## Re-Wear (Upgrading Sustainable fashion)



**By:**

**M. Talha Masood**

**29423**

**Sfwan Ali**

**13408**

### Supervised by:

**Shahid Ali khan**

**Faculty of Computing**

**Riphah International University, Islamabad**

**Fall 2024**

**A Dissertation Submitted To**

**Faculty of Computing,**

**Riphah International University, Islamabad**

**As a Partial Fulfillment of the Requirement for the Award of the Degree of**

**Bachelors of Science in Software Engineering**

##### Faculty of Computing

**Riphah International University, Islamabad**

Date: June 31, 2024

### Final Approval

This is to certify that we have read the report submitted by **Muhammad Talha Masood (SAP Id:29423)**, **Sfwan Ali (SAP Id: 13408)** for the partial fulfillment of the requirements for the degree of the Bachelors of Science in Software Engineering (BSSE). It is our judgment that this report is of sufficient standard to warrant its acceptance by Riphah International University, Islamabad for the degree of Bachelor of Science in Software Engineering (BSSE).

### Committee:

|  |
| --- |
| **1** |
| Shahid Ali Khan  (Supervisor) |
| **2** |
| Dr. Musharraf Ahmad  (Head of Department/chairman) |

**Declaration**

We hereby declare that this document **“Re-Wear (Upgrading Sustainable Fashion)”** neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this project with the accompanied report entirely based on our personal efforts, under the proficient guidance of our teachers especially our supervisor **Shahid Ali khan**. If any part of the system is proved to be copied out from any source or found to be reproduction of any project from anywhere else, we shall stand by the consequences.

**Muhammad Talha Masood**

**29423**

**Sfwan Ali**

**13408**

**Dedication**

We dedicate this project to Allah Almighty our creator, our strong pillar, our source of inspiration, wisdom, knowledge and understanding. He has been the source of our strength throughout this program. Also, we dedicate our work to our family, friends and teachers. The unrivalled encouragement from our parents and outstanding support from teachers is what lead to success of this project. We also dedicate our work to our supervisor **Shahid Ali Khan** and the faculty members.

**Acknowledgement**

First of all, we are obliged to Allah Almighty the Merciful, the Beneficent and the source of all Knowledge, for granting us the courage and knowledge to complete this Project.

We are grateful to our supervisor **Shahid Ali Khan** for his enthusiasm, patience, insightful comments, helpful information, practical advice, and unceasing ideas that always helped us tremendously in our project. Without his support and guidance, this project would not have been possible. Also, a special thanks to the rest of the faculty members for their unconditional support

**Muhammad Talha Masood**

**29423**

**Sfwan Ali**

**13408**

**Abstract**

"Re-Wear Upgrading Sustainable Fashion introduces a comprehensive online platform that reimagines consumer engagement with second-hand clothing and tailoring services." The marketplace prioritizes quality and affordability by curating a varied variety of pre-owned clothes products such as wedding gowns, groom dresses, shirts, jeans, sarees, shoes, accessories, and more. Users can create their shop and sell their pre-owned products. Re-Wear transforms the apparel business by allowing people to make eco-conscious choices while enjoying a personalized shopping experience.

### Contents

[List Of Figures 11](#_bookmark0)

[List Of Tables 12](#_bookmark1)

[Chapter 1 2](#_bookmark2)

[Introduction 2](#_bookmark3)

[Opportunity And Stakeholder 2](#_bookmark4)

[Stakeholders 2](#_bookmark5)

Motivation & Challenges 3

[Solution Overview 3](#_bookmark8)

[Project Scope 3](#_bookmark9)

[Report Outline 6](#_bookmark10)

[Chapter 1 6](#_bookmark11)

[Chapter 2 6](#_bookmark12)

[Chapter 3 7](#_bookmark13)

[Chapter 4 7](#_bookmark14)

[Chapter 5 7](#_bookmark15)

[Chapter 2 8](#_bookmark17)

[Literature/Market Survey 9](#_bookmark18)

[Introduction 9](#_bookmark19)

[Literature Review/Market Survey 9](#_bookmark20)

[Brainstorming 11](#_bookmark22)

[Existing Systems 11](#_bookmark24)

[Summary 13](#_bookmark26)

[Chapter 3 14](#_bookmark27)

[Requirement Analysis 15](#_bookmark28)

[Introduction 15](#_bookmark29)

[Problem Scenarios 15](#_bookmark30)

[Problem Statement 15](#_bookmark31)

[Functional Requirements 16](#_bookmark33)

Non-[Functional Requirements 25](#_bookmark33)

[SQA Activities 25](#_bookmark42)

[Defect Identification Through Inspection 25](#_bookmark43)

[Defect Detection 28](#_bookmark45)

[Black Box Test Case Design 42](#_bookmark69)

[Summary 47](#_bookmark86)

[Chapter 4 48](#_bookmark87)

[System Design 49](#_bookmark88)

[Introduction 49](#_bookmark89)

[Architectural 4Design 49](#_bookmark90)

[Detailed Design 50](#_bookmark93)

[Use Case Diagrams 50](#_bookmark94)

[Use Case Fully Dressed Format 54](#_bookmark99)

[Activity Diagrams 86](#_bookmark149)

[Component Diagram 123](#_bookmark200)

[Schema Diagram 124](#_bookmark200)

[Sqa Activities](#_bookmark204)

[Usability Testing Through Heuristic Evaluation](#_bookmark205)

[Chapter 5 132](#_bookmark225)

[Implementation 133](#_bookmark226)

[Endeavour 133](#_bookmark227)

[Team 134](#_bookmark228)

[Work Breakdown Structure 135](#_bookmark229)

[Roles & Responsibility Matrix 143](#_bookmark231)

[Components, Libraries, Web Services, And Stubs 144](#_bookmark232)

[Libraries 144](#_bookmark233)

[Ide, Tools, And Technologies 144](#_bookmark235)

[Software Engineering Methodologies 144](#_bookmark236)

[Deployment Diagram 144](#_bookmark237)

[Sqa Activities: Defect Detection 145](#_bookmark239)

[Test Case Design (White Box) 150](#_bookmark240)

[Summary 151](#_bookmark264)

### List of Figures

Figure 2.0: Response….………………...…………………………………..9

Figure 2.1: Response….………………...…………………………………..10

Figure 2.2: Mind Map….………………...………………………………....12

Figure 4.1: Software Architecture Diagram….………………...…………...49

Figure 4.2: Use Case Diagram – Admin….………………...………………50

Figure 4.3: Use Case Diagram – Tailor….………………...……………….51

Figure 4.4: Use Case Diagram – Buyer….………………...……………….52

Figure 4.5: Use Case Diagram – Seller….………………...……………….53

Figure 4.6: Activity Diagram – Register User….………………...………...85

Figure 4.7: Activity Diagram – Login….………………...………………....86

Figure 4.8: Activity Diagram – Forgot Password….………………...……...87

Figure 4.9: Activity Diagram – Verify ….………………...………………...88

Figure 4.10: Activity Diagram – Deactivate User….………………...……...89

Figure 4.11: Activity Diagram – View User….………………...……………90

Figure 4.12: Activity Diagram – User Querry….………………...…………..91

Figure 4.13: Activity Diagram – Search User….………………...…………..92

Figure 4.14: Activity Diagram – Verify User ….………………...…………..93

Figure 4.15: Activity Diagram – View Product ….………………...………..94

Figure 4.16: Activity Diagram – Search Product….………………...……….95

Figure 4.17: Activity Diagram – Filter Product….………………...………...96

Figure 4.18: Activity Diagram – View Seller….………………...…………..97

Figure 4.19: Activity Diagram – Contact Seller….………………...………..98

Figure 4.20: Activity Diagram – Create Order….………………...………....99

Figure 4.21: Activity Diagram – Create shop….………………...…………100

Figure 4.22: Activity Diagram – List Product….………………...…………101

Figure 4.23: Activity Diagram – Delete Product ….………………..............102

Figure 4.24: Activity Diagram – Edit Product ….………………..................103

Figure 4.25: Activity Diagram – Showcase Product ….……………….....…104

Figure 4.26: Activity Diagram – Delete Showcase ….……………….......…105

Figure 4.27: Activity Diagram – Edit Showcase ….………………….......…106

Figure 4.28: Activity Diagram – Communication….…………………......…107

Figure 4.29: Activity Diagram – Tailoring….……………………...…......…108

Figure 4.30: Activity Diagram – Manage Profile….………….………......…109

Figure 4.31: Component Diagram ….………………………………………..123

Figure 4.32: Schema Diagram ….………………………………….……...…124

Figure 4.33: Heuristic Violation – 01………………………………………..127

Figure 4.34: Solution of Heuristic Violation – 01 (A)………………………128

Figure 4.35: Heuristic Violation – 02 (A)…………………………………128

Figure 4.36: Solution of Heuristic Violation – 02 (B)………………………129

Figure 4.37 Heuristic Violation – 03…………………………………….….129

Figure 4.38: Solution of Heuristic Violation – 03 (A)……………………….131

Figure 4.39: Solution of Heuristic Violation – 03 (B)……………………….131

Figure 4.40: Solution of Heuristic Violation – 03 (C)………………………132

Figure 5.1: Deployment Diagram….…………………………………..……145

### List of Tables

Table 2.1: Exiting System….…………...…………………………………..13

Table 3.0: Problem Statement…..……...…………………………………...15

Table 3.1: FR-01-Authentication…..……..………………………………...16

Table 3.2: FR-01-Tailor Module…..……..………………………………...19

Table 3.3: FR-01-Buyer Module…..……..………………………………...20

Table 3.4: FR-01-Admin Module.....……..………………………………...22

Table 3.5: FR-01-Seller Module…...……..………………………………...24

Table 3.6: Defect identification through inspection…....…………………...25

Table 3.7: Test Data-Register……………………..…....…………………...28

Table 3.8: Test Data-Login(Email)…………... …..…....…………………...29

Table 3.9: Test Data-Login(Password)…………... …..…....…………….....30

Table 3.10: Test Data-Product Name...…………... …..…....…………….....30

Table 3.11: Test Data-Product Description………..…..…....…………….....31

Table 3.12: Test Data-Product Type……..………..…..…....……………......32

Table 3.13: Test Data-Product picture…….………..…..…....…………….....33

Table 3.14: Test Data-Create Shop...……...………..…..…....…………….....33

Table 3.15: Test Data-Image………………………..…..…....…………….....34

Table 3.16: Test Data-Order Details………………..…..…....…………….....35

Table 3.17: Test Data-Order Description…….……..…..…....…………….....36

Table 3.18: Test Data-Add Order………….………..…..…....…………….....37

Table 3.19: Test Data-Phone no……………………..…..…....…………….....37

Table 3.20: Test Data-Address……………………...…..…....…………….....38

Table 3.21: Test Data-Payment………..…..…....…………….........................39

Table 3.22: Test Data-Buyer Seller………..………..…..…....…………….....40

Table 3.23: Test Data-Profile Verification………..…..…....……………........41

Table 3.24: Test Data-Search Product……..………..…..…....…………….....41

Table 3.25: Test Data-List Product (Name)……..…...…..…...…………….....41

Table 3.26: Test Data- List Product (Image) ……..………..…..…........……...42

Table 3.27: Test Data- List Product (Condition) ……..………..…..…....….....43

Table 3.28: Test Case-Login………………………..…..…....…………….....42

Table 3.29: Test Case-Create Showcase…..………..…..…....…………….....44

Table 3.30: Test Case-Create Order………………..…..…....……………......45

Table 3.31: Test Case-Name………..…..…………………....…………….....45

Table 3.31: Test Case-Number..………..…………………....…………….....46

Table 3.31: Test Case- Password ………..…..……………....…………….....47

Table 3.31: Test Case- Profile Photo..…..…………………....…………….....47

Table 3.32: Test Case-List Product..…....…....…………….............................48

Table 3.33: Test Case-Register……………………..…..…....…………….....46

Table 4.0: Fully Dressed Use Case -Register……….…..…....…………….....54

Table 4.1: Fully Dressed Use Case -Login....……….…..…....…………….....55

Table 4.2: Fully Dressed Use Case -Forgot Password.….........…………….....57

Table 4.3: Fully Dressed Use Case -Verify User……….…….…………….....58

Table 4.4: Fully Dressed Use Case -Deactivate User ……......…………….....59

Table 4.5: Fully Dressed Use Case -View User…….…..…....…………….....61

Table 4.6: Fully Dressed Use Case -User query…..…..…....……………......62

Table 4.7: Fully Dressed Use Case -Search User….…..…....……………......63

Table 4.8: Fully Dressed Use Case -View Order….…..…....……………......63

Table 4.9: Fully Dressed Use Case -View Product….…..…....…….……......64

Table 4.10: Fully Dressed Use Case -Search Product…….….....………….....65

Table 4.11: Fully Dressed Use Case -Filter Product.…..…....……………......67

Table 4.12: Fully Dressed Use Case -View Seller Profile…...……………......68

Table 4.13: Fully Dressed Use Case -Contact Seller…… .....……………......69

Table 4.14: Fully Dressed Use Case -Create Order….……………………......69

Table 4.15: Fully Dressed Use Case -Create shop….…..…....……………......72

Table 4.16: Fully Dressed Use Case -List Product….…..…....……….............74

Table 4.17: Fully Dressed Use Case -Delete Product….…..…....……….........76

Table 4.18: Fully Dressed Use Case -Edit Product….…..…………..……......76

Table 4.19: Fully Dressed Use Case - Manage Product ….…..…......…….......77

Table 4.20: Fully Dressed Use Case - Showcase Product ..…....……………...79

Table 4.21: Fully Dressed Use Case - Delete Showcase ….…..…....………....81

Table 4.22: Fully Dressed Use Case - Edit Showcase ….…..…....…………...81

Table 4.23: Fully Dressed Use Case -Communication….…..…....………......83

Table 4.24: Fully Dressed Use Case -Tailoring….…..…....………………......83

Table 4.25: Heuristic Violation -01………………………………………….127

Table 4.26: Heuristic Violation -02…………………………………………129

Table 4.27: Heuristic Violation -03…………………………………………130

Table 5.1: Work Breakdown Structure………..….…..…....……………......135

Table 5.2: Roles and Responsibility…………..….…..…....……………......143

Table 5.3: W.B Test Case- Register……………….…..…....……………....145

Table 5.4: W.B Test Case- Login………………….…..…....……………....146

Table 5.5: W.B Test Case- Edit Profile(Name)………..…....……………....147

Table 5.6: W.B Test Case- Edit Profile(Photo)………..…....……………....148

Table 5.7: W.B Test Case- Edit Profile(Password)………..…..…………....148

Table 5.8: W.B Test Case- Edit Profile(Contact Number)....……………....149

Table 5.9: W.B Test Case- List Product………………..…....……………....149

Table 5.10: W.B Test Case- Edit Product………………..…....……………....150

# Chapter 1 Introduction

### Chapter 1

## INTRODUCTION

Re-Wear is a web-based platform that transforms the way individuals engage with used clothing and custom tailoring services. In addition to allowing sellers and tailors to open their own shops to market their goods, it lets consumers buy both new and used clothing.

#### OPPORTUNITY AND STAKEHOLDER

Now a days in the fashion sector, people encountered multiple challenges. Environmental dumping was a major problem, with many clothing ending up in trash due to a lack of useful reuse options. Additionally, customers found it difficult to locate used clothing and tailoring services because they had to use many platforms, which made purchasing difficult and wasteful. Because there were limited customizing choices, it was difficult to find secondhand clothing that fit properly. Independent retailers and tailors also had trouble expanding their customer base and efficiently handling their inventory. By developing a single platform that facilitates sustainable shopping, helps users locate fitted apparel, and supports vendors and tailors.

* + 1. **Stakeholders**
       1. **Administrators**

They will have the opportunity and authority to manage the user and products.

* + - 1. **Buyer**

They are individuals who want to be stylish without harming their pocketbook by searching for affordable, trendy clothing.

* + - 1. **Seller**

Users who want to earn money from gently used items can sort through their wardrobes with Re-Wear.

* + - 1. **Tailor**

These are the individuals who the site offers to help close the gap between perfectly fitting garments and gently used items.

#### Motivation & Challenges

#### Motivation:

The goal of "Re-Wear Upgrading Sustainable Fashion" aims to provide a marketplace where customers can buy, sell, and acquire tailored clothing in addition to giving them the option to open their own stores to promote their goods. This strategy encourages sustainable fashion by providing an area for used clothing and tailored clothing.

#### Challenges:

The hurdles for "Re-Wear Upgrading Sustainable Fashion" include negotiating diverse second-hand marketplaces, assuring access to specialized services, and overcoming ingrained customer preferences for new apparel.

#### SOLUTION OVERVIEW

Our Proposed Solution is a Re-Wear Web Application for Upgrading Sustainable Fashion which provides a solution for the following:

* Seeks to create a unique online Buying and Selling platform that reimagines the fashion industry's approach to second-hand clothes and tailoring services.
* The tailoring services that allow clients to purchase stitched clothes based on their unique needs for increased comfort and style.
* Allow seller and tailor to create their shop in the market place.
  + 1. **Project Scope**

Re-Wear have four main modules which are then subdivided into multiple submodules.

**Admin Module**

**Authentication**

Admin can login to the system.

Admin will also be able to change their passwords through forgot password functionality.

* Admin will be able to block/unblock user’s profile.
* Admin should be able to maintain tracking records of buyer’s & seller’s orders.
* Admin should be able to answer user’s queries.

**Seller Module**

**Authentication**

Seller can login to the system.

Seller will also be able to change their passwords through forgot password functionality.

* Seller will be able to sell products after getting verified.
* Seller can create his shop and can sell products
* Seller will be able to keep track of their selling orders.
* Seller will be able to chat with buyer.

**Product Management**

Different Product will be managed by the Seller.

**Order Management**

Number of Order on Panel will be viewed and the reports will be generated by the Seller and Tailor.

**Buyer Module**

**Authentication**

Buyer can login to the system.

Buyer will also be able to change their passwords through forgot password functionality.

* Buyer will be able to search and filter products/items by their price, time (latest, old).
* Buyer will be able to explore products from different categories.
* Buyer will be able to use tailor service on specific products (event dresses) for alteration by providing their measurements according to product nature.
* Buyer will be able to place order and pay for product, using card or cash on deliver.

**Tailor Module**

**Authentication**

Buyer can login to the system.

Buyer will also be able to change their passwords through forgot password functionality.

* Tailor will be able showcase his product
* Tailor will be able to create shop and sell his product.

**Profile Management**

Buyers, Seller and Tailor profile will be able to manage to their profile details i.e. name, email, picture and password.

**Buyer Module**

Product Searching

Buyer will be able to search Product.

**Product Filtering**

Buyer will be able to filtered Product by category.

**View Profile**

Seller and Tailor profile will be viewed by Buyer.

#### REPORT OUTLINE

This report covers the detail of all aspects of the system, for understanding and clarity. This report has been divided into five chapters.

##### Chapter 1

This chapter introduces our system, it covers the opportunity we are looking for while building the system, Motivation and challenges, overview of the proposed solution and scope of the system.

##### Chapter 2

This chapter focuses on the current situation of the market including the market survey, and how our system stands out tall from all the other systems.

##### Chapter 3

This chapter is all about need analysis as it examines the real requirements, problem scenario addressed by the developed system and, quality assurance techniques upon requirements. It also identifies the real user of the system and those affected by the system.

##### Chapter 4

This chapter provides all the information related to design factor of the developed system by describing the system architecture design consideration, detailed design containing different diagrams that model the working behavior of the system and quality assurance upon design.

##### Chapter 5

This chapter keeps track of the information related to the actual development of the system by describing the system flow and the environment in which the proposed system was developed.

##### Chapter 6

This chapter includes our Achievements and improvements during this project also we elaborated the challenges faced by us in this document

# Chapter 2 Literature/Market Survey

### Chapter 2

## LITERATURE/MARKET SURVEY

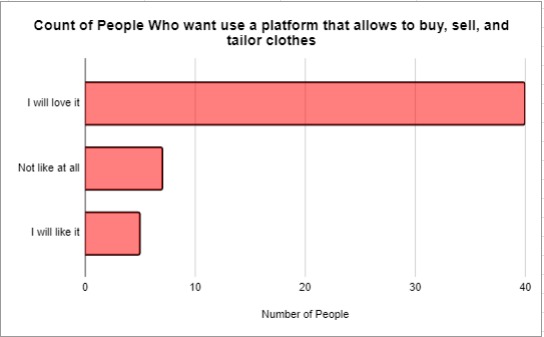
#### INTRODUCTION

The existing system encourages the manufacturing and purchase of products such as bicycles, laptop computers, automobiles, and other goods, while placing minimal attention on sustainable clothing methods. Meanwhile, those that do focus on sustainable apparel face several constraints, including high shipping costs and other logistical obstacles. This mismatch highlights a substantial market gap in which sustainable fashion is undervalued and may encounter accessibility and cost challenges. As a result, there is an urgent need for the fashion industry to adopt a more sustainable approach that addresses social concerns and promotes ethical consumer patterns.

