**On Demand Car Wash System:-**

*Low Level Design (LLD)*

**Table of Contents**

***1.0*** ***Document Purpose***

***2.0*** ***Intended Audience***

***3.0*** ***Project Background, Objective(s)***

***4.0*** ***Design Pattern***

***5.0*** ***Solution Steps***

***6.0*** ***Classes/function name***

***7.0*** ***Solution Diagram***

***8.0*** ***Architecture Diagram***

***9.0*** ***Microservices Architecture Diagram***

***10.0 Er Diagram***

***11.0 Flow Diagram***

***12.0 Use Case Diagram***

***12.0 Class Diagram***

***14.0 Data Model/Tables***

***15.0 Unit Testing***

2

## 1.0 Document Purpose

The documents contain a detailed description of the solution architecture of the on-demand Car Wash System.

## 2.0 Intended Audience

|  |  |
| --- | --- |
| **Role** **Nature of Engagement in the On Demand Car Wash System Architecture** | |
| **Product**  **Owner/SME** | Key stakeholder to ensure that the architecture is aligned  with business goals. |
| **Business**  **Analysts** | Business analysts are one of the stakeholders who are  informed with the key architectural decisions. |
| **Enterprise Architects** | To enforce Customer management Platform Architecture is aligned to business goals and architecture, architectural  guidelines. |
| **Developers** | Use Technical Architecture Document as the guiding document for detail design and implantation approach to  align with Customer management Microservice |
| **End-User** | An End- user can check the bookings done, cost for the booking and other package information and book/cancel a  Wash. |

* 1. **Project Background & Objectives**

### Project Background

On Demand Car Wash System leads to perform Management of Car wash details where one can register themselves and perform various operations related to washing Cars.

### Project Objectives

On Demand Car Wash System will perform various operations like instantly booking the car wash service or scheduling it for later.

The user of this system should first register with his/her email ID and their password for any interaction with the system. Once registered and after logging in the user should select the kind of activity, he would like to perform.

3

### Technologies Used

* + - Frontend:- Angular
    - Backend:- Web-API (ASP.NET CORE WEB API) ASP.NET MVC and SWAGGER
    - Database:- SQL SERVER MANAGEMENT STUDIO(SSMS)

## 4.0 Design Pattern

|  |  |  |
| --- | --- | --- |
| **Serial no.** **Name** **Description** | | |
| 1 | Angular | Creating a user interface  (Front-end), and consuming API services. |
| 2 | Database | For storing, maintaining and accessing customer, admin, car washer and booking details. |
| 3 | API | Using HTTP requests, we will use the respective action to trigger various  operations |

* 1. **Solution Steps**

### Customer

#### Registering Customer

1. User will be able to register himself by entering the details like Name, address, Phone, email, and Password.

4

1. After filling the user credentials the form is validated.
2. If the validation is successful, by clicking the submit button browser directs the request to customer registration API.
3. The call reaches the API gateway.
4. API gateway does the routing and saves the data in the database.
5. Once a user is successfully registered an alert is displayed and the user is redirected to the login page.

#### Viewing Wash package and Payment Details

1. After login, users can book their first car wash within few taps.
2. They can even add a new car and include some extra wash add-ons if they wish to.
3. users can schedule car wash for upcoming dates by selecting their preferred date, time, location, and package.
4. A user can confirm the booking and pay washers through their debit/credit cards.
5. User gets a payment receipt from washer after the successful car wash.
6. After the successful car wash, users can share their thoughts and opinions for washers through reviews and ratings.
7. Users can view and edit their profile information and can view order details like Current orders and Past orders.

### Car Washer

* + 1. Washer can login using – Email
    2. Wash Request is sent to the washer along with the user details. The washer can either accept or decline the request.
    3. On accepting the request, a washer can navigate through customer's address by Google Maps Integration.
    4. Washers can view and update their profile information like profile picture and contact information.
    5. Washers will be notified in the below scenario: -

5

* + - 1. Scheduled wash notification before 2 hours.
      2. New wash order.
      3. When users cancel wash request.
      4. On successful Payment by users.
    1. Washer and can view order details like Current orders and Past orders.

