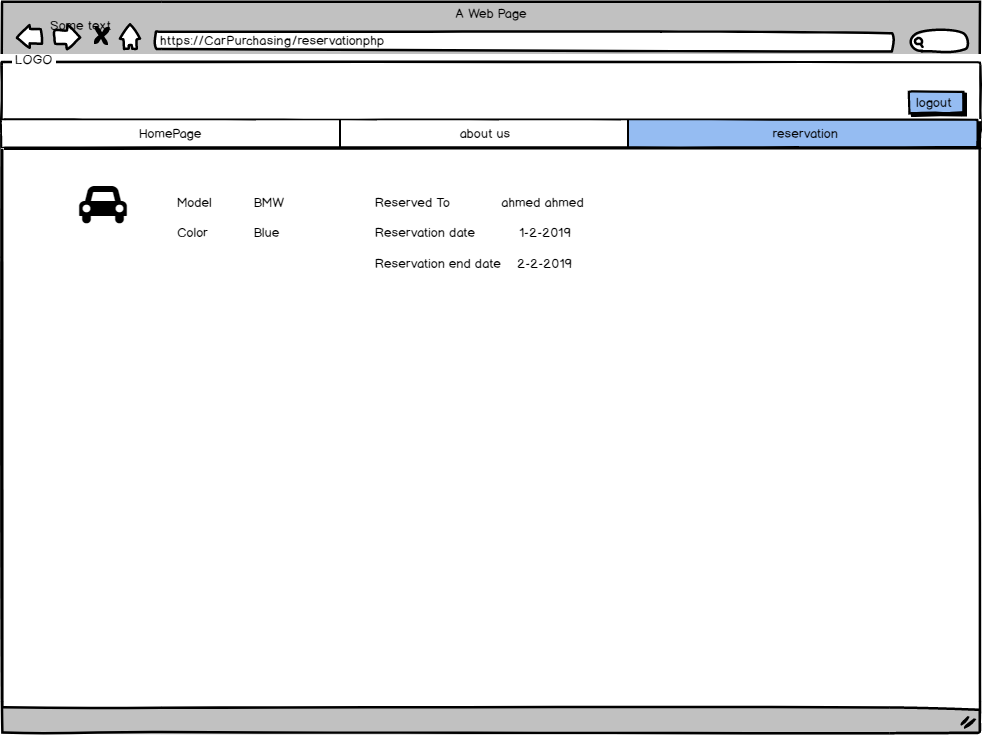
****

**Description:**

The reservation page displays the reserved cars.