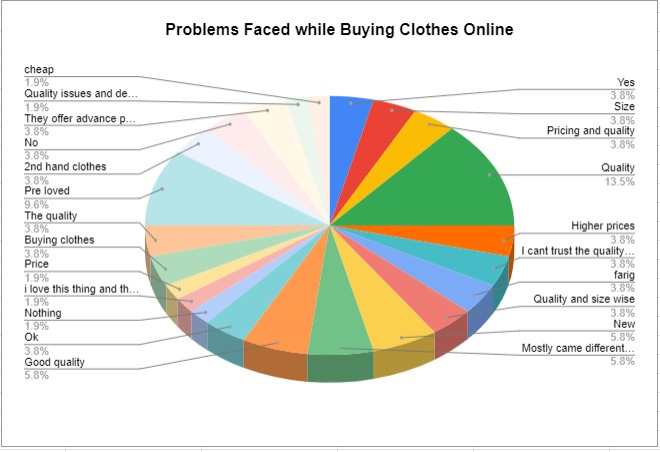
#### LITERATURE REVIEW/MARKET SURVEY

##### Online Survey

For this project, we used the Online Survey technique for requirements elicitation. The Survey was designed in such a way so that we get a complete understanding of the project on which is being worked on, the project and the problem to be solved.



**Figure 2.0: Response**



**Figure 2.1: Response**

* + - 1. **Reason for Selecting Online Survey Technique**

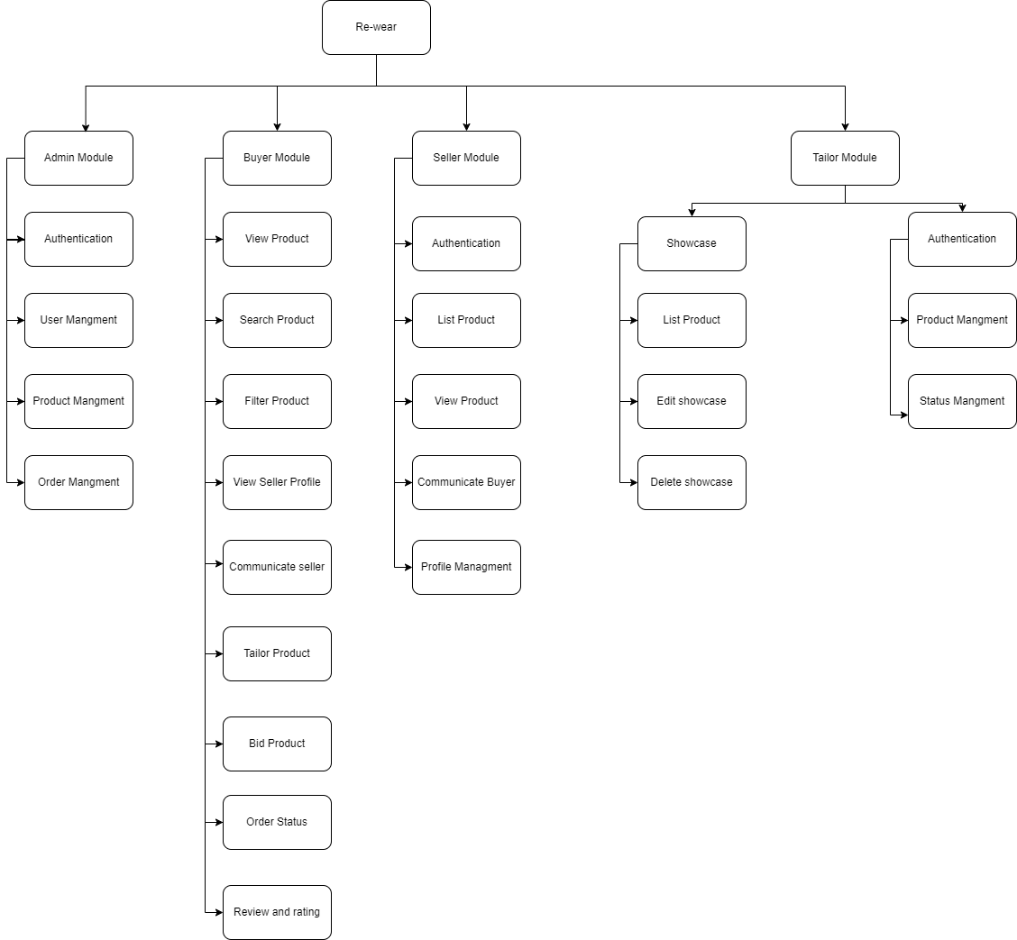
Surveys are an effective way to collect a lot of user data on preferences and buying/selling behaviors. This increases the target audience's comprehension. Deeper investigations are then possible through focused conversations with smaller groups (separate sellers, tailor and existing Market). These in-depth discussions reveal needs, obstacles, and motivations that surveys might miss. This combination offers a comprehensive understanding of user requirements, which is essential for developing an effective Re-Wear platform.

The following portion of this section describes in detail, the Online Survey process which was conducted for requirements elicitation.

* + - 1. **Important Questions**
         * What makes you want to sell pre-loved items?
         * We only get dresses to fit or stitch for tailors on Eid. Otherwise not, we don't receive enough orders and we are unaware of any apps that would allow us to acquire enough sales so that we can make a good living?
      2. **Current Software**
         * What system are used for buying, selling and tailoring?
         * Can you describe or give demo of your current workflow?
      3. **Key Features**
         * What features do you consider essential for a Platform to be effective in your role?
      4. **Security and Authentication**
         * How concerned are you about the security of your Seller Details within the system?

#### BRAINSTORMING

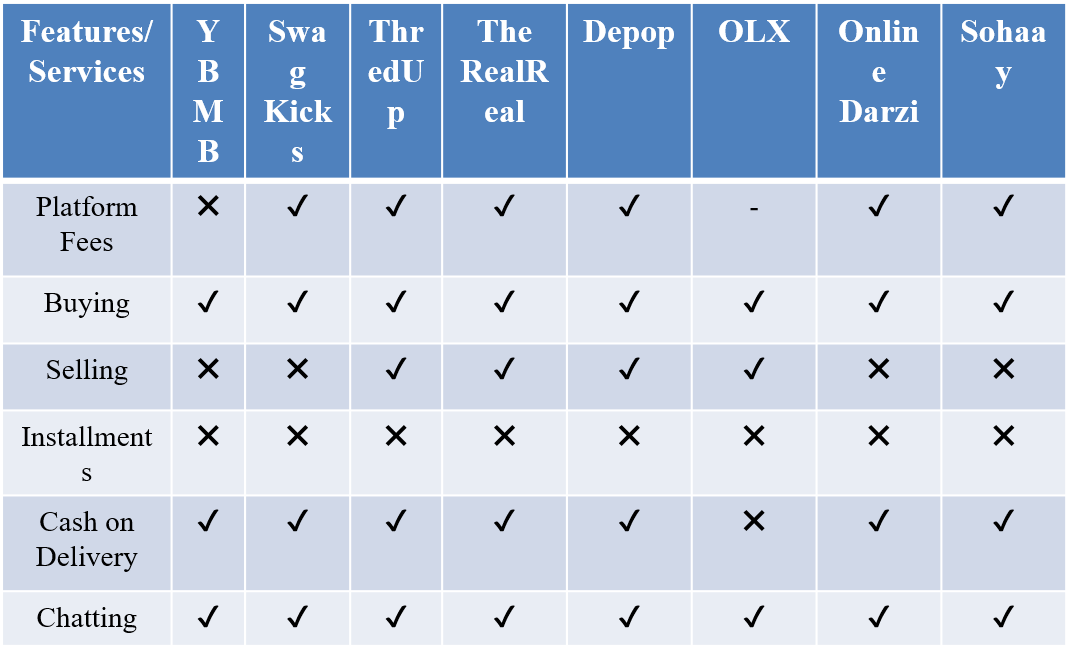
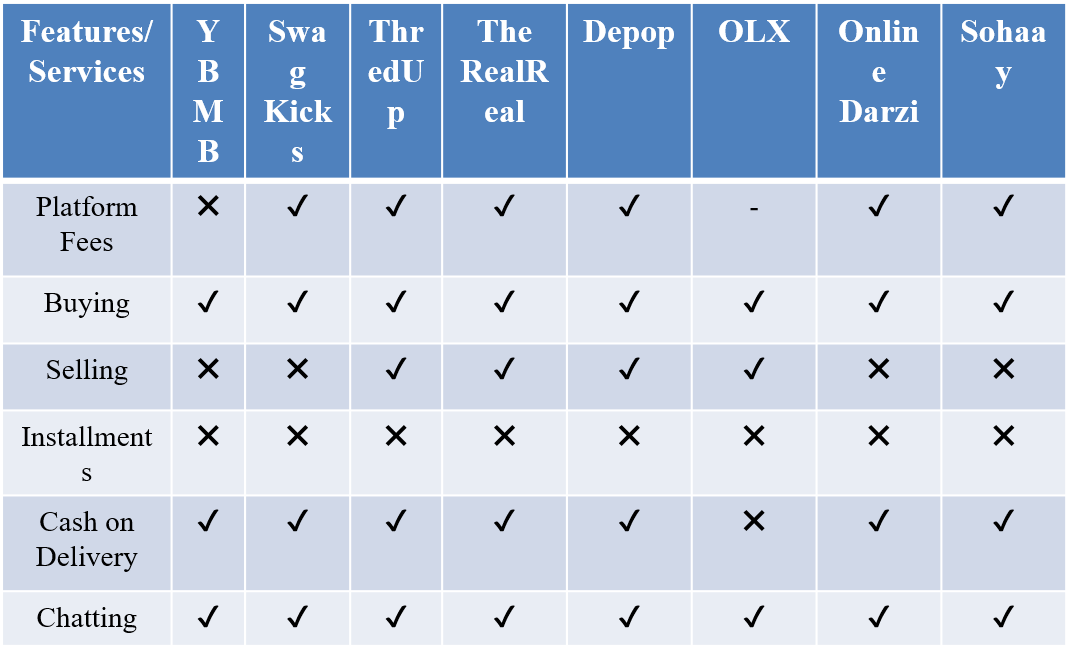
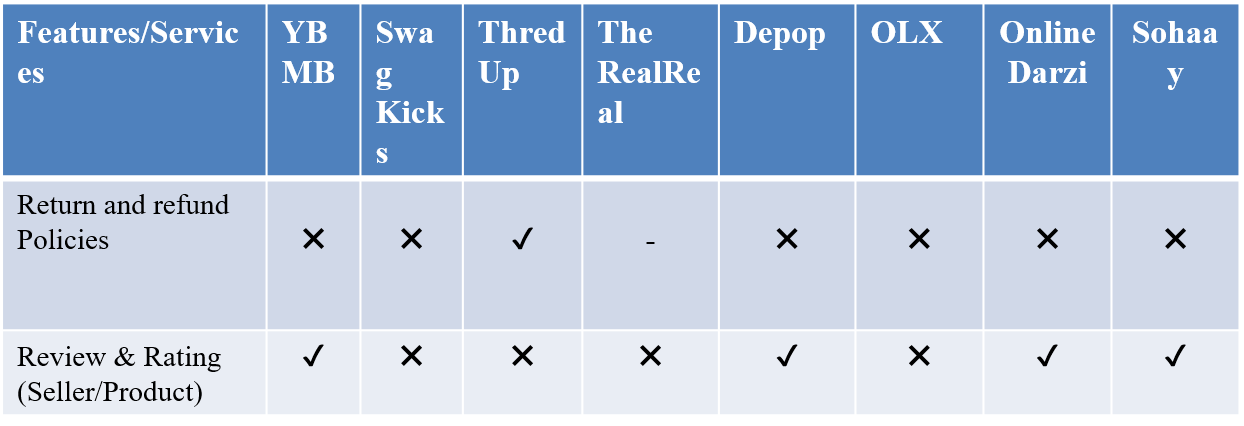
The figure 2.1 represents the major modules like Admin module, Buyer module, Seller module and Tailor module and their sub modules identified after Online Survey and brainstorming. This visual summary serves as a roadmap for the development and organization of our system.



**Figure 2.2: Mind Map**

#### EXISTING SYSTEMS

**Table 2.1: Existing Systems**



#### SUMMARY

From the above discussion Re-Wear provides a novel online platform that will alter the fashion business by encouraging sustainability. Re-wear distinguishes itself with cutting-edge features like user can create their own shops on Re-wear and customization services that are unavailable on other platforms. These new capabilities enable consumers to modify their items for a perfect fit, therefore improving cost, accessibility, and customer happiness. With its dedication to sustainability and continual innovation, Re-Wear establishes a new benchmark for ethical and ecologically responsible fashion consumption.

# Chapter 3 Requirement Analysis

### Chapter 3

## REQUIREMENT ANALYSIS

#### INTRODUCTION

In This chapter will go over the functional and non-functional criteria for our project re-wear. In advance of that, we'll go over all of the issue statements we discovered when researching the project concept. These functional requirements are acquired from the customer using a variety of methodologies, including interviews and brainstorming. The non-functional requirements are collected by monitoring the functional requirements.

#### PROBLEM SCENARIOS

* + 1. **Problem Statement**

**Table 3.0: Problem Statement**

|  |  |
| --- | --- |
| **Elements** | **Description** |
| **The Problem of** | Lack of Visibility for New Tailors on online platform |
| **Affects** | Tailor |
| **The Result of Which** | Amateur tailors find it difficult to draw in customers and make money on Online platforms |
| **Benefits** | * Attract customers * Generate income |

|  |  |
| --- | --- |
| **Elements** | **Description** |
| **The Problem of** | The cost of traditional wedding gowns and grooms might be exorbitant. |
| **Affects** | Buyer |
| **The Result of Which** | Due of the high expense, some individuals who cannot afford to attend weddings or other significant cultural and personal events may not be able to fully participate. |
| **Benefits** | * Providing an affordable and comprehensive solution * Facilitating easier access to the essential groom and bride attire for formal occasions. |

#### FUNCTIONAL REQUIREMENTS

* + 1. **Common Functional Requirements**

**Table 3.1: FR-01-Authentication**

|  |  |  |
| --- | --- | --- |
| **ID** | **Requirement** | **Priority** |
| * **Authentication** | | |
| FR-01.1 | Admin, Buyer, Tailor and Seller shall be able to login in to their account by entering their registered email and password. | High |
| FR-01.2 | Admin, Buyer, Tailor and Seller shall be able to recover their password. | High |
| * **Profile Updating** | | |
| FR-01.3 | Admin, Buyer and Seller shall be able to update their profile details i.e. name, password and profile picture. | High |

##### Tailor Module

**Table 3.2: FR-02-** **Tailor Module**

|  |  |  |
| --- | --- | --- |
| **ID** | **Requirement** | **Priority** |
| * **Show-Case Product** | | |

|  |  |  |
| --- | --- | --- |
| FR-02.1 | Tailor will be able to showcase their expertise, experience, and service offerings. | High |
| FR-02.2 | Buyer will be able to view users shop | High |
| FR-02.3 | The Tailor shall be able to view his/her Products. | High |
| FR-02.4 | The Tailor shall be able to search his Products. | High |
| FR-02.5 | The Tailor shall be able to upload his/her Products. | High |
| FR-02.6 | The Tailor shall be able to delete his/her Products. | High |
| FR-02.7 | The Tailor shall be able to view last Status of the Products. | Low |
| FR-02.8 | The Tailor shall be able to edit Products Details. | Low |
| FR-02.9 | The Tailor shall be able to Create Shop | High |

|  |  |  |
| --- | --- | --- |
| * **Order Management** | | |
| FR-02.10 | Buyers will send proceed the order, including order details, customer information, and garment specifications. | High |
| FR-02.11 | Buyer can give reviews after the purchase of product. | High |

|  |  |  |
| --- | --- | --- |
| * **Edit Profile** | | |
| FR-02.12 | Tailor will be able to edit his/her details. | High |
| FR-02.13 | Tailor will be able to view Buyer profile which will show: username, email & Collection. | High |
| FR-02.14 | The Tailor shall be able to set the status of the availability i.e. Active or away. | Medium |
| * **Profile Management** | | |
| FR-02.15 | The tailor shall be able to setup his/her profile by providing details i.e. name, profile picture, password, contact number and address | Medium |

##### Buyer Module

**Table 3.3: FR-03-Buyer Module**

|  |  |  |
| --- | --- | --- |
| **ID** | **Requirement** | **Priority** |
| * **Search** | | |
| FR-03.1 | The buyers shall be able to search product from the available category. | High |
| FR-03.2 | The buyers shall be able to filter product list by applying filters i.e. Product type, date. | High |
| * **Product** | | |
| FR-03.3 | Buyer will be able to view and purchase the Products available for sale on the marketplace. | High |
| FR-03.4 | Buyers will be able to see product condition if the product is used | High |
| FR-03.5 | Buyers will be able to select a Product and view its following details: Id, Location, Status, Size, Description, Price. | High |

|  |  |  |
| --- | --- | --- |
| * **View Seller Profile** | | |
| FR-03.6 | The buyer shall be able to view the profiles of sellers. | High |
| FR-03.7 | The buyers shall be able to contact sellers via message. | High |

|  |  |  |
| --- | --- | --- |
| * **Pricing** | | |
| FR-03.8 | The Buyer shall be able to Pay in order to purchase | High |
| FR-03.9 | The Buyer can pay from card or cash | High |
| FR-03.10 | The Buyer shall be able to proceed order after completing his profile details such as contact number, name, Location | High |

##### Admin Module

**Table 3.4: FR-04-Admin Module**

|  |  |  |
| --- | --- | --- |
| **ID** | **Requirement** | **Priority** |
| * **User Management** | | |
| FR-04.1 | The admin shall be able to verify new sellers into the system. | High |
| FR-04.2 | The admin shall be able to view users list. | High |
| FR-04.3 | The admin shall be able to edit existing user’s details. | High |
| FR-04.4 | The admin shall be able to monitor the status of users | High |
| FR-04.5 | The admin shall be able to search specific users from list. | High |
| * **Order Management** | | |
| |  |  |  | | --- | --- | --- | | FR-04.6 | The admin shall be able to view orders list. | High | | FR-04.7 | The admin shall be able to edit existing order details like delivered or processing. | High | | FR-04.8 | The admin shall be able to update the status of a order  i.e. delivered and un-delivered. | High | | FR-04.9 | The admin shall be able to search order from list | High | | | |

##### Seller Module

**Table 3.5: FR-05-Seller Module**

|  |  |  |
| --- | --- | --- |
| * **Product** | | |
| FR-05.0 | Seller will be able to select from the Product he owns, and Product to sell. | High |
| FR-05.1 | Seller will be able to list his Product which will include Product name, Type, Product material, category, size, Description, Image, Price, Condition and Quantity upload on the marketplace, for a certain price. | High |
| FR-05.2 | Users will be able to select an Product and view its following details: Id, Location, Status, Size, Description, Price. | High |
| FR-05.3 | Seller will be able to delete product from collection | High |
| FR-05.4 | Seller will be able to Sell after getting verified | High |

|  |  |  |
| --- | --- | --- |
| * **Shop** | | |
| FR-05.5 | Seller will be able to create his/her shop which includes her product | High |

|  |  |  |
| --- | --- | --- |
| * **Edit Profile** | | |
| FR-05.6 | Seller will be able to edit details. | High |
| FR-05.7 | Seller will be able to view Buyer profile which will show: username, email. | High |

#### Non-FUNCTIONAL REQUIREMENTS

|  |  |  |
| --- | --- | --- |
| * **Security in term of authorization** | | |
| FR-05.2  FR-02.15  FR-02.5 | Only registered users will be able to sell, buy and tailor like upload product, Create Shop etc. | High |

#### SQA ACTIVITIES

* + 1. **Defect Identification through Inspection**
       1. Technique: Checklist Inspection

**Table 3.6: Defect Identification through Inspection**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **R. ID** | **Requirement** | **Checklist Point** | **Defect** | **Corrected Requirement** |
| FR- 02.5 | The Tailor shall be able to upload his/her Products. | Is information provided in requirement is complete or incomplete? | Incomplete | The Tailor shall be able to upload his/her Products.by providing product details i.e. name, description, contact, pictures. |
| FR-05.6  FR-02.12 | Seller will be able to edit.  Tailor will be able to edit his/her details. | Is information provided in requirement is complete or incomplete? | Incomplete | User will be able to edit his/her details like username, address, contact Number and change password  And seller will be able to edit order which will include Product name, Type, Product material, category, size, Description, Image, Price, Condition and Quantity |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FR-04.4 | The admin shall be able to update the status of users. | Are the requirements stated simply and completely so that they are unambiguous. | Ambiguous | The admin shall be able to update the status of users activate or deactivate |
| FR- 3.8 | The Buyer shall be able to Pay in order to purchase | Is information clear or unclear? | Incomplete | The Buyer shall be able to Pay from card or cash on delivery in order to purchase |

Table 3.6 represents the defects identified from requirements through checklist inspection technique.

* + 1. **Defect Detection**

In this section we will perform defect detection upon requirements through black box testing using techniques like Equivalence Class Partitioning (ECP), Boundary Value Analysis (BVA).

* + - 1. List of Test Scenarios

It contains the test data that will be later on used in test cases.