### Admin

* + 1. Admin will be able to edit and add new car washer, add news car details, and payment details.
    2. Admin has access to all users and car washer’s details.
    3. Admin can Add/Edit service plan details and Active/Inactive it.
    4. Admin can Add/Edit Add-On list, Promo codes and can inactive also.
    5. Admin can view pending, accepted, under process, completed and cancelled orders.
    6. GetAllCars() will let the admin view all the car details.
    7. GetCustomerById() will allow Admin to view customer details by ID.

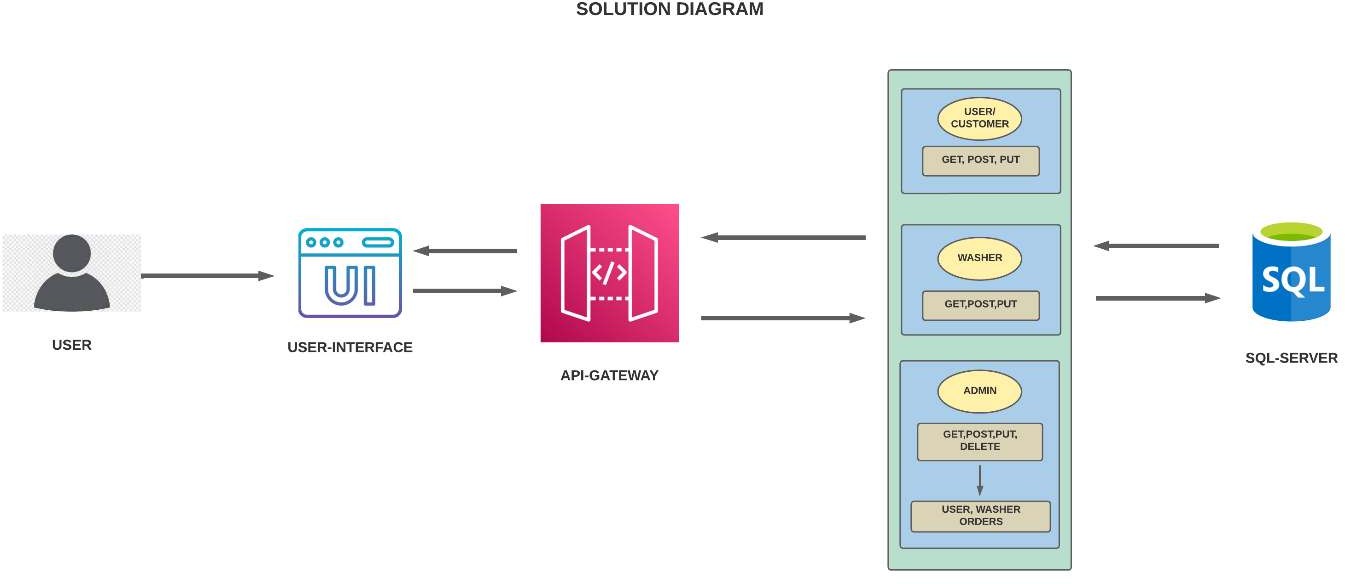
## 6.0 Classes/Functions

|  |  |  |
| --- | --- | --- |
| **Serial** **Class** **Description**  **no.** | | |
| **1** | Model Class | Model for holding the booking schema details for user. |
| **2** | Repository | The Interface in Data Access Layer for the user. |
| **3** | Controller | Controller handles the incoming HTTP requests and send the response back to the caller. |

