

BULT Batch Case Study

Requirements Specification for On-Demand Car Wash Company

**Prepared by
(Group 1)**

Korrai Bhargav

Vutukuri Naga Venkata kasi Amarnadh kumar

Imam Gouse Peddarumala shaik

Sujeet Kumar

jami mutyalarao

Sudesh K

Adrija Biswas

Priyank Joshi

Neetu kumari

DURGESH KUMAR JHA

Table of Contents

1.Introduction.....	
1.1 Document Purpose	
1.2 Intended Audience.....	
1.3 Proposed System.....	
1.4 Product Scope.....	
1.5 Overview & Overall Description.....	
1.6 Functionalities of the Project.....	
2.Analysis and SRS Document	
2.1 System requirements specification.....	
2.2 Introduction.....	
2.3 Purpose.....	
2.4 Functional Requirements.....	
2.5 Non-Functional requirements.....	
2.6 Software Requirements.....	
2.7 Hardware Requirements.....	
2.8 Methodology.....	
3 Design.....	
3.1 Class Diagram.....	
3.2 Use Case.....	
3.3 Sequence Diagram.....	
4.Implementation	
5.Testing.....	
6.Conclusion.....	
7.Bibliography.....	

1. Introduction

1.1 Document Purpose

This requirements specification document describes the features and behaviour of a technical solution for a On Demand car wash company. On Demand car wash company will enable customers to create an account via company's webpage, add details and opt for services in present or future the account and make payment for paying in company's self-service car wash. Main purpose of on Demand car wash company is to offer additional option for clients to avail service anywhere. The specification document also defines at a high-level the main business process that will be supported.

1.2 Intended Audience

Document is intended for:

- The Case study for their review and evaluation of the group project.
- The car wash company for ordering respective software solution.
- The software development team for their use in developing the software solution

1.3 Proposed System:

In the existing system customer must visit nearest service centre in order to get the service done that consume their time finds difficult to manage office work along with this and service provider won't get the benefit needed hence there is a need to give the customer best experience needed.

1.4 Product Scope

Today, with the changing preferences and growing income, people are likely to roam around in sedans and SUV's.

- With the increasing car consumer market, car wash segment is all set to join the race. Forecasting the expanding markets of the car care services, on Demand car wash company approached Hyperlink Info System to develop an interface that connects customers and service providers to avail the best wash services within the nearest proximity.
- The main aim of the client is to explore the untapped market opportunities by satisfying increasing consumer demands through professional car wash services.
- The app is developed on Android and iOS platforms and designed for the USA audience only.
- To enable car wash company's clients to create an Account on company's website.
- To enable car wash company's clients to place orders of client card via their Account.
- To enable car wash company's clients to use Account to pay for services at self-service car wash stations.
- To enable car wash company to manage client cards and any issues/procedures associated with Account. Software solution is related to and forms an integral whole with following other software solutions/systems:
 - Company webpage
 - Wash Now.
 - schedule car wash for upcoming dates.
 - Braintree Payment Gateway.
 - Review and Ratings.
 - Notification

1.5 Overview

- The remaining sections of this document provide an overall description (including product perspective, product functions, user classes and characteristics, assumptions, dependencies and constraints while designing the OnDemand car wash system), a description of system features, including the functional requirements, and the descriptions of non-functional requirements, scenarios and use cases. Overall description of the product is discussed in section 2 of this document. Section 3, the system features, gives detailed description of functional requirements. Section 4 describes non-functional requirements of the product. Section 5 gives scenarios and use cases. Appendices A and B contain supporting information: process models and issues list.

Overall Description

- **User Classes and Characteristics**

1. Customers Clients using company's OnDemand Car Wash. Can be adults of any age, education, technical expertise, or experience. Frequency of system use can vary from frequent to occasional. Can choose the Braintree payment gateway as one of the payment options. Customer may be either a private person or a person who is an authorized representative of a company.
2. Service manager Car wash company's employee responsible for handling customer relationships, managing any possible issues and problems and ensuring continuous service is provided. Familiar with the system (administrator level). Service manager is representing the car wash company.
3. Car wash company who offers car wash self-service to its customers. Company wants their customers to have different options available for car wash service. Company wants customer to a OnDemand service to the currently available options, aiming to increase customer satisfaction and possibly attract more customers. Company is paying for the development and maintenance

1.6 Functionalities of the project: -

- ❖ **For Customers**

- **Sign up/Login:** Sign up with basic details. A user can login using Email or Facebook.
- **Car & payment Details:** Add car details along with its images. Also, add payment details for further transaction process.
- **Wash Now:** After filling appropriate details, users can book their first car wash within few taps. All they need to do is select their car and wash package. They can even add a new car and include some extra wash add-ons if they wish to. Users can also guide washers by adding some important notes and information.
- **Schedule Later:** An app also allows users to schedule car wash for upcoming dates. Just select their preferred date, time, location and package and its done. The wash request is sent to the washers. If no acknowledgement is received from washer regarding approve/decline of was request, a random washer is assigned to the user by admin.
- **Payment:** A user can confirm the booking and pay washers along their debit/credit cards. Braintree payment gateway is integrated for payments.
- **Receipt:** After the successful car wash, washer sends payment receipt to the user which includes after wash car photo, package, and extra add-on details. Users are also facilitated to share information on various social media platforms.

