**On Demand Car Wash**

**POC**  
**Low Level Design(LLD)**

Date: 20/05/2022

Current Document Version: [*1.0*]

DOCUMENT APPROVAL

**Approvers of this document**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Department** | **Role** | **Signature** | **Date** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Document Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Version #** | **Author** | **Date** | **Description** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1.0 Document Purpose 4](#_Toc94636300)

***1.1* *Scope*…………………………………………………………………………………………………………….…4**

[2.0 Intended Audience 4](#_Toc94636301)

[3.0 Project Background, Objective(s) 4](#_Toc94636302)

[3.1 Project Background 4](#_Toc94636303)

[3.2 Project Objective 4](#_Toc94636304)

[4.0 Design Pattern 4](#_Toc94636305)

[5.0 Solution Diagram](#_Toc94636306)

[6.0 Solution Steps 4](#_Toc94636307)

[7.0 Classes/function name 8](#_Toc94636308)

[8.0 Validations 8](#_Toc94636309)

[9.0 Data model/Tables 9](#_Toc94636310)

[10.0 HTTP Status Code 9](#_Toc94636314)

[11.0 Unit Testing 9](#_Toc94636314)

# Document Purpose

This document describes the solution architecture for On Demand Car Wash.

**1.1 Scope**

On Demand Car Wash Servicing System is a Web based System where user can easily find out with nearby car wash servicing garage in Online. This project will be beneficial for those people who don’t want to go to the garage. It is an easy and time saving System. This online system provides home delivery of car wash service for that users have to needs to register in this System. Then user needs to Login. By Login user decide which service he/she has needs.

# Intended Audience

This document is intended as a reference for the following roles and stakeholders who are interested in the On Demand Car Washtechnical architecture.

# Project Background, Objective(s)

## Project Background

On Demand Carwashleads to perform

Management of Customers details where one can register themselves and perform various operations

## Project Objective

On Demand Car​ will perform various operations like request for car wash , schedule wash and payment

Customers can first register themselves and then they can perform all the operations

# Design Pattern

|  |  |  |
| --- | --- | --- |
| # | Name | Description |
| 1 | CarWashWebAPIService | Using HTTP requests, we will use the respective action to trigger various operations |
| 2 | CarWash\_BAL | Using User Services to provide the service to Repository |
| 3 | CarWash\_DAL | Here we Imports and Exports the whole Data from Database and whatever the logic to use for that we are designing repository in that |

**Hardware and Software Requirements**

**Operating Systems**

Windows /Linux/Mac.

**Hardware Environment**

Processor: x86 or x64

Hard Disc: up to 3 GB of free space may be required

**Development Environment**

Microsoft Visual Studio 2019, 2022.

Visual Studio Code (Text-editor)

.NET Core 3.1 and above

Command Line (Optional)

Internet Information Services (IIS) 7.0+

Microsoft SQL Server 2019

**Browser support**

Chrome Firefox Opera Edge IE

Latest Latest Latest 13 + 11 +

**Specification and other Technologies**

Angular as front end and ASP.NET Core Web API as Backend.