6

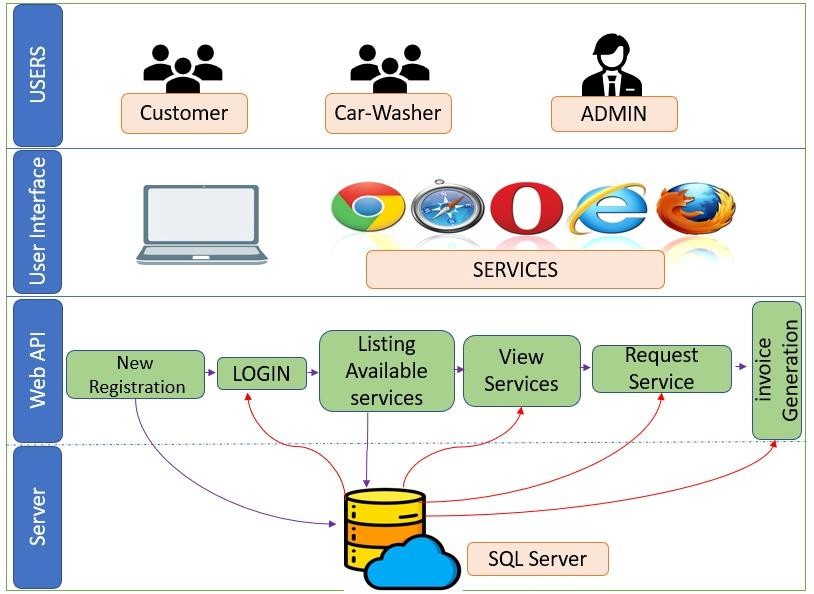
|  |  |  |
| --- | --- | --- |
| **4** | Services | It’s the Business Access Layer holding the Business Logic and meditates the communication between the controller and repository (Data Access) Layer. |
| **5** | Exception Handlers | Exception Handlers handles all the exceptions that which are revealed during runtime. |

