**Software Requirements Specification for Car Purchasing Web Application**

Table of Contents

[**1.** **Introduction** 4](#_Toc129953945)

[1.1 Purpose 4](#_Toc129953946)

[1.2 References 4](#_Toc129953947)

[**2.** **Overall Description** 5](#_Toc129953948)

[2.1 User Classes and Characteristics 5](#_Toc129953949)

[2.2 Operating Environment 5](#_Toc129953950)

[2.3 Design and Implementation Constraints 5](#_Toc129953951)

[2.4 Assumptions 5](#_Toc129953952)

[**3.** **System Features** 6](#_Toc129953953)

[3.1 Registration 6](#_Toc129953954)

[**3.1.1 Description** 6](#_Toc129953955)

[**3.1.2 Stimulus/Response Sequences** 7](#_Toc129953956)

[**3.1.3 User Registration Page include:** 7](#_Toc129953957)

[**3.1.4 Functional Requirements** 7](#_Toc129953958)

[**3.1.5 Technical Requirements** 7](#_Toc129953959)

[3.2 Login 9](#_Toc129953960)

[**3.2.1** **Description** 9](#_Toc129953961)

[**3.2.2 Stimulus/Response Sequences** 9](#_Toc129953962)

[**3.2.3 User/Admin login page include:** 9](#_Toc129953963)

[**3.2.4 Functional Requirements** 9](#_Toc129953964)

[**3.2.5 Technical Requirements** 9](#_Toc129953965)

[3.3Logout 10](#_Toc129953966)

[**3.3.1** **Description** 10](#_Toc129953967)

[**3.3.2 Stimulus/Response Sequences** 10](#_Toc129953968)

[**3.3.3 Functional Requirements** 10](#_Toc129953969)

[**3.2.4 Technical Requirements:** None 10](#_Toc129953970)

[3.4 View Car List 11](#_Toc129953971)

[**3.4.1** **Description** 11](#_Toc129953972)

[**3.4.2 Stimulus/Response Sequences** 11](#_Toc129953973)

[**3.4.3 Functional Requirements:** None 11](#_Toc129953974)

[**3.4.4 Technical Requirements:** None 11](#_Toc129953975)

[3.5 Search 11](#_Toc129953976)

[**3.5.1** **Description** 11](#_Toc129953977)

[**3.5.2 Stimulus/Response Sequences** 11](#_Toc129953978)

[**3.5.3 Search feature include:** 11](#_Toc129953979)

[**3.5.4 Functional Requirements** 12](#_Toc129953980)

[3.6 Reserve 12](#_Toc129953981)

[**3.6.1** **Description** 12](#_Toc129953982)

[**3.6.2 Stimulus/Response Sequences** 12](#_Toc129953983)

[**3.6.3 Functional Requirements** 12](#_Toc129953984)

[**3.6.4 Technical Requirements: None** 13](#_Toc129953985)

[3.7 Add car 13](#_Toc129953986)

[**3.7.1** **Description** 13](#_Toc129953987)

[**3.7.2 Stimulus/Response Sequences** 13](#_Toc129953988)

[**3.7.5 Technical Requirements** 14](#_Toc129953989)

[3.9 Delete car 14](#_Toc129953990)

[**3.8.1** **Description** 14](#_Toc129953991)

[**3.8.2 Stimulus/Response Sequences** 14](#_Toc129953992)

[**3.8.3 Functional Requirements** 14](#_Toc129953993)

[**3.8.4 Technical Requirements**: None. 15](#_Toc129953994)

# **Introduction**

## **Purpose**

The purpose of this software requirement specification is to provide a clear, documented model of the requirements for the online car purchasing web application. This system will host a huge database of vehicles, allowing vendors to advertise and market their vehicles and buyers to find the vehicles they're looking for. This document provides high-level description for customers who want to purchase cars. The system provides the customer a way to browse easily through various brands and price ranges. So that customer will easily reserve the car that he wants with an intuitive and easy to use portal.

## **References**

**1.2.1** Customer requirements specification.

**1.2.2** Sample in Question (SIQ).

# **Overall Description**

## **User Classes and Characteristics**

The system includes two types of users:

**Customer:** the client should be able to view and buy from the available cars on the website, The features associated with client will be discussed in detail in section **(3.1).**

**Administrator:** the administrator should be able to:

* Approve customer reservation for the car.
* Decline customer reservation for the car.
* Add car data.
* Delete car data.

## **Operating Environment**

The web application can be run on:

* Windows 10: Google Chrome (v.110.0.5481.104).
* Mozilla Firefox (v.110.0).
* Microsoft Edge (v. 110.0.1587.50).