* + - * 1. Test Data Tailor

**Table 3.7: Test Data- Register ( name)**

|  |  |
| --- | --- |
| **ID** | TD – 01.0 |
| **Requirement** | FR-01.1 |
| **Form** | Register |
| **Field** | Name |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * {{a, b, c z}, {A, B, C…Z} |
| **Invalid** | * Is Empty * {0,1,2,3…9} * Arithmetic Symbols {+, -, % …/} |

Table 3.8 represents the test data for the Register functionality, focusing on the Name field, and using Equivalence Class Partitioning.

**Table 3.8: Test Data- Login ( Email)**

|  |  |
| --- | --- |
| **ID** | TD – 01.1 |
| **Requirement** | FR-01.1 |
| **Form** | Login/ Register |
| **Field** | Email |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * Format = [username@mailserver.com](mailto:username@mailserver.com) * {{a, b, c z}, {A, B, C…Z} * {0,1,2,3…9} * Period {.} * At the rate of {@} |
| **Invalid** | * Is Empty * Email ∉ Format (doesn’t follow this format) * Arithmetic Symbols {+, -, % …/} |

Table 3.8 represents the test data for the Login functionality, focusing on the Email field, and using Equivalence Class Partitioning.

**Table 3.9: Test Data- Login (Password)**

|  |  |
| --- | --- |
| **ID** | TD – 01.2 |
| **Requirement** | FR-01.2 |
| **Form** | Login/ Register |
| **Field** | Password |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * 8 ≤ Password length ≤ 30 * {{a, b, c z}, {A, B, C…Z} * {0,1,2,3…9} * Arithmetic Symbols {+, -, % …/} * Special characters {@, #, …&} * Password ∈ Confirm Password |
| **Invalid** | * Is Empty * Password length < 8 * Password length > 30 |

Table 3.9 represents the test data for the Login functionality, focusing on the Password field, and using Equivalence Class Partitioning.

**Table 3.10: Test Data- Show-case Product ( Product Name)**

|  |  |
| --- | --- |
| **ID** | TD – 01.3 |
| **Requirement** | FR-02.1 |

|  |  |
| --- | --- |
| **Form** | Show-Case Product |
| **Field** | Product Name |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * {{a, b, c …z}, {A, B, C …Z}} * White spaces {“ ”} |
| **Invalid** | * Empty field { } * Arithmetic Symbols {+, -, % …/} * Special characters {@, #, …&} * Numeric value {0,1 2, 3 …9} |

Table 3.10 represents the test data for the Show-Case Product, focusing on the Product Name field, and using Equivalence Class Partitioning.

**Table 3.11: Test Data -Show-case Product ( Product Description)**

|  |  |
| --- | --- |
| **ID** | TD – 01.4 |
| **Requirement** | FR-02.1 |
| **Form** | Show-Case Product |
| **Field** | Description |
| **Technique** | * Equivalence Class Partitioning (ECP) |
| **Valid** | * {{a, b, c …z}, {A, B, C …Z}} * White spaces {“ ”} |

|  |  |
| --- | --- |
|  | * Numeric value {0,1 2, 3 …} * 20 ≤ Characters ≤ 150 |
| **Invalid** | * Empty field { } * 20 ≥ Characters ≥150 |

Table 3.11 represents the test data for the Show-Case Product, focusing on the Description field, and using Equivalence Class Partitioning.

**Table 3.12: Test Data- List-Product ( Product Type)**

|  |  |
| --- | --- |
| **ID** | TD – 1.6 |
| **Requirement** | FR-02.1 |

|  |  |
| --- | --- |
| **Form** | Create Product |
| **Field** | Product Type |
| **Technique** | Boundary Value Analysis (BVA) |
| **Valid** | Product Type ∈ Valid Product Type   * Valid Product Type {Wedding, Shits, Pent, Footwear, Accessories} |
| **Invalid** | Product Type ∉ Valid Product Type   * Valid Product Type {Wedding, Shits, Pent, Footwear, Accessories} |

Table 3.12 represents the test data for the Product type, focusing on the Product Type field, and using Boundary Value Analysis.

**Table 3.13: Test Data- Show-case Product ( Picture Post)**

|  |  |
| --- | --- |
| **ID** | TD – 01.5 |
| **Requirement** | FR-02.5 |
| **Form** | Show-Case Product |  |
| **Field** | Picture Upload |
| **Technique** | Boundary Value Analysis (BVA) |
| **Valid** | File Type = {PNG, JPG, JPEG } |
| **Invalid** | File Type = {RAR, ZIP, PDF, DOCS } |

Table 3.13 represents the test data for the Show-Case Product, focusing on the Picture field, and using Equivalence Class Partitioning.

**Table 3.14: Test Data- Pricing**

|  |  |
| --- | --- |
| **ID** | TD – 01.6 |
| **Requirement** | FR-02.2 |
| **Form** | Product Pricing |
| **Field** | price |
| **Technique** | Boundary Value Analysis (BVA) |
| **Valid** | * Numeric value {0,1 2, 3 …} * Rupee |
| **Invalid** | * {{a, b, c …z}, {A, B, C …Z}} * White spaces {“ ”} * Null |

Table 3.14 represents the test data for the Price, focusing on the Image field, and using Boundary Value Analysis.

**Table 3.15: Test Data- Add Image from Device ( Image)**

|  |  |
| --- | --- |
| **ID** | TD – 01.7 |
| **Requirement** | FR-05.1 |
| **Form** | Add Image from device |
| **Field** | Image |
| **Technique** | Boundary Value Analysis (BVA) |
| **Valid** | File Type = {PNG, JPG, JPEG} |
| **Invalid** | * File Type! = {PNG, JPG, JPEG} * Is Empty |

Table 3.15 represents the test data for the Add Image (from device), focusing on the Image field, and using Boundary Value Analysis.

**Table 3.16: Test Data - Order Details (Name)**

|  |  |
| --- | --- |
| **ID** | TD – 01.8 |
| **Requirement** | FR-03.4 |
| **Form** | Order details |
| **Field** | Name |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * {{a, b, c …z}, {A, B, C …Z}} * White spaces {“ ”} |

|  |  |
| --- | --- |
| **Invalid** | * Is Empty * Numeric value {0,1 2, 3 …9} * Arithmetic Symbols {+, -, % …/} * Special characters {@, #, …&} |

Table 3.16 represents the test data for the Order details, focusing on the Name field, and using Equivalence Class Partitioning.

**Table 3.17: Test Data- Order details (Description)**

|  |  |
| --- | --- |
| **ID** | TD – 01.9 |
| **Requirement** | FR-03.6 |
| **Form** | Order details |
| **Field** | Description |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * {{a, b, c …z}, {A, B, C …Z}} * White spaces {“ ”} * Numeric value {0,1 2, 3 …} * 1 ≤ Characters ≤ 150 |
| **Invalid** | * Is Empty * Email ∉ Format (doesn’t follow this format) * 1 ≥ Characters ≥150 * Arithmetic Symbols {+, -, % …/} |

Table 3.17 represents the test data for the Order Details, focusing on the Description field, and using Equivalence Class Partitioning.

**Table 3.18: Test Data- Add Order details (Email)**

|  |  |
| --- | --- |
| **ID** | TD – 2.0 |
| **Requirement** | FR-03.6 |
| **Form** | Order details |
| **Field** | Email |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * Format = [username@mailserver.com](mailto:username@mailserver.com) * {{a, b, c z}, {A, B, C…Z} * {0,1,2,3…9} * Period {.} * At the rate of {@} |
| **Invalid** | * Is Empty * Numeric value {0,1 2, 3 …9} * Arithmetic Symbols {+, -, % …/} * Special characters {@, #, …&} |

Table 3.18 represents the test data for the Order Details, focusing on the Email, and using Equivalence Class Partitioning.

**Table 3.19: Test Data- Add Order details (Contact no)**

|  |  |
| --- | --- |
| **ID** | TD – 02.1 |
| **Requirement** | FR-03.6 |
| **Form** | Order details |
| **Field** | Phone no |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * Country Code, {+92} * Numeric value {0,1 2, 3 …9} * number. Length == 12 |
| **Invalid** | * Is Empty * Special characters {@, #, …&} * {{a, b, c z}, {A, B, C…Z} * number. Length > 12 * number. Length < 12 |

Table 3.19 represents the test data for the Order Details, focusing on the phone no, and using Equivalence Class Partitioning.

**Table 3.20: Test Data- Add Order details (Adress)**

|  |  |
| --- | --- |
| **ID** | TD – 02.2 |
| **Requirement** | FR-03.6 |
| **Form** | Order details |
| **Field** | Location |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * {City only} * {{a, b, c z}, {A, B, C…Z} * {Postal code} * Special characters {@, #, …&}, {street address} |
| **Invalid** | * Is Empty * {country} * Arithmetic Symbols {+, -, % …/} |

Table 3.20 represents the test data for the Order Details, focusing on the location, and using Equivalence Class Partitioning.

**Table 3.21: Test Data- Add Order details (payment)**

|  |  |
| --- | --- |
| **ID** | TD – 2.3 |
| **Requirement** | FR-02.10, FR-02.10 |
| **Form** | Order details |
| **Field** | card |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * Pin= {0,1,2,3…9} * CVC == 4 Digit || CVC ==3 Digit * Cash-holder name= {a, b, c z}, {A, B, C…Z} * Expiry date status= {not expired} |
| **Invalid** | * Is Empty * Arithmetic Symbols {+, -, % …/} * Special characters {@, #, …&} |

Table 3.21 represents the test data for the Order pricing, focusing on the Email, and using Equivalence Class Partitioning.

**Table 3.22: Test Data- Buyer, Seller Profile ( Name)**

|  |  |
| --- | --- |
| **ID** | TD – 02.3 |
| **Requirement** | FR-02.14 & FR-5.6 |
| **Form** | Buyer & Seller Profile |
| **Field** | Name |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * {{a, b, c …z}, {A, B, C …Z}} * White spaces {“ ”} |
| **Invalid** | * Empty field { } * Numeric value {0,1 2, 3 …9} * Arithmetic Symbols {+, -, % …/} * Special characters {@, #, …&} |

Table 3.22 represents the test data for the buyer and seller profile functionality, focusing on the Name field, and using Equivalence Class Partitioning.

**Table 3.23: Test Data- Seller Profile ( verification)**

|  |  |
| --- | --- |
| **ID** | TD – 02.4 |
| **Requirement** | FR-5.5 |
| **Form** | Seller Verification |
| **Field** | Identity |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * {{Profile picture}} * {ID card front}, {ID card back} * File Type = {PNG, JPG, JPEG} |
| **Invalid** | * Empty field { } * File Type =! {PNG, JPG, JPEG} |

Table 3.23 represents the test data for the seller verification, focusing on the Verification, and using Equivalence Class Partitioning.

**Table 3.24: Test Data- Search Product ( Search Bar)**

|  |  |
| --- | --- |
| **ID** | TD – 02.5 |
| **Requirement** | FR-03.1 |
| **Form** | Search product |
| **Field** | Search Bar |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * {{a, b, c …z}, {A, B, C …Z} * Numeric value {0,1 2, 3 …9} * White spaces {“ ”} * Arithmetic Symbols {+, -, % …/} * Special characters {@, #, …&} |

|  |  |
| --- | --- |
| **Invalid** | * Empty field { } |

Table 3.24 represents the test data for the Search Product functionality, focusing on the Search Bar field, and using Equivalence Class Partitioning.

**Table 3.25: Test Data- List Product ( Name)**

|  |  |
| --- | --- |
| **ID** | TD – 02.6 |
| **Requirement** | FR-5.2 |
| **Form** | List Product |
| **Field** | Product Name |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * {{a, b, c …z}, {A, B, C …Z}} * White spaces {“ ”} |
| **Invalid** | * Is Empty * Numeric value {0,1 2, 3 …9} * Arithmetic Symbols {+, -, % …/} * Special characters {@, #, …&} |

Table 3.25 represents the test data for the Product Name, focusing on the Name field, and using Equivalence Class Partitioning.

**Table 3.26: Test Data- List Product ( Image)**

|  |  |
| --- | --- |
| **ID** | TD – 02.7 |
| **Requirement** | FR-5.2 |
| **Form** | List Product |
| **Field** | Image |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * File Type = {PNG, JPG, JPEG } |
| **Invalid** | * Is Empty * File Type =! {PNG, JPG, JPEG } |

Table 3.26 represents the test data for the Product Image, focusing on the Image, and using Equivalence Class Partitioning.

**Table 3.27: Test Data- List Product ( Products type)**

|  |  |
| --- | --- |
| **ID** | TD – 02.8 |
| **Requirement** | FR-5.2 |
| **Form** | List Product |
| **Field** | Product type |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * Used * New |
| **Invalid** | * Is Empty |

Table 3.27 represents the test data for the Product Condition, focusing on the Product Type, and using Equivalence Class Partitioning.

**Table 3.28: Test Data- List Product ( Category)**

|  |  |
| --- | --- |
| **ID** | TD – 02.9 |
| **Requirement** | FR-5.2 |
| **Form** | List Product |
| **Field** | Category |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * “Required = choose" |
| **Invalid** | * Is Empty |

Table 3.28 represents the test data for the Product Condition, focusing on the Category, and using Equivalence Class Partitioning.

**Table 3.29: Test Data- List Product ( Material)**

|  |  |
| --- | --- |
| **ID** | TD – 03.0 |
| **Requirement** | FR-5.2 |
| **Form** | List Product |
| **Field** | Material |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * {{a, b, c …z}, {A, B, C …Z}} * White spaces {“ ”} |
| **Invalid** | * Is Empty * Numeric value {0,1 2, 3 …9} * Arithmetic Symbols {+, -, % …/} * Special characters {@, #, …&} |

Table 3.29 represents the test data for the Product Condition, focusing on the Material, and using Equivalence Class Partitioning.

**Table 3.30: Test Data- List Product ( Size)**

|  |  |
| --- | --- |
| **ID** | TD – 03.1 |
| **Requirement** | FR-5.2 |
| **Form** | List Product |
| **Field** | Size |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * Numeric value {0,1 2, 3 …9} * {{a, b, c …z}, {A, B, C …Z}} |
| **Invalid** | * Is Empty * Arithmetic Symbols {+, -, % …/} * Special characters {@, #, …&} |

Table 3.30 represents the test data for the Product Condition, focusing on the Size, and using Equivalence Class Partitioning.

**Table 3.31: Test Data- List Product ( Description)**

|  |  |
| --- | --- |
| **ID** | TD – 03.3 |
| **Requirement** | FR-5.2 |
| **Form** | List Product |
| **Field** | Description |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * Numeric value {0,1 2, 3 …9} * {{a, b, c …z}, {A, B, C …Z}} * Arithmetic Symbols {+, -, % …/} * Special characters {@, #, …&} * 1 ≤ Characters ≤ 150 |
| **Invalid** | * Is Empty * 1 ≥ Characters ≥150 |

Table 3.31 represents the test data for the Product Condition, focusing on the Description, and using Equivalence Class Partitioning.

**Table 3.32: Test Data- List Product ( Price)**

|  |  |
| --- | --- |
| **ID** | TD – 03.4 |
| **Requirement** | FR-5.2 |
| **Form** | List Product |
| **Field** | Price |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * Numeric value {0,1 2, 3 …9} |
| **Invalid** | * Is Empty * {{a, b, c …z}, {A, B, C …Z}} * Arithmetic Symbols {+, -, % …/} * Special characters {@, #, …&} |

Table 3.32 represents the test data for the Product Condition, focusing on the Price, and using Equivalence Class Partitioning.

**Table 3.33: Test Data- List Product ( Quantity)**

|  |  |
| --- | --- |
| **ID** | TD – 03.5 |
| **Requirement** | FR-5.2 |
| **Form** | List Product |
| **Field** | Quantity |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * Numeric value {0,1 2, 3 …9} |
| **Invalid** | * Is Empty * {{a, b, c …z}, {A, B, C …Z}} * Arithmetic Symbols {+, -, % …/} * Special characters {@, #, …&} |

Table 3.33 represents the test data for the Product Condition, focusing on the Quantity, and using Equivalence Class Partitioning.

**Table 3.34: Test Data- List Product ( Condition)**

|  |  |
| --- | --- |
| **ID** | TD – 03.6 |
| **Requirement** | FR-5.2 |
| **Form** | List Product |
| **Field** | Condition |
| **Technique** | Equivalence Class Partitioning (ECP) |
| **Valid** | * Numeric value {0,1 2, 3 …10} |
| **Invalid** | * Is Empty * {{a, b, c …z}, {A, B, C …Z}} * Arithmetic Symbols {+, -, % …/} * Special characters {@, #, …&} |

Table 3.34 represents the test data for the Product Condition, focusing on the Condition, and using Equivalence Class Partitioning.

* + 1. **Black Box Test Case Design**

Login

**Table 3.28: B.B. Test Case - Login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TC.ID** | **Input Fields** | | **Reference**  **- ECP** | **Expected Output** |
| **Email** | **Password** |
| **1.1** | N/A | N/A | Table 3.8,  3.9 | Please fill all fields. |
| **1.2** | Talha@gmail.com | Talha@12345 | Table 3.8,  3.9 | Registered Successfully |
| **1.3** | xyz | 12345678 | Table 3.8,  3.9 | Please Enter Valid Email. |
| **1.4** | Talha@gmail.com | 12345678 | Table 3.8,  3.9 | Email already  Registered. |
| **1.5** | sfwan@gmail.com | sfwan@12345 | Table 3.8,  3.9 | Registered successfully |

Create showcase

**Table 3.29: B.B. Test Case - Create showcase**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Input Field** | | | | | |  |  |
| **TC ID** | **Name** | **Descript ion** | **Descript ion Length** | **Product**  **Type** | **Picture** | **Price** | **Reference-ECP** | **Excepted Output** |
| 1.1 | Wedding dress | Wedding Dress with gold water | 15 | N/A | PDF | **25,000**  **35,000** | Table 3.10-3.14 | Product File type is incorrect and product type is not selected |
| 1.2 | @!  &^ | Dress with precious work | 15 | N/A | N/A | Table 3.10-3.14 | Product  name not  valid.  Please  enter  valid  name  name. |
| 1.3 | N/A | N/A | 0 | N/A | N/A | N/A | Table 3.10-3.14 | Please fill all fields |
| 1.4 | Wedding dress | Wedding Dress with gold water | 15 | Women-wear | PDF | **25,000** | Table 3.10-3.14 | Showcase created successfully |

Create Order

**Table 3.30: B.B. Test Case - Create Order**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Input Field** | | | | | | | |  |  |
| **TC ID** | **Name** | **Description** | **Email** | **Phone no** | **Adress** | **Payment** | | | **Reference-ECP** | **Excepted Output** |
| **Exp-date** | **Name** | **CVC** |
| 1.1 | Talha | Order will be delivered in 3 days | talhagmail.com | +92315 | UK | 2019 | Sfwan | **201** | Table 3.16-3.21 | The Provided field are incorrect |
| 1.2 | N/A | Order will be delivered in 3 days | N/A | N/A | N/A | N/A | N/A | N/A | Table 3.16-3.21 | Please fill all fields |
| 1.3 | Talha | Order will be delivered in 3 days | talha@gmail.com | +92315 | isl | 2026 | Talha | **2013** | Table 3.16-3.21 | Order created successfully |

Edit Profile

Name

**Table 3.31: B.B. Test Case - Name**

|  |  |  |  |
| --- | --- | --- | --- |
| **TC.ID** | **Input** | **Reference - ECP** | **Expected Output** |
| **Name** |
| 1.1 | N/A | Table 3.16 | Please enter name |
| 1.2 | Talha | Table 3.16 | Valid |
| 1.3 | !$%#^@ | Table 3.16 | Invalid name. name not updated |

Edit Profile

Contact Number

**Table 3.32: B.B. Test Case – Contact Number**

|  |  |  |  |
| --- | --- | --- | --- |
| **TC.ID** | **Input** | **Reference - ECP** | **Expected Output** |
| **Number** |
| 1.1 | N/A | Table 3.19 | Please enter Number |
| 1.2 | +923365940220 | Table 3.19 | Valid |
| 1.3 | 0322cjdnjc | Table 3.19 | Invalid Number. |

Edit Profile

Change Password

**Table 3.33: B.B. Test Case - Password**

|  |  |  |  |
| --- | --- | --- | --- |
| **TC.ID** | **Input** | **Reference - ECP** | **Expected Output** |
| **Password=confirm password** |
| 1.1 | N/A | Table 3.9 | Please enter Password |
| 1.2 | Talha@123 | Table 3.9 | Valid |

Profile Photo

**Table 3.34: B.B. Test Case - Profile Photo**

|  |  |  |  |
| --- | --- | --- | --- |
| **TC.ID** | **Input** | **Reference - ECP** | **Expected Output** |
| **Image** |
| **1.1** | N/A | Table 3.13 | No pic selected |
| **1.2** | Abc.io | Table 2.5 | Invalid image format. Image not updated. |
| **1.3** | Pic.jpeg | Table 2.5 | Valid image format. |

Register

**Table 3.35: B.B. Test Case - Register**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TC.ID** | **Input Fields** | | | **Reference**  **- ECP** | **Actual Output** |
| **Name** | **Email** | **Password** |
| **1.1** | Talha Masood | [Talha@gmail.com](mailto:Talha@gmail.com) | Talha@12345 | Table 3.8,  3.9 | Register successful. System redirected the user to respective Dashboard. |
| **1.2** | Talha | [Talhagmail.com](mailto:Talha@gmail.com) | 123 | Table 3.8,  3.9 | Password or email is incorrect |
| **1.3** | H20 | xyz | 12345678 | Table 3.8,  3.9 | Name and Email is incorrect |
| **1.4** | N/A | N/A | N/A | Table 3.8,  3.9 | Please fill the field |