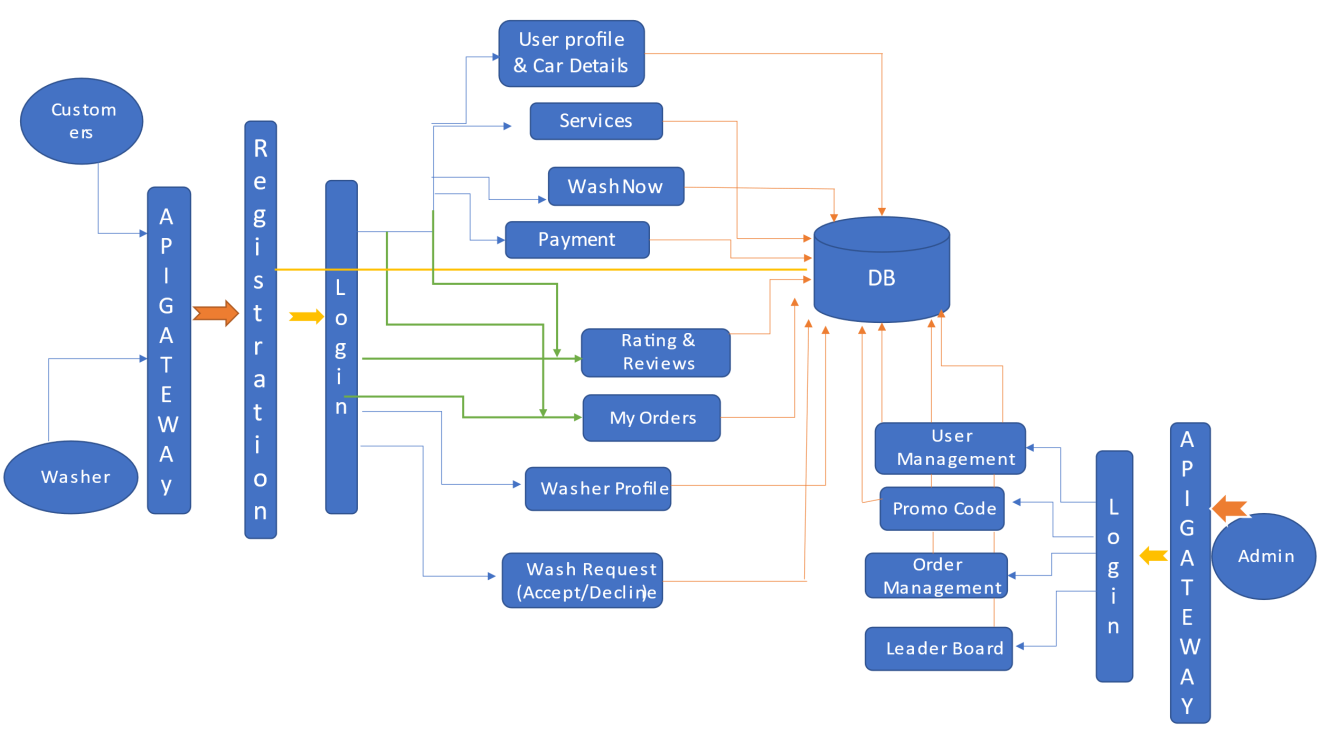
HTML 5

CSS

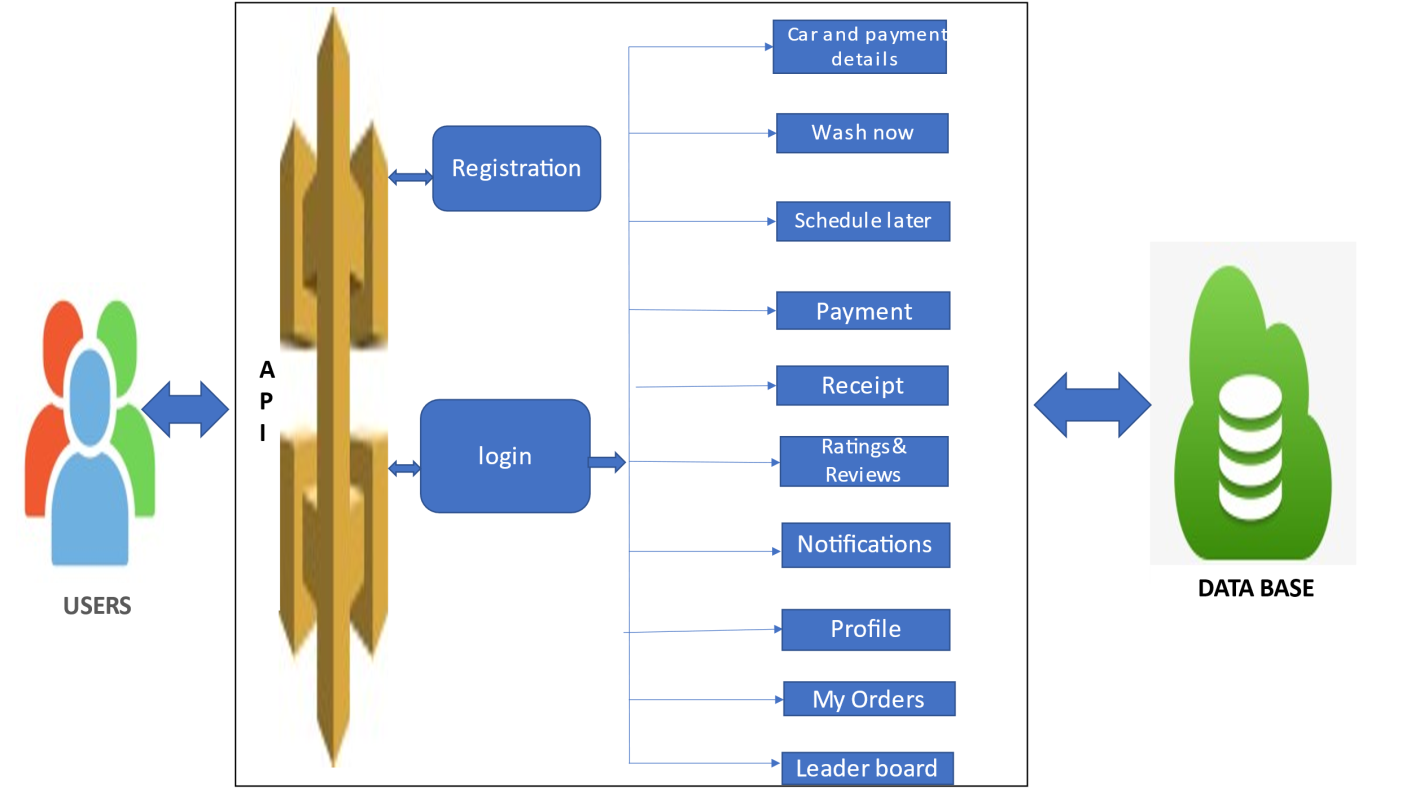
JavaScript

Typescript

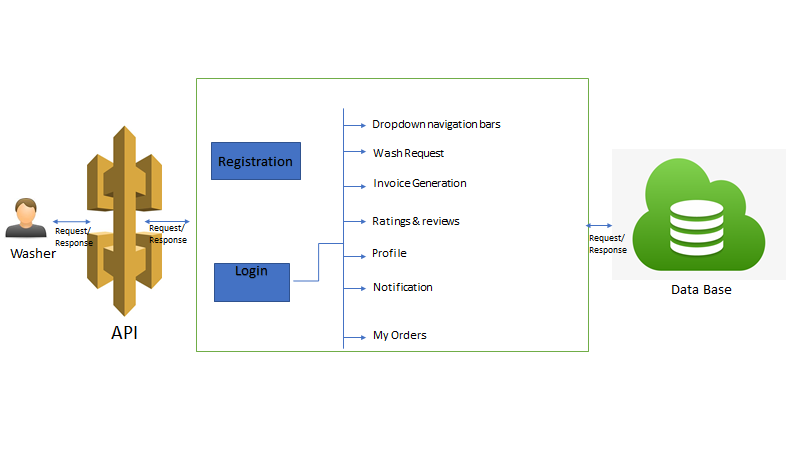
# SolutionDiagram



- **Customer View**



**- Washers View**

****

# -Admin ViewUseCase DiagramDiagram Description automatically generated

**ER Diagram**

Diagram

Description automatically generated

# 6.0 Solution Steps

**6.1 Customer Registration**

1. Customer will enter the required details such as first Name, last name, email, phone number, gender, age and click submit button browser directs the request to customer registration API
2. call reaches the Api gateway
3. API gateway does the routing and forwards the request to CustomerController.cs Andthis handle function will call the doProcess()
4. doProcess () will call the customerschemaValidator.dovalidate() function to do the input validation it will have thecustomerValidator.validateCustomerRegistrationas argument to perform the validation
5. If validation fails, then it will return the error code and error description. with status code
6. If validation is successful, then the handler will call the CustomerService.registerCustomer() which will call the CustomerRepository.registerCustomer() to store the data in database
7. It sends a response body with HTTP Success response to CustomerController.cs
8. CustomerControllerRegister() returns JSON Response
9. Success JSON response and HTTP status code 200 with corresponding success message.