The Internet is a basic necessity for the system to be accessed.

## **Design and Implementation Constraints**

* Web - based system / PC based.
* Unique user IDs.
* Admin features needed.

## **Assumptions**

Admin Features:

* Add the car data.
* Delete car data.
* Approve reservation.

Drop List Feature:

* Car brand.
* Price range

# **System Features**

The Purchasing car website will have 2 roles.

1. Admin
2. User

Following features will be available to these 2 different roles

| **Admin** | **User** |
| --- | --- |
| Login & Logout | Login & Logout |
| View car list | View car list |
| Search | Search |
| Add car | Register |
| Delete car | Reserve car |

## **3.1 Registration**

### **3.1.1 Description**

User must have an account to be able to view the home page. Users who don’t have an account will not be able to view or reserve any of the website products. Users' accounts will hold information about their first name, last name, email, password, phone number and address. Users can view and login to their accounts anytime in the future after their registration.

### **3.1.2 Stimulus/Response Sequences**

In the login page, users can login if they already have an account and if not, there would be a sign-up button to register his information. Upon matching the required criteria (**3.1.4,3.1.5**), the account will let the user log in to his/her account and the home page is displayed.

# **3.1.3 User Registration Page include:**

* First Name field.
* Last Name field.
* E-mail field.
* Password field.
* Phone
* Address
* Register Button

### **3.1.4 Functional Requirements**

**REQ-1: Register**

● User: Users

● Input: ○ In sign up page

○ User enters first name, last name, email, phone number, address, and password.

○ User enters on the sign-up button.

● Output: ○ Successfully registered, the home page is displayed

● Alternative flow(s):

* + If the user enters an email that already exist, ask user to re-enter a valid one.
  + In case the user enters email that doesn’t match the email format, ask user to re-enter a valid one.
  + In case of repeated phone number, ask user to re-enter another one.
  + If the user clicks on the login button, the system should navigate to the login.

page (**3.2**).

### **3.1.5 Technical Requirements**

**T001** User First Name - field must not be blank.

**T002** User First Name - special characters are not allowed.

**T003** User Last Name - must not be blank.

**T004** User Last Name - special characters are not allowed.

**T005** Email - must not be blank.

**T006** Email - must be in email format (ex: abc@gmail.com).

**T007** Password - password must not be blank.

**T008** Password- must be at-least 8 characters.

**T009** Password - numbers are allowed.

**T010** Password - special characters are allowed.

**T011** Password - password must be encrypted.

**T012** Phone Number –must not be blank.

**T013** Phone Number – special characters are not allowed.

**T014** Phone Number – characters are not allowed.

**T015** Address - Address field must not be blank.

**T016** Address - Numbers are allowed.

**T017** Error-If the data the user entered in the fields doesn't follow the

previously mentioned constraints from (**T001: T016**), an error message should appear.

## **3.2 Login**

### **3.2.1 Description**

Login feature will help in authentication of user and admin accounts. Users who have valid email and password can only login into respective accounts. After that users will be able to navigate to the home page successfully.

### **3.2.2 Stimulus/Response Sequences**

In the login page, users and admins can login, if they already have an account. Upon matching the required criteria mentioned in (**3.2.4**, **3.2.5**), the account will let the user log in to his/her account and the home page is displayed. For the users who don’t have a registered account, there would be a sign-up button to register his information.

### **3.2.3 User/Admin login page include:**

* E-mail field.
* Password field.
* Login button.
* Sign up button.

### **3.2.4 Functional Requirements**

**REQ-2: Login**

● User: User and Admin

● Input: ○ In login page

○ User- enters registered email and password

○ Admin- enters registered email and password

● Output: ○Successfully logged in, the home page is displayed

● Alternative flow(s):

○ In case the user enters email that doesn’t match the email format, ask the

user to re-enter a valid credential.

○ In case of a mismatch between user id and password, ask the

user to re-enter a valid credential.

### **3.2.5 Technical Requirements**

**T018** Email - must not be blank.

**T019** Email - must be in email format (ex: abc@gmail.com).

**T020** Email – first character can’t have space.

**T021** Password - password must not be blank.

**T022** Password- must be at-least 8 characters.

**T023** Password - numbers are allowed.

**T024** Password - special characters are allowed.

**T025** Password - password must be encrypted.

**T026** Error-If the data the user entered in the fields doesn't follow the

previously mentioned constraints from (**T018: T025**), an error message should appear.

## **3.3Logout**

### **3.3.1 Description**

All users and admins will be able to logout from their accounts from the home page.