List Product

**Table 3.36: B.B. Test Case - List Product**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Input Field** | | | | | | | |  |  |  |
| **TC ID** | **Name** | **Type** | **Material** | **Category** | **Size** | **Description** | **Image** | **Quantity** | **condition** | **Reference-ECP** | **Excepted Output** |
| 1.1 | Dress | Used | Silicon | Girl dress | M | Good dress | Pdf | 1 | 3.7 | Table 3.25-3.34 | Product File type is  incorrect |
| 1.2 | Dress | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | Table 3.25-3.34 | Fiel is empty |
| 1.3 | @1 | used | 35 | Girls dress | 2222 | N/A | pdf | N/A | N/A | Table 3.25-3.34 | Incorrect field |
| 1.4 | Dress | New | Cotten | Girl dress | M | Good dress | jpeg | 1 | 3.7 | Table 3.25-3.34 | Listed Successfully |

Edit Product

**Table 3.37: B.B. Test Case - Edit Product**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Input Field** | | | | | | | |  |  |  |
| **TC ID** | **Name** | **Type** | **Material** | **Category** | **Size** | **Description** | **Image** | **Quantity** | **condition** | **Reference-ECP** | **Excepted Output** |
| 1.1 | Dress | Used | Silicon | Girl dress | M | Good dress | Pdf | 1 | 3.7 | Table 3.25-3.34 | Product File type is  incorrect |
| 1.2 | Dress | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | Table 3.25-3.34 | Fiel is empty |
| 1.3 | @1 | used | 35 | Girls dress | 2222 | N/A | pdf | N/A | N/A | Table 3.25-3.34 | Incorrect field |
| 1.4 | Dress | New | Cotten | Girl dress | M | Good dress | jpeg | 1 | 3.7 | Table 3.25-3.34 | Product Edited Successfully |

##### Summary

As demonstrated above, we used brainstorming and the current system to write all of the functional requirements. Additionally, after examining the functional requirements of the system, we drafted the non-functional requirements that we felt were essential

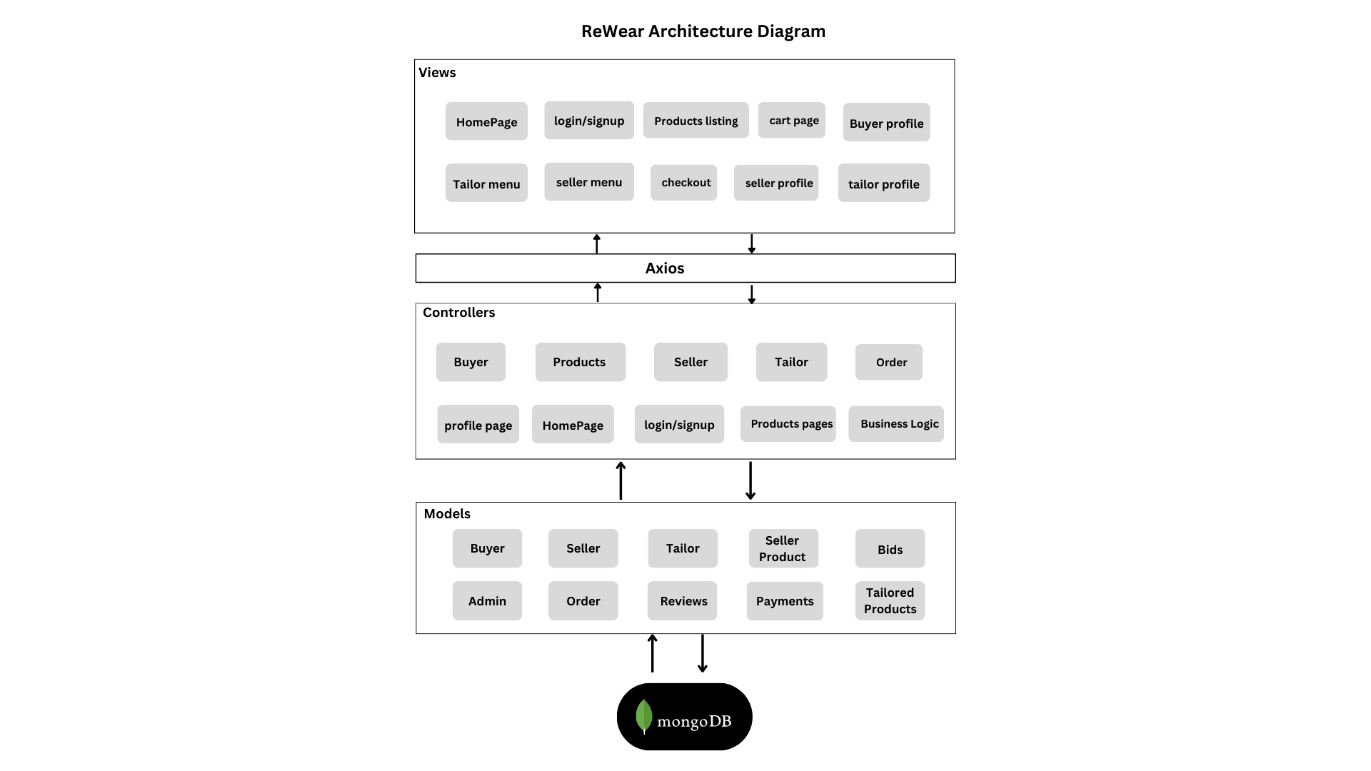
# Chapter 4 System Design

#### INTRODUCTION

The system designer is a role who defines the responsibilities, attributes, operations, and relationships of components of the software and determines how a component should be adopted to conform to the implementation environment. Design is based upon the requirements elicited from the user. In this will chapter, we will show all the design diagrams like architectural design, use case design, and activity design of our project “Re-wear”. We will illustrate use case diagrams as well as detailed use case textual details in a fully dressed format. In all the diagrams we will try to draw out the workflow and technical design of system.

#### ARCHITECTURAL DESIGN

##### Software Architecture Diagram

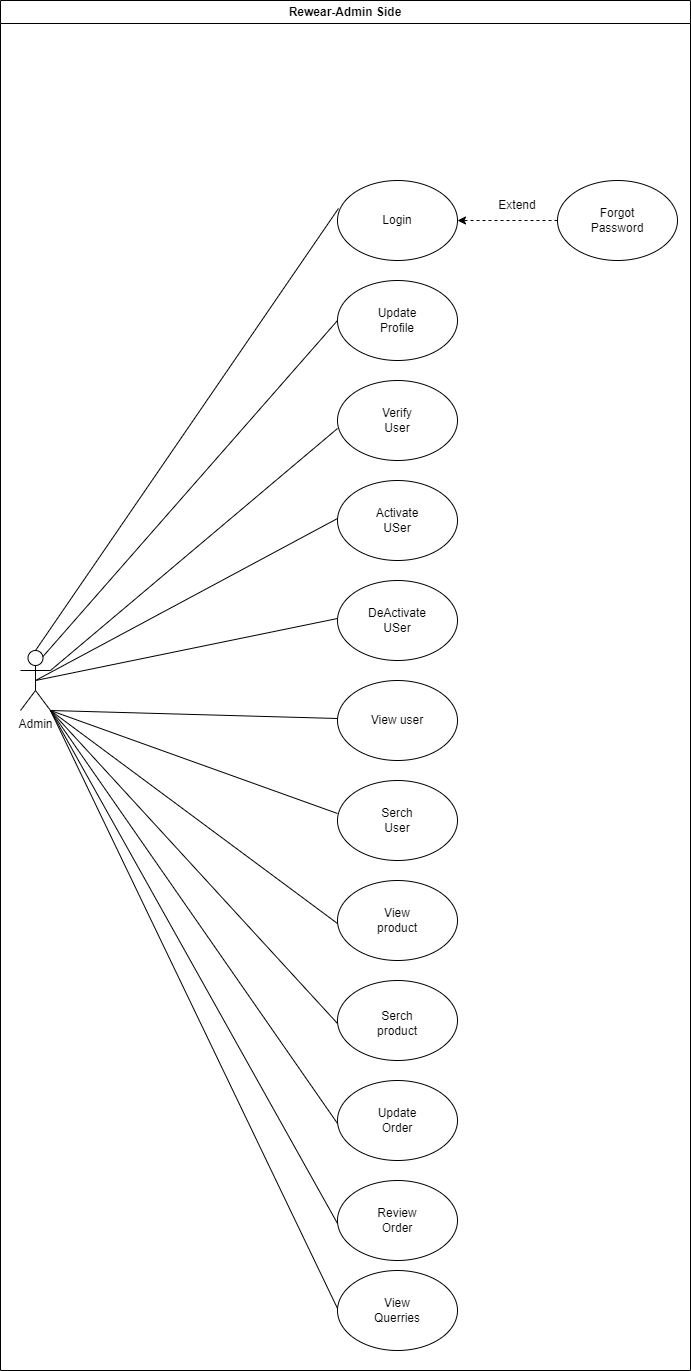
****

**Figure 4.1: Software Architecture Diagram**

#### DETAILED DESIGN

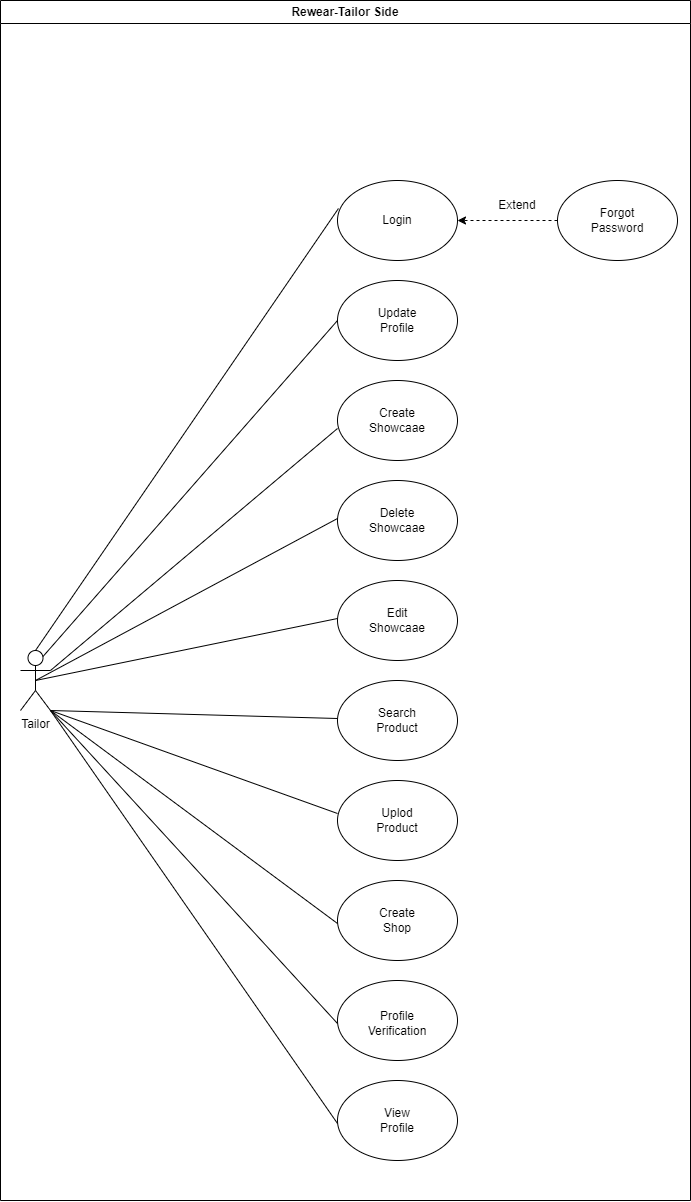
##### Use Case Diagrams

* + - 1. Use Case Diagram - Admin



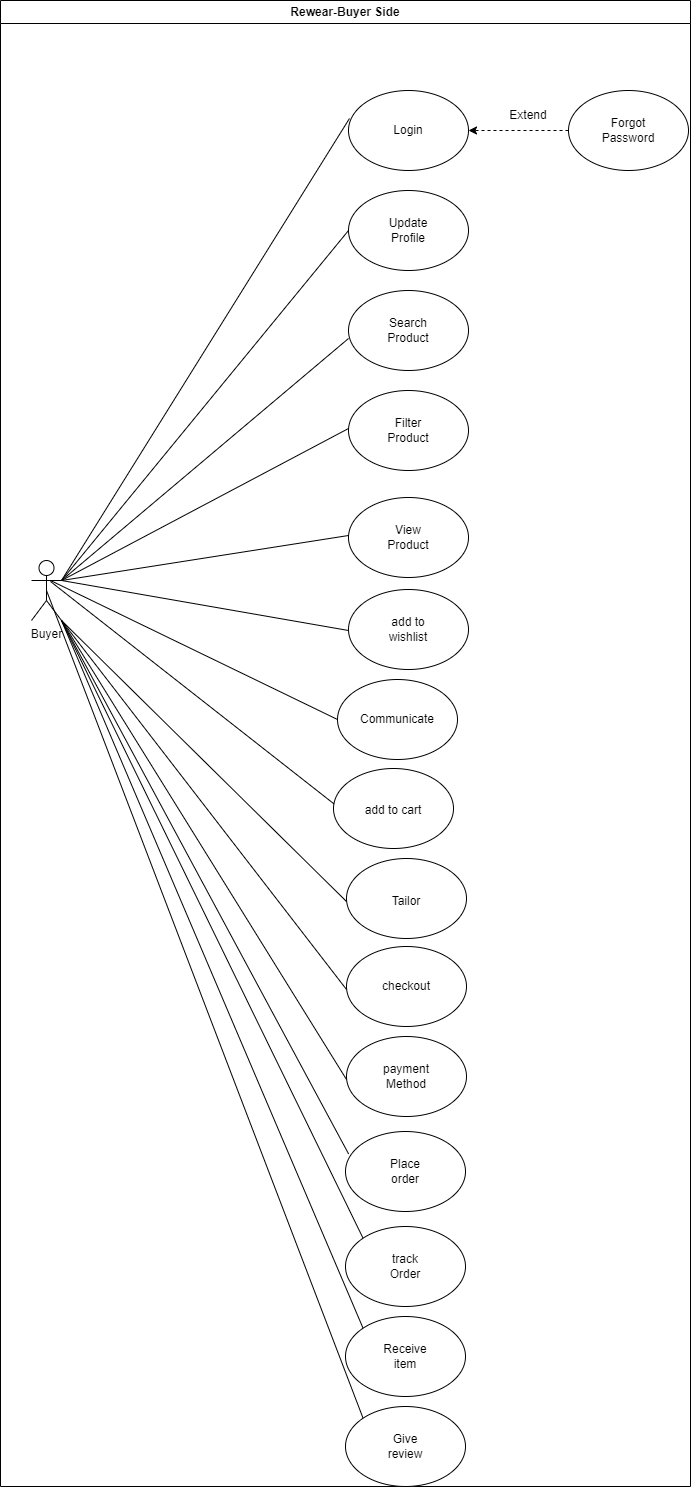
**Figure 4.2: Use Case Diagram - Admin**

* + - 1. Use Case Diagram – Tailor



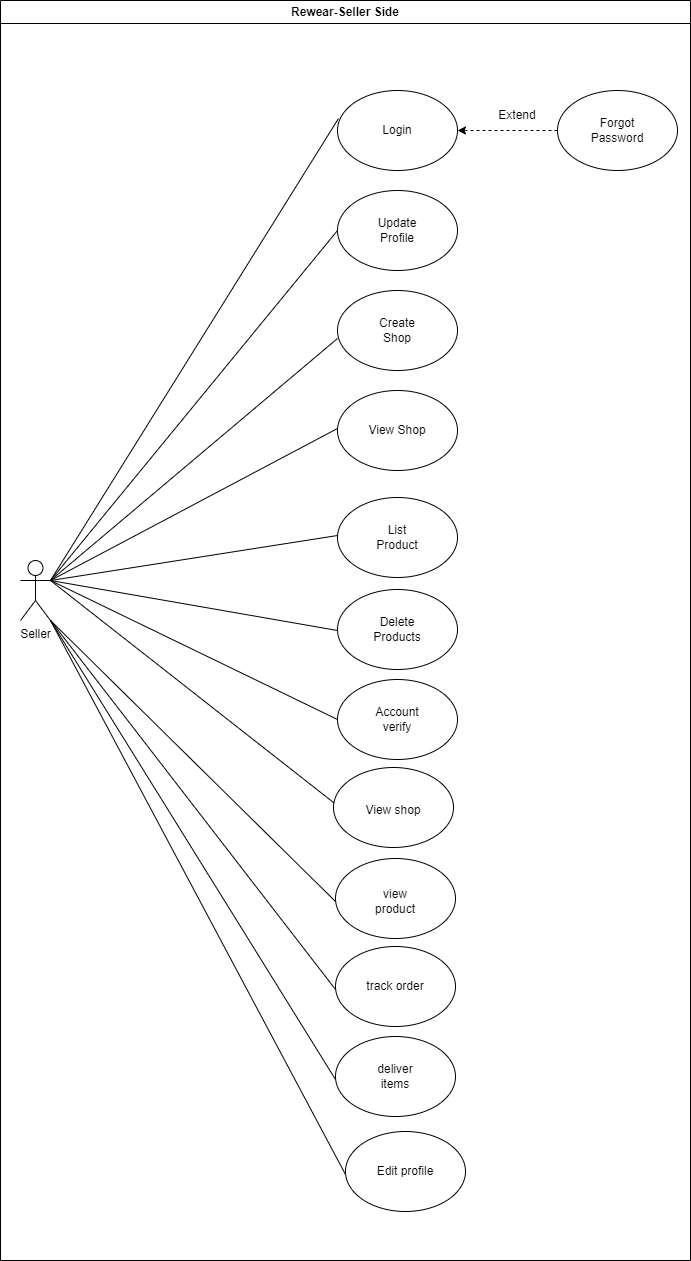
**Figure 4.3: Use Case Diagram - Tailor**

* + - 1. Use Case Diagram – Buyer



**Figure 4.4: Use Case Diagram - Buyer**

* + - 1. Use Case Diagram – Seller



**Figure 4.5: Use Case Diagram - Seller**

##### Use Case Fully Dressed Format

**Register**

**Register Account**

**Table 4.0: Use Case Fully Dressed – Register**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-00 | |
| **Use Case Name** | Register | |
| **Actors** | Buyer | |
| **Preconditions** | System must be running, and user must have enter name, email, name and password in the system. | |
| **Basic Flow** | User will be able to register by entering email, name and password | |
| **Main Success Scenario** | **Action**   1. User will enter his/her email, name and password. 2. User presses the “Submit” button. | **Response**   1. The system validates and authenticates the email, name and password. 2. System will ask user to login into the system |
| **Exceptional Flow** | * 1. User enters invalid Email.      1. System displays error message and ask the user to enter correct email.      2. User re-enters email and press login.   2. User left empty field.      1. System displays error message and ask the user to fill all fields.      2. User renters all field and press login.   3. User enter password that does not match with confirm password      1. System displays error message and ask the user to enter correct password   4. User enters invalid Name.      1. System displays error message and ask the user to fill Name.      2. User renters name field and press login.   5. User enter invalid Password      1. System displays error message and ask the user to fill Password.      2. User renters Password field and press login. | |
| **Post Conditions** | User successfully registers. | |

**Login**

**Table 4.1: Use Case Fully Dressed - Login**

|  |  |
| --- | --- |
| **Use Case ID** | UC-01 |
| **Use Case Name** | Login |
| **Actors** | User |

|  |  |  |
| --- | --- | --- |
| **Preconditions** | System must be running, and user must have email &  password. | |
| **Basic Flow** | User will be able to login by entering email and  password. | |
| **Main Success Scenario** | **Action**   1. User will enter his/her credentials (Email & Password). 2. User presses the “Login” button. | **Response**  3. The system validates and authenticates the credentials and displays a Login success message. |
| **Exceptional Flow** | * 1. User enters invalid email or password      1. System displays error message and ask the user to enter correct email or password.      2. User renter’s email or password and press login. | |
| **Post Conditions** | User will be re-directed to their respective dashboard. | |

**Forgot Password**

**Table 4.2: Use Case Fully Dressed - Forgot Password**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-02 | |
| **Use Case Name** | Forgot Password | |
| **Actors** | User | |
| **Preconditions** | System must be running, and user must have same email  which is registered in the system. | |
| **Basic Flow** | User will be able to login by entering username and  password. | |
| **Main Success Scenario** | **Action**   1. User will enter his/her email. 2. User presses the “Submit” button. | **Response**   1. The system validates and authenticates the email. 2. System sends an email to the user to reset his password. |
| **Exceptional Flow** | * 1. User enters email.      1. System displays message No user exists and ask the user to enter correct email.      2. System will send recovery Email to user. | |
| **Post Conditions** | User successfully recover his password. | |