- **Review and Ratings:** After the successful car wash, users can share their thoughts and opinions for washers through review and ratings.
- **Notifications:** Users will be notified the below scenario:
 - ❖ When washer accepts/rejects the request.
 - ❖ When washer starts and complete service.
 - ❖ Receipt received.
- **Profile:** Users can view and edit their profile information.
- **My Orders:** Here users can view order details in below form:
 - ❖ Current Orders: View list of pending, accepted and in-process orders.
 - ❖ Past Orders: view list of completed and cancelled orders.
- **Leader board:** This section displays the list of leaders who saved maximum water gallons by washing cars.

❖ **For Car Washers: -**

- **Login:** Login using Email or Facebook
- **Wash Request:** Wash Request is sent to the washer along with the user details. The washer can either accept or decline the request. On accepting the request, a washer can navigate through customer's address by google maps Integration.
- **Invoice Generation:** After successful car wash service, an invoice is generated and sent to the customers which include washed car image, package, and add-on details.
- **Rating and Reviews:** After the successful car wash, washers can share their experience with the users through reviews and ratings.
- **Profile:** Washers can view and update their profile information like profile picture and contact information. In this section, user can view the ratings for washers too.
- **Notification:** Washers will be notified in the below scenario: Scheduled wash notification before 2 hours.
 - New wash order.
 - When users cancel wash request.
 - On successful Payment by users.
- **My Orders:** Here washers can view orders in below form:
 - Current Orders: View pending accepted and in-process order list.
 - Past Orders: View completed and cancelled orders

❖ **Admin Panel:**

- **User Management:** It includes below modules:
- **Customer:**
 - -Edit profile
 - -view ratings
 - -Active/Inactive customers
- **Washers:**
 - -Add/Edit washer's profile

- -Active/Inactive customers
- -view customer's ratings and reviews
- -Export washer's report to excel
- **Car Management:**
 - Add/Edit car details
 - Active/Inactive Car
- **Service Plan Management:**
 - Add/Edit service plan details
 - Active/Inactive service plans
- **Add-On Management:**
 - Add/Edit Add-On List
 - Active/Inactive Add-On List
- **Promo Code Management:**
 - Add/Edit Promo Code
- **Order Management:**
 - Wash Now & Schedule:
 - View order details
 - Assign pending order request to washers
 - Shows the list of pending, accepted, under process, completed and cancelled orders.
- **Report Management:**
 - Admin can filter and generate reports based on order number, washer name, type, service, and date.
- **Leader board:**
 - Shows the list of leader boards.
- **Advance Report Management:**
 - Admin can generate advanced wash reports based on business, sales, users and locations.

2.ANALYSIS AND SRS DOCUMENT

2.1 SYSTEM REQUIREMENTS SPECIFICATION

SRS defines as the Software Requirements Specification which describes what the software will do and how it will be expected to perform actions and execute it. It also describes the functionality of the project needs to fulfil all stakeholder needs.

2.2 INTRODUCTION

The main objective of the SRS is to completely explain and documents every actions performs in the back end part also the project deals with the both client and server side processes, action and executes based on the required components and needs. Payment parts involved in the project should be carefully managed and maintained for the user goodwill so that small errors may cause huge impact in the user side as well as client side.

Product Requirements:

1. Login
2. New Registration
3. Account Section
4. Displaying Current Resource
5. Creating New Resource
6. Managing Resource

Process Requirements

1. Login Page
2. Registration Page
3. Account Section Page
4. Report Generation Page
5. Payment Details Page
6. Managing Resources Page
7. Displaying New Resource Page

2.3 Purpose

The main aim of the client is to explore the untapped market opportunities by satisfying increasing consumer demands through professional car wash services also as the increasing demand for car wash in the current world, the car wash management system should be updated day to day basis and needs for the customers.

2.4 Functional Requirements

SRS No	SRS Name	Description
01	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.
02	Login	Admin and User and Service Provider can login in this system by using Email Address and password.
03	Manage Profile	By Login to this System Admin and User and Service Provider Can Manage his /her Profile changing Name, User Name, Email & Phone Number.
04	Request For Service	User and Service Provider can request for service. User send request to Service Provider he/she needs car wash or repairing and Service Provider send request to admin if he wants to new service added.
05	Cancel Request	If User and Service Provider want to cancel their request theycan cancel it.
06	Manage User Category	Admin can view and edit User category, delete user.
07	Manage User	Admin can view service and request.
08	Manage Service	Admin can view, edit and delete service category.