**Car Details :**

1. customer wants to include the car details. enters the car detailslike car number model and car image .browser directs the request to Car detailspost method.
2. Call reaches the API gateway.
3. API gateway does the routing and forwards the request to CustomerController.cs this handle function calls the Addcar ()
4. addCar () will call the CustomerService.AddCarDetails() which calls the CustomerRepository.AddCarDetails() to fetch the data from database.
5. It sends response body with HTTP Success response code to CustomerController.cs
6. CustomerControllerAddCar() returns JSON Response
7. Success JSON response and HTTP status code 200 with corresponding success message.

**Wash Now**

1. customer wants to washthe car at that timehe/she needs to raise the requestwith relevant details. Browser directs the request to customer Request API
2. Call reaches the API gateway.
3. API gateway does the routing and forwards the request to CustomerController.cs this handle function calls the washNow()
4. washNow() will call the processWashNow() function to do request raised.
5. If washer accept the wash request customer will get the notification.
6. If no washer available customer should raise another request
7. After washer accepting the request browser redirects to payment window. Payment can be done before service or can be done on later.
8. It sends response body with HTTP Success response code to CustomeController.cs
9. CustomerControllerwashNow() returns JSON Response
10. Success JSON response and status HTTP code 200 with corresponding success message.

**Schedule Later**

1.customer wants to schedule the car wash request he/she can raise the request with schedule date and time. Browser directs the request to customer Request API

2.Call reaches the API gateway.

3.API gateway does the routing and forwards the request to CustomerController.cs this handle function calls the scheduleWash()

4.scheduleWash() will call the processScheduleWash() function to do request raised.After washer accepting the request browser redirects to payment window. Payment can be done before service or can be done on later.

5.It sends response body with HTTP Success response code to CustomeController.cs

6.ustomerController scheduleWash() returns JSON Response

7.Success JSON response and status HTTP code 200 with corresponding success

**6.2 Washer Steps**

**Washer Registration**

1. Washer will enter the required details such as first Name, last name, email, phone number, gender, age and click submit button browser directs the request to Washer registration API
2. call reaches the api gateway
3. API gateway does the routing and forwards the request to WasherController. Handle. And this handle function will call the do Process ()
4. do Process () will call the customerschemaValidator.dovalidate() function to do the input validation it will have the WasherValidator.validateWasherRegistration as argument to perform the validation
5. If validation fails, then it will return the error code and error description. with status code
6. If validation is successful, then the handler will call the WasherService.registerWasher() which will call the WasherRepository.registerWasher() to store the data in database
7. It sends a response body with HTTP Success response to WasherController.
8. WasherController Register() returns JSON Response
9. Success JSON response and HTTP status code 200 with corresponding success message.
10. listCustomerHandler returns JSON Response
11. Success JSON response and HTTP status code 200 with corresponding success message.

**Wash Orders:**

1. Washer wants to checkall the orders raised by the customers ,he/she request the browser to list wash orders.
2. Call reaches the API gateway.
3. API gateway does the routing and forwards the request to WasherController. Handle this handle function calls the getOrders ()
4. getOrders () will call the processGetOrders() function to do the input validation.
5. If validation fails, then it will return the error code and error description. with status code
6. If validation is successful, then the handler will call the WasherService.processGetOrders() which will call the WasherRepository.processGetOrders() to update the data in database
7. It sends response body with HTTP Success response code to WasherController
8. WasherController.getOrders() returns JSON Response
9. Success JSON response and status HTTP code 200 with corresponding success message.

# Classes

|  |  |  |
| --- | --- | --- |
| **#** | **Class** | **Description** |
| 1 | Models.cs | Model holds the customers and washers schema details |
| 2 | Controller.cs | The controller to handle the Workflow of All Operations which calls the UserService class |
| 3 | Service.cs | It contains the core business logic for the All Operations Which calls the UserRepository for Updating Data in Database |
| 4 | Repository.cs | This class deals with the data accessibility for User Operations |
| 5 | Interface.cs | This interface declaring the User Methods |
| 6 | Admin.cs | Model holds the Users schema details |

# Validations

For Registration:

User have to do registration with valid information like Name, , Email Address, Phone Number, Select User Type, Password, Confirm-password are inserted to register this System.