**Verify User**

**Table 4.3: Fully Dressed Use Case - Verify User**

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC-03 | |
| Use Case | verify User | |
| Actors | Admin, Seller | |
| Preconditions | The admin must be logged into his account and should  View Seller profile and verify them. | |
| Basic Flow | Admin will be able verify seller into the system. | |
| Main Success Scenario | **Action**  1. Seller will click on “Profile” button.   1. Seller will fill the form fields i.e. Name, front image, back image, face identity, Contact Number, Location. 2. The Seller presses the “Submit” button. | **Response**   1. System will display the “Seller Verification” form. 2. System Validates the entered information. 3. System will display “verification” success message. |
| Exceptional Flow | * 1. Seller left empty fields.      1. System displays error message and ask the seller to fill all empty fields.      2. Seller fill all the fields and press Submit.   2. The Name format for seller identity verification is not supported      1. Seller will re-enter name again.   3. Seller enters invalid Contact Number. | |

|  |  |
| --- | --- |
|  | * + 1. System displays error message and ask the user to enter correct Contact no.     2. Seller renters Contact no.   1. Seller enters Name that does not match with CNIC or invalid name.      1. System will display error message and ask the user to enter Correct name.      2. Seller will reenter the Name.   2. Seller Upload Face identity that does not Match with CNIC      1. System will display error message that Face dose not match with CNIC      2. Seller will Upload the Face Identity again. |
| Post Conditions | The Seller will be successfully Verified and able to sell |

**Deactivate User**

**Table 4.4: Fully Dressed Use Case –Deactivate User**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-04 | |
| **Use Case** | Deactivate User | |
| **Actors** | Admin | |
| **Preconditions** | The admin must be logged into his account and should be  on manage user page. | |
| **Basic Flow** | Admin will be able to Activate or Deactivate user from the system. | |
| **Main Success Scenario** | **Action**  1. Admin presses the “Disable” button against a particular user.  3. Admin clicks “Confirm” button on re-confirmation  message. | **Response**  2. System will display a re-confirmation message.  4. The system will disable the user. |

|  |  |
| --- | --- |
| **Alternative Flow** | * 1. System will ask Reconfirm deactivation      1. Admin selects ‘Deactivate upon reconfirmation. Use-case ends.      2. Admin selects ‘Cancel’ upon reconfirmation. Use-case ends. |
| **Post Conditions** | The admin has successfully disabled user from the  system. |

**View User**

**Table 4.5: Fully Dressed Use Case - View User**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-05 | |
| **Use Case** | View User | |
| **Actors** | Admin | |
| **Preconditions** | The admin must be logged into his account and should be  on home page. | |
| **Basic Flow** | Admin will be able to view user’s details. | |
| **Main Success Scenario** | **Action**   1. Admin will access ‘Manage User’ page. 2. Admin clicks “Confirm” button on re-confirmation   message. | **Response**   1. System will re-direct user to Manage User’ page. 2. System will display the details of that user. |
| **Alternative Flow** | * 1. System will Display about User.      1. System will display that no user exists      2. System will display all the user details | |
| **Post Conditions** | Admin views the User’s details. | |

**User Querry**

**Table 4.6: Fully Dressed Use Case – User Querry**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-06 | |
| **Use Case** | User Querry | |
| **Actors** | Admin | |
| **Preconditions** | The admin must be logged into his account and should  be on manage user page. | |
| **Basic Flow** | Admin will be able to edit user’s details. | |
| **Main Success Scenario** | **Action**  **1** Admin will click ‘view button against a particular user.  **3** Admin will look into the problem of that user & clicks on “Done” button. | **Response**  **2** System will display the queries.  **4** System will solve issue update the user.  System will display “Success” success message. |
| **Exceptional Flow** | * 1. System will display View button against user query      1. System will display notify user if Query is solved      2. If the query is solved use case will end. | |
| **Post Conditions** | Admin successfully address the user’s query. | |

**Search User**

**Table 4.7: Fully Dressed Use Case - Search User**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-07 | |
| **Use Case** | Search User | |
| **Actors** | Admin | |
| **Preconditions** | The admin must be logged into his account and should  be on manage user page. | |
| **Basic Flow** | Admin will be able to search user from the list. | |
| **Main Success Scenario** | **Action**  1. Admin will enter query in the search bar and click on search button. | **Response**  2. System will display result set. |
| **Exceptional Flow** | * 1. The user will enter query      1. The System will display no result found      2. The system will display search user details | |
| **Post Conditions** | Admin successfully searches the user from the list. | |

**Verify Order**

**Table 4.8: Fully Dressed Use Case - Verify Order**

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC-08 | |
| Use Case | Verify Order | |
| Actors | Admin, Seller, Buyer | |
| Preconditions | The admin must be logged into his account and should  View Order and verify. | |
| Basic Flow | Admin will be able verify Order into the system. | |
| Main Success Scenario | **Action**  1. Admin will click on “view Order” button.   1. Admin will view the form fields i.e. Name, email, Contact No, Location, Product name, Product quantity. 2. The admin presses the “Submit” button. | **Response**   1. System will display the “Order” form. 2. System Validates the entered information. 3. System will display “verification” success message. |
| Exceptional Flow | * 1. Seller Enter name      1. Admin will confirm name and enter correct Name.   2. Seller Enter address      1. Admin will confirm address and enter correct address.   3. Seller Enter email      1. Admin will confirm email and enter correct email.   4. Seller Enter Contact      1. Admin will confirm contact and enter correct location.   5. System will show product name      1. Admin will confirm Product and enter correct Product.   6. System will show product quantity      1. Admin will confirm Product Quantity and enter correct Product Quantity. | |
| **Post Conditions** | Admin successfully verify the user | |

**View Product**

**Table 4.9: Fully Dressed Use Case - View Product**

|  |  |
| --- | --- |
| **Use Case ID** | UC-9 |
| **Use Case** | View Product |
| **Actors** | Admin |
| **Preconditions** | The admin must be logged into his account and should be  on manage Product page. |
| **Basic Flow** | Admin will be able to view subject’s information |
| **Main Success Scenario** | |  |  | | --- | --- | | **Action**  1. Admin will click on  manage Product option. | **Response**  2. System will display all the Products. | |
| **Exceptional Flow** | * 1. System will show product exist   2. 1 System will show no product exist   1.1.2 System will show product details |
| **Post Conditions** | Admin views the Product details. |

**Search Product**

**Table 4.10: Fully Dressed Use Case - Search Product**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-10 | |
| **Use Case** | Search Product | |
| **Actors** | Admin | |
| **Preconditions** | The admin must be logged into his account and should  be on manage Product page. | |
| **Basic Flow** | Admin will be able to search Product from the list. | |
| **Main Success Scenario** | **Action**  1. Admin will enter query in the search bar and click on search button. | **Response**  2. System will display result set. |
| **Exceptional Flow** | * 1. System will Search product exist   2. 1 System will show Search product   1.1.2 System will show no Search exist | |
| **Post Conditions** | Admin successfully searches the products from the list. | |

**Filter Product**

**Table 4.11: Fully Dressed Use Case - Filter Product**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-11 | |
| **Use Case** | Filter Product | |
| **Actors** | Buyer | |
| **Preconditions** | Buyer must be on website’s “view Product” page. | |
| **Basic Flow** | Client must be able to filter the Product search result. | |
| **Main Success Scenario** | **Action**  1. Client will apply the filters from the given categories i.e. Date,  location, Type. | **Response**  2. System filters & displays the Product list accordingly. |

|  |  |
| --- | --- |
| **Alternative Flow** | 1.1. Filtered product  1.1.1. The system will display the filter result  1.1.2. There is no Product against the chosen filters, the system displays a No Product found message. |
| **Post Conditions** | Client successfully filtered the Product list from search  result. |

**View Seller Profile**

**Table 4.12: Fully Dressed Use Case - View Seller Profile**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-12 | |
| **Use Case** | View Seller Profile | |
| **Actors** | Buyer | |
| **Preconditions** | The Buyer must be on website’s landing page. | |
| **Basic Flow** | Buyer will be able to see Seller profile. | |
| **Main Success Scenario** | **Action**  1. Client will visit website’s “Landing page”. | **Response**  2. System will display the list of all the Sellers. |

|  |  |  |
| --- | --- | --- |
|  | 3. Buyer will click on profile of a particular Seller. | 4. System will display the detailed profile of that Seller. |
| **Alternative Flow** | 3.1. Seller profile  3.1.1. The system will display Message no seller exists.  3.1.2. The system will display Profile of Seller | |
| **Post Conditions** | The User can view Profiles of Seller. | |

**Contact Seller**

**Table 4.13: Fully Dressed Use Case - Contact Seller**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-13 | |
| **Use Case** | Contact Seller | |
| **Actors** | Buyer | |
| **Preconditions** | The Buyer must be on website’s “Profile” page. | |
| **Basic Flow** | Buyer will be able to contact Seller. | |
| **Main Success Scenario** | **Action**  1. Buyer will click on profile of a particular Seller.  3. Buyer will click on “Send Message” button to Seller. | **Response**  2. System will display the detailed profile of that Seller.  4. System will re-direct Buyer to Message. |
| **Exceptional Flow** | 3.1. Message Send  3.1.1. System will display message can’t be send to a particular user  3.1.2. System will display message send to a particular user | |
| **Post Conditions** | The Buyer will successfully contact Seller. | |

**Create Order**

**Table 4.14: Fully Dressed Use Case - Create Order**

|  |  |
| --- | --- |
| **Use Case ID** | UC-14 |
| **Use Case** | Create Order |
| **Actors** | Buyer |
| **Preconditions** | The Buyer must be logged into his account and  should be on dashboard page. |
| **Basic Flow** | Buyer will be able to create a order. |

|  |  |  |
| --- | --- | --- |
| **Main Success Scenario** | **Action** | **Response** |
| 1. Buyer clicks on ‘create Order button. | 1. System will re-direct the Buyer to ‘create Order page. 2. System will ask Buyer to select one of the two options i.e. Auto Fill or Write. |
| 4. Creator selects one of the two options. | 5. System displays the Order form according to selected option. |
| 6. Buyer fills in the form fields i.e. name, description, Email Contact, Tailor(optional), Payment Type and clicks the ‘Proceed Order  button’. | 1. System validates and create the Order. 2. System re-directs the buyer to the Dashboard. |
| **Alternative Flow** | * 1. Buyer has left empty fields      1. The system will ask the user to fill in all the form fields.   2. Invalid Address      1. System will display error message and ask the user to enter correct Address.   3. ‘Name’ field contains special characters or arithmetic operations      1. System will display error message.   4. Description’s length is not in between 1 and 500      1. System will display error message and ask the user to enter correct Description.   5. Invalid email, empty or blank spaces      1. System will display error message and ask the user to enter correct Description.   6. Invalid Contact no, alphabet or special character.      1. System will display error message and ask the user to enter correct Contact.      2. Seller renters Contact no.   7. Seller chooses payment method      1. System will display error message and ask the user to Choose Payment method.   8. Seller will choose Tailor service | |
| **Post Conditions** | The Buyer successfully creates a order. | |

**Create Shop**

**Table 4.15: Fully Dressed Use Case – Create shop**

|  |  |
| --- | --- |
| **Use Case ID** | UC-15 |
| **Use Case** | Create shop |
| **Actors** | Seller, Tailor |
| **Preconditions** | The User must be logged into his account and  should be on Dashboard. |
| **Basic Flow** | User will be able to Create a shop. |

|  |  |  |
| --- | --- | --- |
| **Main Success Scenario** | **Action** | **Response** |
| 1. User clicks on ‘dashboard’ button. | 1. System will re-direct the user to ‘dashboard menu’ Page. |
|  | 4. System displays the Shop form. |
| 2. After completing profile verification the user will be able to create shop which require a name of shop, Shop description, Category, shop logo, term and policies, marketing & SEO, Member fees . | 5. System validates and create the shop.  6. System re-directs the buyer to the dashboard. |
| **Alternative Flow** | * 1. user has left empty fields      1. the system will ask the user to fill in all the form fields.   2. ‘Name’ field contains special characters or arithmetic operations      1. system will display error message.   3. If the user didn’t pay Member fee user wont be able to create shop   price | |
| **Post Conditions** | The User will successfully Create a shop. | |

**List Product**

**Table 4.16: Fully Dressed Use Case – List Product**

|  |  |
| --- | --- |
| **Use Case ID** | UC-16 |
| **Use Case** | List Product |
| **Actors** | Seller |
| **Preconditions** | The Seller must be logged into his account and  should be on dashboard page. |
| **Basic Flow** | Seller will be able to List a product. |

|  |  |  |
| --- | --- | --- |
| **Main Success Scenario** | **Action** | **Response** |
| 1. Seller clicks on ‘List Product’ button. | 1. System will re-direct the Seller to ‘List Product page. |
|  |  |
| 2. Seller fills in the form fields i.e. name, description, Category, Type, Size, Material, Product Picture, Price, Quantity, Condition and clicks the ‘Proceed List  button’. | 1. System Validate and create the List. 2. System re-directs the Seller to the Product. |
| **Alternative Flow** | * 1. Seller has left empty fields      1. the system will ask the user to fill in all the form fields.   2.2. ‘Name’ field contains special characters or arithmetic operations   * + 1. system will display error message.   1. ‘Product Material’ field contains special characters, Numeric or arithmetic operations      1. system will display error message   2. ‘Type’ field is required if user does not fill      1. system will display error message   3. ‘Category’ field is required if user does not fill      1. system will display error message   4. ‘Size’ field is required if user does not fill      1. system will display error message   5. ‘Description’ field Length Should be from 1 to 500 if user exceed length.      1. system will display error message   6. ‘Image’ field Length Should have valid image format      1. system will display error message   7. ‘Price’ field contains special characters, Alphabet or arithmetic operations      1. system will display error message   8. ‘Quantity’ field contains special characters, Alphabet or arithmetic operations      1. system will display error message   9. ‘Condition’ field contains special characters, Alphabet or arithmetic operations      1. system will display error message | |
| **Post Conditions** | The Buyer successfully creates an order. | |

**Delete Product**

**Table 4.17: Fully Dressed Use Case - Delete Product**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-17 | |
| **Use Case** | Delete Product | |
| **Actors** | Seller | |
| **Preconditions** | The Seller must be logged into his account and  should be on ‘Product’ section. | |
| **Basic Flow** | Seller will be able to delete his Product. | |
| **Main Success Scenario** | **Action**  1. Seller on ‘Delete’ button against a particular Product.  3. Creator will re-confirm to delete the Product. | **Response**  2. System will ask the Seller to re-confirm.   1. System deletes the respective Product and attached resources. 2. System will display the success message. |
| **Exceptional Flow** | * 1. User will reconfirm Delete Product      1. User cancel the deletion upon re-confirmation   5.1.2. the showcase will not be deleted | |
| **Post Conditions** | The admin successfully deleted the Product. | |

**Edit Product**

**Table 4.18: Fully Dressed Use Case - Edit Product**

|  |  |
| --- | --- |
| **Use Case ID** | UC-18 |
| **Use Case** | Edit Product |
| **Actors** | Seller |
| **Preconditions** | The Seller must be logged into his account and should be  on ‘Products ‘page. |
| **Basic Flow** | Seller will be able to edit Product. |

|  |  |  |
| --- | --- | --- |
| **Main Success Scenario** | **Action** | **Response** |
| 1. Seller clicks on the respective Product to be edited. | 3. System will display the detail view of that Product. |
| 2. Seller will edit the Product. | 4. System will validate and update the Product accordingly. |
| **Exceptional Flow** | 2.1 Seller has left empty fields  2.1.1 the system will ask the user to fill in all the form fields.  2.2. ‘Name’ field contains special characters or arithmetic operations  2.2.1. system will display error message.  2.3. ‘Product Material’ field contains special characters, Numeric or arithmetic operations  2.3.1. system will display error message  2.4. ‘Type’ field is required if user does not fill  2.4.1. system will display error message  2.5. ‘Category’ field is required if user does not fill  2.5.1. system will display error message  2.6. ‘Size’ field is required if user does not fill  2.6.1. system will display error message  2.7. ‘Description’ field Length Should be from 1 to 500 if user exceed length.  2.7.1. system will display error message  2.8. ‘Image’ field Length Should have valid image format  2.8.1. system will display error message  2.9. ‘Price’ field contains special characters, Alphabet or arithmetic operations  2.9.1. system will display error message  2.10. ‘Quantity’ field contains special characters, Alphabet or arithmetic operations  2.10.1. system will display error message  2.11. ‘Condition’ field contains special characters, Alphabet or arithmetic operations  2.11.1. system will display error message | |
| **Post Conditions** | Seller successfully edited the Product. | |

**Manage Profile**

**Table 4.19: Fully Dressed Use Case – Manage Profile**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-19 | | |
| **Use Case** | Manage Profile | | |
| **Actors** | Seller, Buyer, Tailor | | |
| **Preconditions** | The creator must be logged into his account and should be  on homepage. | | |
| **Basic Flow** | User will be able to manage his profile. | | |
| **Main Success Scenario** |  | **Action** | **Response** |
| 1) User will click on ‘My Profile’ page. | 2) System will redirect User to the ‘My Profile’ page. |
| 3) User can set/ update his details i.e. name, profile photo, contact, address, password. and  click on save button. | 4) System will update changes in user’ profile. |
| **Exceptional Flow** | 3.1 Click on save after Edit  3.1.1 System will display the changes have been saved  3.1.2 System will display no changes have been done | | |
|  | 3.2 New password doesn’t match confirm password field  3.2.1 System will display the error message. | | |
| **Post Conditions** | User successfully manage and update his profiles  details. | | |

**Showcase Product**

**Table 4.20: Fully Dressed Use Case – Showcase Product**

|  |  |
| --- | --- |
| **Use Case ID** | UC-20 |
| **Use Case** | Showcase Product |
| **Actors** | Tailor |
| **Preconditions** | The Tailor must be logged into his account and  should be on dashboard page. |
| **Basic Flow** | Tailor will be able to showcase a product. |

|  |  |  |
| --- | --- | --- |
| **Main Success Scenario** | **Action** | **Response** |
| 1. Tailor clicks on ‘List Product’ button. | 1. System will re-direct the Seller to ‘List Product page. |
|  |  |
| 2. Seller fills in the form fields i.e. name, description, Location, Product Picture, Price and clicks the ‘Proceed List  button’. | 4. System validates and create the List.  5. System re-directs the Seller to the Product. |
| **Alternative Flow** | 2.1. Seller has left empty fields  2.1.1 the system will ask the user to fill in all the form fields.  2.2. ‘Name’ field contains special characters or arithmetic operations  2.2.1. system will display error message.  2.3. ‘Product Material’ field contains special characters, Numeric or arithmetic operations  2.3.1. system will display error message  2.4. ‘Category’ field is required if user does not fill  2.4.1. system will display error message  2.5. ‘Size’ field is required if user does not fill  2.5.1. system will display error message  2.6. ‘Description’ field Length Should be from 1 to 500 if user exceed length.  2.6.1. system will display error message  2.7. ‘Image’ field Length Should have valid image format  2.7.1. system will display error message  2.8. ‘Price’ field contains special characters, Alphabet or arithmetic operations  2.8.1. system will display error message  2.9. ‘Quantity’ field contains special characters, Alphabet or arithmetic operations  2.9.1. system will display error message | |
| **Post Conditions** | The Buyer successfully showcases a product. | |

**Delete Showcase**

**Table 4.21: Fully Dressed Use Case - Delete Showcase**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-21 | |
| **Use Case** | Delete Showcase | |
| **Actors** | Tailor | |
| **Preconditions** | The Tailor must be logged into his account and  should be on ‘Product’ section. | |
| **Basic Flow** | Tailor will be able to delete his Product. | |
| **Main Success Scenario** | **Action**  1. Tailor on ‘Delete’ button against a particular Product.  5. tailor will re-confirm to delete the Product. | **Response**  2. System will ask the tailor to re-confirm.   1. System delete the respective Product and attached resources. 2. System will display the success message. |
| **Exceptional Flow** | * 1. User will reconfirm Delete showcase      1. User cancel the deletion upon re-confirmation      2. the showcase will not be deleted | |
| **Post Conditions** | The tailor successfully deleted the showcase. | |

**Edit showcase**

**Table 4.22: Fully Dressed Use Case - Edit showcase**

|  |  |
| --- | --- |
| **Use Case ID** | UC-22 |
| **Use Case** | Edit showcase |
| **Actors** | tailor |
| **Preconditions** | The tailor must be logged into his account and should be  on ‘showcase ‘page. |
| **Basic Flow** | Tailor will be able to edit showcase. |

|  |  |  |
| --- | --- | --- |
| **Main Success Scenario** | **Action** | **Response** |
| 1. Tailor clicks on the respective Product to be edited. | 3. System will display the detail view of that Showcase. |
| 2. Tailor will edit the showcase. | 4. System will validate and update the showcase accordingly. |
| **Exceptional Flow** | 2.1. Seller has left empty fields  2.1.1 the system will ask the user to fill in all the form fields.  2.2. ‘Name’ field contains special characters or arithmetic operations  2.2.1. system will display error message.  2.3. ‘Product Material’ field contains special characters, Numeric or arithmetic operations  2.3.1. system will display error message  2.4. ‘Category’ field is required if user does not fill  2.4.1. system will display error message  2.5. ‘Size’ field is required if user does not fill  2.5.1. system will display error message  2.6. ‘Description’ field Length Should be from 1 to 500 if user exceed length.  2.6.1. system will display error message  2.7. ‘Image’ field Length Should have valid image format  2.7.1. system will display error message  2.8. ‘Price’ field contains special characters, Alphabet or arithmetic operations  2.8.1. system will display error message  2.9. ‘Quantity’ field contains special characters, Alphabet or arithmetic operations  2.9.1. system will display error message | |
| **Post Conditions** | Tailor successfully edited the Showcase. | |