Table 2. 3: Functional Requirements

2.5 Non-Functional Requirements

SRS No	SRS Name	Description
01	Privacy	All of the Users information is saved Private and anyone can't view it.
02	Robustness	If user's device destroys, a backup of services is stored in database.
03	Performance	The System must be fast to Response.

Table 2. 4: Non-Functional Requirements

2.6 Software Requirements

1. Programming Languages: Angular
2. Database: Microsoft SQL Server Management Studio (SSMS)
3. IDE: Visual Studio
4. Operating System: Windows

2.7 Hardware Requirements

1. Processor: Intel
2. RAM: 8GB
3. HARD DISK: 1TB

2.8 Waterfall Methodology

The Waterfall methodology is a sequential approach that works with set deadlines, requirements, and outcomes. Individual execution teams are not required to be in constant touch with one another and are normally self-contained unless special integrations are necessary. Team members perform more independently and aren't required to submit progress reports as frequently as they would in an agile methodology. Normally, one phase does not start until the previous one has been completed.

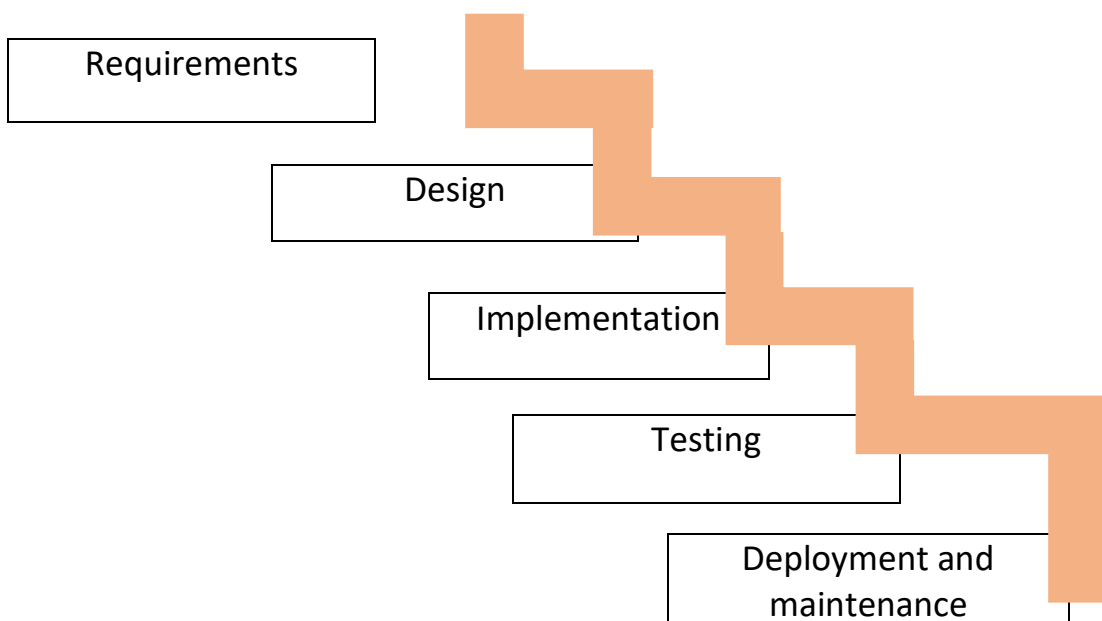


Table 2.8: The Waterfall method involves stages as mentioned below

2.8.1 Requirements

The Waterfall approach assumes that all project requirements can be acquired and understood in advance. The project manager makes every effort to fully comprehend the client's expectations. Written requirements are used to explain each stage of the project, including expenses, assumptions, risks, dependencies, success metrics, and completion timeframes, and are usually contained in a single document.

2.8.2 Design

Software developers create set-ups, layouts, and data models to solve the difficulties posed by the product requirements. In beginning, a higher-level or logical design is made, which outlines the project's objective and scope, as well as the overall traffic flow of each component and the integration points. After that, it's turned into a physical design employing specialized hardware and software technologies.

2.8.3 Implementation

Following the completion of the design, implementation begins. This phase involves programmers writing application programs based on project needs and specifications, as well as some testing and implementation. If major changes are required during this stage, it may be necessary to return to the design step.

2.8.4 Testing

Testing is required before a product can be distributed to clients to guarantee that it is free of faults and that all requirements have been met, ensuring a positive user experience with the product. To develop their test cases, the testing team will leverage the product manager's design documents, personas, and user case scenarios.

2.8.5 Deployment and Maintenance

The maintenance phase begins after the project has been distributed to the market or to consumers. As bugs are discovered and user requests for changes are received, a team will be formed to handle updates and the deployment of new software versions.

