**On-Demand Car Wash**

**POC**

**Low Level Design (LLD)**

Date: 15/07/2022

Current Document Version: [*1.0*]

DOCUMENT APPROVAL

**Approvers of this document**

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

**Document Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Version #** | **Author** | **Date** | **Description** |
| 1.0 | Pooja U | 07/07/2022 | On-Demand Car Wash LLD |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1.0 Document Purpose 1](#_Toc94636300)

[2.0 Intended Audience 1](#_Toc94636301)

[3.0 Project Background, Objective(s) 1](#_Toc94636302)

[3.1 Project Background 1](#_Toc94636303)

[3.2 Project Objective 1](#_Toc94636304)

[4.0 Design Pattern 2](#_Toc94636305)

[5.0 Solution Diagram 3](#_Toc94636306)

[5.1 Data Flow Diagram/usecase diagram 3-6](#_Toc94636303)

[6.0 Solution Steps 7-9](#_Toc94636307)

[7.0 Hardware and Software Requirements 9](#_Toc94636307)

[8.0 Classes/function name 9-10](#_Toc94636308)

[9.0 Validations](#_Toc94636309)

[10.0 Data model/Tables 6](#_Toc94636310)

[11.0 API Canvas 10](#_Toc94636311)

# Document Purpose:

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

# Intended Audience:

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

|  |  |
| --- | --- |
| Role | Nature of Engagement in WB Classics Portal Technical Architecture |
| Product Owners/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 on-demand car wash Platform Architecture is aligned to business goals and architecture, architectural guidelines. |
| Solution Architects | To ensure solution design and architecture is aligned to business requirements, architectural guidelines. |
| Developers | Use Technical Architecture Document as the guiding document for detail design and implantation approach to align with on-demand car wash microservice |

# Project Background, Objective(s)

## **Project Background**

On-Demand Car Wash microservice leads to perform online car services at their doorsteps, where each end users register themselves and perform various operations.

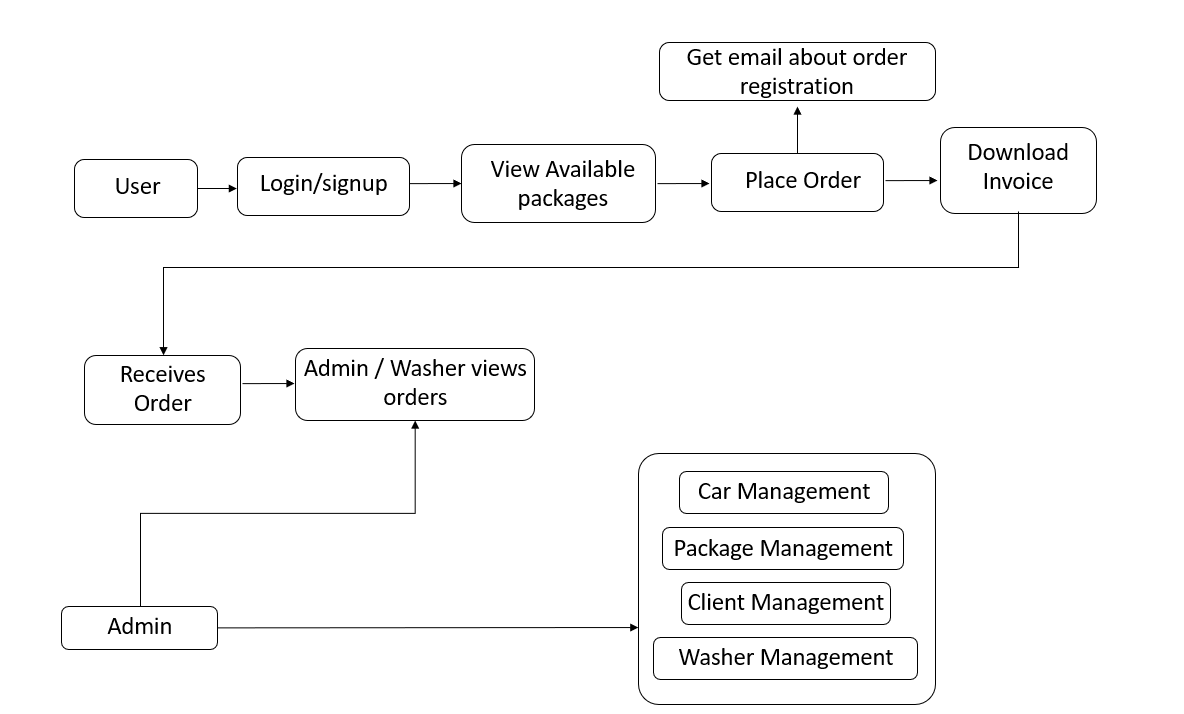
## **Project Objective**

The main objective of On-Demand Car Wash microservice is to simplify the traditional way of going to car service centers by replacing it with online car wash service. The system will be able to handle many services to take care of all customers in a quick manner.