**Communication**

**Table 4.23: Fully Dressed Use Case - Communication**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-23 | |
| **Use Case** | Communication | |
| **Actors** | Buyer, Tailor | |
| **Preconditions** | The tailor must be on website’s “Profile” page. | |
| **Basic Flow** | Buyer will be able to contact Tailor. | |
| **Main Success Scenario** | **Action**  1. Buyer will click on profile of a particular Tailor.  3. Buyer will click on “Send Message” button to Seller. | **Response**  2. System will display the detailed profile of that Seller.  4. System will re-direct Buyer to Message. |
| **Exceptional Flow** | 3.1 Message will be sent  3.1.1 System will display message sending failed  3.1.2 System will display message will be send | |
| **Post Conditions** | The Buyer will successfully contact Tailor. | |

**Tailoring**

**Table 4.24: Fully Dressed Use Case - Tailoring**

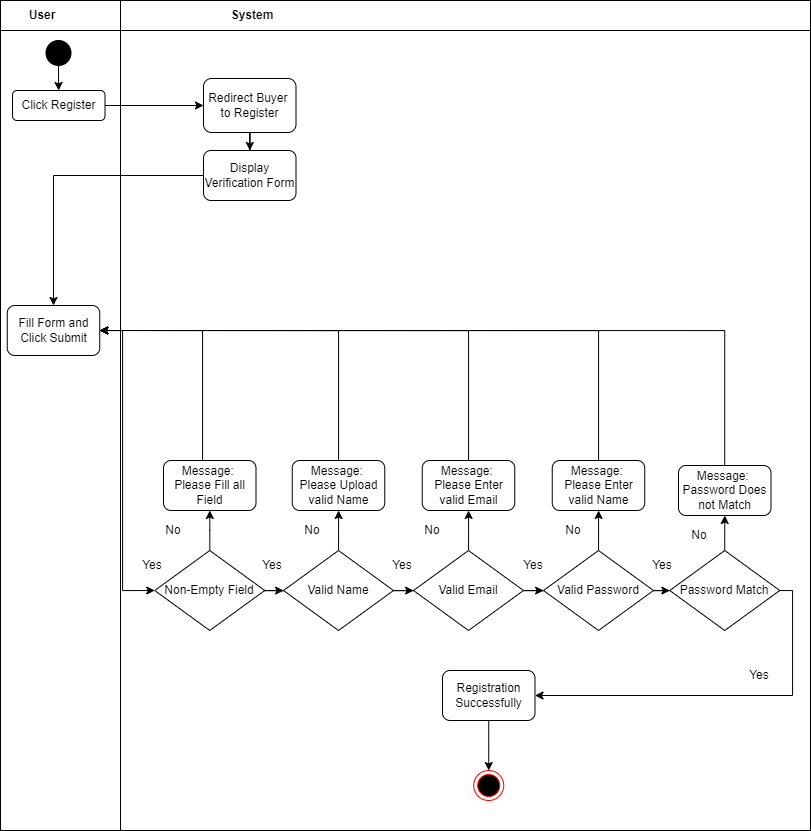
|  |  |
| --- | --- |
| **Use Case ID** | UC-24 |
| **Use Case** | Tailor |
| **Actors** | Buyer, Tailor |
| **Preconditions** | The Tailor must be logged into his account and should be  on ‘Request ‘page. |
| **Basic Flow** | Buyer will be able to send Tailoring Request. |

|  |  |  |
| --- | --- | --- |
| **Main Success Scenario** | **Action** | **Response** |
| 1. Tailor clicks on the Request to Read | 2. System will display the detail view of that Product. |
| 3. Tailor will Accept tailoring request | 4. System will validate and assign the request. |
| **Exceptional Flow** | 3.1. tailoring request upon re-confirmation  3.2. User will accept the request in given time and budget  3.3. User will delete the tailoring request and use case ended | |
| **Post Conditions** | Tailor successfully accepts the request. | |

##### 

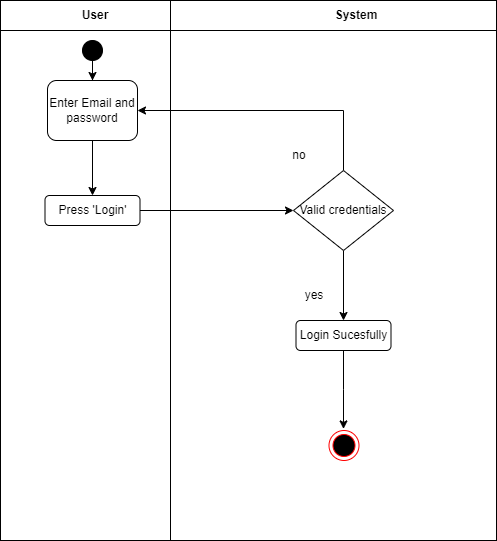
##### Activity Diagrams

**Register**

****

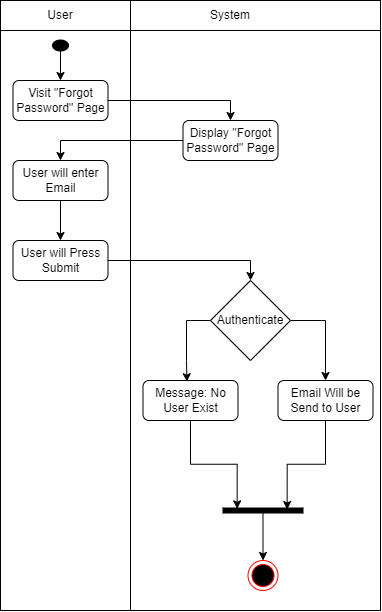
**Figure 4.6: Activity Diagram – Register**

**Login**

****

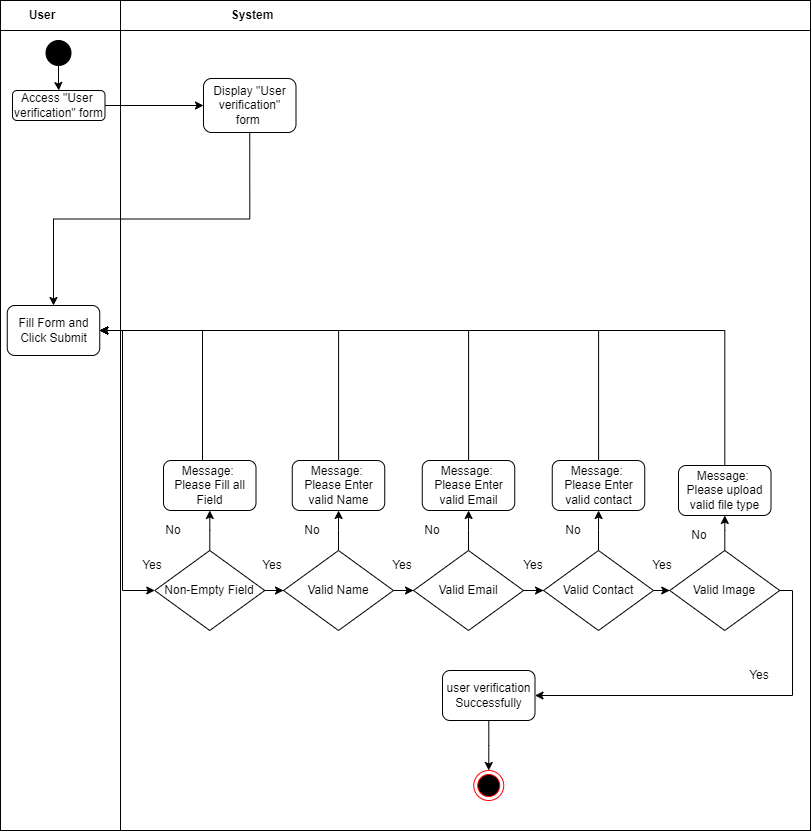
**Figure 4.7: Activity Diagram – Login**

**Forgot Password**

****

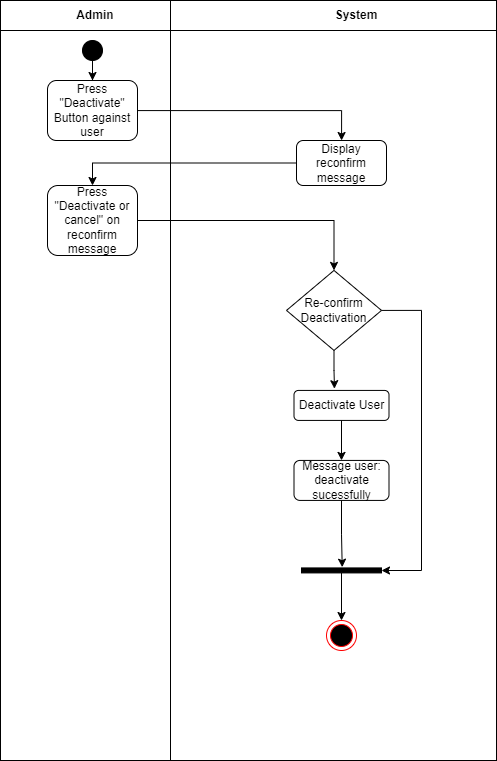
**Figure 4.8: Activity Diagram – Forgot Password**

**Verify User**

****

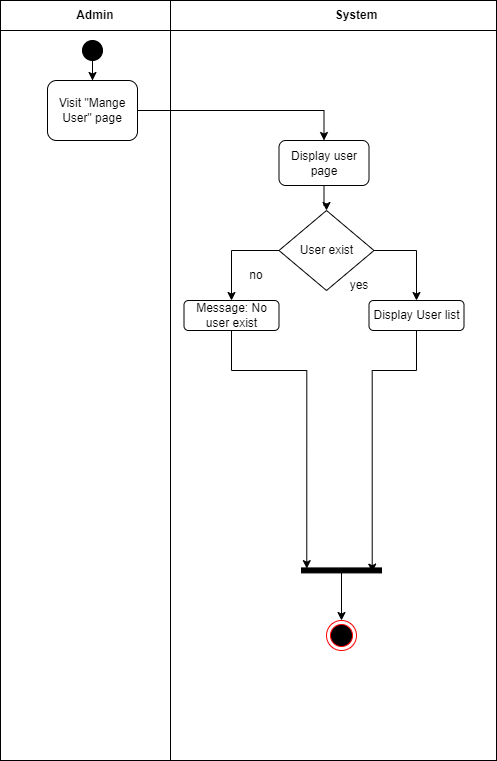
**Figure 4.9: Activity Diagram – Verify User**

**Deactivate User**

****

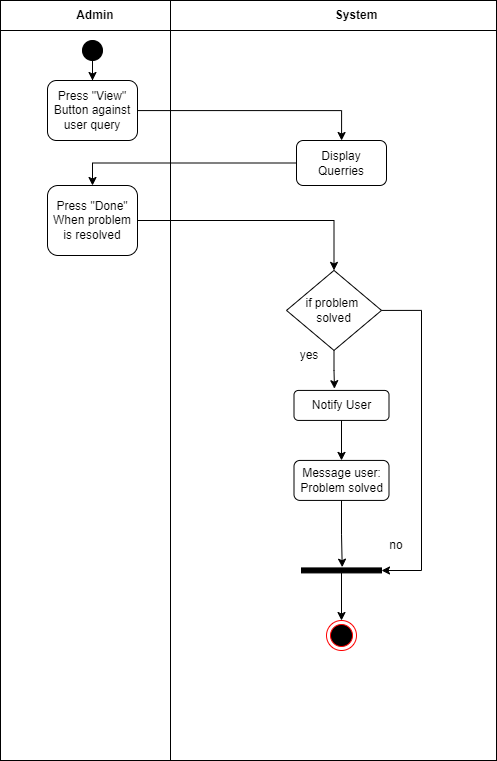
**Figure 4.10: Activity Diagram - Deactivate User**

**View User**

****

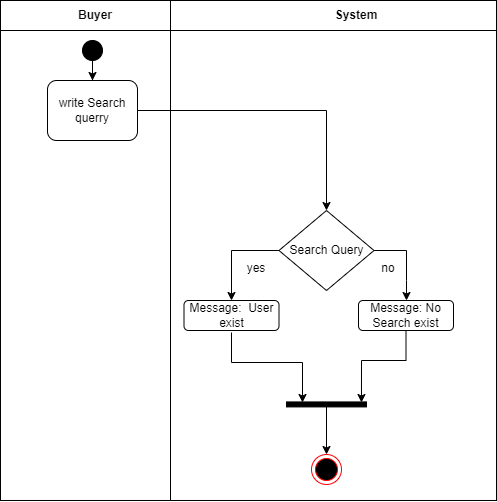
**Figure 4.11: Activity Diagram – View User**

**User Querry**

****

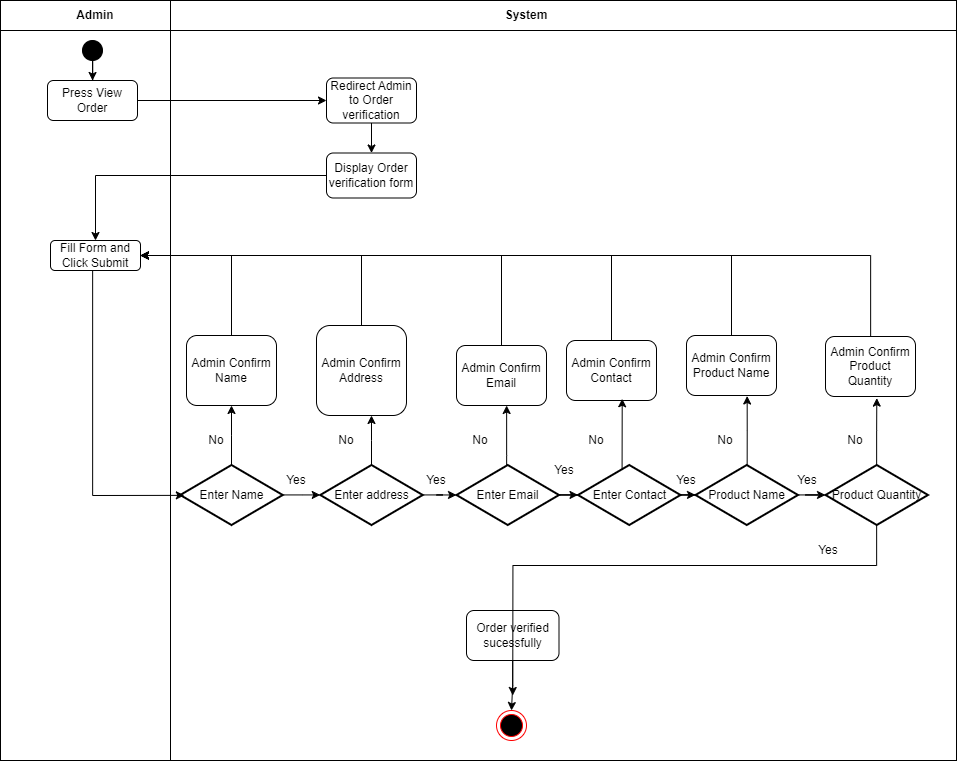
**Figure 4.12: Activity Diagram – User query**

**Search User**

****

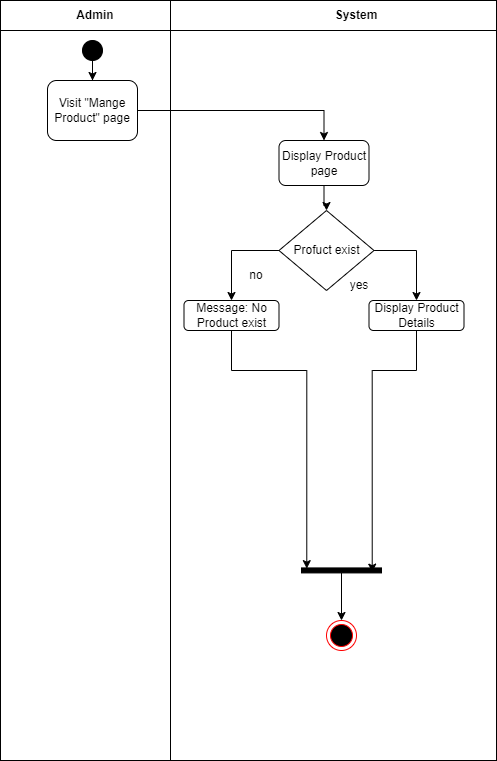
**Figure 4.13: Activity Diagram – Search User**

**Verify order**

****

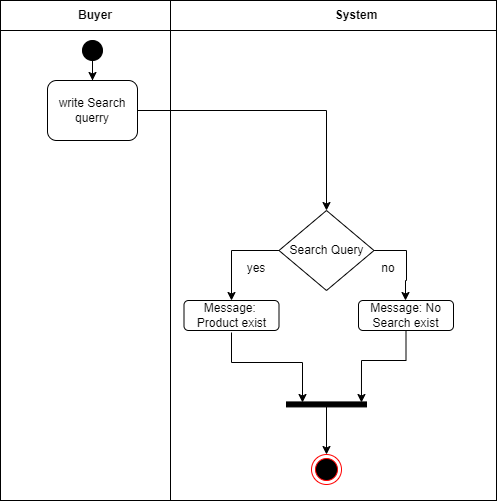
**Figure 4.14: Activity Diagram – verify User**

**View product**

****

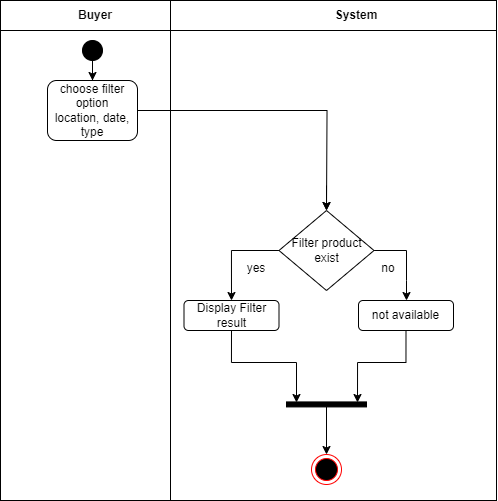
**Figure 4.15: Activity Diagram – View product**

**Search Product**



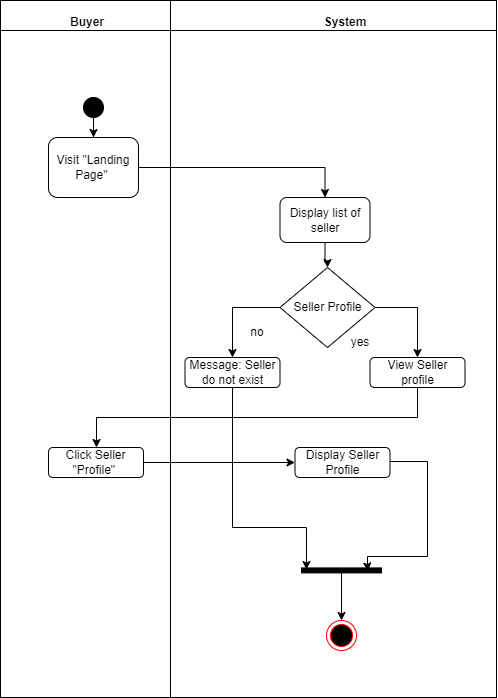
**Figure 4.16: Activity Diagram – Search Product**

**Filter Product**

****

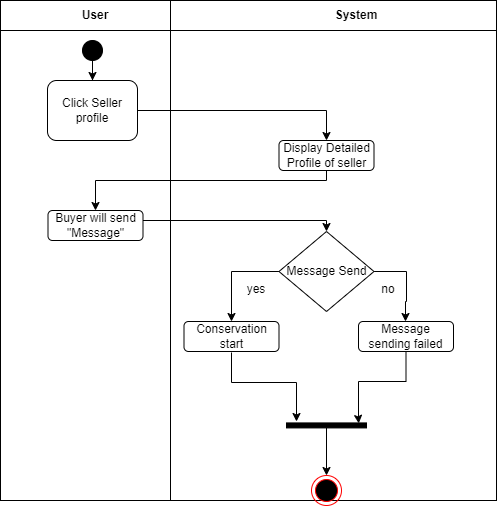
**Figure 4.17: Activity Diagram – Filter Product**