### **3.3.2 Stimulus/Response Sequences**

In the home page, users and admins can logout, then the system will navigate them to the login page again**.**

### **3.3.3 Functional Requirements**

**REQ-3: Logout**

**●** User: Users and admins

● Input: ○ Click ‘logout’ button in home page

● Output: ○ User and admins are successfully logged out, the login page will be displayed

● Alternative flow(s): None

### **3.2.4 Technical Requirements:** None

## **3.4 View Car List**

### **3.4.1 Description**

In the home page, the users can navigate from an enormous data of available cars and below each car he will be able to know the cars’ (price, brand, model, the reaming number of each car and description).

### **3.4.2 Stimulus/Response Sequences**

When the user, he/she should be able to reach the home page and view all the list of the available cars including the details of each one mentioned in the description (**3.4.1**)**.**

### **3.4.3 Functional Requirements:** None

### **3.4.4 Technical Requirements:** None

## **3.5** **Search**

### **3.5.1 Description**

Customers can search for an item from a large catalog of available cars in the car purchasing system. Users can search through a dropdown list.

### **3.5.2 Stimulus/Response Sequences**

In the home page, users and admins can search by two dropdown lists to choose from the available cars by their brand or price.

### **3.5.3 Search feature include:**

* Drop down list for brands.
* Drop down list for prices.
* Search button.

### **3.5.4 Functional Requirements**

**REQ-4: Search item**

● User: Users and admins

● Input: ○ In the home page

○ click on the price dropdown list.

○ users can choose the wanted price.

○ or click on the brand dropdown list.

○ users can choose a specific brand.

○ press search button.

● Output: ○List of cars according to the chosen brand or range of price.

● Alternative flow(s):

○ If the cars from a specific brand or range of price are not available,

a message should appear please try another brand or range of price.

**3.5.5 Technical Requirements:**  None

## **3.6 Reserve**

### **3.6.1 Description**

Once the customer proceeds to reserve any car from the home page, the user will find a reservation button below each car.

### **3.6.2 Stimulus/Response Sequences**

In the home page, the users can navigate from enormous data of available cars. Once the user chooses any of the available cars, then he/she could press on the reserve button which is found below each car and then a message will appear that reservation successfully placed.

### **3.6.3 Functional Requirements**

**REQ-5: Buy**

● User: Users

● Input: ○In home page

○ Click on reserve button

● Output: ○ Reservation successfully placed.

● Alternative flow(s):

○In case the car that the user has choose is already reserved then a message

appearthat “this car is already reserved, please choose another one”**.**

# **3.6.4 Technical Requirements: None**

## **3.7 Add Car**

### **3.7.1 Description**

Admins are authorized to add new cars to the system by simply filling out the "Add car" form which includes a photo, description, price. Brand, and model for the car.

### **3.7.2 Stimulus/Response Sequences**

In the admin home page, the admin can click on the “Add car” button, a form is displayed on a new page. The admin shall enter the details for each car from (photo, description, price. Brand, and model) and after filling the form the admin will press on the “save” button to add the car to the system.

**3.7.3 Admin car page include:**

* Upload photo button.
* Price field.
* Brand field.
* Model field.
* Description field.
* Save button.
* Cancel button.

**3.7.4 Functional Requirements**

**REQ-6: Add**

● User: Admins

● Input: ○ In home page

○ Click on add button

○ Fill the add car form (photo, price, brand, model, description)

○ Click on save button

● Output: ○ Car added successfully.

● Alternative flow(s):

○In case the admin presses the cancel button instead of save button, the home

page displays**.**

# **3.7.5 Technical Requirements**

**T027** brand- must not be blank.

**T028** model- must not be blank.

**T029** Price– Special characters are not allowed.

**T030** Price- must not be blank.

**T031** Description- must not be blank.

**T032** Error-If the data the admin entered in the fields doesn't follow the

previously mentioned constraints from (**T026: T031**), an error message should appear.

## **3.9 Delete Car**

### **3.8.1 Description**

Admins are authorized to delete any car from the system by simply deleting the car from the “Delete cars” page.

### **3.8.2 Stimulus/Response Sequences**

From the admin home page, admin can click on the “delete car” button which will navigate the admin to delete cars page which includes a delete button below each car post.

### **3.8.3 Functional Requirements**

**REQ-7: Delete**

● User: Admins

● Input: ○ In home page

○ Click on “Delete car” Button, the delete car page ‘ll display.

○ Click on delete button

● Output: ○ Car deleted successfully.

● Alternative flow(s): None.

# **3.8.4 Technical Requirements**: None.