For Login:

Admin,Customer and washer can login in this system by using Email Address and password.

# Data model/Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | CWUserProfile | | |  | |
| PK | | UserId | | | INT | |
|  | | UserFirstName | | | VARCHAR(25) | |
|  | | UserLastName | | | VARCHAR(25) | |
|  | | UserRole | | | VARCHAR(50) | |
|  | | UserEmail | | | VARCHAR(50) | |
|  | | UserMobileNumber | | | VARCHAR(10) | |
|  | | UserGender | | | VARCHAR (25) | |
|  | | UserCreatedDate | | | Timestamp with time zone | |
|  | | CWAddress |  | |
| PK | | UserId | INT | |
| FK | | AddressId | INT | |
|  | | AddressState | VARCHAR (25) | |
|  | | AddressCity | VARCHAR (25) | |
|  | | AddressPincode | VARCHAR (25) | |
|  | | AddressLine1 | VARCHAR (25) | |
|  | | AddressLine2 | VARCHAR (25) | |
|  | | AddressLandmark | VARCHAR (25) | |
|  | | CreatedDate | Timestamp with time zone | |

|  |  |  |
| --- | --- | --- |
|  | CWCarRecords |  |
| PK | CarId | INT |
| Fk | UserId | INT |
|  | CarRegistrationNumber | VARCHAR(15) |
|  | CarCompany | VARCHAR(15) |
|  | CarModel | VARCHAR(10) |
|  | CarColor | VARCHAR(10) |
|  | CreatedDate | Timestamp with time zone |

|  |  |  |
| --- | --- | --- |
|  | CWWashNow |  |
| PK | WashNowId | INT |
| Fk | UserId | INT |
|  | WashNowRequestTime | Timestamp with time zone |
|  | WashNowSelectedCar | VARCHAR(15) |
|  | PackageName | VARCHAR(10) |
|  | WashNowWashNotes | VARCHAR(25) |
|  | CreatedDate | VARCHAR(25) |

|  |  |  |
| --- | --- | --- |
|  | CWScheduleLater |  |
| PK | SchedulelaterId | INT |
| Fk | UserId | INT |
|  | SchedulelaterRequestTime | Timestamp with time zone |
|  | SchedulelaterSelectedcar | VARCHAR(15) |
|  | SchedulelaterWashNotes | VARCHAR(15) |
|  | SchedulelaterCarLocation | VARCHAR(15) |
|  | PackageName | VARCHAR(10) |
|  | UsercreatedDate | Timestamp with time zone |

|  |  |  |
| --- | --- | --- |
|  | CWWashRequest |  |
| PK | RequestId | INT |
| FK | UserId | INT |
|  | SelectedCar | VARCHAR(25) |
|  | CarLocation | VARCHAR(15) |
|  | CreatedDate | Timestamp with time zone |
|  | RequestTime | Timestamp with time zone |
| Fk | PackageId | VARCHAR(255) |
|  | WashNotes | VARCHAR(255) |

# 10.0 HTTP Status Code

201 – CustomerRegistered

200 - Request succeeded

400 – Inputs are invalid

404 – Customer Not found

502 – Bad gateway

# 11.0 Unit Testing

|  |  |
| --- | --- |
| Project Name | On Demand Car Wash |
| Created by |  |
| Date of Creation |  |
| Date of review |  |

**For Registration of customers**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test CASE ID | TEST CASE SCENARIO | TEST CASE | PRE CONDITION | TEST STEPS | TEST DATA | EXPECTED RESULT | Actual Result |
| TC\_o1 | Customer registration | Enterthe valid data to get registered | Customer needs to enter all the valid details | 1) Enter  customer\_first\_name: John  customer\_last\_name:doe  emai:customer@gmail.com  Phone:0123456789  age:22  address:delhi  gender:male  2) Enter Submit | <Valid Details | Successful registration | Successful registration |
| TC\_o4 | Customer registration | Enter all the required fields to get registered | If customer misses one of the fields during registration which is marked as required in schema | 1) Enter  customer\_first\_name: John  customer\_last\_name:doe  Phone:0123456789  age:22  address:delhi  gender:male  2) Enter Submit | <email is missing> | You need to enter email | You need to enter email |