**View Seller Profile**

****

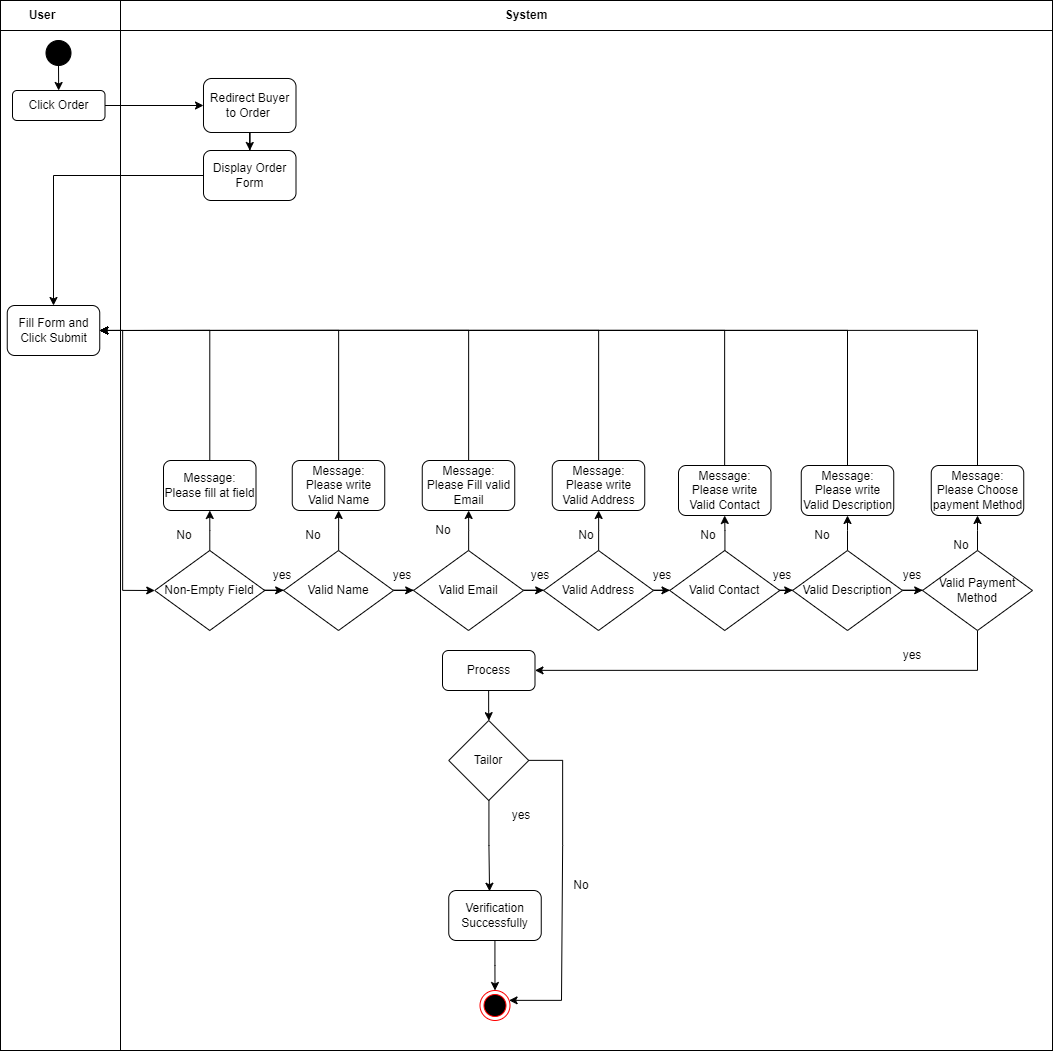
**Figure 4.18: Activity Diagram – View Seller**

**Contact Seller**

****

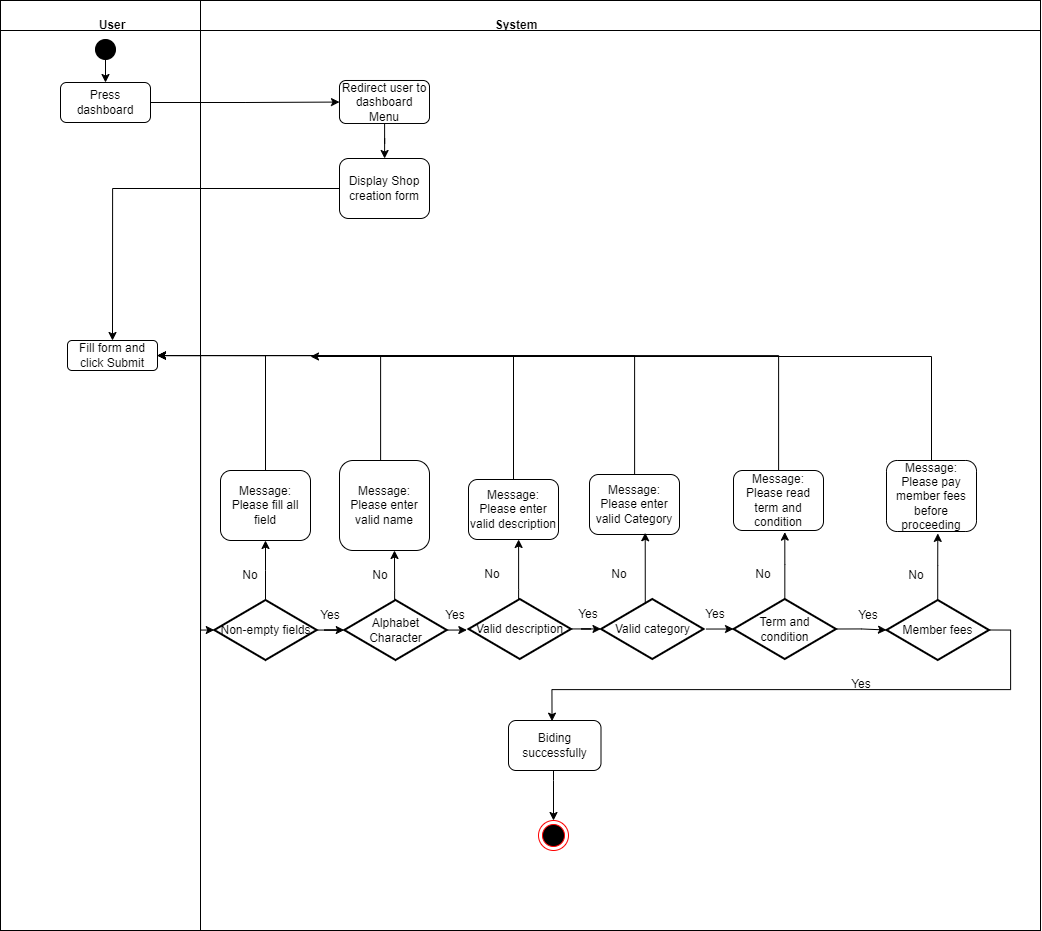
**Figure 4.19: Activity Diagram – Contact Seller**

**Create Order**

****

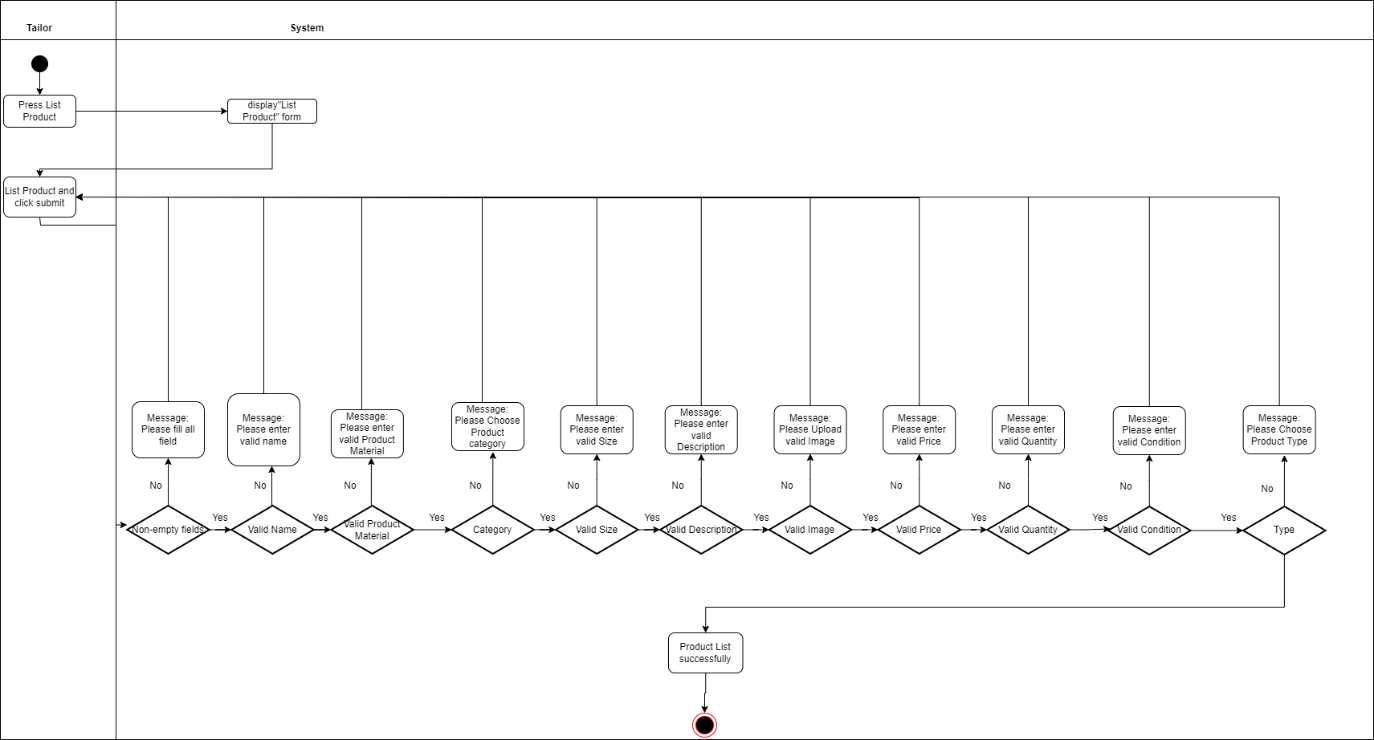
**Figure 4.20: Activity Diagram – Create Order**

**Create Shop**

****

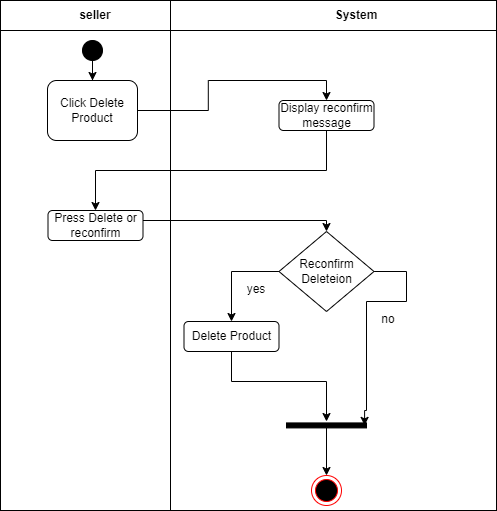
**Figure 4.21: Activity Diagram – Create Shop**

**List Product**

****

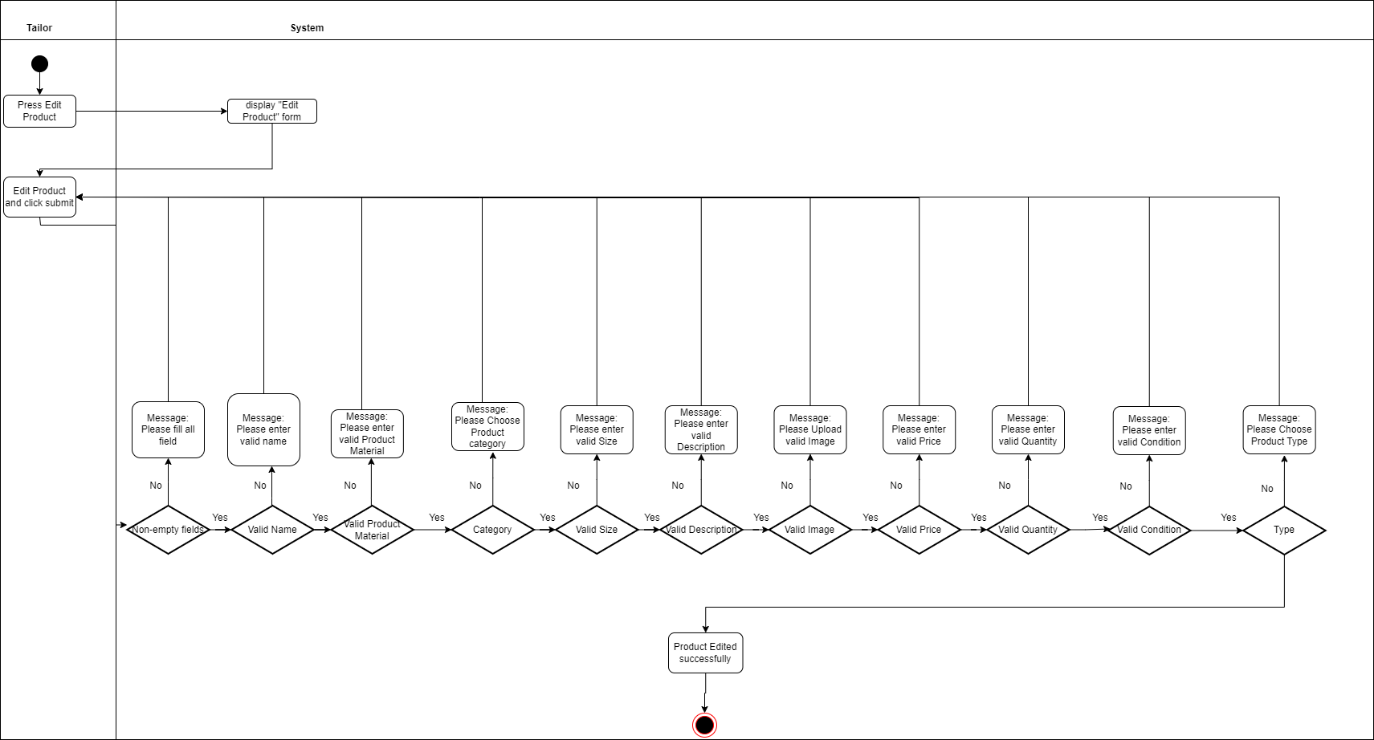
**Figure 4.22: Activity Diagram – List Product**

**Delete Product**

****

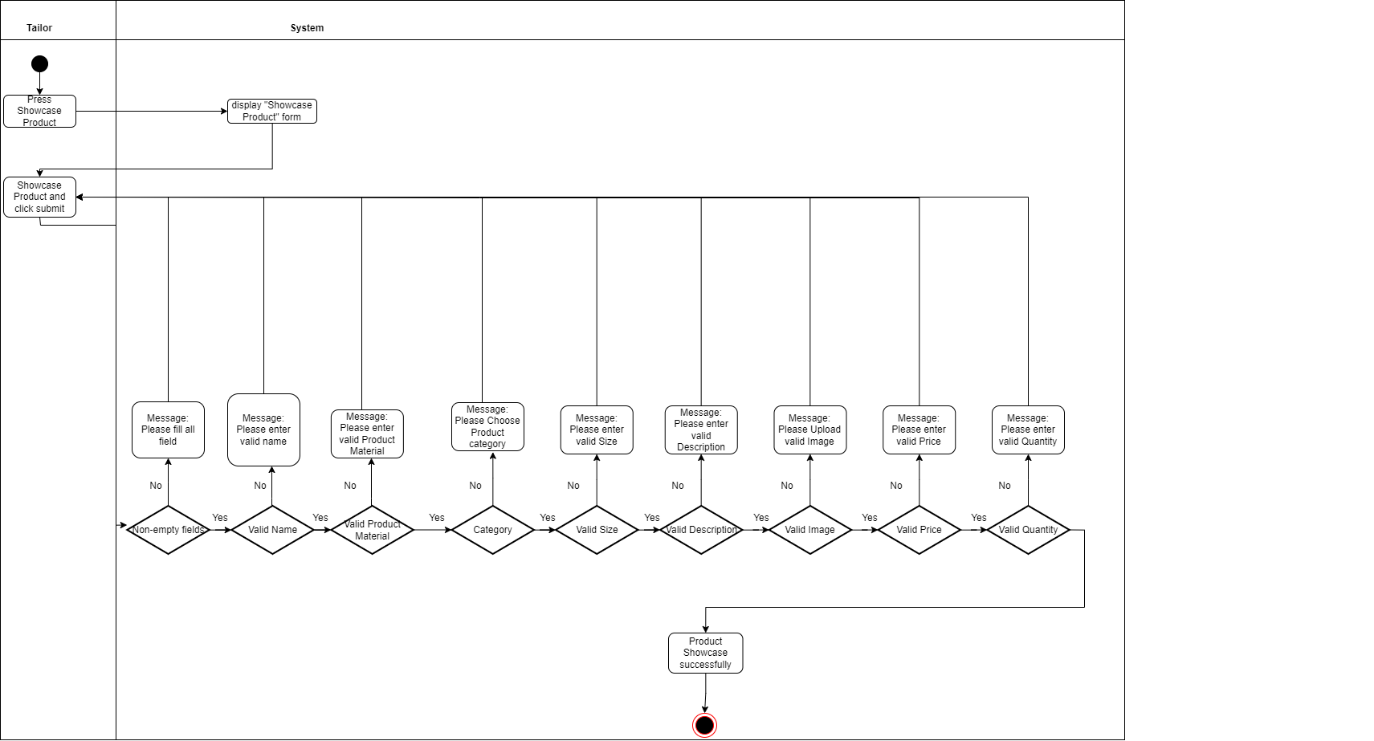
**Figure 4.23: Activity Diagram –Delete product**

**Edit Product**

****

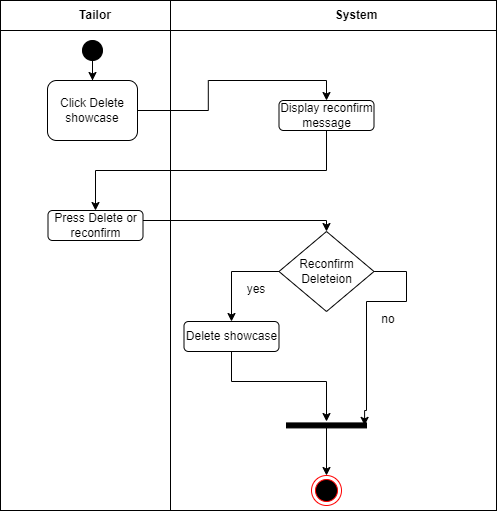
**Figure 4.24: Activity Diagram –Edit product**

**Showcase Product**

****

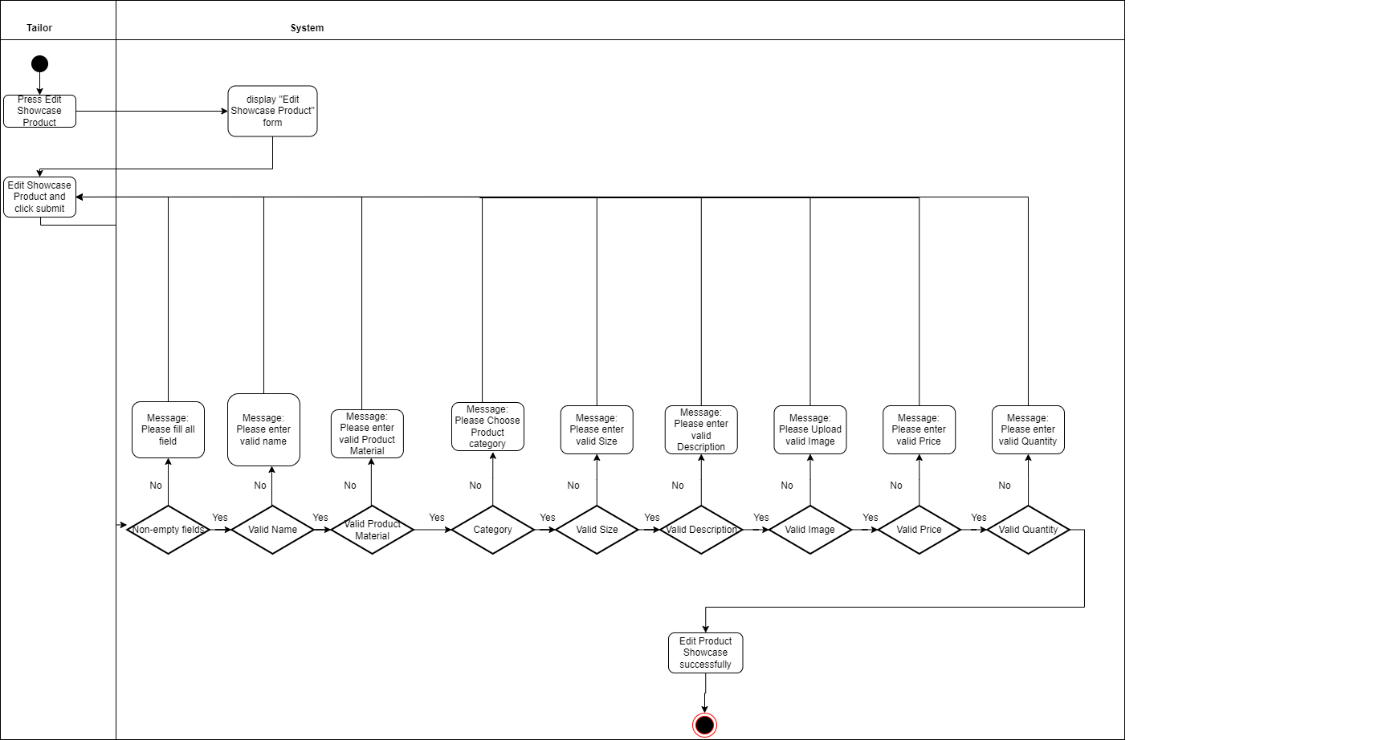
**Figure 4.25: Activity Diagram –Showcase product**

