

ONEMARKET



Ikram Aslam

Awais Ur Rehman

Muhammad Ibrahim

Supervised by

Ms. Fatima Gillani

*Submitted for the partial fulfilment of BS Software Engineering degree to the
Faculty of Engineering & Computer Science*

NATIONAL UNIVERSITY OF MODERN LANGUAGES

ISLAMABAD

DECEMBER, 2022

ABSTRACT

In the modern world, almost all businesses have moved online especially after COVID-19 has accelerated digital transformation. Some systems provide the facility to buy the accessories of the vehicles however there is no such system that connects the user with the physical existing shops, especially for Rawalpindi and Islamabad vehicle markets.

The proposed system will provide the user with the facility to explore the registered vehicle shops in Rawalpindi and Islamabad and find the right shop for their vehicles' product(s). The proposed system will ensure the presence of all the shops available in the twin cities vehicle markets to improve the services of automobile businesses. Moreover the system will divide the vehicles shops into different categories and it will provide the facility for the shop owner to register their shops according to their nature of businesses.

OneMarket is developed in react native which is JavaScript framework. Mongo db is used for storing text data and Cloudinary for media files. All the shops which are registered in OneMarket will be shown in real-time.

FINAL APPROVAL CERTIFICATE

It is certified that project proposal titled ‘**ONEMARKET**’ submitted by **Awais-ur-Rehman, Muhammad Ibrahim & Ikram Aslam** for the partial fulfilment of the requirement of “**Bachelor’s Degree in Software Engineering**” is approved.

COMMITTEE

HOD Software Engineering:

Dr. Muzafar Khan

Signature: _____

Head Project Committee:

Naveed Ahamad

Signature: _____

Project Supervisor:

Ms. Fatima Gillani

Signature: _____

Dated: 13th December, 2022

UNDERTAKING

We hereby undertake that we are marking this group for our final year project by our own choice. We entirely must complete this project in time. In case of failure or underperforming, neither the university nor the supervisor will be held responsible for unsatisfactory performance.

Group Members

Signature

1. Awais-ur-Rehman

2. Muhammad Ibrahim

3. Ikram Aslam

DECLARATION

We hereby declare that our dissertation is entirely our work and genuine /original. We understand that in case of discovery of any PLAGIARISM at any stage, our group will be assigned an F (FAIL) grade and it may result in withdrawal of our Bachelor's degree.

Group Member

Signature

1. Ikram Aslam

2. Awais ur Rehman

3. Muhammad Ibrahim

TABLE OF CONTENTS

Chapter	Page
CHAPTER 1 INTRODUCTION	1
1.1 INTRODUCTION	2
1.2 MOTIVATION	2
1.3 PROBLEM STATEMENT	2
1.4 ONE MARKET MOBILE APPLICATION.....	3
1.5 GOALS AND OBJECTIVES	3
1.6 DEVELOPED SYSTEM FEATURES	4
1.6.1 REGISTER SHOP.....	4
1.6.2 EDIT SHOP.....	4
1.6.3 ADD PRODUCT	4
1.6.4 SEARCH SHOP	4
1.6.5 SEARCH PRODUCT.....	4
1.6.6 CHAT	4
1.6.7 ADMIN CONSOLE.....	4
1.7 SCOPE OF THE STUDY	5
1.7.1 ANDROID APPLICATION.....	5
1.7.2 REACT NATIVE.....	5
1.7.3 MONGODB AND CLOUDINARY	5
1.8 PROCESS MODEL	5
1.9 NATURE OF THE PROJECT	6
1.10 OVERVIEW OF THE REPORT	6
CHAPTER 2 BACKGROUND & EXISTING WORK.....	7
2.1 INTRODUCTION	8
2.2 IMPORTANT CONSTRUCTS OF THE APPLICATION.....	8
2.3 ANDROID APP DEVELOPMENT	8
2.4 CLIENT SERVER ARCHITECTURE.....	8
2.5 REACT NATIVE.....	8
2.6 MONGODB	8
2.7 EXISTING SYSTEMS	9
2.8 COMPARISON OF EXISTING SYSTEMS.....	9

2.9 SUMMARY	10
CHAPTER 3 REQUIREMENT SPECIFICATION.....	11
3.1 INTRODUCTION	12
3.2 INTERFACE REQUIREMENT	12
3.2.1 HARDWARE INTERFACE REQUIREMENT	12
3.2.2 SOFTWARE INTERFACE REQUIREMENT	13
3.3 FUNCTIONAL REQUIREMENTS	13
3.3.1 REGISTER SHOP.....	13
3.3.2 LOGIN	13
3.3.3 HOME	14
3.3.4 SEARCH	14
3.4 NON-FUNCTIONAL REQUIREMENTS	14
3.4.1 MAINTAINABILITY	14
3.4.2 PERFORMANCE	14
3.4.3 SECURITY	14
3.4.4 USABILITY	15
3.4.5 FLEXIBILITY	15
3.5 USECASE DIAGRAM.....	15
3.5.1 REGISTER SHOP USECASE DESCRIPTION	16
3.5.2 LOGIN USECASE DESCRIPTION.....	17
3.5.3 VIEW PRODUCT USECASE DESCRIPTION.....	18
3.5.4 VIEW SHOP USECASE DESCRIPTION	19
3.5.5 SEARCH SHOP USECASE DESCRIPTION	20
3.5.6 EDIT SHOP USECASE DESCRIPTION	21
3.5.7 DELETE SHOP USECASE DESCRIPTION.....	22
3.5.8 ADD PRODUCT USECASE DESCRIPTION	23
3.5.9 EDIT PRODUCTS USECASE DESCRIPTION	24
3.5.10 DELETE PRODUCT USECASE DESCRIPTION	25
3.5.11 DO CHAT USECASE DESCRIPTION	26
3.5.12 DATABASE REQUIREMENT	27
3.6 PROJECT FEASIBILITY	27
3.7 TECHNICAL FEASIBILITY	27
3.8 OPERATIONAL FEASIBILITY.....	27
3.9 LEGAL & ETHICAL FEASIBILITY	27

3.10 SUMMARY	28
CHAPTER 4 SYSTEM MODELLING	29
4.1 INTRODUCTION	30
4.2 SYSTEM DESIGN	30
4.3 DESIGN APPROACH	30
4.4 INTERFACE DESIGN	30
4.5 HIGH FIDELITY PROTOTYPE.....	31
4.6 VIEW MODEL	41
4.6.1 LOGICAL VIEW	41
4.6.2 PROCESS VIEW DIAGRAM	41
4.6.3 DEVELOPMENT VIEW	59
4.6.4 PHYSICAL VIEW	60
4.7 SUMMARY	61
CHAPTER 5 IMPLIMENTATION	62
5.1 INTRODUCTION.....	63
5.2 MODULES	63
5.2.1 HOME MODULE	63
5.2.2 SHOP MODULE	63
5.2.3 VIEW ITEM MODULE.....	63
5.2.4 ALL SHOP MODULE	63
5.2.5 CUSTOMIZE SHOP MODULE	63
5.2.6 SIGN IN MODULE.....	63
5.2.7 SIGN UP MODULE	64
5.2.8 ADMIN MODULE.....	64
5.2.9 LIBRARIES	64
5.2.10 DATABASE.....	64
5.3 FRAMEWORKS	65
5.3.1 REACT NATIVE.....	65
5.3.2 EXPRESS JS	65
5.4 EXTERNAL API'S	66
CHAPTER 6 TESTING, ANALYSIS & VALIDATION.....	67
6.1 INTRODUCTION	68
6.2 TEST METHODOLOGY	68
6.3 TEST BED	69

6.4 SYSTEM TEST CASE	69
6.5 TEST CASES.....	69
6.5.1 REGISTER TEST CASE	70
6.5.2 LOGIN TEST CASE.....	71
6.5.3 SEARCH TEST CASE	72
6.5.4 ADD SHOP TEST CASE.....	73
6.5.5 EDIT SHOP TEST CASE	74
6.5.6 DELETE SHOP TEST CASE.....	75
6.5.7 ADD PRODUCT TEST CASE	76
6.5.8 EDIT PRODUCT TEST CASE.....	77
6.5.9 DELETE PRODUCT TEST CASE.	78
6.5.10 CHATTING TEST CASE	79
6.5.11 ADMIN TEST CASE.....	80
6.6 SUMMERY	81
CHAPTER 7 CONCLUSION & FUTURE WORK.....	82
7.1 OVERVIEW OF CHAPTER.....	83
7.2 SYSTEM OVERVIEW.....	83
7.3 MILESTONES ACHIEVED	83
7.4 LIMITATIONS	83
7.5 FUTURE WORK.....	84
7.6 SUMMARY	84
7.6 SUMMARY	84
REFERENCES.....	82

LIST OF FIGURES

Figure	Page
3.1 Usecase Diagram of System	15
4.1: Splash Screen Prototype	31
4.2: Login Screen Prototype	32
4.3: Create Account Prototype	33
4.4: Home Screen Prototype	34
4.5: Edit Shop Prototype	35
4.6: Shop Prototype.....	36
4.7: Search Product Prototype.....	37
4.8: Search Shop Prototype.....	38
4.9: Selected Product Prototype.....	39
4.10: Admin Page Prototype	40
4.11: Activity Diagram	42
4.12: Chat State Machine	43
4.13: Login State Machine.....	44
4.14: Create Account State Machine	45
4.15: Manage Shop State Machine	46
4.16: Manage Category State Machine	47
4.17: Manage Product State Machine	48
4.18: View State Machine.....	49
4.19: Search State Machine.....	50
4.20: Login Sequence.....	51
4.21: Registration Sequence.....	52
4.22: Manage Shop Sequence	53
4.23: Manage shop by admin sequence	54
4.24: Manage Product Sequence.....	55
4.25: Chat Sequence.....	56
4.26: Search Shop Sequence	57
4.27: Search Product Sequence.....	58
4.28: Component Diagram.....	59
4.29: Deployment Diagram.....	60

LIST OF TABLES

Table	Page
2.1: Comparison of Existing System and Proposed System Features	9
3.1: Create Account Usecase	16
3.2: Login Usecase.....	17
3.3: View Usecase.....	18
3.4: View Usecase.....	19
3.5: Search Usecase	20
3.6: Edit Shop Usecase	21
3.7: Delete Shop Usecase.....	22
3.8: Add product Usecase	23
3.9: Edit Product Usecase	24
3.10: Delete Product Usecase	25
3.11: Chat Usecase.....	26
6.1: Registration Test Case	70
6.2: Login Test Case	71
6.3: Search Test Case	72
6.4: Add Shop Test Case.....	73
6.5: Edit Shop Test Case.....	74
6.6: Delete Shop Test Case	75
6.7: Add Product Test Case	76
6.8: Edit Product Test Case.....	77
6.9: Delete Product Test Case	78
6.11: Chat Test Case	79
6.12: Admin Page Test Case.....	80

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The ever-growing trends amongst fanatics of cars have given a boost to the car-related industry. People are becoming fond of upgrading their automobiles with high quality products thus getting the added facilities out of their cars. While some car enthusiast love their car to an extent that they keep on looking for cleaning products to get the shiny look out of their cars. Many are concerned about their car's security tool. There is no such place to get all the products you may need for your car. Many online shops provide the facility to sell or buy vehicle accessories. However, there is no such system that connects the user to local shops in the physical existing markets

OneMarket is an online car accessory store, car care and auto part store in Pakistan that connects vehicle users to local shops of Rawalpindi and Islamabad. Our Proposed system will be different from all of these available systems. The proposed system will allow the local vehicles shop owner to register their shop and create an online presence for their shop. For now, we have selected only two locations Rawalpindi and Islamabad. All the shops in that area will be able to register their shops through our system. There will be different categories of shops according to the nature of their business-like Spare Parts Shop, Tyre Shop, and Decoration Shops etc. Every category of the shop will be able to register their shop in their specific category. They will create the account in the system. After that, they will be able to make the profile for their shop according to the nature of their business.

A shop owner will also be able to upload their product. The system will help the people of twin cities who have vehicles because they will be able to search and explore the whole market in one place. They will be able to list down all the shops in the specific markets, their categories, their info and products. The system will also help the shop owner too because in this way there are chances that their business will grow much faster because of the online presence.

1.2 MOTIVATION

We aim to build a system that will ensure, all accessible physical markets are present online. Such a system doesn't exist, particularly not for the twin cities.

1.3 PROBLEM STATEMENT

In Pakistan, almost in every city, there are specific markets for cars where there are a lot of shops of different categories. Some shops lie under the category of Internal parts and some

are under External parts. There are big giant stores where that offer to buy vehicle accessories online but there is no such platform where the local shops register their shops and provide their services and products to the customer. So when the customer goes to the market, they don't have any idea about the product quality and price. Even they didn't know that which products should be bought from which shops.

The proposed system will allow the customers to explore the whole vehicle markets in Rawalpindi and Islamabad and then according to what they want to buy or what services they want to avail of, select specific shops and list down all their products and can directly contact the shop owner and ask anything about car accessories or products they want to buy for their car. In this way, they can learn all the car accessories and other necessary information about the product like product price, other information about product etc.

1.4 ONE MARKET MOBILE APPLICATION

There exists some application that is related to cars but they do not provide all the features that our system contain. The system will help to reduce the limitation and drawbacks of that system. One Market App will provide all the products related to the car on a single platform for people who want to buy parts of the car. The system will help to reduce the time and effort users that they spend on finding the product or finding a specific shop. OneMarket is a Mobile based application that can be easily accessible by any android user.

The system will provide all the features for the Market which include Shops, Products and Shops Categories. Features included are: list down all the shops in specific location, search product, search shop, register shops; this feature allows the shop's owner to register their shops. After registration they can add product to their shop. The chatting feature is also included in this system that allows the customer to chat with the shop owner about the product.

1.5 GOALS AND OBJECTIVES

The goal and objective of the proposed system is to develop a car related app for people which will helps them to find the shops and products for their vehicles. It will provide the features for shops owner to register their shops. It is a Mobile based application and it would help many users from different filed of life. This app will save the time of people that they waste on finding the shops.

1.6 DEVELOPED SYSTEM FEATURES

Features of the developed system are as follows:

1.6.1 REGISTER SHOP

OneMarket provide this for the shop owner. So that the shop owner can register their shops. Shop owner will have to give their details and then shop category and location.

1.6.2 EDIT SHOP

After registration shop owner can edit their shop detail. Shop detail include shop name, description, profile photo and cover photo.

1.6.3 ADD PRODUCT

This feature is also for the shop owner. After registration and shop detail, the shop owner will be able to add product to their shop. Maximum user can add 10 products for now.

1.6.4 SEARCH SHOP

This feature is for every user. There are two filter available to search shops. These features are “location” and “category”. User can enable anyone or both features to search shops. After the selection of the category of shop then there will be a list of all the shops which lie under that specific category. For example, if the customer selects the tyre-shop category, then all the tyre shops will be listed in that location.