3.Design

3.1 Class Diagram:

Class diagram is a static diagram. It represents the static view of an application. The class diagram is not only used for visualizing, describing, and documenting various aspects of a system but also for constructing executable code of the software application.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

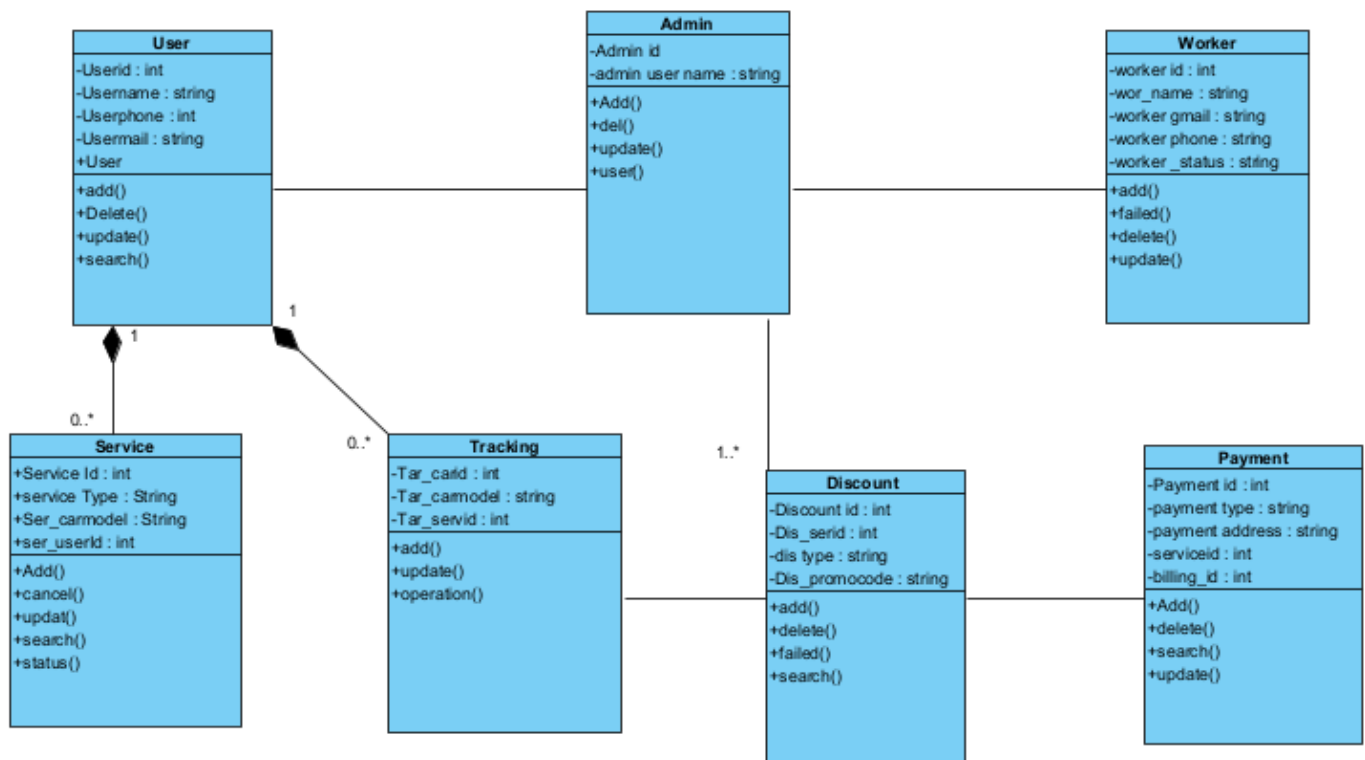


Table 3.1: Class Diagram

3.2: Use case Diagram: -

A use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you will use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

- **Actors:** The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.
- **System:** A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.
- **Goals:** The result of most use cases. A successful diagram should describe the activities and variants used to reach the goal.