**ID:** **Car\_Wireframe\_12**

**Logout**

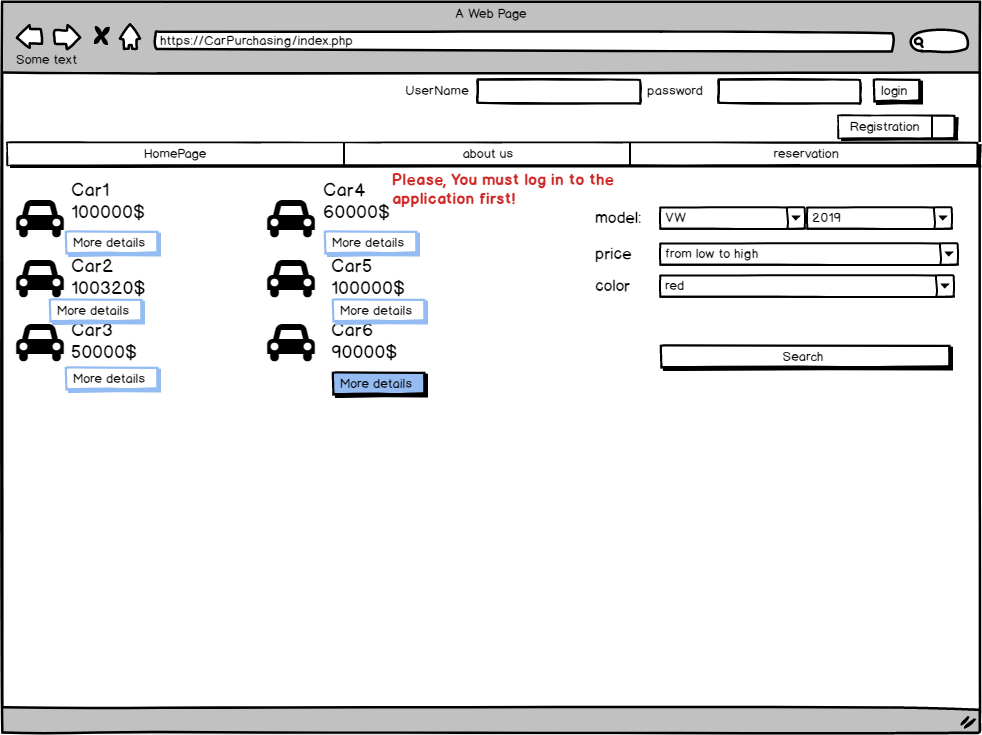
****

**Description:**

The signed user can log out from the application

**ID:** **Car\_Wireframe\_13**

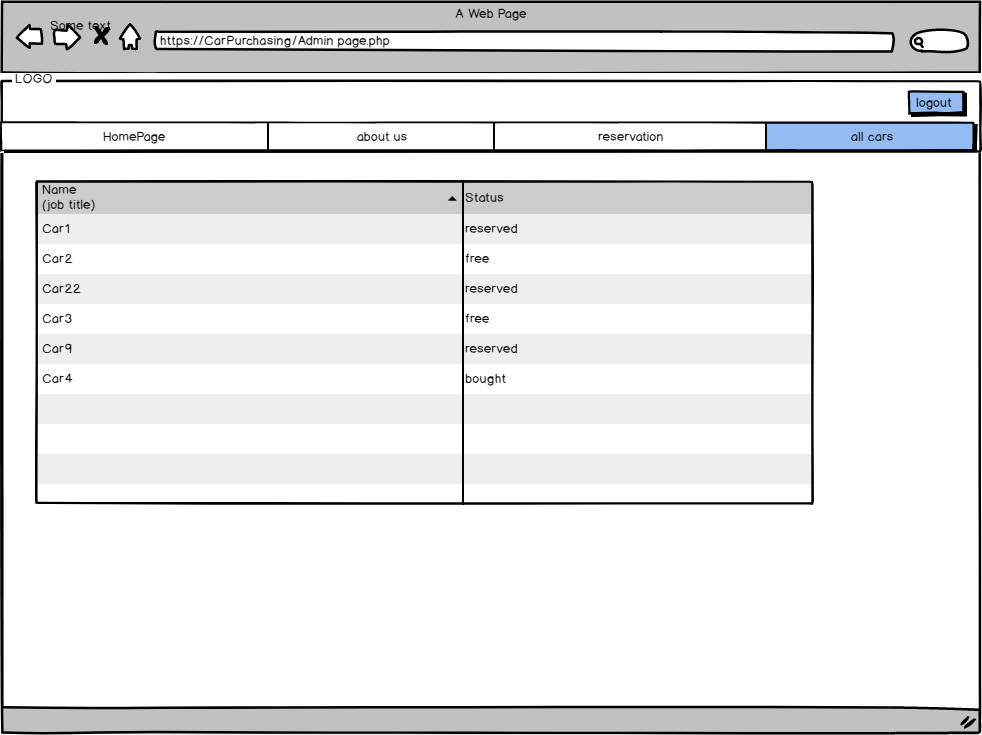
Error Message for unregistered user.



**Description:** unregistered users can’t access the tab of more details.

**ID:** **Car\_Wireframe\_14**

**View all cars**

****

**Description:** Admin can view all cars in the application