1.6.5 SEARCH PRODUCT

This feature is for every user also. User can enter the name of the product and the OneMarket will list down all the products related to that search term.

1.6.6 CHAT

When the user selects a specific product then they will be able to see the product detail. Along with the detail there will be icon of WhatsApp [12]. By clicking on that icon user can chat with the shop owner on the WhatsApp.

1.6.7 ADMIN CONSOLE

This module is for admin. Admin can login to OneMarket using their credentials and from there they can see all the shops registered into the OneMarket. They can also delete the shop and visit them.

1.7 SCOPE OF THE STUDY

We are targeting to provide the online presence of physical markets in Rawalpindi and Islamabad.

1.7.1 ANDROID APPLICATION

OneMarket is an android based application and it will be available for any user to download and install it.

1.7.2 REACT NATIVE

We used React Native [2] to build OneMarket which is a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android. It is based on React, Facebook's JavaScript library for building user interfaces, but instead of targeting the browser, it targets mobile platforms. Most of the code you write can be shared between platforms, React Native makes it easy to simultaneously develop for both Android and iOS. React Native currently supports both iOS and Android, and has the potential to expand to future platforms as well.

1.7.3 MONGODB AND CLOUDINARY

For backend we used MongoDB [7] which is a document database with the scalability and flexibility that you want with the querying and indexing that you need. All the media file are stored in Cloudinary [8] which is cloud-based storage provider. The remaining data which is in the text form are stored in MongoDB Atlas.

1.8 PROCESS MODEL

The development of this project is based on the Incremental model. With the help of this model, the project begins with the requirement specification of the system that defines all the functional and non-functional requirements. Then based on the defined requirements, designs of the application are designed with help of various UML models. Relying on the UML designs, different components and modules of the project are developed one by one and are then integrated into the previously developed module. On the completion of the project, it is tested for its accuracy, time consumption in generating the results and the generation of results on how the system is going to react and inform the user.

1.9 NATURE OF THE PROJECT

This project is an Android based application that is developed using React Native and MongoDB.

1.10 OVERVIEW OF THE REPORT

This chapter is about the problem statement, introduction, developed system purpose, developed system functionalities and the possibility of the developed system. The approaches and tools that are used to develop OneMarket are discussed in the following chapter. The second chapter is about the reason behind the idea. The chapter describes an existing system and its functionalities, and the flaws in the developed system, its assessment, comparison with other existing systems and its limitation are also conferred. In the third chapter, we have discussed the modeling approaches for the system. First of all, we develop use cases for the system, then we design a use case diagram and then finally we made sequence diagrams for the process view of the system and enable the reader to get a better understanding of the system. In a further chapter, the system requirement specifications are described. It covers all the requirement of the system which includes interface, functional, non-functional, database and resource requirement. It highlights the feasibility of the system by describing the use cases with a diagram and detailed description. Design approaches of the system are also discussed in detail in going further with the document. The interface design of the system, 4+1 view models, logical view, physical view, process view and the development view diagram.

CHAPTER 2

BACKGROUND & EXISTING WORK

2.1 INTRODUCTION

In this chapter, we give a detailed description of the existing system their features and limitations, and how our proposed system overcomes the existing system limitations. Important constructs of OneMarket, in which tools and technology this application builds.

2.2 IMPORTANT CONSTRUCTS OF THE APPLICATION

One Market is developed using the React Native Platform. In simple words, this app allows the user to see car-related parts and allows the shop owner to register their shops. Important constructs of this app are discussed here:

2.3 ANDROID APP DEVELOPMENT

Android App Development is a process in which applications are developed by using different platforms like VS Code. For development React Native is used. It would be helpful for many users from different filed of life to view the car-related product and shops.

2.4 CLIENT SERVER ARCHITECTURE

The client-server architecture is developed by using React Native and MongoDB. MongoDB is a document base database that helps to establish a client-server architecture for the One Market.

2.5 REACT NATIVE

React Native is a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android. It is based on React, Facebook's JavaScript library for building user interfaces, but instead of targeting the browser, it targets mobile platforms. Most of the code you write can be shared between platforms, React Native makes it easy to simultaneously develop for both Android and iOS. React Native currently supports both iOS and Android, and has the potential to expand to future platforms as well.

2.6 MONGODB

MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need. MongoDB's document model is simple for developers to learn and use, while still providing all the capabilities needed to meet the most complex requirements at any scale.

2.7 EXISTING SYSTEMS

A lot of work has been done in this area. Lots of sites and apps are available that list down the parts or accessories of vehicles where the user can buy or sell auto accessories. Here are some of the apps and sites that are related to the OneMarket are explained below: These apps or sites include

- Sehgal Motors
- AutoX
- AutoJin
- Daraz

2.8 COMPARISON OF EXISTING SYSTEMS

The table given below shows the comparison between the existing and our system.

Table 2.1: Comparison of Existing System and Proposed System Features

Features	Sehgal Motors [6]	AutoX [4]	AutoJin[3]	Daraz[5]	OneMarket
Car Accessories	Yes	Yes	Yes	Yes	Yes
Local vehicle Shops Reg	No	No	No	No	Yes
Chat with Shops	Yes	No	No	Yes	Yes
List down shops in specific category	No	No	No	No	Yes
Search Shop	No	No	No	No	Yes
Buy product Online	Yes	Yes	Yes	Yes	No
Search Product	Yes	Yes	Yes	Yes	Yes
Add Product	Yes	Yes	Yes	Yes	Yes

2.9 SUMMARY

This chapter covers existing systems' features and limitations and compares existing systems and proposed systems. In the end, a comparison matrix is provided to show why this system is needed even though there are already few similar systems available.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 INTRODUCTION

In this chapter, the Requirement Specifications of an OneMarket are discussed in detail. This chapter provides a detailed overview of “ONE Market” that includes the system’s flow diagram, interface requirements, functional requirements, non-functional requirements, use case, resource requirements, database requirements, and project feasibility. The requirement specification chapter provides complete details of OneMarket, its features and the importance of implementing the system. These details show the requirements to implement the system and how the system will be implemented. Use case diagram explains the functionalities of the system and these functionalities will help users with the great and most satisfying online shopping experience.

3.2 INTERFACE REQUIREMENT

Interface requirements are classified as those requirements which need to be met by the system. By interface requirements, hardware and software component communicate efficiently. The system requirements where the interface is involved between all the system components are the essential part and are described in detail. Also, the description should be easy to understand. Interfacing is an important part of a system to provide smooth interaction between different components of the system and its interaction with other systems.

OneMarket is an online application and is consists of software components majorly, so it is important to design the interface of these components in such a way that makes the application easy to interact with for better performance. The requirements for software and hardware interface are given below:

3.2.1 HARDWARE INTERFACE REQUIREMENT

The hardware interface requirements that are essential to run OneMarket application successfully are:

3.2.1.1 MOBILE PHONE

OneMarket app is a Mobile Application that runs on Android. An android smartphone that has at least 1GB RAM is required to run this application. An active internet connection is required to use this Application.

3.2.1.2 PC (LAPTOP OR DESKTOP)

A laptop/desktop with specifications (RAM 4GB or higher, dual-core 2.5 GHz processor or higher, minimum 200GB hard drive, and window 8.1 or above) is required to develop the system.

3.2.2 SOFTWARE INTERFACE REQUIREMENT

As OneMarket is an online Mobile application therefore internet connection is essential to run this application.

3.2.2.1 VISUAL STUDIO CODE

For coding editor, we used visual studio code which is a lightweight and powerful source code editor which can runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages and runtimes (such as C++, C#, Java, Python, PHP, Go, .NET). It is also use for the development of the mobile application using react native.

3.2.2.2 REACT NATIVE

React Native is a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android. It is based on React, Facebook's JavaScript library for building user interfaces, but instead of targeting the browser, it targets mobile platforms. Most of the code you write can be shared between platforms, React Native makes it easy to simultaneously develop for both Android and iOS. React Native currently supports both iOS and Android, and has the potential to expand to future platforms as well.

3.3 FUNCTIONAL REQUIREMENTS

3.3.1 REGISTER SHOP

OneMarket provide this functionality for the shop owner. So that the shop owner can register their shops. Shop owner will have to give their details and then the shop category and location.

3.3.2 LOGIN

The OneMarket must have login activity with a field for a username and a password. After login, the user will be able to

- View/modify his shop
- Manage Products.
- Manage Shops

3.3.3 HOME

The OneMarket must have a home screen visible for all the users of the application (whether login or not). The home screen should contain the latest products added to the database.

3.3.4 SEARCH

The OneMarket should have a screen for product search and shop search. In shop search user can apply these filters:

- Location
- Category

3.4 NON-FUNCTIONAL REQUIREMENTS

3.4.1 MAINTAINABILITY

We have created separate directories and files for backend and frontend. We have used all the recommended preference approach to create folder structure and store structure. We used Redux recommended folder and file structure. So that's why our code is maintainable. System features can be edited and new features can be added easily.

3.4.2 PERFORMANCE

Our code is very well optimized and its performance is very good. We used mongo db for backend which is far better in performance as compare to firebase. MongoDB is a more robust document database known for high performance and best-in-class security, and has several advantages over Firebase. MongoDB can be operated on premise or in the cloud (using MongoDB Atlas), while Firebase is purely a cloud database service.

3.4.3 SECURITY

Our system is secure. It provides integrity and authentication. Only the shop owner who register their shop can edit their shop and its products. One shop owner can't delete and edit other shops. In the same way only, the admin can delete the shops. So that's why data of the shops are secure.

3.4.4 USABILITY

Our system is simple and easy to use. Its interface is very simple and easy, user can learn all of its functionality within a few minutes.

3.4.5 FLEXIBILITY

Our system is flexible enough to deliver space to increase new features and to grip them suitably.

3.5 USECASE DIAGRAM

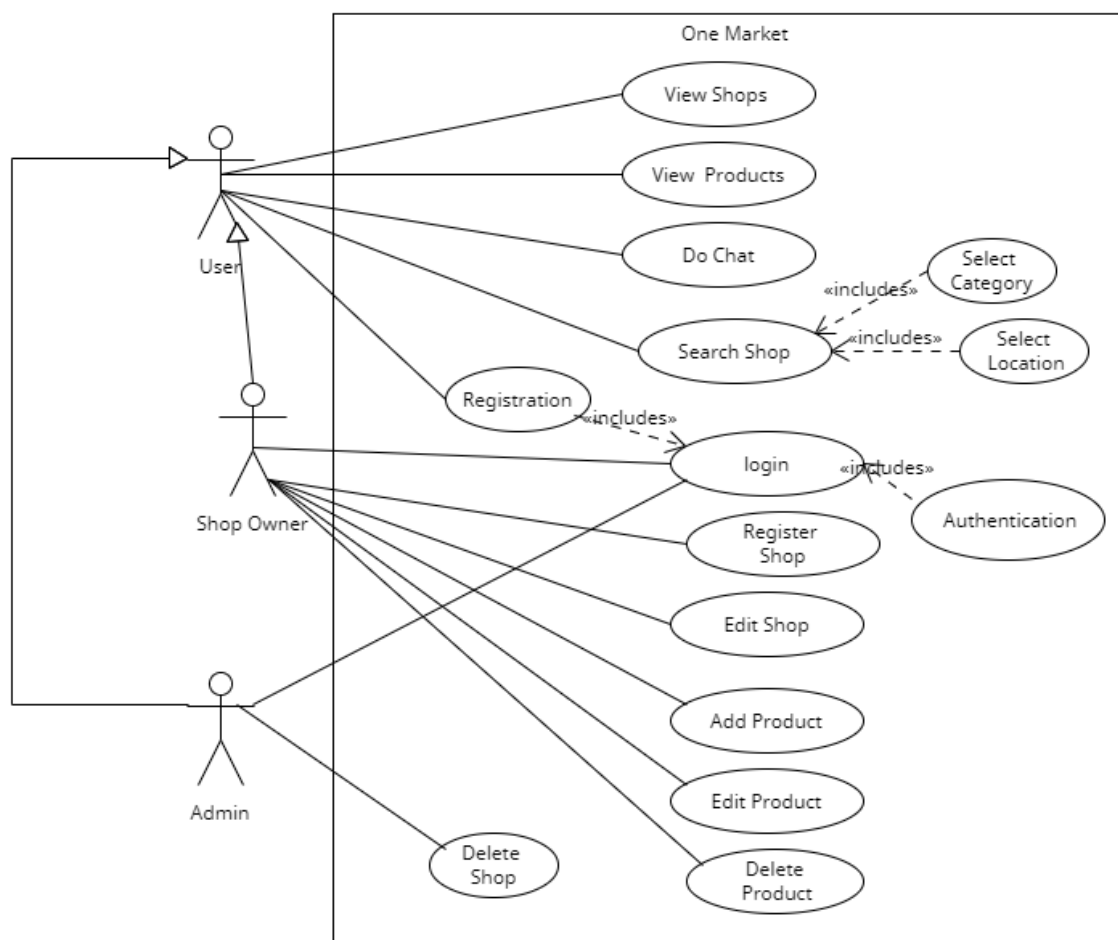


Figure 3.1 Usecase Diagram of System

3.5.1 REGISTER SHOP USECASE DESCRIPTION

Table 3.1: Create Account Usecase

Usecase name	Register Shop		
Usecase ID	UC-01		
Goal	It explains how user registered their shop		
Brief Description	User will provide the basic details to register their shop into system		
Pre-Condition	User must have to open system		
Post Condition	User successfully registered they're into system and log in into the system successfully		
Failed condition	When user leave some field empty or doesn't meet password criteria		
Primary Actor	User		
Dependency	None		
Basic Flow	Steps	Actions	System Response
	1	User starts the system	System will show the home page
	2	User choose to register their shop	System will open the register shop page
	3	User provide the required information and enter submit	System will check if the required fields are set and then store the information into db.
Alternative Flow	Steps	Actions	System Response
	1	User may cancel the register shop option	System will redirect the user to home
	2	User may have some field empty	System will show error

3.5.2 LOGIN USECASE DESCRIPTION

Table 3.2: Login Usecase

Usecase name	Login		
Usecase ID	UC-02		
Goal	This use case will tell how user will login into the system		
Brief Description	User enter the username and password to login to the system		
Pre-Conditions	User must have a shop registered into the system		
Post Conditions	User login successfully		
Failed condition	User unable to login		
Primary Actor	Shop owner, Admin		
Dependency	Include Register Shop Usecase, Authentication Usecase		
Basic Flow	Steps	Actions	System Response
	1	User login into the system	System asks for login information
	2	User enter the login information into the system.	System check and verify the provided information from database
Alternative Flow	Steps	Actions	System Response
	2a	If invalid user name or password is entered into the system	System will not allow the user or to login and redirect to login page again
	2a	User doesn't have an account in the system	System will show the user to create account first to login