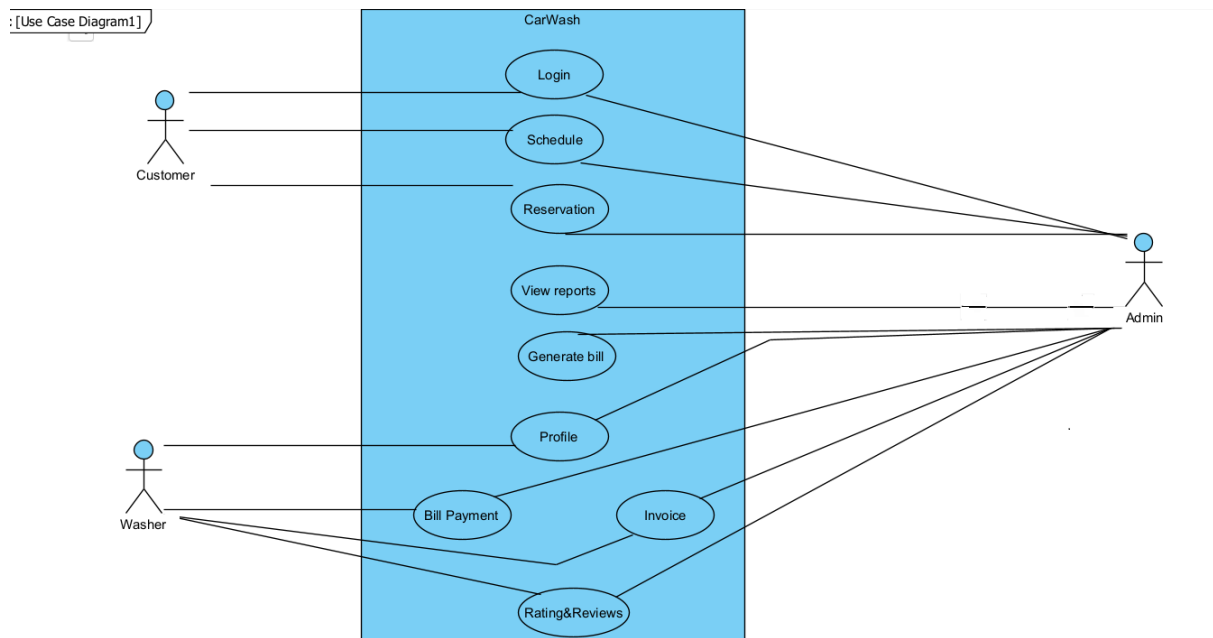


Table 3.2: Use case Diagram

3.3: Sequence Diagram:

Most used **interaction** diagram.

Interaction diagram – An interaction diagram is a system can be a cumbersome task, we use diverse types of interaction diagrams to capture various features and aspects of interaction in a system.

Sequence Diagrams – A sequence diagram simply depicts the interaction between objects in sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order objects are in a system function. These diagrams are widely used by businesspeople and software developers to document and understand requirements for new and existing systems.

interactive behavior of a system.