## 7.0 Solution Diagram



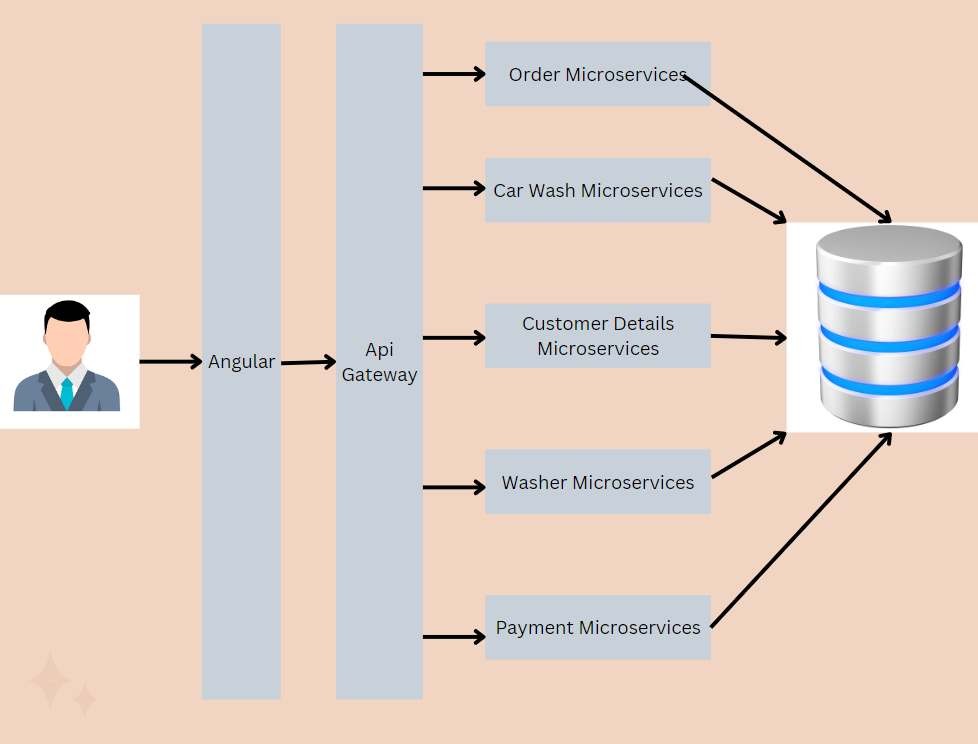
7

## 8.0 Architecture Diagram

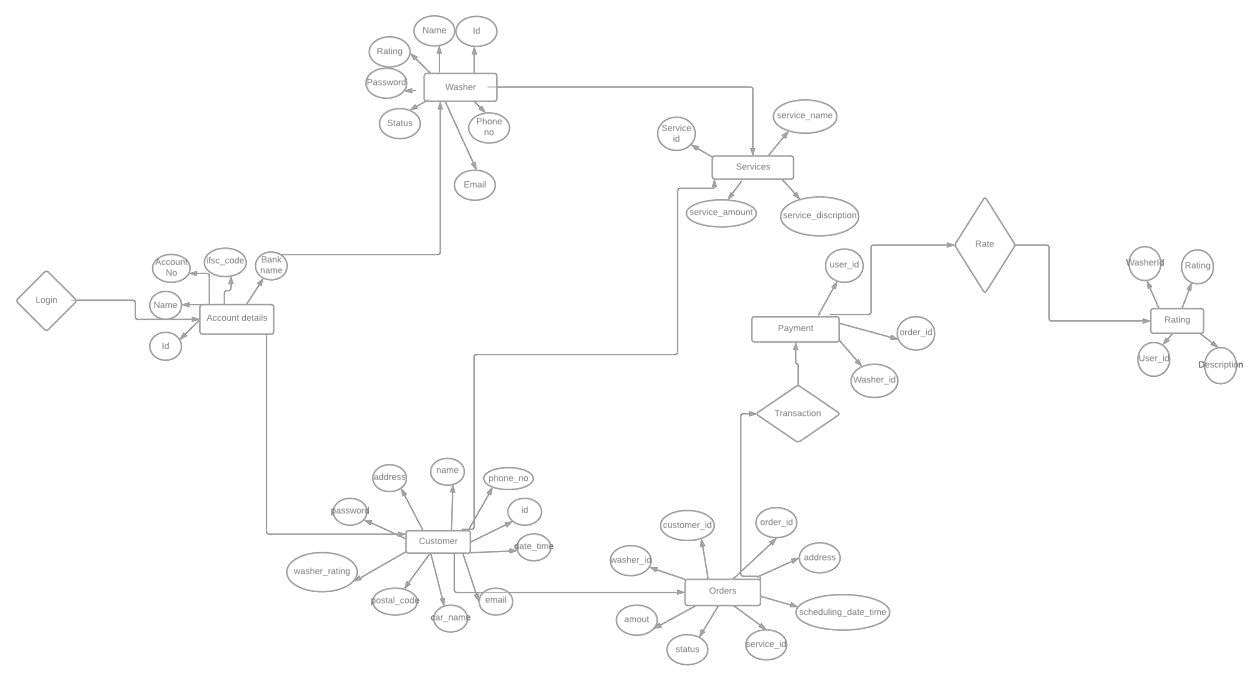


8

## 9.0 Microservices Architecture

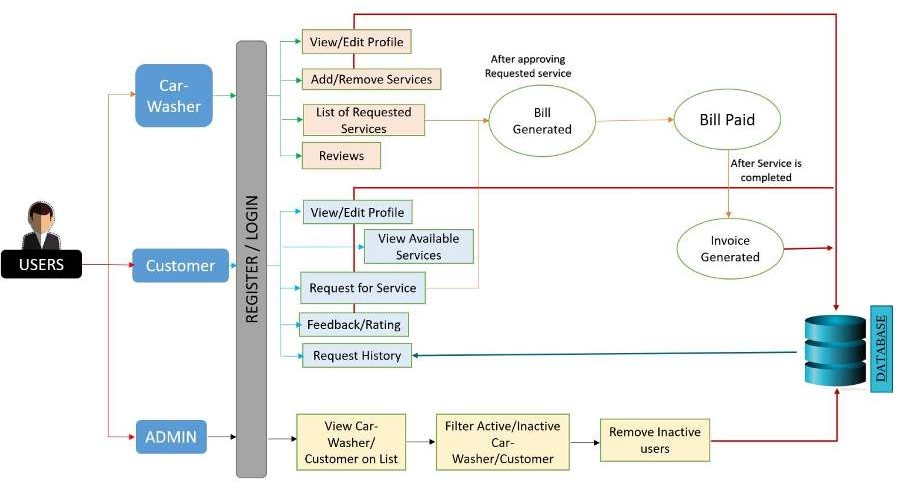


**10.0 Er Diagram**

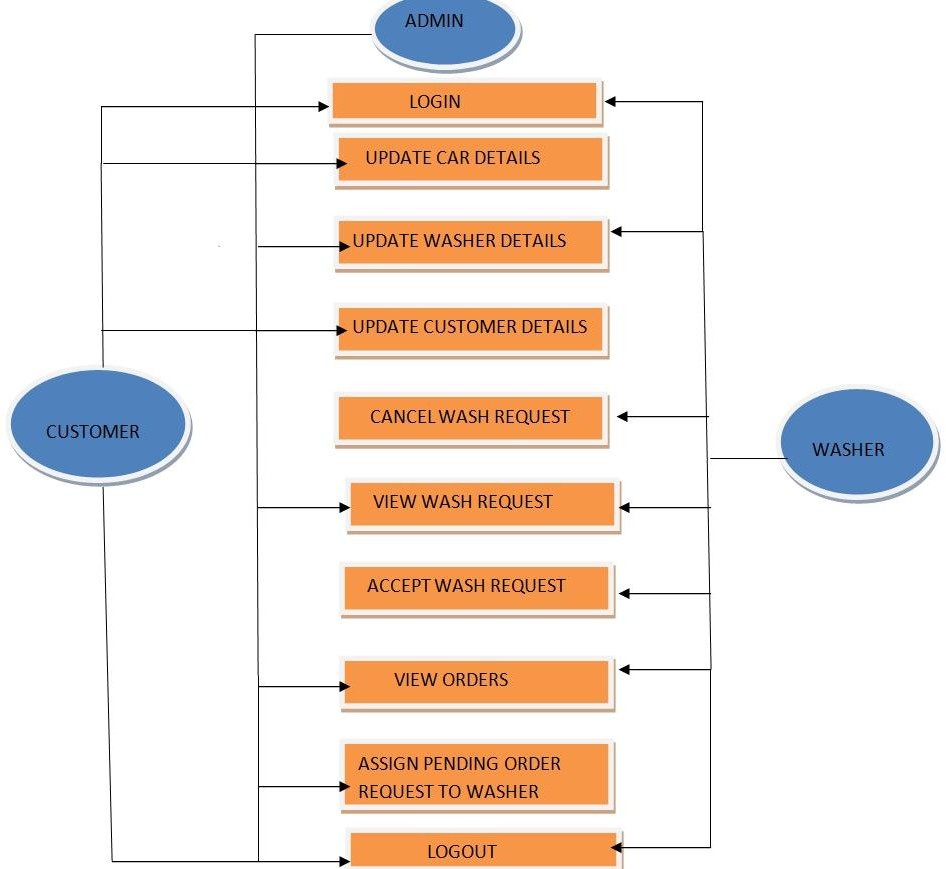


9

## 11.0 Data Flow Diagram

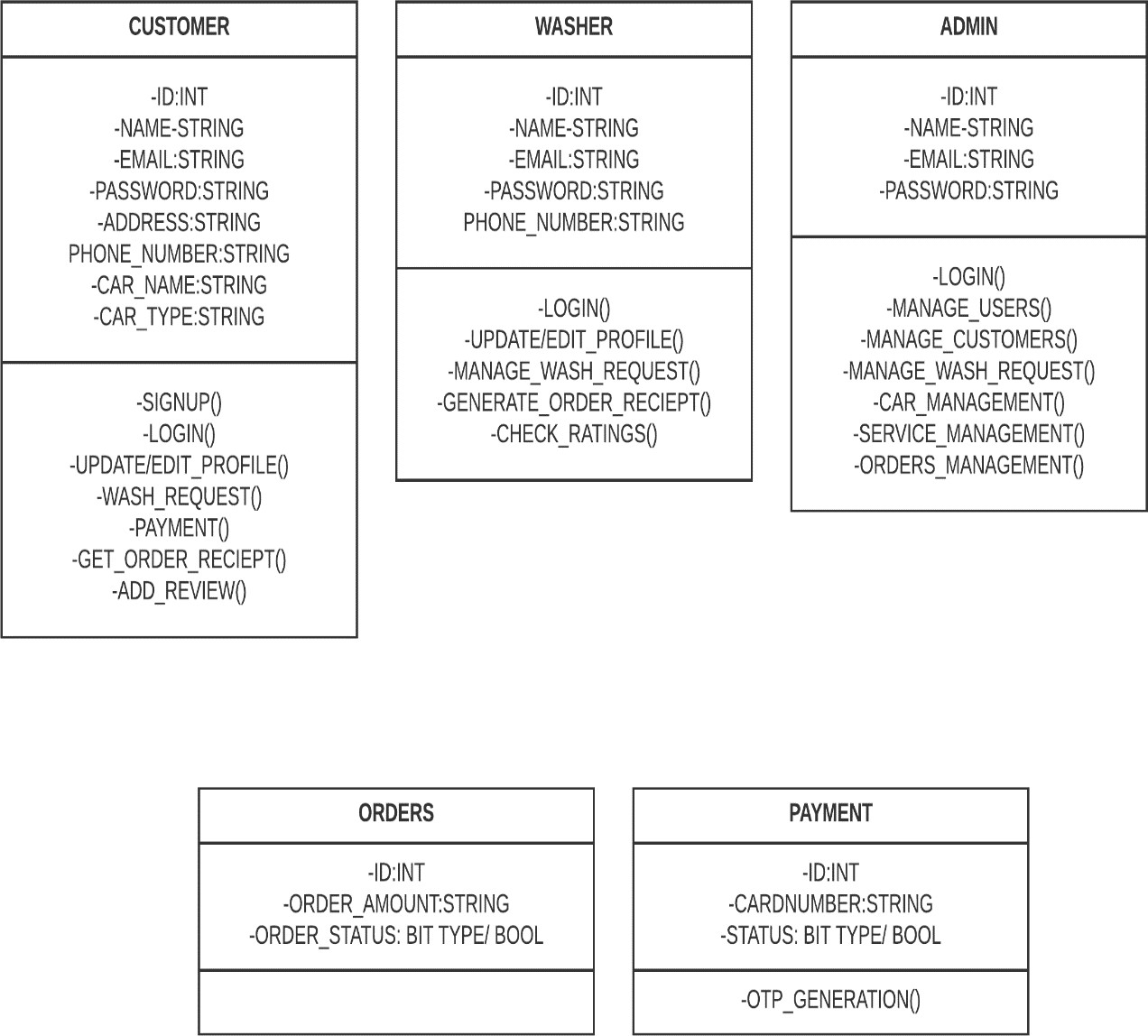


**12.0 Use Case Diagram**



10

## 13.0 Class Diagram



11

## 14.0 Database Diagram

## 

12

**15.0 Unit Test Cases**

# For Customer Registration

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test CASE ID | TEST CASE SCENARIO | TEST CASE | PRE- CONDITION | TEST STEPS | TEST DATA | EXPECTED RESULT | ACTUAL RESULTS |
| 1 | customer registration | enter all the required fields to get  registered | that id needs to be present in database | 1. Enter name=john email=[abc@gmail.com](mailto:abc@gmail.com) phone=1234567891 address=delhi 2. Enter Submit | Valid Details | Customer Registration Successfully | customer Registration Successfully |
