

---

# Software Requirements Specification

**For: Car Bookings**

**Version** 1.6

**Prepared by:**

- Aya Tarek

**Reviewed by:**

- Eslam Fawzy
- Aml Nasser
- Basma Galal

**Date:** 17-5-2022

# Table of Contents

<b>Table of Contents</b>	<b>ii</b>
<b>Revision History</b>	<b>ii</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Purpose	1
1.2 Intended Audience and Reading Suggestions	1
1.3 Project Scope	1
1.4 References	1
<b>2. Overall Description</b>	
2.1 Product Features	
2.2 Design and Implementation Constraints	2
2.3 System components	2
2.4 Assumptions	2
2.5 Use Case	2
<b>3. System Features</b>	<b>3</b>
Customer Requirements	
User Functional Requirements	
3.1 Registration	
3.2 Login	
3.3 View	
3.4 Reservation	
Admin Functional Requirements	
3.5 Login	
3.6 Add car	
3.7 Update car	
3.8 Delete car	
<b>Appendix B: Issues List</b>	<b>10</b>

## Revision History

Name	Date	Main change	Version
Basma Galal	12-4-2022	Added more features and assumptions	1.1
Aya Tarek	29-4-2022	Requirements changes from customer	1.2
Aya Tarek	5-5-2022	Applying customer comments	1.3
Aya Tarek	6-5-2022	Changing requirements IDs	1.4
Aya Tarek	11-5-2022	Change in Design	1.5
Aya Tarek	17-5-2022	Remove of three features	1.6

# **1. Introduction**

## **1.1 Purpose**

This documentation is written to describe the web application (Car Reservation System) and what is the foundation of this project. It gives an overall view of how the system is supposed to work and what is to be expected by the user.

## **1.2 Intended Audience and Reading Suggestions**

- 1.2.1 Customer.
- 1.2.2 Developers.
- 1.2.3 Testers.
- 1.2.4 Project manager.

## **1.3 Project Scope**

This platform will host an enormous database of cars enabling the sellers to advertise and market their cars and to help the buyer to search for their desired cars. It is a web application that will be used by users and admins. The admin will be able to add, delete and update cars and also add users and other admins. The user will be able to reserve the desired car and search by color, brand and price.

## **1.4 References**

- 1.4.1 Customer requirements specification.
- 1.4.2 Sample in Question (SIQ).

# **2. Overall Description**

## **2.1 Product features**

The system allows users to register, login, search for a car by specific criteria and reserve the desired car and allows admins to add, delete, update cars and also add users or other admins.

## **2.2 Design and Implementation Constraints**

- 2.2.1 Web-based / PC based.

2.2.2 Unique username for each user or admin.

2.2.3 Adding additional features for the admin.

## **2.3 System components**

2.3.1 Registration page

2.3.2 Login page

2.3.3 Available cars list page for users.

2.3.4 Available cars list page for Admins.

2.3.5 Add new car page.

2.3.6 Update car page.

## **2.4 Assumptions**

[Project Management Plan](#)

## **2.5 Use Case**

[Use case](#)

# **3. System Features**

<b>Priority</b>	High
	Medium
	Low

## Customer requirements

- **CR-001**

Priority: high

**Registration:** The user can register.

- **CR-002**

Priority: high

**User login:** The user can login with a unique user ID.

- **CR-003**

Priority: high

**Reservation:** The user has the option to reserve a car from the available cars.

- **CR-004**

Priority: high

**Admin login:** The Admin can login with a unique user ID.

- **CR-005**

Priority: medium

**Add new car:** Admin can add new car.

- **CR-006**

Priority: medium

**Update car:** Add can update car details.

- **CR-007**

Priority: medium

**Delete car:** Admin can delete a car from the system.

## **User Functional Requirements**

### **3.1 CR-001: Registration**

#### 3.1.1 Description

The user will register using email, password and mobile phone.

#### 3.1.2 Stimulus/Response Sequences

After user Registration an e-mail containing a unique username will be sent to the user to confirm registration.

#### 3.1.3 Functional Requirements

Req-Regs-001: The user shall enter only valid format email that does not already exist in the database ex: [email@gmail.com](mailto:email@gmail.com).

Req-Regs-002: The user shall enter password. The password field can't be blank.

Req-Regs-003: The user shall confirm password. The password should match the previously entered password in Req-Regs-002.

Req-Regs-004: The user shall enter a valid mobile number containing eleven digits.

Req-Regs-005: The user shall be able to press on the register button.

Req-Regs-006: If any of the entered data in requirements (Req-Regs-001, Req-Regs-002, Req-Regs-003, Req-Regs-004) is incorrect a error message shall appear and fields with wrong data appear surrounded by red.

Req-Regs-007: If all entered data in requirements (Req-Regs-001, Req-Regs-002, Req-Regs-003, Req-Regs-004) is correct the system will confirm registration and send a unique username through pre entered e-mail.

Req-Regs-008: When there is no entered data in the email, password, confirm password or phone number fields there should be default texts in each field.

Req-Regs-009: The data fields default texts mentioned in (Req-Regs-008) should be "Enter your email", "phone number", "password", "confirm password" in

the fields username, phone number, password and confirm password respectively.