# Design Pattern:

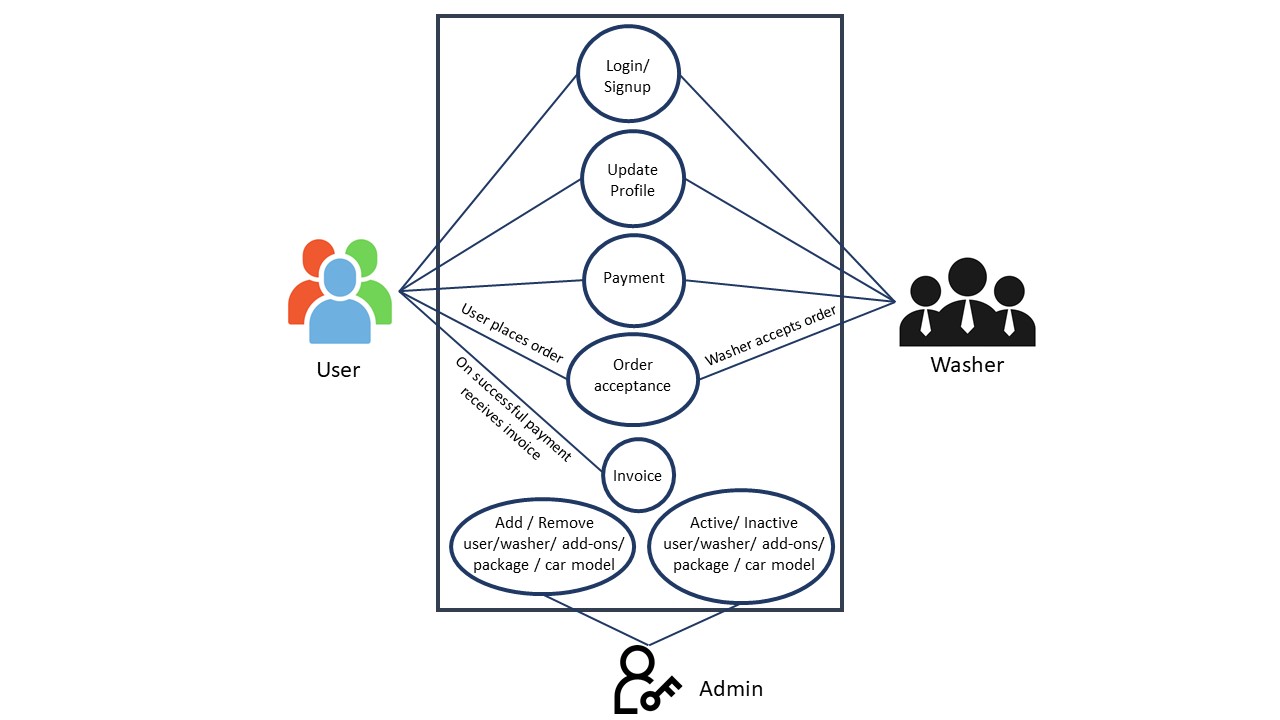
|  |  |  |
| --- | --- | --- |
| # | Name | Description |
| 1 | Web API | Using HTTP requests, we will use the respective action to trigger various operations. |
| 2 | Web App | A web app is created using Angular and connected with web API to perform various operations. |
| 3 | Database | A database is created with various tables with constraints, keys. |

# 5.0 Solution Diagram



# 

# 5.2 Use Case Diagram:



# 6.0 Solution Steps

**Customer Registration:**

1. User will enter the required details such as firstname, lastname, email, phone number, car details and click register 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 Customer register Handler. And this handler function will redirect to the login.
4. All the fields entered by the customer will be validated and if validation succeeds then customer will be redirected to the login page.
5. If validation fails then the customer will be redirected to the registration page to enter the valid details.
6. It sends a response body with HTTP Success response to Customer register Handler.

**Customer Dashboard:**

1. After the login the user will be redirected to the customer dashboard.
2. In the dashboard the customer can select the wash now or he can schedule it later by entering car details, wash package and add-ons with the extra information.
3. If wash now clicked the user has to provide location.
4. If user wants to schedule it later, he can click on wash later button and provide the details such as location, date and time.
5. When user clicks on either wash or wash it later the user will be redirected to the payment page where user needs to add his credit/debit card details in order to make payment.
6. Once the car wash is done the user can give ratings to the washers and can share his opinions through the review.

**Customer Profile:**

1. User can view his personal information in this section.
2. If user wants to update his details an edit option is provided for the user
3. All the orders by the user will be displayed here.
4. User will be notified when the washer accepts/rejects his requests.
5. When the service is completed, user will be notified once again.

**Car washers:**

1. Washer must enter his credentials in order to login
2. In case of successful login, the washer will be able to view all the requests by the user and perform necessary operations
3. If the washer accepts the request from the user, he can view the user’s location by clicking the user’s location button.
4. washer will be notified when the user schedules the wash and when he completes the payment.