| 2 | customer registration | enter all the required fields to get  registered | That id needs to be present in database | 2) Enter name= john email=[abcmail.com](mailto:abc@gmail.com) phone=1234567891 address= delhi  2) Enter Submit | InValid email | In email there should be @ | In email there should be @ |
| 3 | customer registration | enter all the required fields to get registered | customer must enter all the required field as per specified in schema | 3) Enter name= john email=[abc@gmail.com](mailto:abc@gmail.com) phone=1234567891 address=delhi  2) Enter Submit | <All the Require d fields are available> | successful registration | successful registration |

13

# For Customer Listing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test CASE ID | TEST CASE SCENARIO | TEST CASE | PRE- CONDITION | TEST STEPS | TEST DATA | EXPECTED RESULT | ACTUAL RESULTS |
| 1 | Customer List | Enter the valid Customer Id of customer in parameter to get details | That id needs to be present in database | 1)  Enters Valid Id  2)  Enter Submit | Valid Id | Customer Details | Customer Details |
| 2 | Customer List | enters the wrong Customer Id which is not there in database to get customer details | That id needs to be present in database | 1)  Enters invalid Id  2)  Enter Submit | Invalid Id | Customer Details with this id is not present | Customer Details with this id is not present |

14

# For Customer Updation

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test CASE ID | TEST CASE SCENARIO | TEST CASE | PRE- CONDITION | TEST STEPS | TEST DATA | EXPECTED RESULT | ACTUAL RESULTS |
| 1 | Customer Updation | enter the valid id of customer in parameter to get details | that id needs to be present in database | 1)  Enters Valid Id  2)  Enter Submit | Valid Id | customer Update Successfully | customer Update Successfully |
| 2 | Customer Updation | Enter the wrong id which is not there in database to get customer details | That id needs to be present in database | 1)  Enters Valid Id  2)  Enter Submit | Valid Id | customer Details with this id you want to update is not present | customer Details with this id you want to update is not present |