3.5.3 VIEW PRODUCT USECASE DESCRIPTION

Table 3.3: View Product Usecase

Usecase Name	View Product		
Usecase ID	UC-03		
Goal	This use case will tell the user how to view products		
Brief Description	User will be able to view all the products available in the system		
Pre-Condition	User must have to open the system		
Post Condition	Products displayed successfully		
Failed condition end	None		
Primary actor	User		
Dependency	None		
Basic Flow	Steps	Actions	System Response
	1	When the user opens the system, random products will be shown. The user can select and view products	System will show all the products of shops in the system

3.5.4 VIEW SHOP USECASE DESCRIPTION

Table 3.4: View Shop Usecase

Usecase Name	View Shop		
Usecase ID	UC-03		
Goal	This use case will tell the user how to view shop		
Brief Description	User will be able to view all the shops available in the system		
Pre-Condition	User must have to open the system		
Post Condition	Shops displayed successfully		
Failed condition end	None		
Primary actor	User		
Dependency	None		
Basic Flow	Steps	Actions	System Response
	1	When the user opens the system, random shops will be shown. The user can select and view specific shop	System will show all the shops and or specific shop in the system

3.5.5 SEARCH SHOP USECASE DESCRIPTION

Table 3.5: Search Usecase

Usecase_Name	Search Shop		
Usecase ID	UC-04		
Goal	This use case will tell how the user will search shops		
Brief Description	User will be able to search a specific category of shop and in specific location by applying filter		
Pre-Condition	User must have to open the system		
Post Condition	Filtered shops results displayed successfully		
Failed condition end	None		
Primary actor	User		
Dependency	None		
Basic Flow	Steps	Actions	System Response
	1	The user open the system and search the shop or products by applying filter	System will show the filtered shops or products
Alternative Flow	Steps	Actions	System Response
	1	When the searched shops or products is not available	System will display the empty result message

3.5.6 EDIT SHOP USECASE DESCRIPTION

Table 3.6: Edit Shop Usecase

Usecase_Name	Edit Shop		
Usecase ID	UC-06		
Goal	User can edit their shop		
Brief Description	This use case will let the user edit shop their information		
Pre-Condition	User must have a registered shop in the system		
Post Condition	Shop edited successfully		
Failed condition	When user doesn't have shop in the system		
Primary actor	Shop owner		
Dependency	Registered Shop Usecase		
Basic Flow	Steps	Actions	System Response
	1	The user login to the system by giving his/her credentials	The system will verify the credentials
	2	The user will then edit their shop by updating the details for the shop	The system will update the shop information
Alternative Flow	Steps	Actions	System Response
	1a	Username or password is incorrect	System will redirect the login page
	1b	User doesn't have shop registered	System will show to registered shop first

3.5.7 DELETE SHOP USECASE DESCRIPTION

Table 3.7: Delete Shop Usecase

Usecase_Name	Delete Shop		
Usecase ID	UC-07		
Goal	Admin can delete user's shop		
Brief Description	Admin can delete the registered shop in the system		
Pre-Condition	Shop must be registered in the system		
Post Condition	Shop deleted successfully		
Failed condition end	When user doesn't have shop registered in the system		
Primary actor	Admin		
Dependency	Registered Shop Usecase		
Basic Flow	Steps	Actions	System Response
	1	Admin login to the system by giving his/her credentials	The system will verify the credentials
	2	Admin will then delete the shop	The system will check if the user have shop registered in the system and then delete it
Alternative Flow	Steps	Actions	System Response
	1a	Username or password is incorrect	System will redirect the login page
	1b	User doesn't have an account	System will show to create account notification

3.5.8 ADD PRODUCT USECASE DESCRIPTION

Table 3.8: Add product Usecase

Usecase_Name	Add Product		
Usecase ID	UC-08		
Goal	To add product		
Brief Description	User can add products into their shop		
Pre-Condition	User must have a shop for adding the product		
Post Condition	Products are added successfully		
Failed condition end	When max limit reached		
Primary actor	Shop owner		
Dependency	Registered Shop Usecase		
Basic Flow	Steps	Actions	System Response
	1	The user login to the system by giving his/her credentials	The system will verify the credentials
	2	The user will add products into the their shop by giving details for the product	The system will add the product into system and displayed products added successfully
Alternative Flow	Steps	Actions	System Response
	1a	Username or password is incorrect	System will redirect the login page
	1b	User doesn't have a shop registered	System will tell the user to registered the shop first
	2	When the maximum limit of product addition reached	System will tell the user maximum limit is reached.

3.5.9 EDIT PRODUCTS USECASE DESCRIPTION

Table 3.9: Edit Product Usecase

Usecase Name	Edit Products		
Usecase ID	UC-09		
Goal	To edit the products		
Brief Description	User can edit the description of the products which is already available into the shop		
Pre-Condition	User must have some products into the shop		
Post Condition	Products updated successfully		
Failed condition end	When user doesn't have any products		
Primary actor	Shop owner		
Dependency	Add Products Usecase		
Basic Flow	Steps	Actions	System Response
	1	The user login to the system by giving his/her credentials	The system will verify the credentials
	2	The user will then edit their product	The system will edit the product and displayed product edited successfully
Alternative Flow	Steps	Actions	System Response
	1a	Username or password is incorrect	System will redirect the login page
	1b	User doesn't have a shop registered	System will show to registered shop first
	2b	User doesn't have any products in their shop portfolio	Prompt the user to add the products first

3.5.10 DELETE PRODUCT USECASE DESCRIPTION

Table 3.10: Delete Product Usecase

Usecase_Name	Delete Product		
Usecase ID	UC-10		
Goal	To delete the products which are available in the shop		
Brief Description	User can delete their products		
Pre-Condition	User must have products to delete		
Post Condition	Products deleted successfully		
Failed condition end	When user doesn't have any products		
Primary actor	Shop owner		
Dependency	Add products use case		
Basic Flow	Steps	Actions	System Response
	1	The user login to the system by giving his/her credentials	The system will verify the credentials
	2	The user will then delete their products	The system will delete the product and displayed product deleted successfully
Alternative Flow	Steps	Actions	System Response
	1a	Username or password is incorrect	System will redirect the login page
	1b	User doesn't have shop registered	System will show to registered shop first
	2b	User doesn't have any products in their shop portfolio	Prompt the user to add the products first

3.5.11 DO CHAT USECASE DESCRIPTION

Table 3.11: Chat Usecase

Usecase_Name	Do Chat		
Usecase ID	UC-12		
Goal	It explains how user will chat with the shop owner		
Brief Description	User will be able to chat with the shop owner		
Pre-Condition	User must have to open the system		
Post Condition	Redirect the user to WhatsApp		
Failed condition end	None		
Primary actor	User		
Dependency	None		
Basic Flow	Steps	Actions	System Response
	1	User starts the system	System will show the home page
	2	User choose the product and then click on the chat option	User will be redirected to WhatsApp

3.5.12 DATABASE REQUIREMENT

A database is used to store the data in an organized collection and then retrieve or access it electronically from an android phone or a computer system. It is used to store and retrieve user data. MongoDB is used in this system.

3.5.12.1 MONGODB

MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need. MongoDB's document model is simple for developers to learn and use, while still providing all the capabilities needed to meet the most complex requirements at any scale.

3.5.12.2 CLOUDINARY

Cloudinary is an end-to-end image- and video-management solution for websites and mobile apps, covering everything from image and video uploads, storage, manipulations, and optimizations to delivery. We used Cloudinary as a storage for media files.

3.6 PROJECT FEASIBILITY

Feasibility study of the project is done to analyze whether the project is feasible within the allocated time and budget.

3.7 TECHNICAL FEASIBILITY

Project OneMarket is a complete android application. The main technologies that are associated with OneMarket are react native and all of its features and functionalities, mongo db, Cloudinary. Each of these technologies are freely available and the technical skills required are manageable. Time limitations of the product development and the ease of implementing using these technologies are synchronized.

Initially we utilize the free space which provide mongo db atlas and Cloudinary, but in the future it will be hosted in a paid cloud storage.

3.8 OPERATIONAL FEASIBILITY

The system targets just three categories of shop which are Decoration, Tyre and Spare Parts. So, the system will work on just those categories.

3.9 LEGAL & ETHICAL FEASIBILITY

The system is legally and ethically feasible as:

- It doesn't break any rule and regulation of state.
- It is purely designed for the assistance of people, so, there's no way that it could harm them.
- Components data is secure and can be used only by the systems components.

3.10 SUMMARY

This chapter gives the entire requirement specification of our system. The whole system working is described as what the system can do. Every feature of the system is explained to reduce the complexity of the system. The chapter also briefly describes the interface requirements including hardware and software interface specifications. This chapter also covers functional and non-functional requirements. Using use case diagrams and descriptions it also discussed to understand the in-depth working of the system. Furthermore, hardware and software resources are also mentioned which are required to develop and test the system successfully and efficiently. The technical, operational, and 3 legal & ethical feasibility of the system is also discussed for a better understanding of the development of the system.

CHAPTER 4

SYSTEM MODELLING

4.1 INTRODUCTION

Design is the first step in the development phase of any engineered system. It is the process of applying various techniques and principles to define a device, a process or a system in sufficient detail to permit its physical realization. The designer's goal is to produce a model or a representation of an entity that will later be built. The process by which the model is developed combines institution and judgment based on experience in building similar entities, a set of principles or heuristics that guide how a model evolves and a process of iteration that ultimately leads to a final design representation. Modeling for a system act as a foundation for creating any system. Different models developed support system modeling and design. In this chapter, we discuss the system model we facilitated by data modeling and using UML diagrams.

4.2 SYSTEM DESIGN

Software design encompasses the significant decisions about the organization of a software system. System design means the process in which interfaces, components, data modeling and architecture are discussed in detail. System architecture shows how a system's components (hardware and software) are laid down. Our system consists of an android application (database server) and an android application.

4.3 DESIGN APPROACH

Generally, there are two basic design approaches in software engineering top to down and bottom-up. In OneMarket, we use a top-down design approach is used to create the app. As it is a significant part of developing a system to show how much system depth can be a module. One market is developed by using a top-down design approach. Top to down approach involves going through all the phases of requirement engineering then designing and in the end Implementation. Collection of requirements was the first phase then splitting or diving those requirements for sub-tasks so they can be enough for sub-systems. Then testing is conducted. In the end, using an incremental approach, we integrated the modules

4.4 INTERFACE DESIGN

Interface design is one the most important part of the engineering process because the user's overall experience depends on how well your interface is designed. No matter how well a system does its functionality but if a user is unable to do a specific task because of a non-

conformed interface design, it will leave a bad impression. User Interface design works with user activities while they are communicating with the framework.

For One market interface is designed on the android platform. Android is the most commonly used operating system around the world so it helps users to understand the interface.

4.5 HIGH FIDELITY PROTOTYPE



Figure 4.1: Splash Screen Prototype

Welcome

 Phone Number

 Password

[Log in](#)

or
Don't have an account

[Registration](#)

Figure 4.2: Login Screen Prototype



Profile **Details**

Account **Details**

Figure 4.3: Create Account Prototype



Search any part here ...



FINDSHOP

Location ▾

Market ▾

Search Now



Arshad Decoration



Imran Spareparts



Usman TyreWorkshop



Inam SeatCover House



Floor Mat

Dustproof Non Skid 5D All Weather Car Mat Car Floor Mats for Model 3

Rs: 1,100



Head Light

Lights ... C6 LED Headlight Bulbs For Any Car And Bike H4

Rs: 2,100

Figure 4.4: Home Screen Prototype



Search registered product here ...

Add Product

Product	Product Name	Update	Delete
	random_name		
	random_name		
	random_name		
	random_name		
	random_name		
	random_name		

Figure 4.5: Edit Shop Prototype



ONEMARKET



Auto Parts Store

Auto parts store , Auto Parts, Mechanic, Sale of card accessories



Search product here ...



Floor Mat

Dustproof Non Skid 5D All Weather
Car Mat Car Floor Mats for Model 3

Rs: 1,100



Head Light

Lights... C6 LED Headlight Bulbs For
Any Car And Bike H4

Rs: 2,100



Floor Mat

Dustproof Non Skid 5D All Weather
Car Mat Car Floor Mats for Model 3

Rs: 1,100

Figure 4.6: Shop Prototype



ONEMARKET



Search registered product here ...



Head Light

Lights ... C6 LED Headlight Bulbs For Any Car And Bike H4

Rs: 2,100



Floor Mat

Dustproof Non Skid 5D All Weather Car Mat Car Floor Mats for Model 3

Rs: 1,100



Floor Mat

Dustproof Non Skid 5D All Weather Car Mat Car Floor Mats for Model 3

Rs: 1,100



Head Light

Lights ... C6 LED Headlight Bulbs For Any Car And Bike H4

Rs: 2,100



Floor Mat

Dustproof Non Skid 5D All Weather Car Mat Car Floor Mats for Model 3

Rs: 1,100

Figure 4.7: Search Product Prototype



ONE MARKET



Search shops



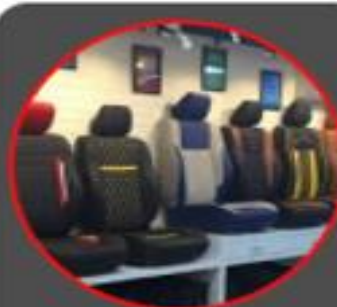
Arshad Decoration



Imran Spareparts



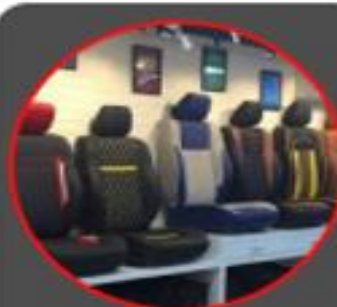
Usman TyreWorkship



Inam SeatCover
House



Usman TyreWorkship



Inam SeatCover
House

Figure 4.8: Search Shop Prototype



ONEMARKET



Rs. 650

Head Light

Usman Auto Decoration

Have a question?



Floor Mat

Dustproof Non-Skid 5D All-Weather
Car Mat Car Floor Mats for Model 3

Rs: 1,100



Floor Mat

Dustproof Non-Skid 5D All-Weather
Car Mat Car Floor Mats for Model 3

Rs: 1,100



Figure 4.9: Selected Product Prototype







Shop Name	Shop Desc	Visit	Delete
shop1	Decoration	▶	
shop2	Tyre	▶	
shop3	Spare Parts	▶	
shop4	Decoration	▶	

Figure 4.10: Admin Page Prototype

4.6 VIEW MODEL

Designing systems nowadays is quite complex as society is becoming more competitive software companies are excluding tasks like requirement gathering and designing resulting in a high number of project failures. This issue introduced software project management which allows engineers to follow each step without wasting any time. 4+1 model allows teams to work on different views so that not all human effort is on one task different teams are assigned to it so that companies can work on multiple tasks at a time and complete them on time. In the 4+1 model, there are 4 different views of the system logical view, process view, physical view and development view and the core of them is that +1 scenarios can be use cases or a small portion of use cases.