## **3.2 CR-002: User login**

### **3.2.1 Description**

The user will login using username and password.

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

When logging in, the user will land on a car listing page with number of currently available cars.

### **3.2.3 Functional Requirements**

Req-User-Log-001: The user shall enter the username that was sent on email on registration to login.

Req-User-Log -002: The user shall enter the password that corresponds to the entered username to login.

Req-User-Log -003: The system shall redirect the user to the car listing page which contains all the currently available cars.

Req-User-Log -004: Each car in the listing page appears by image, price, brand, color, car ID and a button to reserve the chosen car.

Req-User-Log -005: In case user username or password were entered incorrectly, The system shall appear an error message.

Req-User-Log -006: When there is no entered data in the username or password there should be default texts in each field which are "Username" in username field and "Password" in password field .

## **3.3 CR-003: Reservation**

### **3.2.1 Description**

After the user chooses for the available cars in the list the system enables the user to reserve the car.

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

After the user clicks the reservation button the system will show a message to confirm the reservation.

### 3.2.3 Functional Requirements

Req-Res-001: The user shall choose a car in the listing page that contains all the currently available cars and press reserve.

Req-Res-002: The system will show a message “car is reserved successfully” in new page.

Req-Res-003: The user shall be able to go back to home page when pressing back on back button exists in the browser.

## **Admin Functional Requirements**

### **3.4 CR-004: Admin login**

#### 3.4.1 Description

The Admin will login using a predefined username and password.

#### 3.4.2 Stimulus/Response Sequences

When logging in, the admin will land on the car listing page with number of currently available cars and two buttons in the header to add car or update car.

#### 3.5.3 Functional Requirements

Req-Admin-log-001: The admin shall enter the predefined username that was sent through email.

Req-Admin-log-002: The admin shall enter the predefined password that corresponds to the entered username.

Req-Admin-log-003: The system shall redirect the admin to the car listing page containing all available cars.

Req-Admin-log-004: Car ID, color, brand and price shall appear under each car and also option for delete car shall appear as button under each car in the car listing page.

Req-Admin-log-005: Options for add and update cars shall appear in the car listing page header.

Req-Admin-log-006: In case admin username or password were entered incorrectly, The system shall appear an error message.



### **3.5 CR-005: Add new car**

#### **3.5.1 Description**

The admin should be able to add new cars into our system.

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

When the admin presses add new car button the system redirects him to the adding car page.

#### **3.5.3 Functional Requirements**

Req-Add-Car-001: The admin shall be able to press on add new car that exists in the car listing page that the system redirects the admin to after login.

Req-Add-Car-002: The system shall redirect him to the adding new car page.

Req-Add-Car-003: The admin shall choose brand from brands dropdown list.  
(audi is default brand)

Req-Add-Car-004: The admin shall choose color from the car color dropdown list.  
(black is default color)

Req-Add-Car-005: The admin shall enter a valid price which contains Positive digits only in the car price field.

Req-Add-Car-006: The admin shall be able to upload an image for the car by pressing the upload image button.

Req-Add-Car-007: The admin shall be able to press add button and pop message “The car is added successfully” appears.

Req-Add-Car-008: The admin shall be able to press the home button that exists in the add new car page to redirect him to the home page.

### **3.6 CR-006: Update car**

#### **3.6.1 Description**

The admin should be able to update any car’s information.

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

When the admin presses the update car button the system redirects him to the update car page.

#### **3.6.3 Functional Requirements**

Req-Update-Car-001: The admin shall be able to press on the update car button that exists in the car listing page header that the system redirects the admin to after login.

Req-Update-Car-002: The system shall redirect the admin to the updating car page.

Req-Update-Car-003: The admin shall be able to choose the car ID from a dropdown list to identify the car that will be changed.

Req-Update-Car-004: The admin shall be able to change the brand from the dropdown list. (audi is default brand)

Req-Update-Car-005: The admin shall be able to change color from the dropdown list. (black is default color)

Req-Update-Car-006: The admin shall be able to change price in the car price field and the new price entered should contain Positive digits only.

Req-Update-Car-007: The admin shall be able to replace the uploaded image for the car by pressing the upload image button.

Req-Update-Car-008: The admin shall be able to press the update button and a pop up message “The car is updated successfully” appears.

Req-Add-Car-009: The admin shall be able to press the home button that exists in the update car page to redirect him to the home page.

### **3.7 CR-007: Delete car**

#### **3.7.1 Description**

The admin should be able to delete any car.

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

When the admin presses the delete car button the system shall remove the car from the database.

#### **3.7.3 Functional Requirements**

Req-Del-Car-001: The admin shall be able to press on the delete car button that exists in the car listing page under each car that the system redirects the admin to after login.

Req-Del-Car-002: The system shall delete car from the database.

Req-Del-Car-003: The system shall remove the car from the currently available cars listing page.

Req-Del-Car-004: The system shall view a message “The car is deleted successfully” in new page.

Req-Del-Car-005: The admin shall be able to go back to home page when pressing back on back button exists in the browser.

## **Appendix B: Issues List**

No issues