15

# For Customer Deletion

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test CASE ID | TEST CASE SCENARIO | TEST CASE | PRE- CONDITION | TEST STEPS | TEST DATA | EXPECTED RESULT | ACTUAL RESULTS |
| 1 | Customer Deletion | enter the valid id of customer in parameter to get details | That id needs to be present in database | 1)  Enters Valid Id  2)  Enter Submit | Valid Id | Customer Deleted successfully | Customer Deleted successfully |
| 2 | Customer Deletion | enter the valid id of customer in parameter to get details | That id needs to be present in database | 1)  Enters Valid Id  2)  Enter Submit | Valid Id | Customer Details with this id you want to delete is not present | Customer Details with this id you want to delete is not present |

16

# For Washer Registration

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test CASE ID | TEST CASE SCENARIO | TEST CASE | PRE- CONDITION | TEST STEPS | TEST DATA | EXPECTED RESULT | ACTUAL RESULTS |
| 1 | Washer registration | enter all the required fields to  get registered | that id needs to be present in database | 1. Enter name=jack email=[abc@gmail.com](mailto:abc@gmail.com) phone=1234567891 address=delhi 2. Enter Submit | Valid Details | Washer Registration Successfully | Washer Registration Successfully |
| 2 | Washer | enter all | That id needs | 2) Enter name= jack email=[abcmail.com](mailto:abc@gmail.com) phone=1234567891 address= delhi  2) Enter Submit |  | In email | In email |
|  | registration | the | to be present |  | there should | there |
|  |  | required  fields to | in database | InValid  email | be @ | should be  @ |
|  |  | get |  |  |  |  |
|  |  | registered |  |  |  |  |
| 3 | Washer registration | enter all the required fields to get registered | Washer must enter all the required field as per specified in schema | 3) Enter name= jack email=[abc@gmail.com](mailto:abc@gmail.com) phone=1234567891 address=delhi  2) Enter Submit | <All the Require d fields are available> | successful registration | successful registration |

17

# For Washer Listing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test CASE ID | TEST CASE SCENARIO | TEST CASE | PRE- CONDITION | TEST STEPS | TEST DATA | EXPECTED RESULT | ACTUAL RESULTS |
| 1 | Washer List | Enter the valid Washer Id of Washer in parameter to get details | That id needs to be present in database | 1)  Enters Valid Id  2)  Enter Submit | Valid Id | Washer  Details | Washer  Details |
| 2 | Washer List | enters the wrong Washer Id which is not there in database to get Washer details | That id needs to be present in database | 1)  Enters invalid Id  2)  Enter Submit | Invalid Id | Washer Details with this id is not present | Washer Details with this id is not present |

18

# For Washer Updation

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test CASE ID | TEST CASE SCENARIO | TEST CASE | PRE- CONDITION | TEST STEPS | TEST DATA | EXPECTED RESULT | ACTUAL RESULTS |
| 1 | Washer Updation | enter the valid id of Washer in parameter to get details | that id needs to be present in database | 1)  Enters Valid Id  2)  Enter Submit | Valid Id | Washer Update Successfully | Washer Update Successfully |
| 2 | Customer Updation | Enter the wrong id which is not there in database to get Washer details | That id needs to be present in database | 1)  Enters Valid Id  2)  Enter Submit | Valid Id | Washer Details with this id you want to update is not present | Washer Details with this id you want to update is not present |

19

# For Washer Deletion

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test CASE ID | TEST CASE SCENARIO | TEST CASE | PRE- CONDITION | TEST STEPS | TEST DATA | EXPECTED RESULT | ACTUAL RESULTS |
| 1 | Washer Deletion | enter the valid id of customer in parameter to get details | That id needs to be present in database | 1)  Enters Valid Id  2)  Enter Submit | Valid Id | Washer Deleted successfully | Washer Deleted successfully |
| 2 | Washer Deletion | enter the valid id of Washer in parameter to get details | That id needs to be present in database | 1)  Enters Valid Id  2)  Enter Submit | Valid Id | Washer Details with this id you want to delete is not present | Washer Details with this id you want to delete is not present |

20