4.6.1 LOGICAL VIEW

The Logical view is classified into two main types “class diagram” and “entity-relationship diagram”. But as we are using MongoDB the data is store in the form of JSON or in the form of document so we don’t need entity-relationship diagram and class diagram.

4.6.2 PROCESS VIEW DIAGRAM

The process view depicts the dynamic behavior of the system unlike other views it presents how the system will behave, what are the constraints and how they communicate. In this view, the system’s run time behavior is under focus. Process view of the system includes “Activity diagram”, “State Machine diagram” and “Sequence diagram”.

4.6.2.1 ACTIVITY DIAGRAM

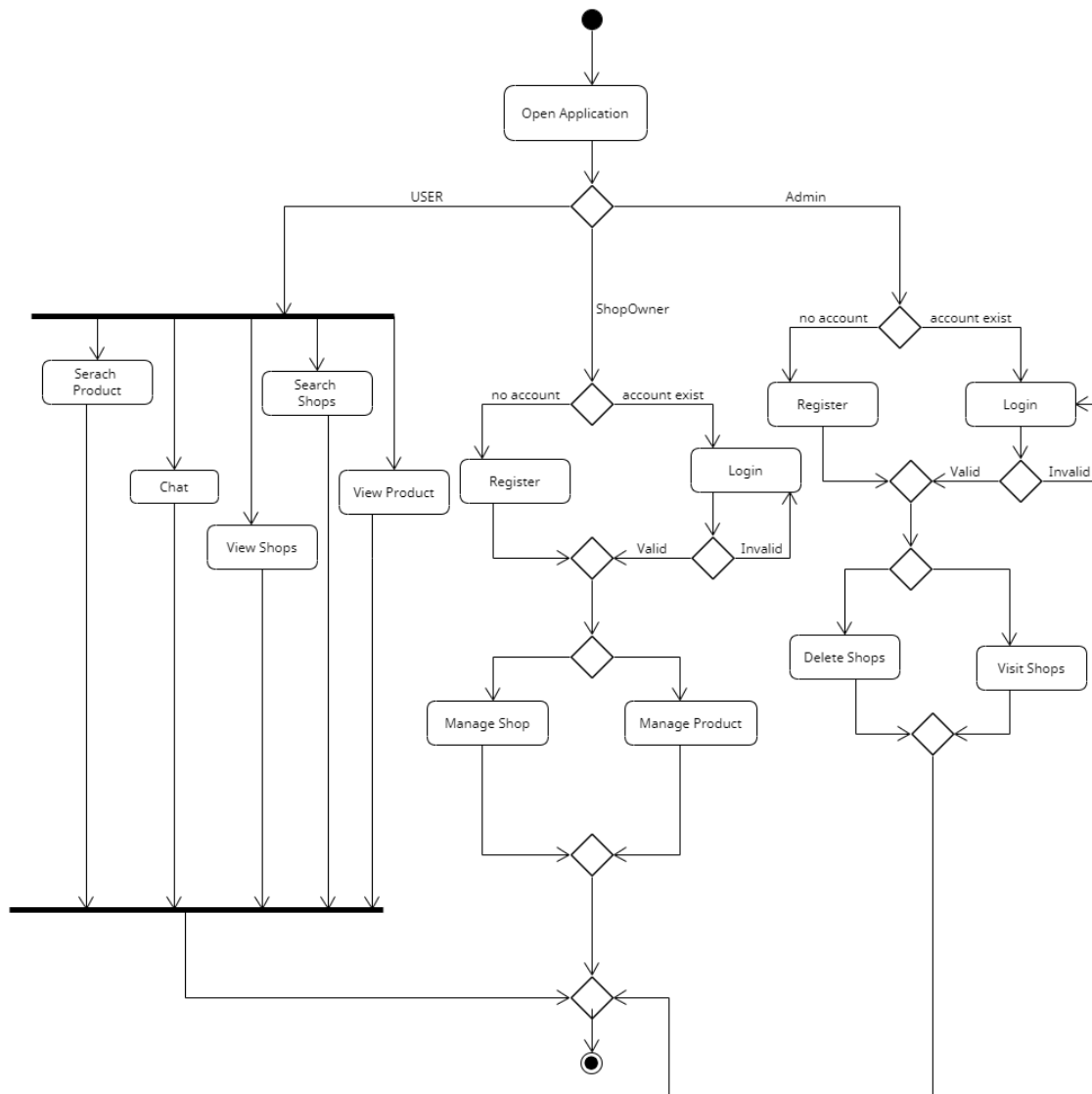


Figure 4.11: Activity Diagram

4.6.2.2 STATE MACHINE DIAGRAM

State machine diagrams represents how system will behave on user's actions or any external action. It shows how systems one state transitions into another upon external affect.

That affect can be trigger, guard or user's action. It is almost similar to activity diagram, but views are completely different. State machine helps developers to deeply understand the working part of system so it can be developed without any flaws.