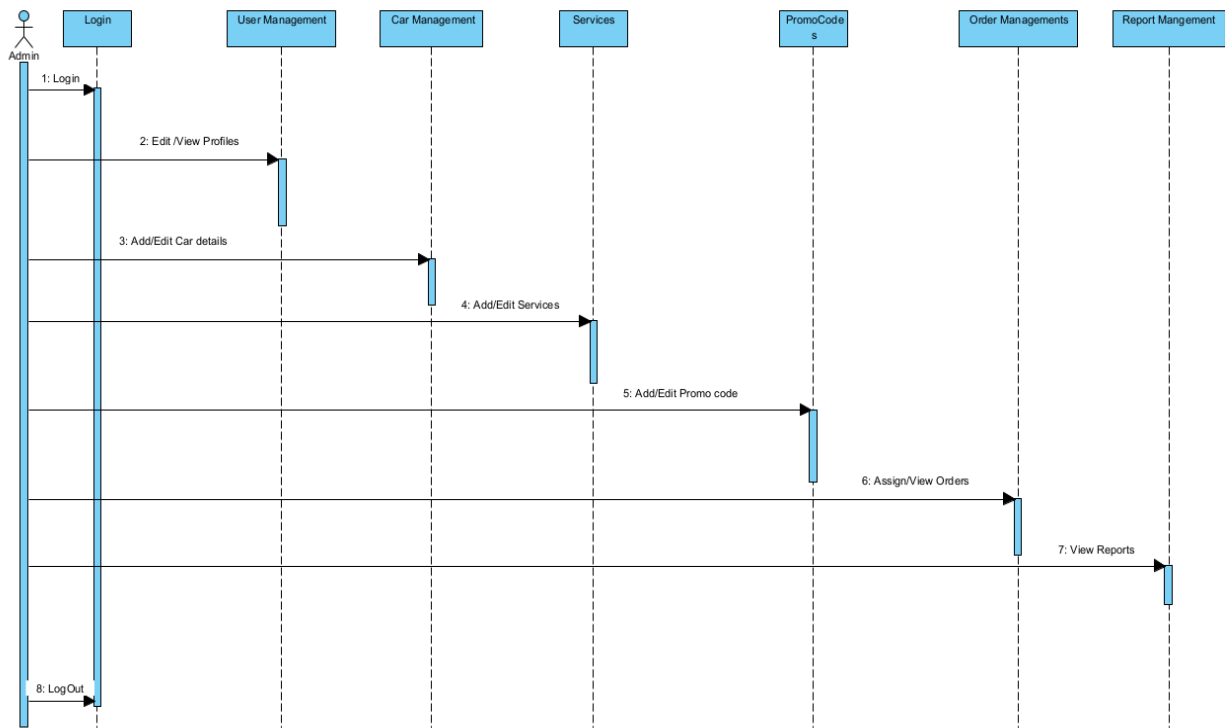


Table 3.3.1: Sequence Diagram User

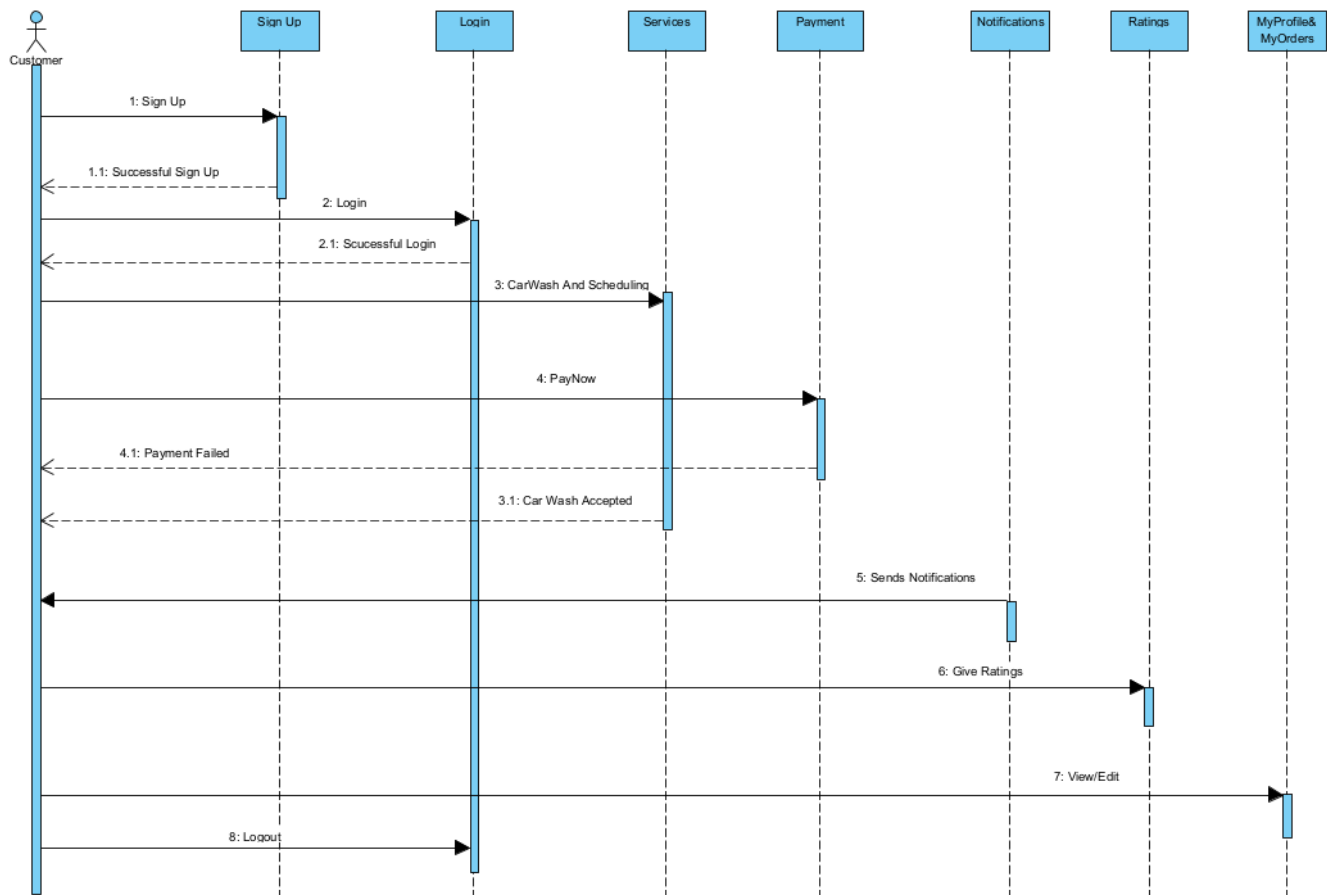


Table 3.3.2: Sequence Diagram Admin

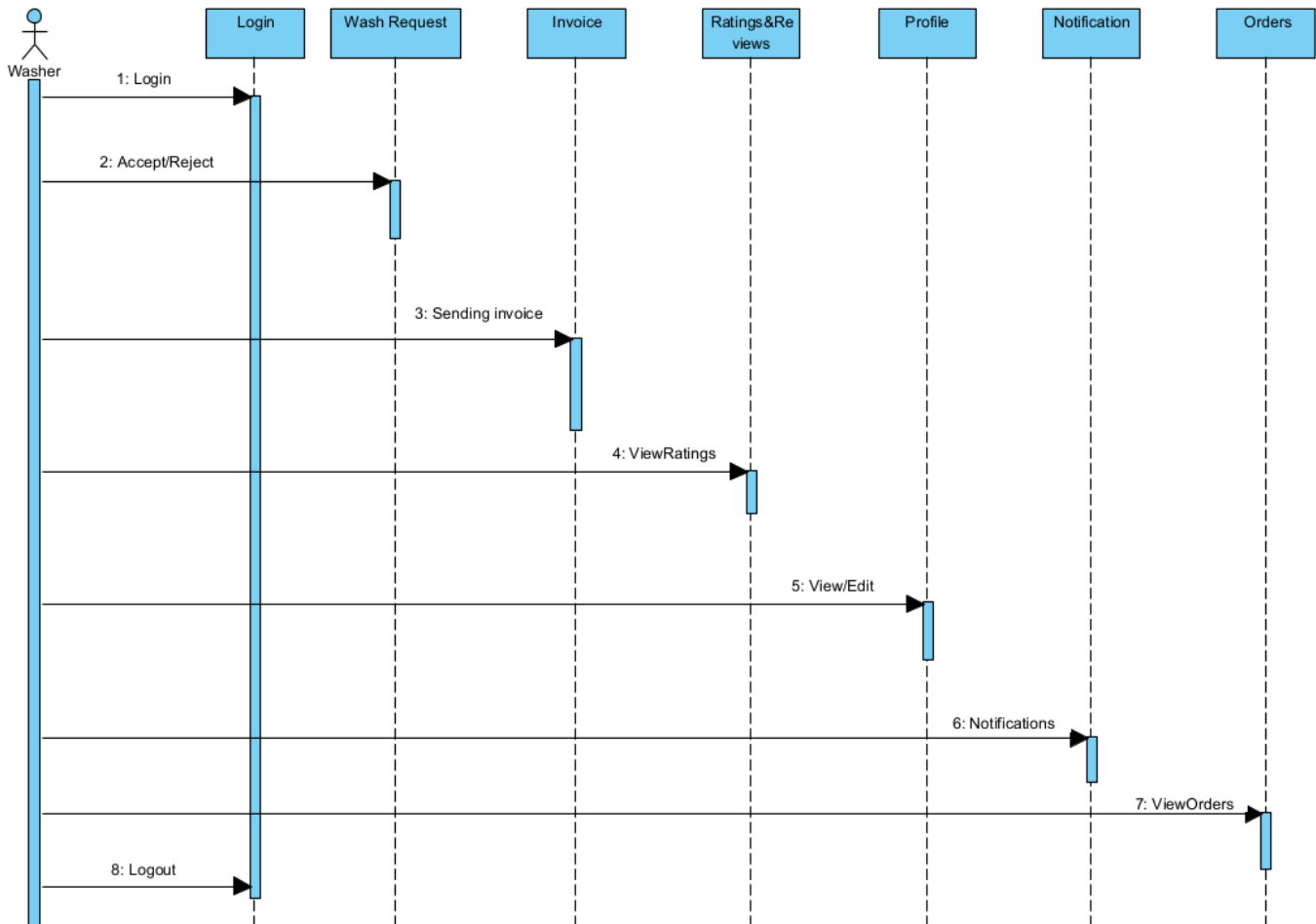


Table3.3.3: Sequence Diagram Washer

4. Implementation:

Perform the step-by-step approach for implanting the entire system.

- Home Page
 - User page
 - Login
 - User Management
 - Car Management
 - Service
 - Promocode
 - Order Management
 - Report Management
 - Admin page
 - Signup
 - Login
 - Service
 - Payment
 - Notification
 - Rating
 - My Profile & My Order

- Car Washer Page
 - Login
 - Wash Request
 - Invoice
 - Rating & Review
 - Notification
 - Profile
 - Orders

5. Testing

Testing Methodologies

Testing is the process of finding differences between the expected behaviour specified by system models and the observed behaviour implemented system. From modelling point of view, testing is the attempt of falsification of the system with respect to the system models. The goal of testing is to design tests that exercise defects in the system and to reveal problems.

The process of executing a program with intent of finding errors is called testing. During testing, the program to be tested is executed with a set of test cases, and the output of the program for the test cases is evaluated to determine if the program is performing as expected. Testing forms the first step in determining the errors in the program. The success of testing in revealing errors in program depends critically on test cases. Testing is very important document for all software related projects. Without Test plan a project is considered low quality.

According to Glen Myers, a software testing has the following objectives: -

- The process of investigating and checking a program to find whether there is error or and does it fulfil the requirements pr not is called testing.
- When the number of errors found during the testing is high, it indicates that the testing was good and is a sign of good test case.
- Finding an unknown error that wasn't discovered yet is a sign of a successful and a good test case.

Strategic Approach to Software Testing:

- The main objective of software testing is to design the tests in such a way that it systematically finds different types of errors without taking much time and effort so that less time is required for the development of the software.

The overall strategy for testing software includes:

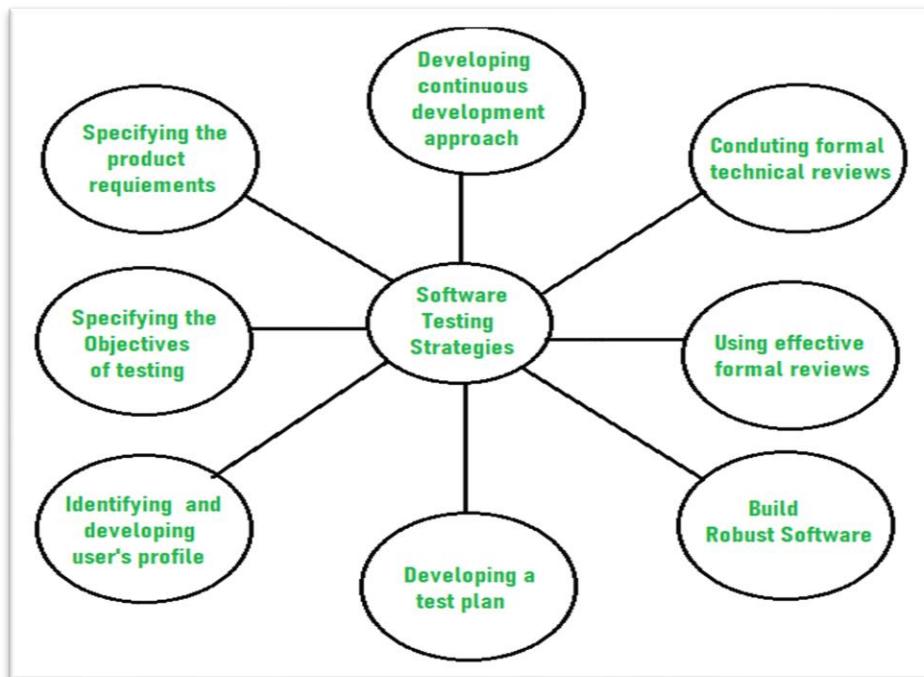


Table 5.1 Overall Strategy for Testing

- ❖ Before testing starts, it's necessary to identify and specify the requirements of the product in a quantifiable manner.
- ❖ Specifying the objectives of testing in a clear and detailed manner.
- ❖ For the software, identifying the user's category and developing a profile for each user.
- ❖ Developing a test plan to give value and focus on rapid-cycle testing.
- ❖ Robust software is developed that is designed to test itself.
- ❖ Before testing, using effective formal reviews as a filter.
- ❖ Conduct formal technical reviews to evaluate the nature, quality or ability of the test strategy and test cases.
- ❖ For the testing process, developing a approach for the continuous development.

Different Levels of Testing

- Client needs
- Design
- Unit Testing
- Acceptance Testing Requirements
- System Testing
- Integration Testing Code

Testing is the process of finding difference between the expected behaviour specified by system models and the observed behaviour of the implemented system.

Testing Activities:

Different levels of testing are used in the testing process, each level of testing aims to test different aspects of the system the basic levels are:

- Unit testing
- Integration testing
- System testing
- Acceptance testing

Unit Testing:

Unit testing focuses on the building blocks of the software system, that is, objects and sub system. There are three motivations behind focusing on components. First, unit testing reduces the complexity of the overall tests activities, allowing us to focus on smaller units of the system. Second, unit testing makes it easier to pinpoint and correct faults given that few components are involved in this test. Third, Unit testing allows parallelism in the testing activities, i.e., each component can be tested independently of one another. Hence the goal is to test the internal logic of the module.

Integration Testing:

In the integration testing, many test modules are combined into sub systems, which are then tested. The goal here is to see if the modules can be integrated properly, the emphasis being on testing module interaction. After structural testing and functional testing, we get error free modules. These modules are to be integrated to get the required results of the system. After checking a module, another module is tested and is integrated with the previous module. After the integration, the test cases are generated, and the results are tested.

System Testing:

In system testing the entire software is tested. The reference document for this process is the requirement document and the goal is to see whether the software meets its requirements. The system was tested for various test cases with various inputs.

Acceptance Testing:

Acceptance testing is sometimes performed with realistic data of the client to demonstrate that the software is working satisfactory. Testing here focus on the external behaviour of the system, the internal logic of the program is not emphasized. In acceptance testing the system is tested for various inputs.

There are six important steps to test a web Application:

❖ Functional testing

The step ensures that the functionalities of a web application are properly functioning or not. Functional testing takes place in the source code.

Functional testing includes:

- Determining the data input and entry
- Test case execution
- Functions need to properly identify because the software runs effectively through the integration of functions
- Actual results must be analysed

❖ Usability testing:

This testing type focuses on how user experiences while using a particular web application. Efforts are put in to ensure that the application is built in-line with user needs. This testing method makes it a point to see that a user can easily navigate through the functions of an application. The content is displayed in the web application must be clearly visible.

6. Conclusion:

The main objective of **On-Demand Car Wash Company System** is to create an effective system for Car Washing experience. The project is totally built for create a better and effective experience for washing the car when it is required. It will provide a part for User who will experience the process and for an Admin also who gone monitor every part how the system gone perform. It will increase the entire experience smooth and effective.

7.Bibliography:

- <https://www.w3schools.com/html/>
- <https://www.w3schools.com/w3css/>
- <https://www.w3schools.com/js/>
- <https://codewithmosh.com/p/the-ultimate-typescript>
- <https://programmingwithmosh.com/angular/>
- <https://www.tutorialsteacher.com/webapi>
- <https://www.pluralsight.com/paths/aspnet-core>
- <https://codewithmosh.com/p/entity-framework>
- <http://www.softwaretestinghelp.com/>
- <https://www.guru99.com/>
- <https://www.geeksforgeeks.org/>