**Admin:**

1. Admin will login using proper id and password.
2. When the login is successful admin can view the functionalities such as user management, service plan management, order management
3. In user management section admin can view all the registered users and washers.
4. In service plan management section admin can add new service plan or edit existing service plan.
5. In order management section admin will assign washers to the users.

# 7.0 Hardware and software Requirements:

|  |  |  |
| --- | --- | --- |
| # | Software | Version |
| 1 | Microsoft SQL Server Management Studio | 12.0.2269.0 |
| 2 | Visual Studio Community 2019 | 16.11.17 |
| 3 | Target Framework | .NET 5.0 |
| 4 | NodeJS | 16.16.0 |
| 5 | Angular CLI | 14.0.6 |

|  |  |  |
| --- | --- | --- |
| # | Hardware | Specifications |
| 1 | Processor | Minimum 1.8 GHz: Recommended 2 GHz and above |
| 2 | RAM | Minimum 4 GB: Recommended 8 GB and above |
| 3 | Hard Drive | Minimum 32 GB: Recommended 64 GB and above |

# 8.0 Classes/function

|  |  |  |
| --- | --- | --- |
| # | Class | Description |
| 1 | Customer | Model holds the **customers schema** details |
| 2 | Admin | Model holds the **admin schema** details |
| 3 | Washer | Model holds the **Washer schema** details |
| 4 | Package | Model holds the **Package schema** details |
| 5 | OrderDetails | Model holds the **Order details schema** details |
| 6 | AddOns | Model holds the **add-ons schema** details |
| 7 | PaymentDetails | Model holds the **payment details schema** details |
| 8 | PromoCode | Model holds the **promo code schema** details |
| 9 | CarModels | Model holds the **car models schema** details |

|  |  |  |
| --- | --- | --- |
| # | Functions | Description |
| 1 | userRegistration.cs | It contains the logic for the registration of new customers |
| 2 | getCustomerDetails.cs | It contains the logic for getting the details of customers |
| 3 | updateCustomer.cs | It contains the logic for updating the details of a customer |
| 4 | removeCustomer.cs | It contains the logic for deleting a customer |
| 5 | Washer.cs | Model holds the Washer schema details |
| 6 | Admin.cs | Model holds the admin schema details |

# API Canvas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Micro-Service** | **Path** | **Verb** | **API Description** | **Role** | **Auth** |
| Car-Wash | /users/id | GET | To get particular user details | User | True |
| Car-Wash | /users/id | PUT | To Update user details | User | True |
| Car-Wash | /users | POST | User Registration | User | True |
| Car-Wash | /users/id | DELETE | To remove user | User | True |
| Car-Wash | /users/ | GET | To get all users | None | True |
| Car-Wash | admin/id | GET | To get particular admin details | Admin | True |
| Car-Wash | /admin/id | PUT | To Update admin details | Admin | True |
| Car-Wash | /admin | POST | admin Registration | Admin | True |
| Car-Wash | /admin /id | DELETE | To remove admin | Admin | True |
| Car-Wash | /admin / | GET | To get all admins | None | True |
| Car-Wash | washer/id | GET | To get particular washer details | Washer | True |
| Car-Wash | /washer /id | PUT | To Update washer details | Washer | True |
| Car-Wash | washer | POST | washer Registration | Washer | True |
| Car-Wash | /washer /id | DELETE | To remove washer | Washer | True |
| Car-Wash | /washer / | GET | To get all washer | Washer | True |

# Data model/Table

|  |  |  |
| --- | --- | --- |
| Admin | | |
| PK | admin\_id | int |
|  | name | varchar(MAX) |
|  | email | varchar(MAX) |
|  | password | varchar(MAX) |

|  |  |  |
| --- | --- | --- |
| user | | |
| PK | user\_id | int |
|  | name | varchar(MAX) |
|  | email | varchar(MAX) |
|  | ph\_no | varchar(MAX) |
|  | password | varchar(MAX) |

|  |  |  |
| --- | --- | --- |
| Washer | | |
| PK | washerid | int |
|  | name | varchar(MAX) |
|  | email | varchar(MAX) |
|  | password | varchar(MAX) |
|  | phone | varchar(MAX) |

|  |  |  |
| --- | --- | --- |
| car | | |
| PK | carid | int |
|  | name | varchar(MAX) |
|  | type | varchar(MAX) |
|  | image | binary |

|  |  |  |
| --- | --- | --- |
| carservice | | |
| PK | serviceid | int |
| FK | carid | int |
|  | service | varchar(MAX) |
|  | price | decimal |
|  | description |  |

|  |  |  |
| --- | --- | --- |
| booking | | |
| PK | bookingid | int |
| FK | serviceid | int |
| FK | userid | int |
|  | slot | binary |
|  | date | date |
|  | add\_on | varchar(MAX) |
|  | price | int |
|  | rating | int |
|  | review | varchar(MAX) |
|  | status | varchar(MAX) |
| FK | washerid | int |
|  | dateassigned | date |
|  | datedelivered | date |