4.6.2.2.1 CHAT STATE MACHINE DIAGRAM

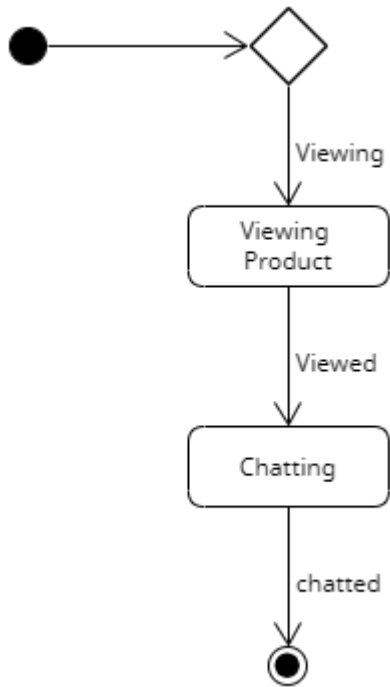


Figure 4.12: Chat State Machine

4.6.2.2.2 LOGIN STATE MACHINE DIAGRAM

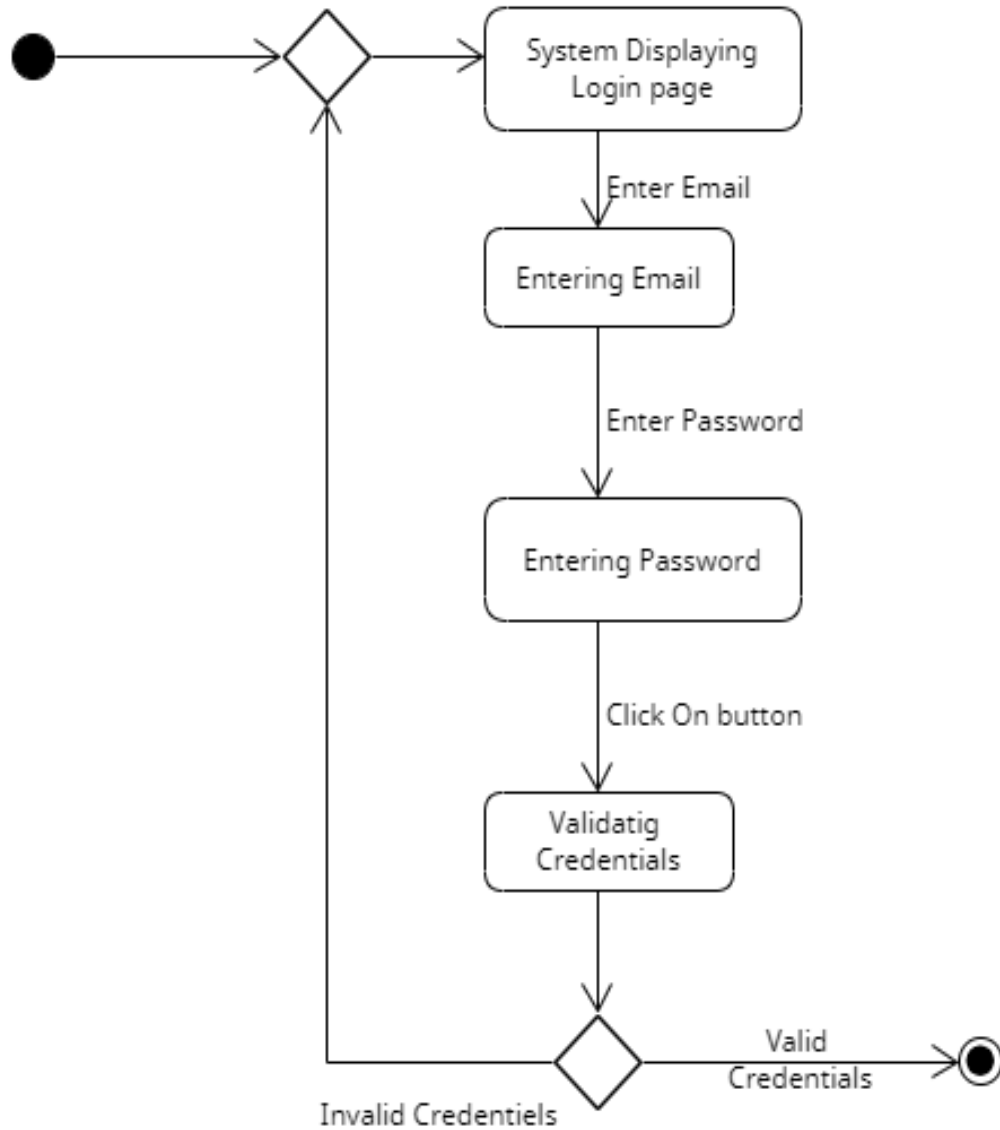


Figure 4.13: Login State Machine

4.6.2.2.3 REGISTRATION STATE MACHINE DIAGRAM

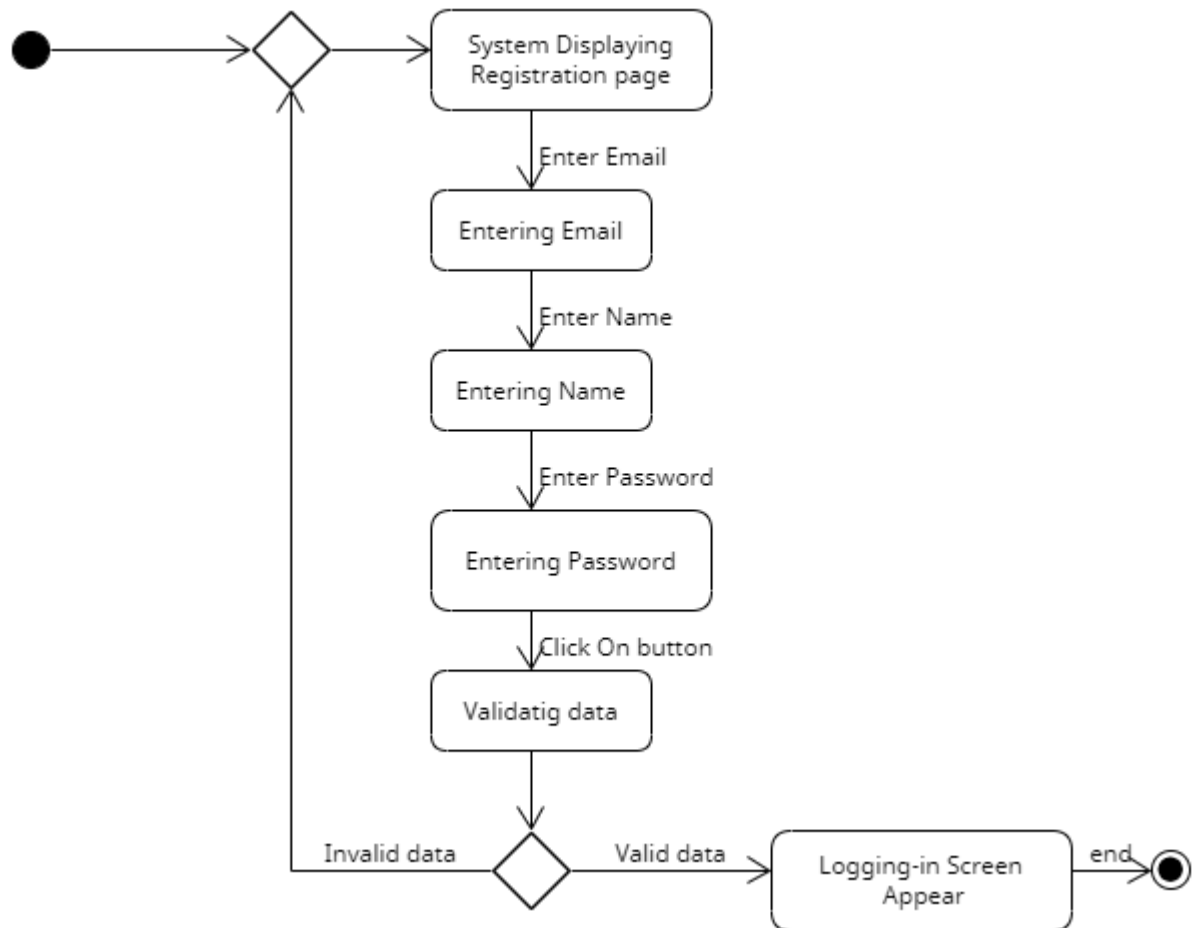


Figure 4.14: Create Account State Machine

4.6.2.2.4 MANAGE SHOP STATE MACHINE DIAGRAM

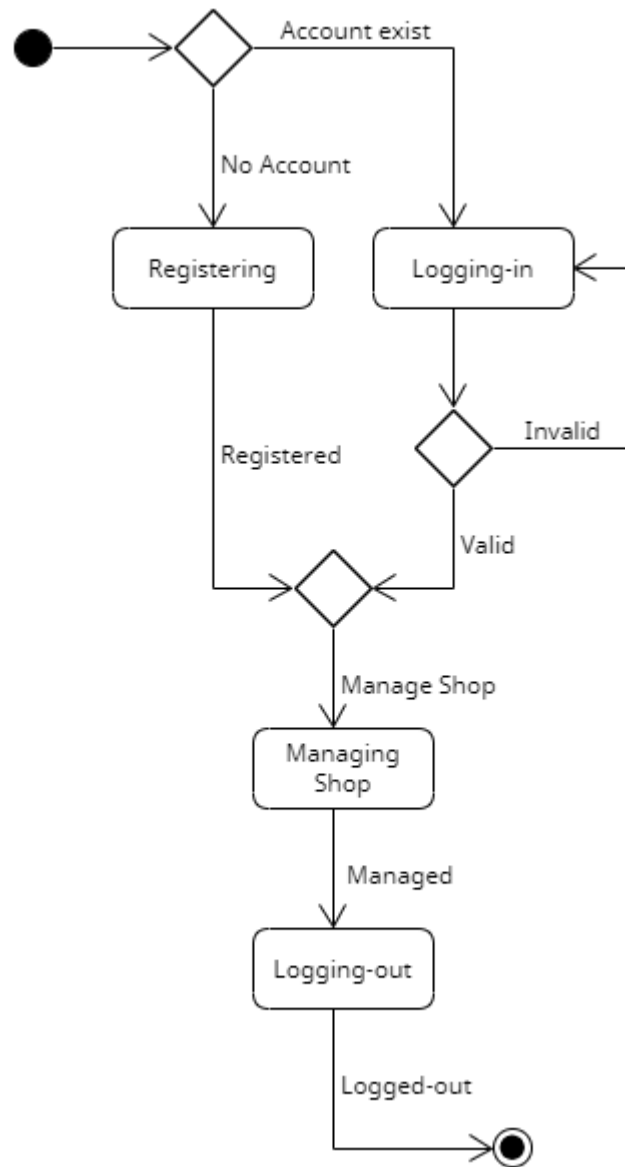


Figure 4.15: Manage Shop State Machine

4.6.2.2.5 MANAGE CATEGORY STATE MACHINE DIAGRAM

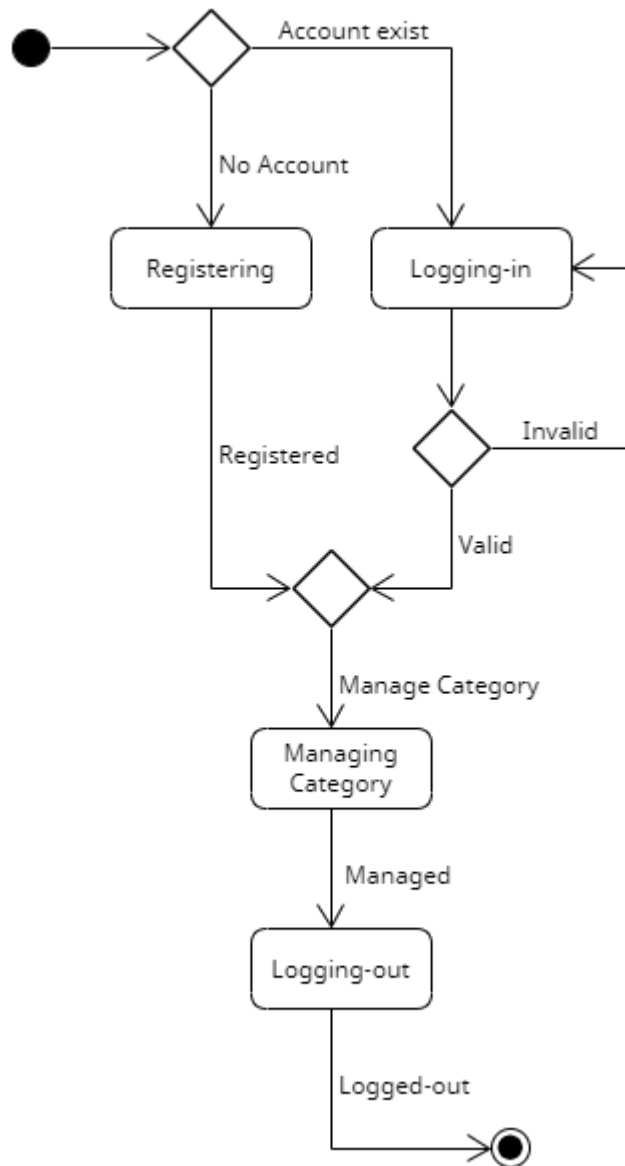


Figure 4.16: Manage Category State Machine

4.6.2.2.6 MANAGE PRODUCT STATE MACHINE DIAGRAM

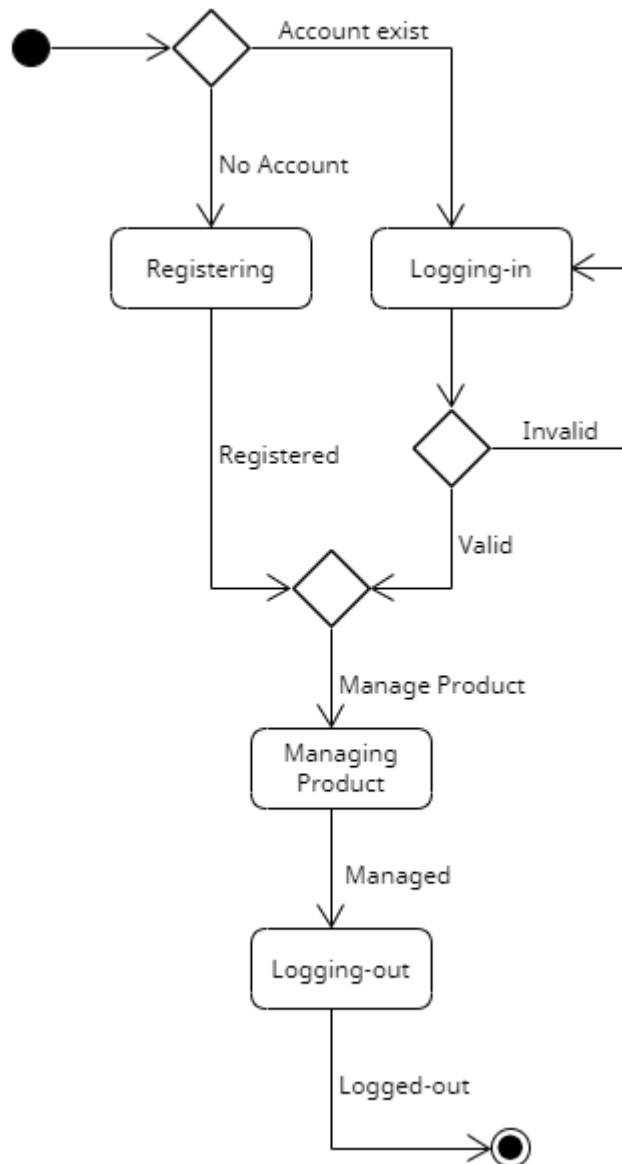


Figure 4.17: Manage Product State Machine

4.6.2.2.7 VIEW STATE MACHINE DIAGRAM

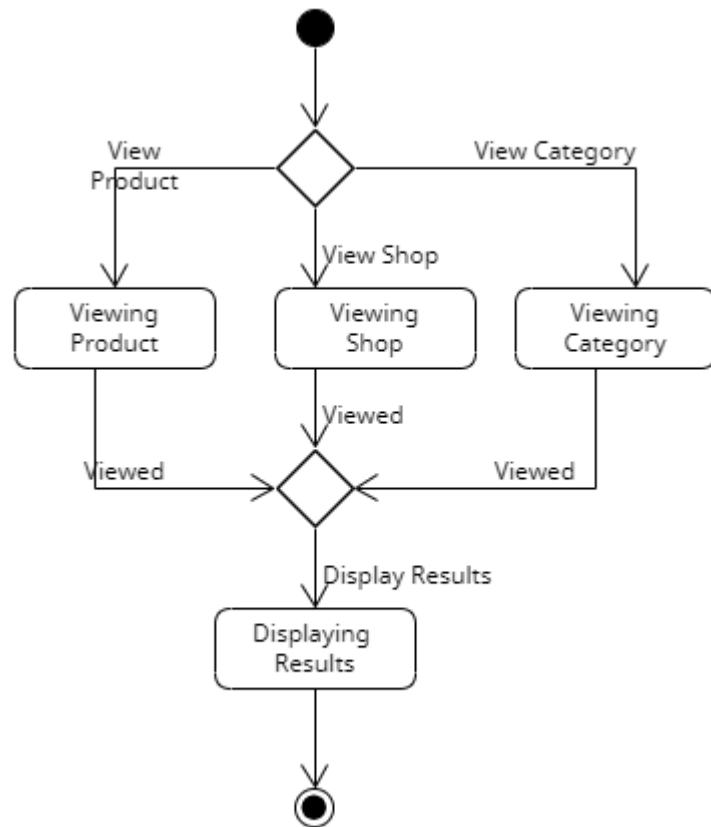


Figure 4.18: View State Machine

4.6.2.2.8 SEARCH STATE MACHINE DIAGRAM

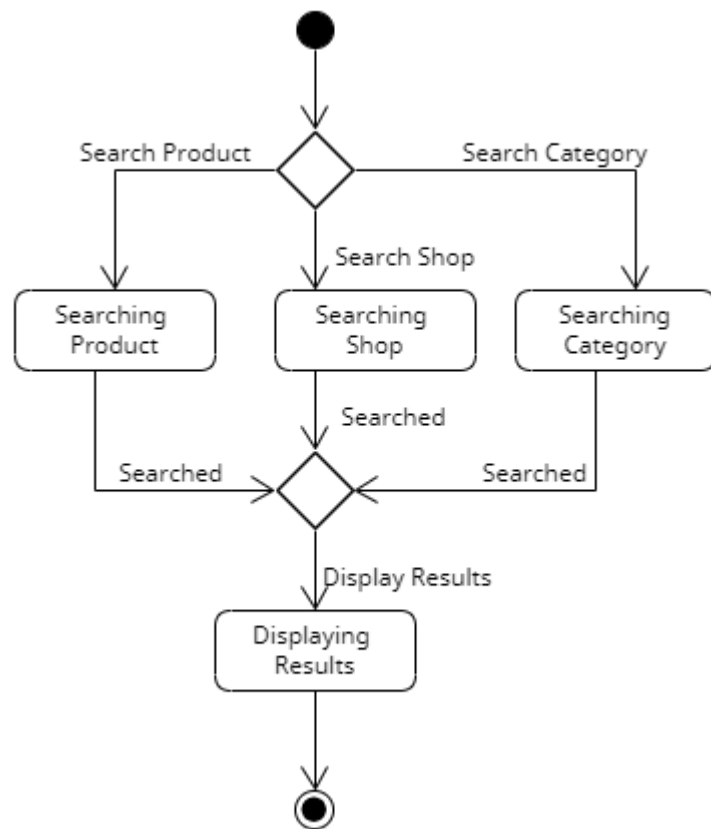


Figure 4.19: Search State Machine

4.6.2.3 SEQUENCE DIAGRAM

A sequence diagram represents the dynamic behavior of the system and shows the interaction of objects with the system in sequential order. It is a graphical representation of the fully dresses use cases. As the name shows, every process is in a sequence. The sequence of interactions of objects with the system enhances the understandability of how actors interact with the system. Sequence diagrams of the system are given below.

4.6.2.3.1 LOGIN SEQUENCE DIAGRAM

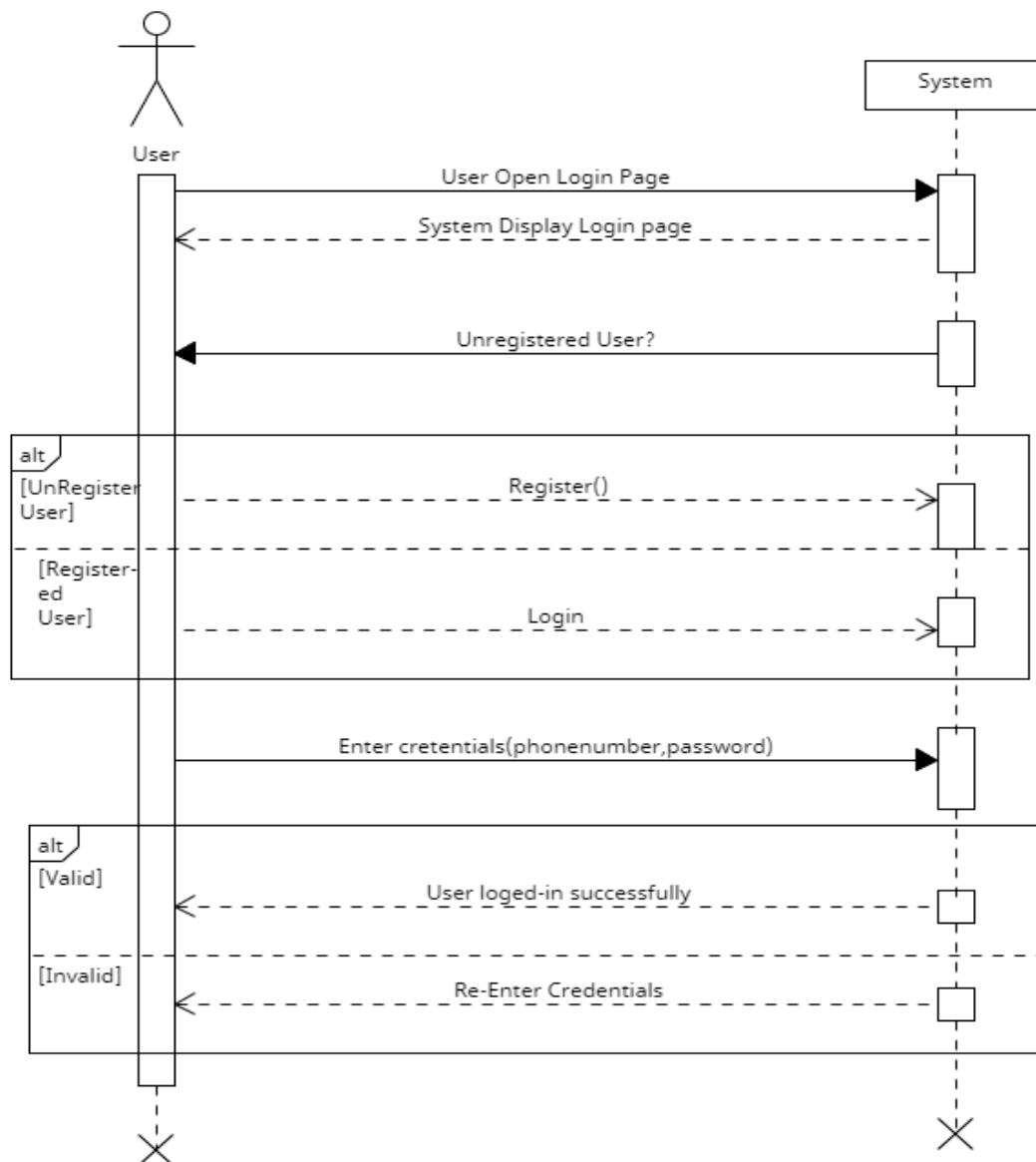


Figure 4.20: Login Sequence

4.6.2.3.2 REGISTRATION SEQUENCE DIAGRAM

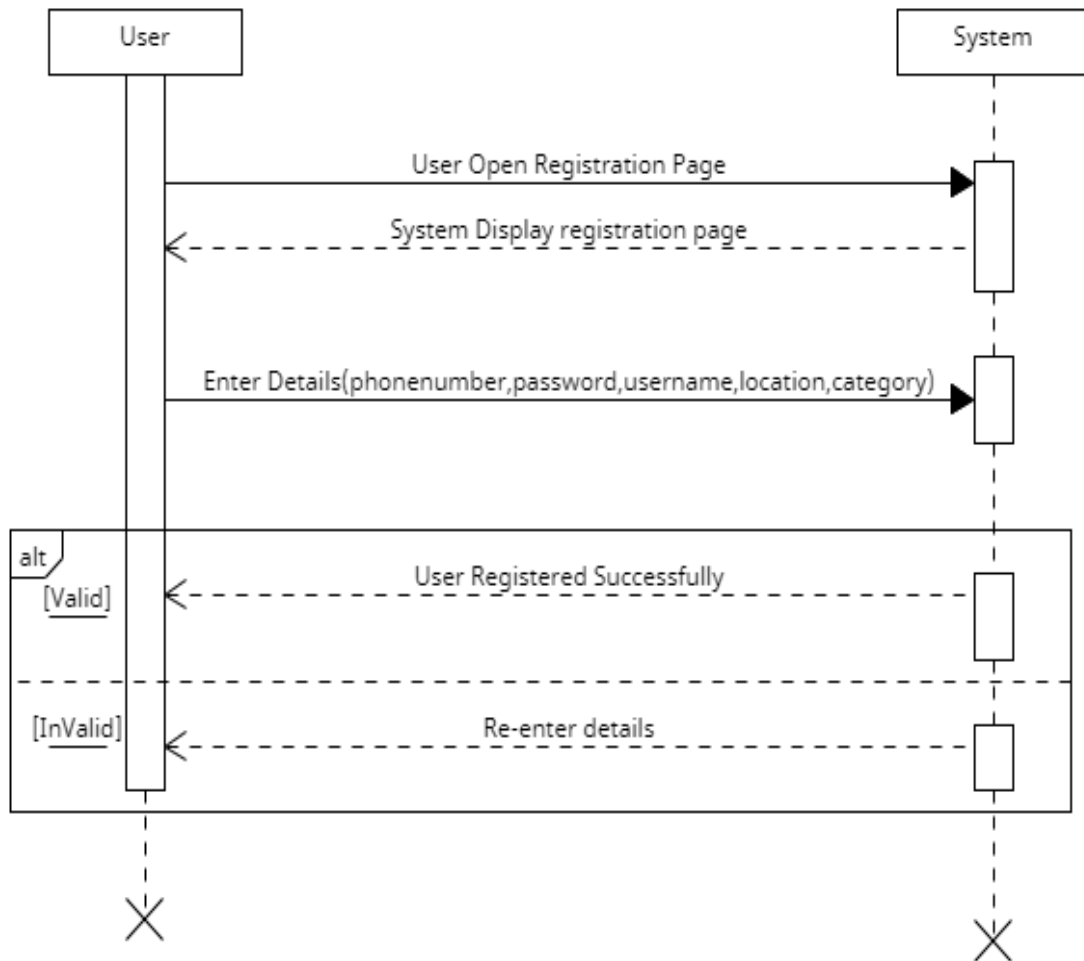


Figure 4.21: Registration Sequence

4.6.2.3.3 MANAGE SHOP BY SHOP OWNER SEQUENCE DIAGRAM

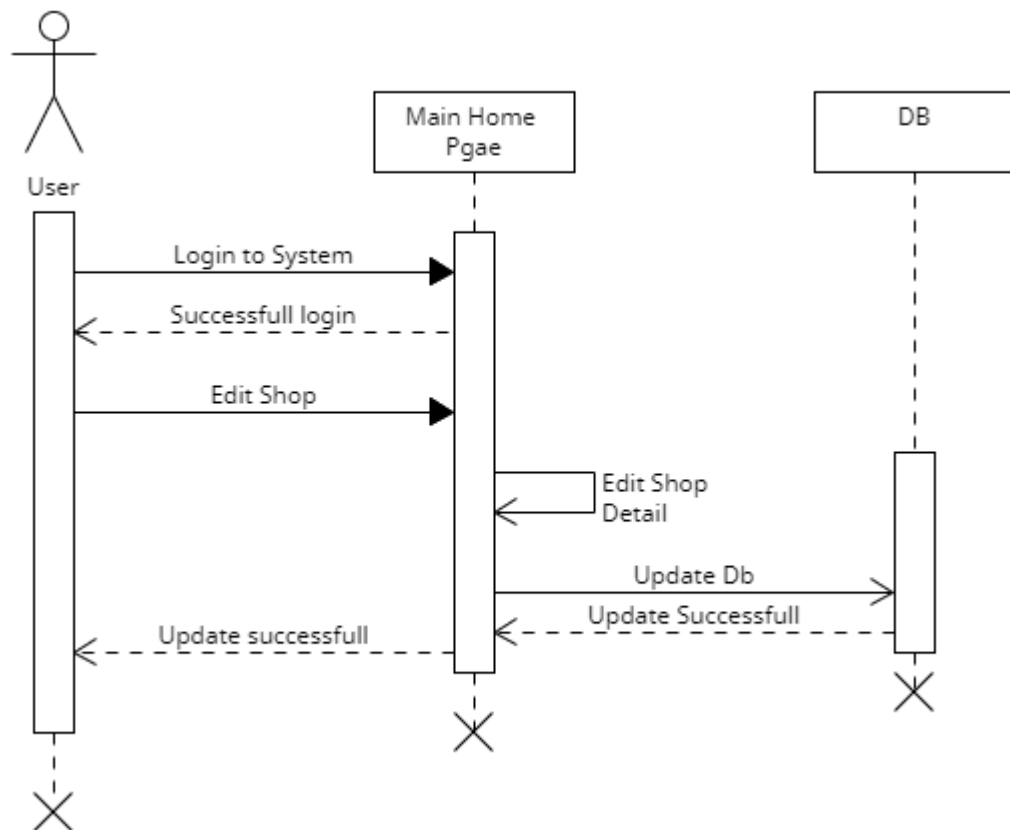


Figure 4.22: Manage Shop Sequence

4.6.2.3.4 MANAGE SHOP BY SHOP ADMIN SEQUENCE DIAGRAM

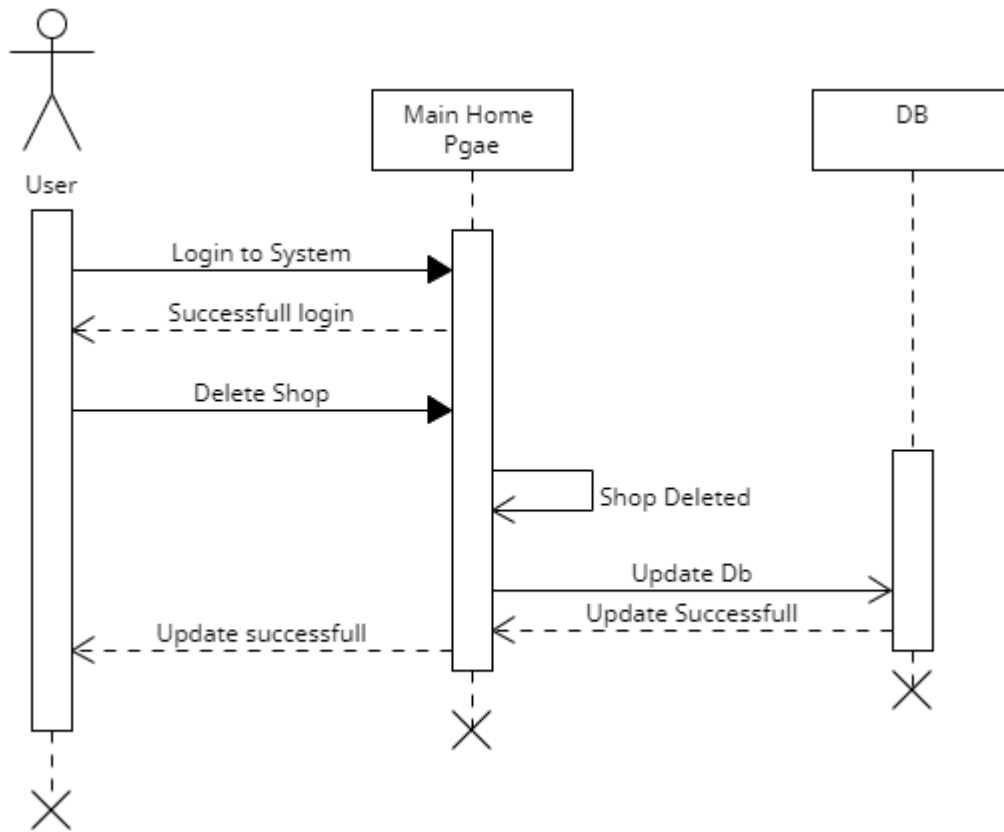


Figure 4.23: Manage shop by admin sequence

4.6.2.3.5 MANAGE PRODUCT SEQUENCE DIAGRAM

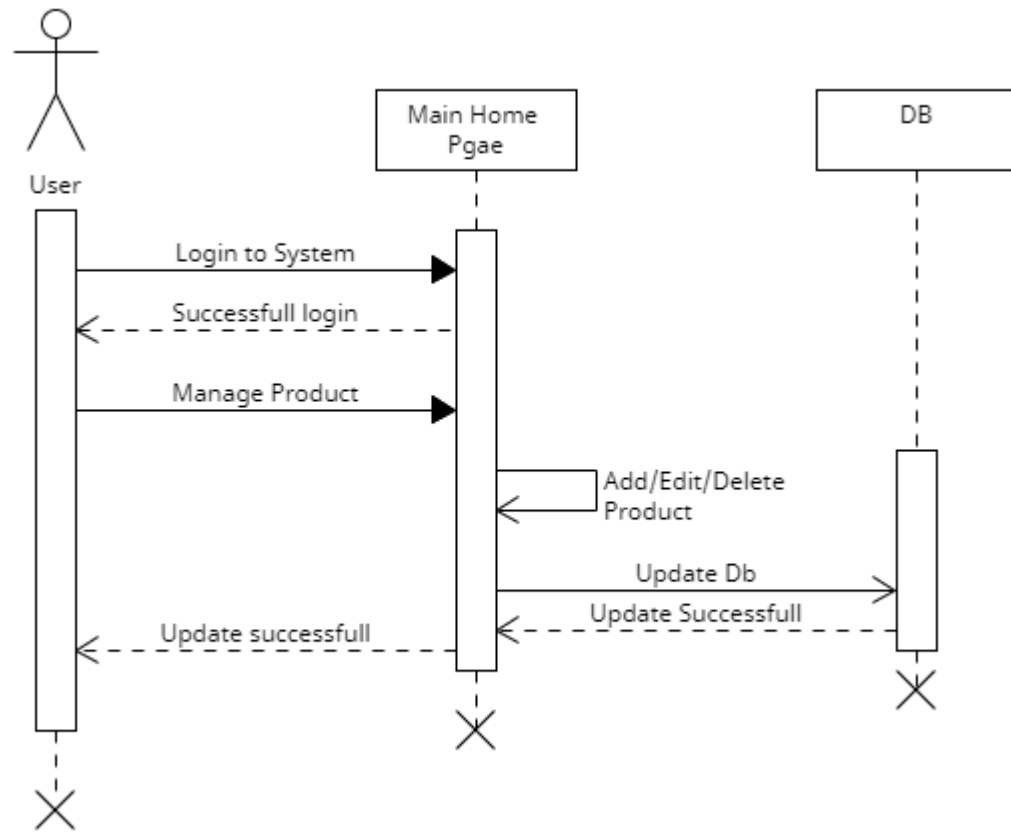


Figure 4.24: Manage Product Sequence

4.6.2.3.6 CHAT SEQUENCE DIAGRAM

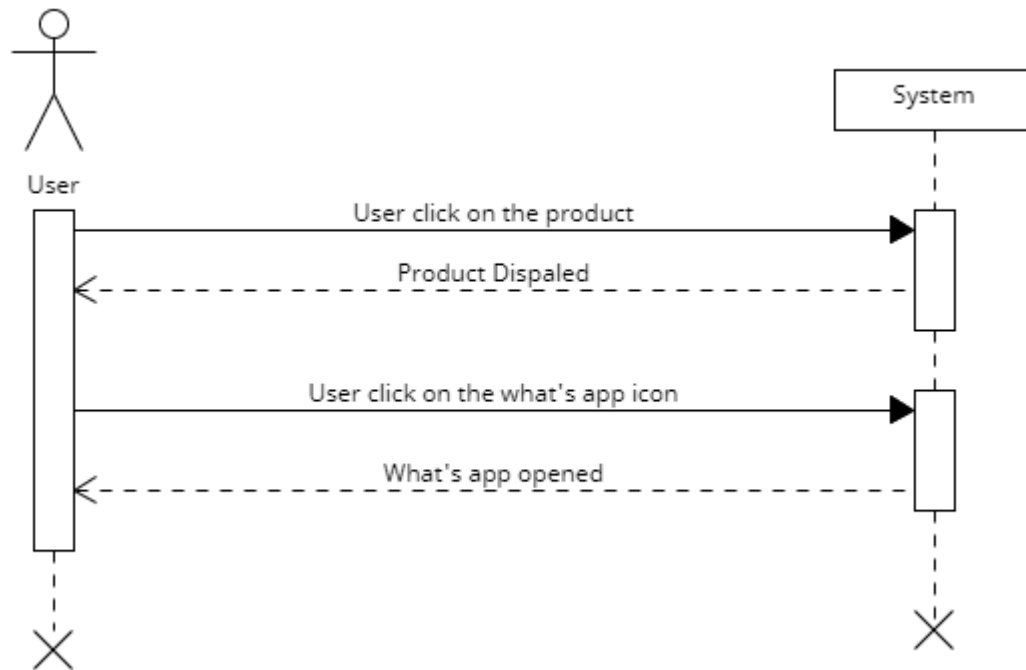


Figure 4.25: Chat Sequence

4.6.2.3.7 SEARCH SHOP SEQUENCE DIAGRAM

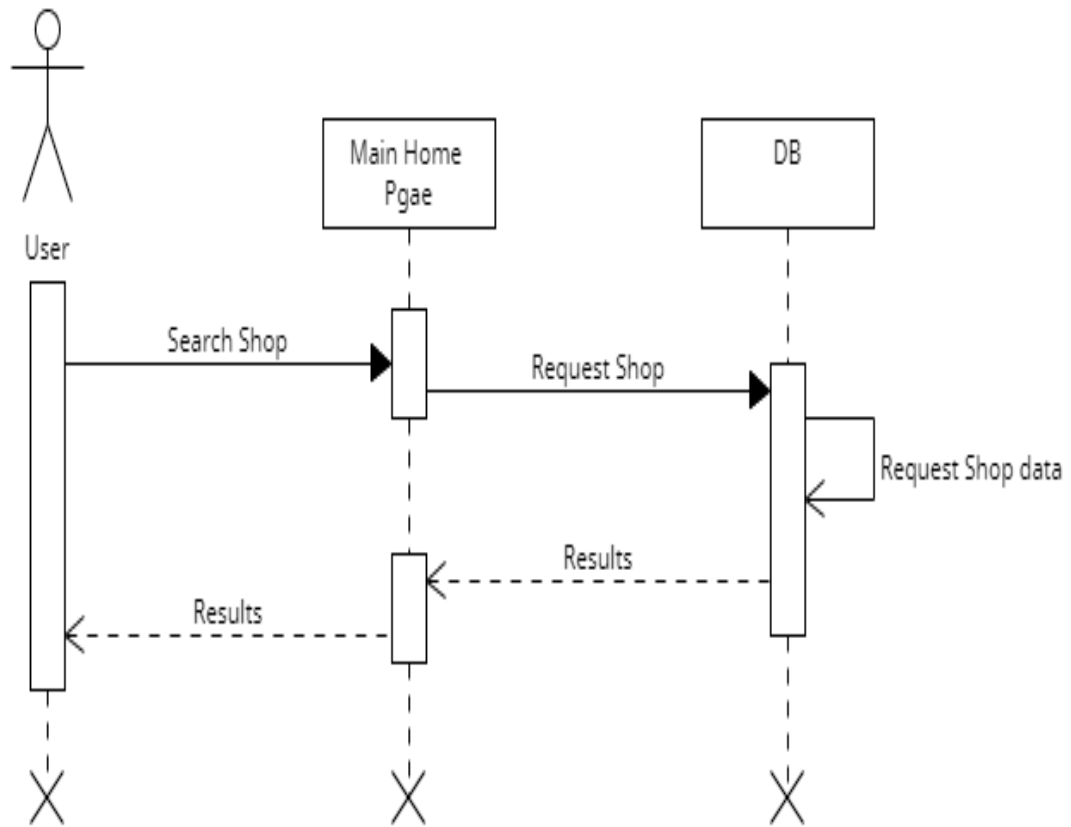


Figure 4.26: Search Shop Sequence

4.6.2.3.8 SEARCH PRODUCT SEQUENCE DIAGRAM

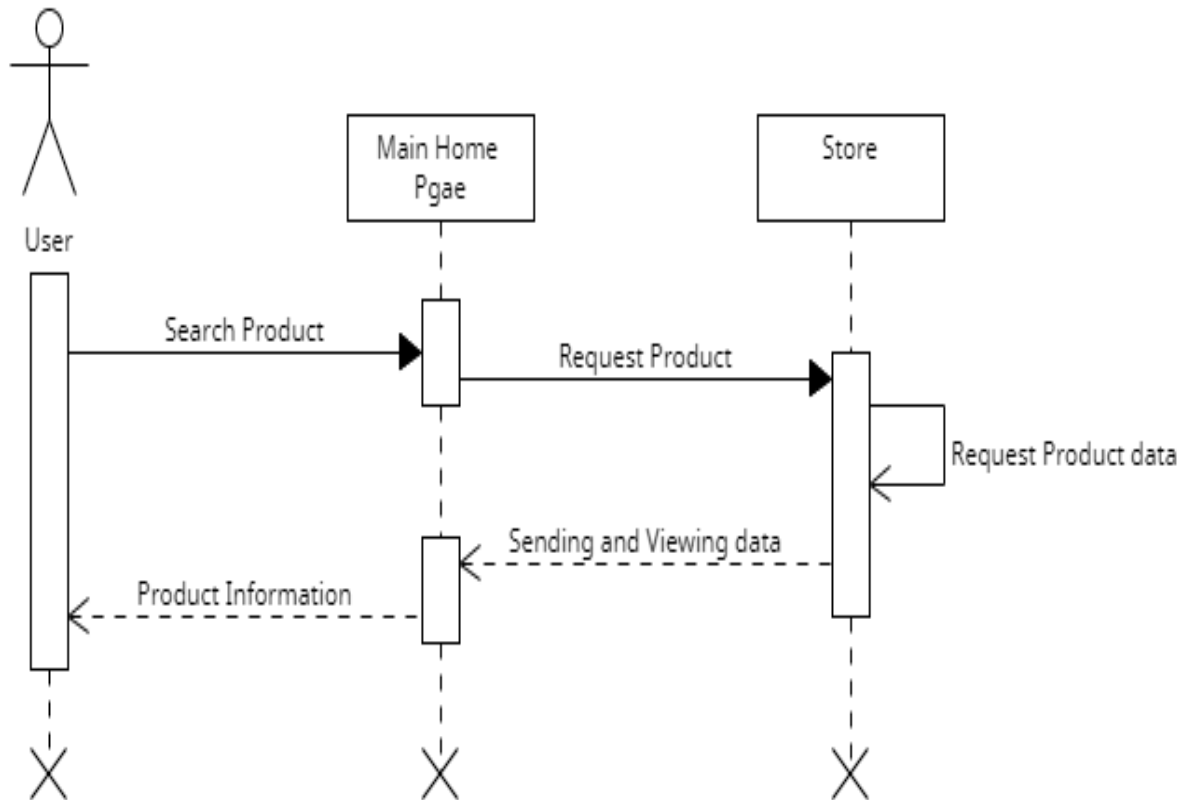


Figure 4.27: Search Product Sequence

4.6.3 DEVELOPMENT VIEW

The development view is a view that explains the software developer's view, and how he/she looks at the system. This view is also known as implementation view because it plays a huge role in the implementation of the system. It represents when the final system will be deployed how it is going to communicate among classes of the system.

4.6.3.1 COMPONENT DIAGRAM

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. It helps to double-check that every aspect of the required functions of the system has been covered in the system development and all the details are checked in implementation.

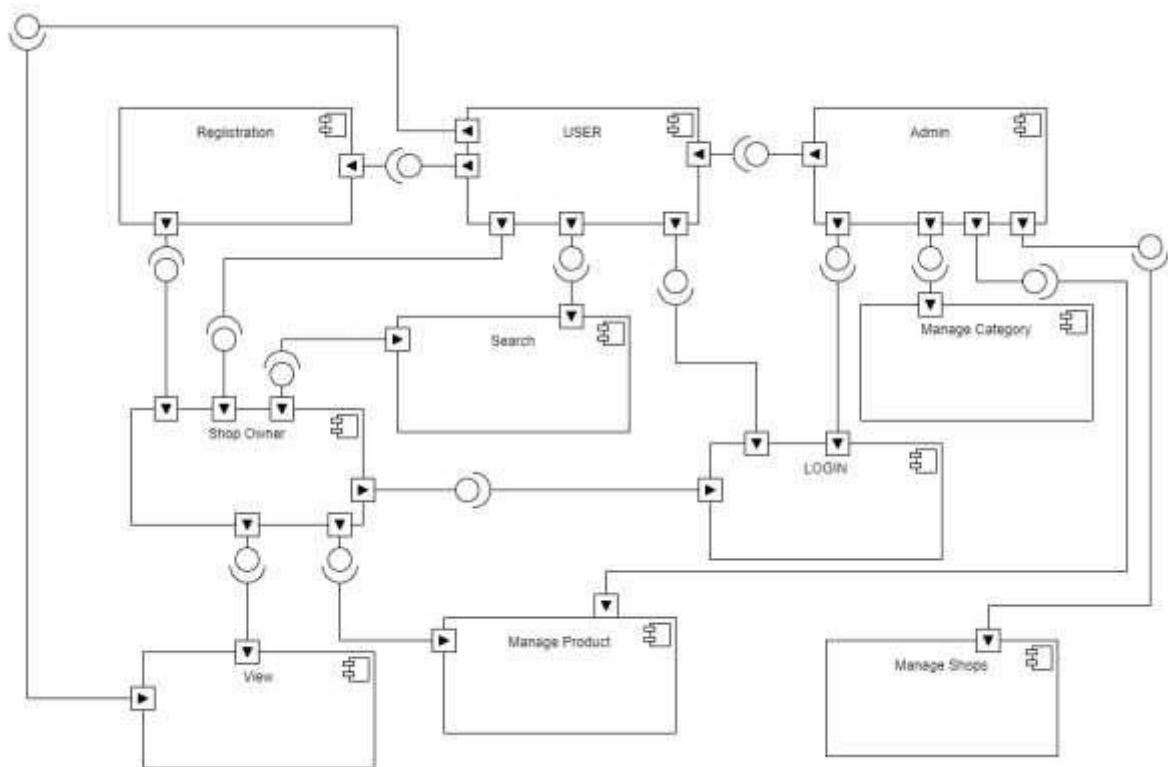


Figure 4.28: Component Diagram

4.6.4 PHYSICAL VIEW

The physical view of the system is designed according to the computer engineer's perspective. How the end product will be deployed helps computer engineers to deploy the system accordingly. A deployment diagram is used for the representation of this view.

4.6.4.1 DEPLOYMENT DIAGRAM

A deployment diagram is used to represent the physical view of the system and is drawn from the system engineer's perspective. The deployment diagram represents the system's execution architecture and depicts the physical software and hardware of the system. The diagram is made up of nodes that describe the physical devices used within the system and the middleware connecting these devices.

The main goal of OneMarket is to provide help to the user in finding the product and shops. From the following figure, the internet is the mainstream which tells that internet is the core of the project.

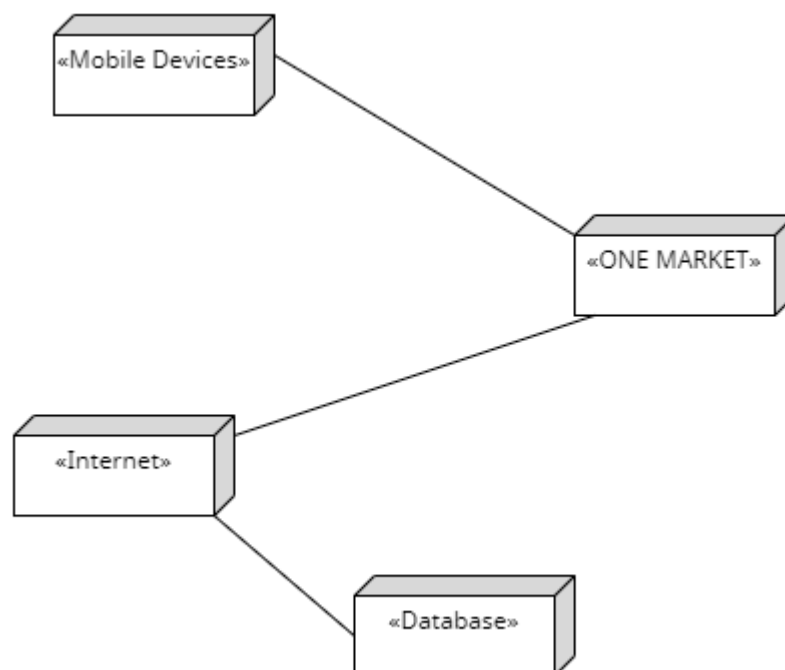


Figure 4.29: Deployment Diagram

4.7 SUMMARY

In this chapter, we discussed the system design, top-down approach and other design aspects. We deeply discussed the 4+1 view architecture model and how it is helpful for different stakeholders of the system. Physical, logical and development views are the static view of the system and the only process view is the dynamic view of the system. The chapter tells us how a system is designed in different ways, which design is important for which stakeholder, and how system design helps in the implementation and enhanced understandability of the system. Why designing is an important phase of system development? The chapter explains all these questions.

CHAPTER 5

IMPLEMENTATION

5.1 INTRODUCTION

The chapter gives a brief explanation of modules of this system. It includes modules, libraries, frameworks and API's and their details that are used in each module. This chapter gives how modules are integrated with each other to perform the functionalities of the system. It briefly describes how the developed system is working and module wise integration with each other.

5.2 MODULES

Following are some of the modules of the system which shows the workflow of the system.

5.2.1 HOME MODULE

When we open OneMarket the home page is opened by default. All the shops and products are shown in this module. The whole app is called from this module. All the other components or modules are included and called from this module.

5.2.2 SHOP MODULE

When we click on a specific shop, the shops details and products is shown. So using this module the shop details and products is shown.

5.2.3 VIEW ITEM MODULE

When you click on a specific product. The product picture and description is shown. This module is used to show details about a single product or item.

5.2.4 ALL SHOP MODULE

When you click on All Shop button on the app, you will be shown all the shop registered on the system. So, all of this is handled by this module.

5.2.5 CUSTOMIZE SHOP MODULE

The shop owner who registers their shops can customize their shops like their shop name and description, profile page and cover page. All of this is handled using this module.

5.2.6 SIGN IN MODULE

In the following module we have three users, a customer (or visitor or a user), a shop owner and an admin. Sign-in module handle two user, admin and shop owner. Signin module handle the sign-in working and authentication.

5.2.7 SIGN UP MODULE

In this module a user gives the user detail and shop detail to register their shop to the OneMarket. So, registration is handle by this module.

5.2.8 ADMIN MODULE

This module is only accessible to the admin. Admin can view and delete the shops according to the requirements.

5.2.9 LIBRARIES

5.2.9.1 AXIOS

Axios is a simple promise-based HTTP client for the browser and node.js. Axios provides a simple to use library in a small package with a very extensible interface.

5.2.9.2 REDUX

Redux is a state management front-end library, which can be used with any library or framework. The primary use of Redux is that we can use one application state as a global state and interact with the state from any react component is very easy whether they are siblings or parent-child.

For example, the basic usage of Redux comes into picture when the app gets large and complex. In such apps, simple data management as parent-child becomes difficult using props. There are multiple components trying to communicate with multiple other components. In such cases, Redux comes in handy.

5.2.9.3 MONGOOSE

Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB.

5.2.10 DATABASE

5.2.10.1 MONGODB

MongoDB is a schema-less NoSQL document database. It means you can store JSON documents in it, and the structure of these documents can vary as it is not enforced like SQL databases. This is one of the advantages of using NoSQL as it speeds up application development and reduces the complexity of deployments.

Mongo DB Atlas is used to stored text form data. 2GB space is given to us by atlas. So we used all the text form data in this cloud storage.

5.2.10.2 CLOUDINARY

Cloudinary is a software solution based on Software-as-a-Service (SaaS). It is primarily used for managing all the media assets present in the cloud. The assets can be anything from the web or mobile application depending on the type. Cloudinary offers a full-fledged solution for all the media assets including image and video needs, upload, administration, optimized delivery, storage, and manipulation.

All the media files are stored in the cloud nary. We used Cloudinary to store images. Cloud nary gives us 1GB free space. So, we utilize this 1GB. It provides us the three things cloud name, api_key, and api_secret_key. We have to use these things and put in our code and send the media files to Cloudinary.

5.3 FRAMEWORKS

5.3.1 REACT NATIVE

React Native is an open-source UI software framework created by Meta Platforms, Inc. It is used to develop applications for Android, Android TV, iOS, macOS, tvOS, Web, Windows and UWP by enabling developers to use the React framework along with native platform capabilities.

Similar to React for the Web, React Native applications are written using a mixture of JavaScript and XML-esque markup, known as JSX.

React Native currently supports both iOS and Android, and has the potential to expand to future platforms as well.

5.3.2 EXPRESS JS

Express js is a node js web application framework that provides broad features for building web and mobile applications. It is used to create a single page, multipage, and hybrid web application. Express js is high-performance, fast, opinionated, and lightweight.

With express js, you can create an API very easily for the server. ExpressJS includes significant features that let you dynamically create a back-end API. It doesn't require any third-party dependencies and doesn't come with any frameworks or plugins. And with the

help of React js, you can create a simple application very easily as react components are easily testable.

5.4 EXTERNAL API'S

We used only one external API that is from WhatsApp. This API is for connecting to the shop owner by contacting through WhatsApp. So, the shop owner number is required during the shop registration. And then we used that number for contacting on WhatsApp.

url: <https://wa.me/<phone no>>

CHAPTER 6

TESTING, ANALYSIS & VALIDATION

6.1 INTRODUCTION

This chapter is a basic segment as it yields the overall portrayal of testing system. The main motivation behind testing the system is to find out errors of the system. We know no program or system is error free each system has some errors that are not found yet. Testing is characterized as actualizing a program or programming framework with an aspiration of recognizing mistakes testing of the system is an important part of developing the system as it finds out how much implementation of the system is successful and is according to the proposed requirements no matter how capable a developer develops a program or system it always contain some of the errors. Purpose of testing is to find out those errors. A gathering of information or data that a product framework will continue as regular information is a test case. A whole programming framework or venture is cut up into pieces and illuminatingly the point of convergence of the analyzer is to test every single lump or unit of the product framework to identify mistake. The aftereffects of testing are the indications of the product or task precision and quality in this chapter we will perform certain test cases on the system and find out whether the system exhibits similar behavior as expected or give errors.

6.2 TEST METHODOLOGY

Testing a product framework is significant on the grounds that it breaks down the exhibition and the value of the product or undertaking being thrived. Testing is a methodology or procedure of identifying or finding misrepresentation in the product framework the accomplishment of any product framework or undertaking lies in its exactness and right execution and any individual or association can accomplish this by testing every module of the product framework.

Differentiating procedures of testing the product framework are utilized yet white box testing and discovery testing merit taking note. Presently there is a need to understand what white box and discovery testing implies. To begin with we characterize the first which gives an appropriate comprehension of the inner structure and working of the product framework. An analyzer should know each and everything or idea of backend programming or source code. This kind of testing is reasonable at different degrees of the product framework testing measure. Unexpectedly Black box testing has no worry about the interior structure, rationale, and working of the product framework.

6.3 TEST BED

The process of testing a specific unit or module of the system is called bed. Test bed is the configuration of software and hardware to test the system. In a present-day a phraseology employed utterly as a reference to the software system test implementation condition or environment is a testbed. Predominantly this is a method of testing all the components of the software system independently test beds are composite reflections of frameworks and are utilized to consider framework segments and associations to pick up additional knowledge into the substance of the genuine framework. They are worked on models and bits of genuine framework parts and are utilized to give knowledge into the activities of elements of a framework. The significant element of a testbed is that it just spotlights on a subset of the complete framework. That is the significant perspective that we wish to contemplate, refine or create is the angle executed in the testbed.

6.4 SYSTEM TEST CASE

System testing conducted to ensure that all requirements and specifications are working accurately it is performed to ensure that system is working accurately and all the functionalities are being performed in accordance to the requirements and specifications provided at the earlier stage of the project. It is conducted to ensure that all requirements and specifications are working accurately while performing testing many errors were identified and those errors are removed later.

6.5 TEST CASES

The circumstances under which it is checked whether the developed system satisfy the proposed requirements precisely the purpose of performing this experiment is to find out mistakes and errors in all the phases of developing the system. The purpose to conduct testing process is to distinguish deception or mistakes in all phases of improvement that are prerequisite plan development and execution. The process of writing a test case can also help reveal errors or defects with the system mostly test cases are the single step or a sequence of steps or actions applied on the system after the testing process the tester decides which test case is passed or which is failed One significant highlight is remembered while making test cases is that all settings should be covered. Whether it is positive or negative. Another value seeing point is that attempt to utilize simple and direct language or words a test case should be exact, brief and sensible. Test cases to test working of different functionalities of this system are given below.

6.5.1 REGISTER TEST CASE

When a user registers on the application and creates account. After creation of account user will become a registered user of the application and will be able to login to the system and use its functionalities.

Table 6.1: Registration Test Case

Test Case ID	TC-01			
Use Case Ref	UC-01			
Test Date	November, 2022			
Objective	To check whether registration module is working			
Environment	Mobile App			
Assumptions	New user Registration			
Pres-Requisite	User is on Registration page and not registered			
Test Procedure	Steps to perform	Input Data	Expected Result	Actual Result
	Click on “Registration” Button	N/A	Warning! Input Required	As expected
		Invalid input data	Warning! Username/Password Invalid	As expected
		Valid input data	User Registered	As expected
Status	Pass			

6.5.2 LOGIN TEST CASE

There are some functionalities which only registered user can perform only for this purpose user must have to register. When user login on the app user will have access to do his desired thing.

Table 6.2: Login Test Case

Test Case ID	TC-02			
Use Case Ref	UC-02			
Test Date	November, 2022			
Objective	To check whether Login module is working properly.			
Environment	Mobile App			
Assumptions	User is on login page			
Pres-Requisite	User must be registered.			
Test Procedure	Steps to perform	Input Data	Expected Result	Actual Result
	Click on “Login” Button	N/A	Warning! Input Required	As expected
		Invalid input data	Warning! Username/Password Invalid	As expected
		Valid input data	Shop profile screen	As expected
Status	Pass			

6.5.3 SEARCH TEST CASE

As app have multiple shops and products registered. So if user wants a specific thing to view/buy then it's better to search it through search module instead of traversing all the app. So user can directly search shop or product from search bar and all the related products and items will be listed.

Table 6.3: Search Test Case

Test Case ID	TC-04			
Use Case Ref	UC-04			
Test Date	November, 2022			
Objective	Search product/shop			
Environment	Mobile App			
Assumptions	App is running and user is on home page			
Pres-Requisite	App is running smoothly			
Test Procedure	Steps to perform	Input Data	Expected Result	Actual Result
	Click on "Search"	N/A	Warning! Input Required	As expected
		Invalid input data	Not relevant product available	As expected
		Valid input data	Product of shop listed	As expected
Status	Pass			

6.5.4 ADD SHOP TEST CASE

App has multiple user included shopkeeper and customer as a shopkeeper they have to register on the app and also add their shop. So that a shopkeeper assign a name and location to shop.

Table 6.4: Add Shop Test Case

Test Case ID	TC-05			
Use Case Ref	UC-05			
Test Date	November, 2022			
Objective	Add shop			
Environment	Mobile App			
Assumptions	App is running and user is on shop profile screen			
Pres-Requisite	App is running smoothly and user is registered			
Test Procedure	Steps to perform	Input Data	Expected Result	Actual Result
	login	N/A	Warning! Input Required	As expected
		Valid input data	Shop name and description shown	As expected
Status	Pass			

6.5.5 EDIT SHOP TEST CASE

When did a shopkeeper (user) add shop on the application and sometime shopkeeper expand their business and add more categories in their business for example a user already selling tires and now he has some products like floor mat and seats cover so now user have to edit category of shop and also can be edited to interior category. Similarly user can edit their locations and etc.

Table 6.5: Edit Shop Test Case

Test Case ID	TC-05			
Use Case Ref	UC-05			
Test Date	November, 2022			
Objective	Edit Shop			
Environment	Mobile App			
Assumptions	App is running and user is on shop profile screen			
Pres- Requisite	App is running smoothly and user is registered			
Test Procedure	Steps to perform	Input Data	Expected Result	Actual Result
		Valid input data	Shop edited	As expected
Status	Pass			

6.5.6 DELETE SHOP TEST CASE

Delete shop test case sometimes user (shopkeeper) decides to delete shop from app there may be multiple reason like he couldn't achieve targeted revenue. User can delete shop from app permanently through delete.

Table 6.6: Delete Shop Test Case

Test Case ID	TC-07			
Use Case Ref	UC-07			
Test Date	November, 2022			
Objective	Delete shop			
Environment	Mobile App			
Assumptions	App is running and user is on shop profile screen			
Pres-Requisite	App is running smoothly and user is registered			
Test Procedure	Steps to perform	Input Data	Expected Result	Actual Result
	Login as Admin	Delete	Deleted	As expected
Status	Pass			

6.5.7 ADD PRODUCT TEST CASE

When a user (Shopkeeper) adds a shop he also need to enlist all the available item on app. So that user (Customer) can view. For this purpose shopkeeper can add all products on app through add product module.

Table 6.7: Add Product Test Case

Test Case ID	TC-08			
Use Case Ref	UC-08			
Test Date	November, 2022			
Objective	Add product			
Environment	Mobile App			
Assumptions	App is running and user is on shop profile screen			
Pres- Requisite	App is running smoothly and user is registered			
Test Procedure	Steps to perform	Input Data	Expected Result	Actual Result
	Click on Add_product Button	Invalid input data	Warning! Only JPG files accepted	As expected
		Valid input data	Product added	As expected
Status	Pass			

6.5.8 EDIT PRODUCT TEST CASE

All the product listed on app if user has some update regarding that product through edit product user can edit product specifications and etc. for example price of product can be change through edit product.

Table 6.8: Edit Product Test Case

Test Case ID	TC-09			
Use Case Ref	UC-09			
Test Date	November, 2022			
Objective	Edit product			
Environment	Mobile App			
Assumptions	App is running and user is on shop profile screen			
Pres-Requisite	App is running smoothly and user is registered			
Test Procedure	Steps to perform	Input Data	Expected Result	Actual Result
	Click on “Edit product” Button	N/A	Warning! Input Required	As expected
		Invalid input data	Warning! Only JPG files accepted	As expected
		Valid input data	Product updated	As expected
Status	Pass			

6.5.9 DELETE PRODUCT TEST CASE.

If the user (shopkeeper) want to remove some items from the list. Through delete product module shopkeeper can remove item there are couple of reasons for removing product from the list one of them is product is no more available in the market.

Table 6.9: Delete Product Test Case

Test Case ID	TC-10			
Use Case Ref	UC-10			
Test Date	November, 2022			
Objective	Delete product			
Environment	Mobile App			
Assumptions	App is running and user is on shop profile screen			
Pres-Requisite	App is running smoothly and user is registered			
Test Procedure	Steps to perform	Input Data	Expected Result	Actual Result
	Click on “Delete product” Button	delete	Deleted	As expected
Status	Pass			

6.5.10 CHATTING TEST CASE

When user (customer) visit the application and decides to buy some thing or may be want more information about that product he can directly talk to shopkeeper through WhatsApp.

Table 6.10: Chat Test Case

Test Case ID	TC-12		
Use Case Ref	UC-12		
Test Date	November, 2022		
Objective	Chatting		
Environment	Mobile App		
Assumptions	App is running and user is on shop profile screen		
Pres-Requisite	App is running smoothly and user is registered		
Test Procedure	Steps to perform	Expected Result	Actual Result
	Click on “Chat” Button	Redirected to Whatsapp	As expected
		Warning! Make sure Whatsapp installed on your device	As expected
Status	Pass		

6.5.11 ADMIN TEST CASE

All the shops registered on ONEMARKET can't be deleted. Just admin of System can delete the shops. Admin page also show all the shops list or registered on ONEMARKET.

Table 6.11: Admin Page Test Case

Test Case ID	TC-101			
Use Case Ref	UC-11			
Test Date	November, 2022			
Objective	Delete product			
Environment	Mobile App			
Assumptions	App is running and user is on Admin page			
Pres-Requisite	App is running smoothly and some shops have been registered.			
Test Procedure	Steps to perform	Input Data	Expected Result	Actual Result
	Click on "Delete shop" Button	Delete button click	deleted	As expected
	Click on View shop	View button click	On shop page	As expected
Status	Pass			

6.6 SUMMERY

Essentially, testing is a means of evaluating and analyzing whether or not all of the system's features are meeting the requirements. Additionally, defects and mistakes in the system are detected throughout the testing phase so that we can determine if the system meets the requirements or not, and this is all done during the development process. This is where the entire system's functioning is tested. More specifically, validation is the process of verifying and identifying the system's validity, and it is a crucial element of the development process. Furthermore, the production of several test cases is incorporated here in order to verify and check a system's functionality.

CHAPTER 7

CONCLUSION & FUTURE WORK

7.1 OVERVIEW OF CHAPTER

This chapter include a brief summary of how the proposed solution has addressed the problem statement specified in the introduction section. Provide an overview of what kind of evaluations were undertaken in order to prove that the solution really solves the problem. Provide an overview of the recommendations and Include a future directions which is required as part of the future work.

7.2 SYSTEM OVERVIEW

The project, OneMarket is based on the real problem faced by vehicle markets. There are many applications which provide the user with the facility to view the vehicle products but there is no such app which provide the facility to the shop owner to register their shops. This application would provide the facility for the shop owner to register their shops and list down their products as well as the car owner to view the shops and products in the twin cities OneMarket is developed using react native.

7.3 MILESTONES ACHIEVED

The milestones that were achieved in Android Application are:

- Registration:
- Login
- Register Shop
- Add Product
- View Shops
- View Product
- Search Shop
- Search Product
- Contact Shop Owner

7.4 LIMITATIONS

Few limitation of OneMarket are: OneMarket doesn't provide the facility to buy the product online. It covers only Rawalpindi and Islamabad vehicle markets. There is no registration for the users who do not want to register their shop. It's only for the android based application. OneMarket requires high level security which cannot be achieved as it is costly.

7.5 FUTURE WORK

In the near future, we will add the functionality to buy the product online. After that we will launch the application to app store. It is also possible that we introduce the IOS and Desktop Version of this application and introduce some security mechanism.

7.6 SUMMARY

OneMarket is based on Android Application for Vehicle Markets of twin cities. Shop owner can register their shops and products while the car owner can see the products and shops in the markets. There are a few limitations of this application, one cannot buy the product online, it is available only for the twin cities and there is no registration for the users who do not want to register their shops. The application requires high level security which cannot be achieved as it is costly. Obviously, there is still a lot to work on in the mobile application if we want to launch it to the market. There is also the possibility of developing the iOS and Desktop version of the application.

REFERENCES

- [1] Navigation Documentation:
<https://reactnavigation.org/docs/getting-started/>
- [2] React Native Documentation
<https://reactnative.dev/docs/navigation#react-navigation>
- [3] Autojin:
<https://autojin.pk/>
- [4] Autox:
<https://autox.pk/>
- [5] Daraz:
<https://www.daraz.pk/>
- [6] Sehgal Motors:
<https://www.sehgalmotors.pk/>
- [7] MongoDB Atlas:
<https://www.mongodb.com/atlas/database>
- [8] Cloudinary:
<https://cloudinary.com/>
- [9] Nodejs:
<https://nodejs.org/en/>
- [10] MongoDB community:
<https://www.mongodb.com/try/download/community>
- [11] React Native Community:
<https://github.com/reactjs>
- [12] Whatsapp API
<https://web.whatsapp.com/>