**Delete Showcase**

****

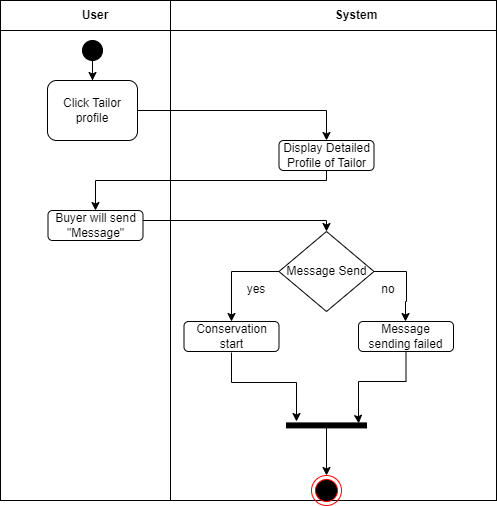
**Figure 4.26: Activity Diagram – Delete Showcase**

Edit **Showcase**

****

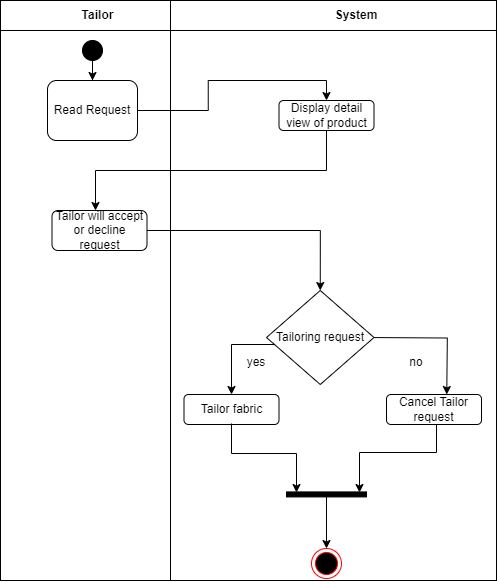
**Figure 4.27: Activity Diagram – Edit Showcase**

**Communication**



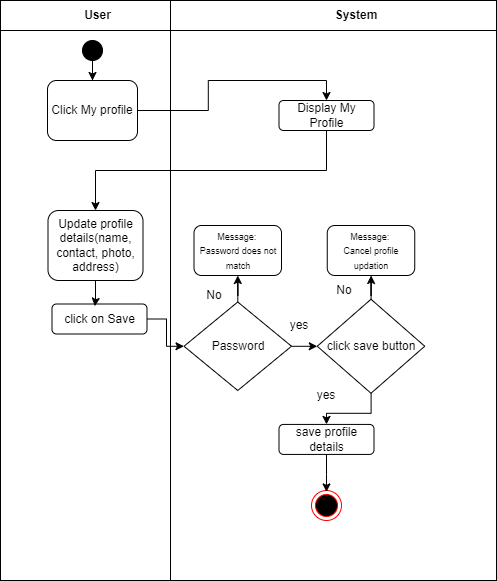
**Figure 4.28: Activity Diagram – Communication**

**Tailoring**

****

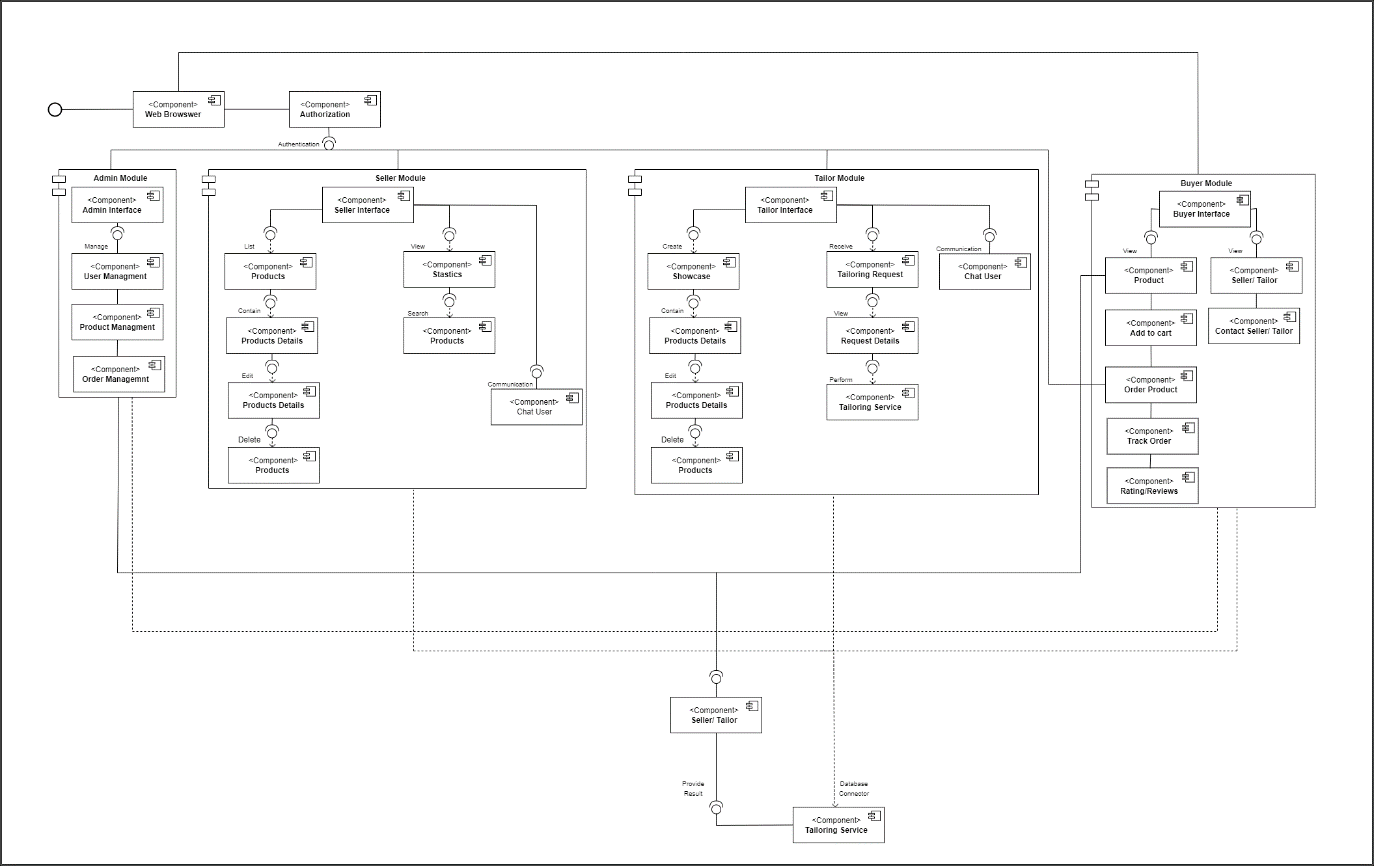
**Figure 4.29: Activity Diagram – Tailoring**

**Manage Profile**

****

**Figure 4.30: Activity Diagram – Manage Profile**

##### Component Diagram

****

**Figure 4.31: Component Diagram of the System**

#### Schema Diagram Figure 4.32: Schema Diagram of the System

### SQA ACTIVITIES

* + 1. **Usability Testing through Heuristic Evaluation**
       1. Introduction

We used Jakob Nielson's 10 requirements to test the usability of the website. A collection of usability guidelines known as Jakob Nielsen's heuristics are used to assess user interface designs with an emphasis on frequent usability problems. We were able to provide instructions for more intuitive and effective digital encounters with the aid of this strategy. The following are these heuristics:

* + - * 1. Visibility of System Status

It states that “The design should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time.”

* + - * 1. Match Between the System and the Real World

The design should speak the users' language. Use words, phrases, and concepts familiar to the user, rather than internal jargon. Follow real-world conventions, making information appear in a natural and logical order.

* + - * 1. User Control and Freedom

Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action without having to go through an extended process.

* + - * 1. Consistency and Standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform and industry conventions.

* + - * 1. Error Prevention

Good error messages are important, but the best designs carefully prevent problems from occurring in the first place. Either eliminate error-prone conditions, or check for them and present users with a confirmation option before they commit to the action.

* + - * 1. Recognition Rather than Recall

Minimize the user's memory load by making elements, actions, and options visible. The user should not have to remember information from one part of the interface to another. Information required to use the design (e.g. field labels or menu items) should be visible or easily retrievable when needed.

* + - * 1. Help Users Recognize, Diagnose, and Recover from Errors

Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution.

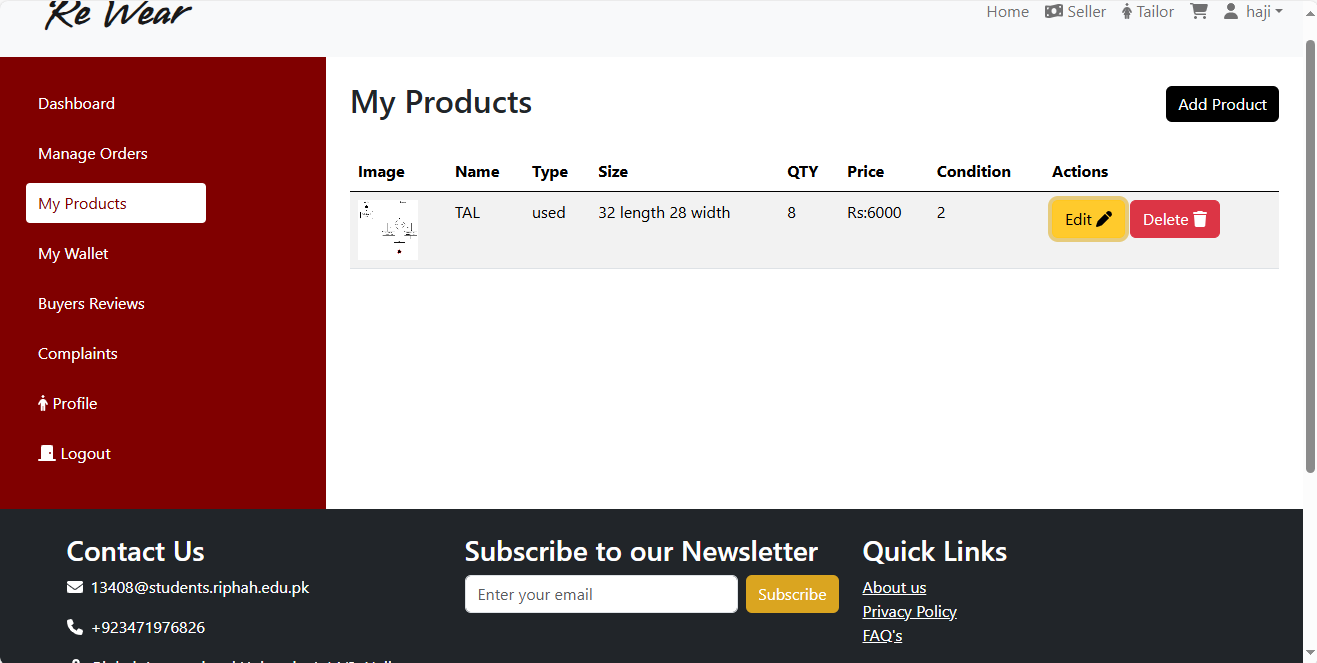
* + - 1. Why Heuristics Evaluation?

Here are reasons why heuristic evaluation is a good approach and why it can be considered better in certain contexts: Cost-effective and Time- efficient, Early Identification of Issues, Standardized Guidelines, Flexible and Adaptable Complements Other Methods.

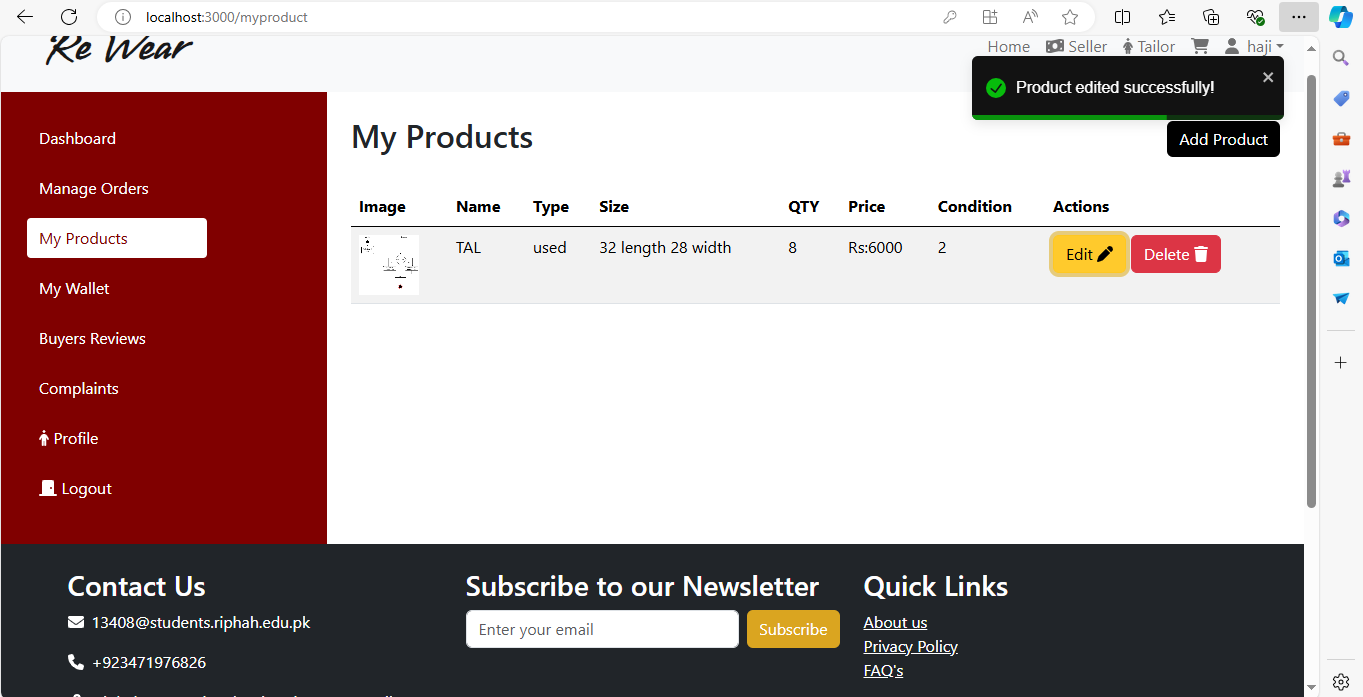
* + - 1. Heuristic Violations

**Table 4.25: Heuristic Violation -01**

|  |  |  |  |
| --- | --- | --- | --- |
| **Heuristic**  **Violations ID** | **Violated Heuristic** | **Issue** | **Solution** |
| **01** | Visibility of System Status. | When user clicks on the “Save Changes” button  no message was shown to him/her. | To address this heuristic, we have added  information messages that  represent the current state  of system, which were missing initially. |

****

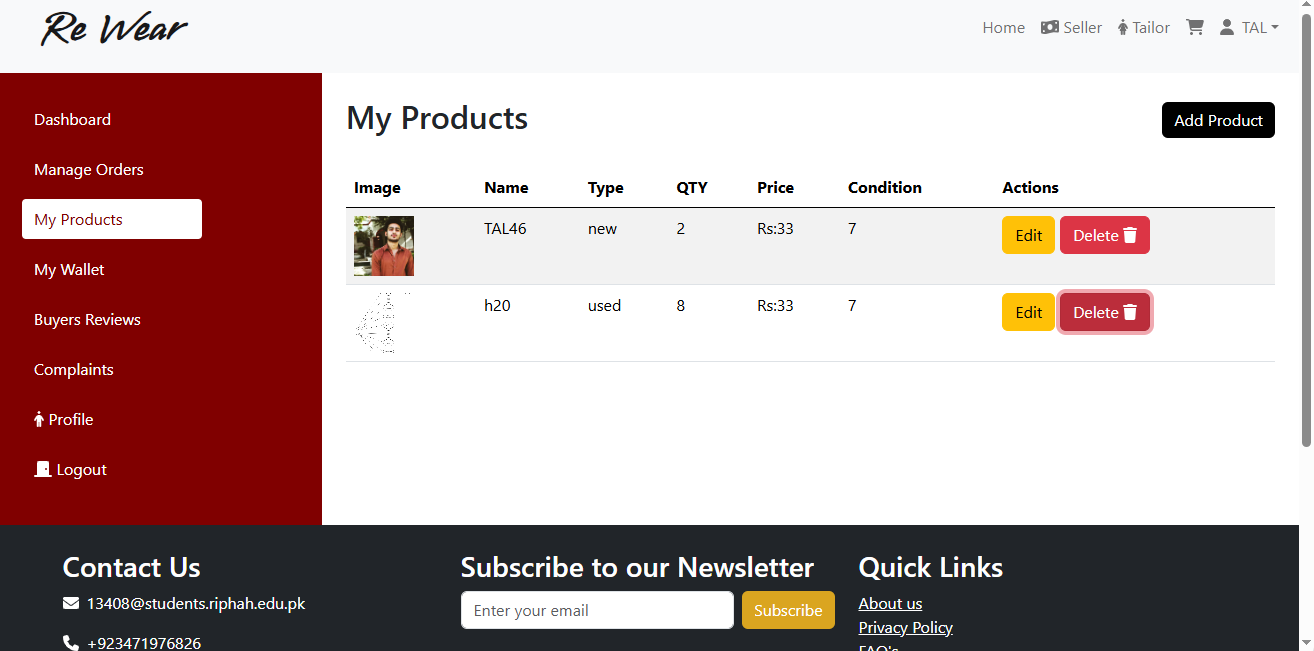
**Figure 4.33: Heuristic Violation - 01**



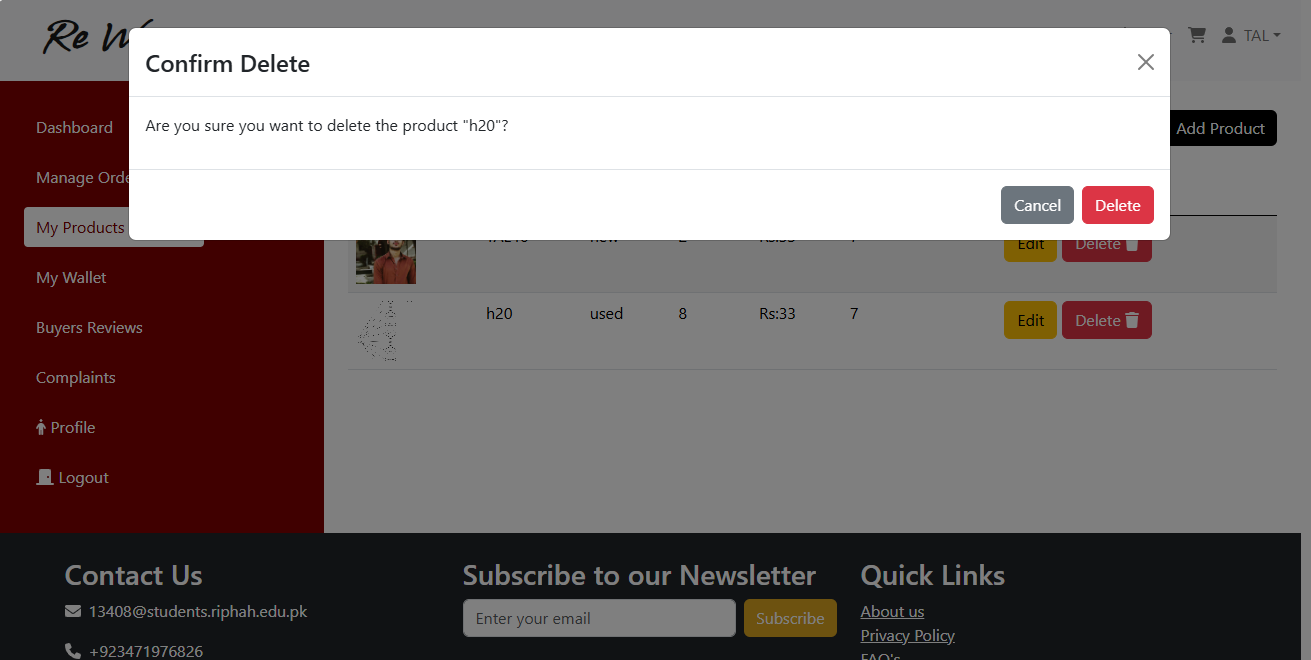
**Figure 4.34: Solution of Heuristic Violation – 01 (A)**

**Table 4.26: Heuristic Violation - 02**

|  |  |  |  |
| --- | --- | --- | --- |
| **Heuristic**  **Violations ID** | **Violated Heuristic** | **Issue** | **Solution** |
| **02** | Error Prevention. | When user clicks on “Delete” icon, the Products gets directly deleted without any reconfirmation. | To address this heuristic, we have added re confirmation message, so that any inconvenience could be avoided. |

****

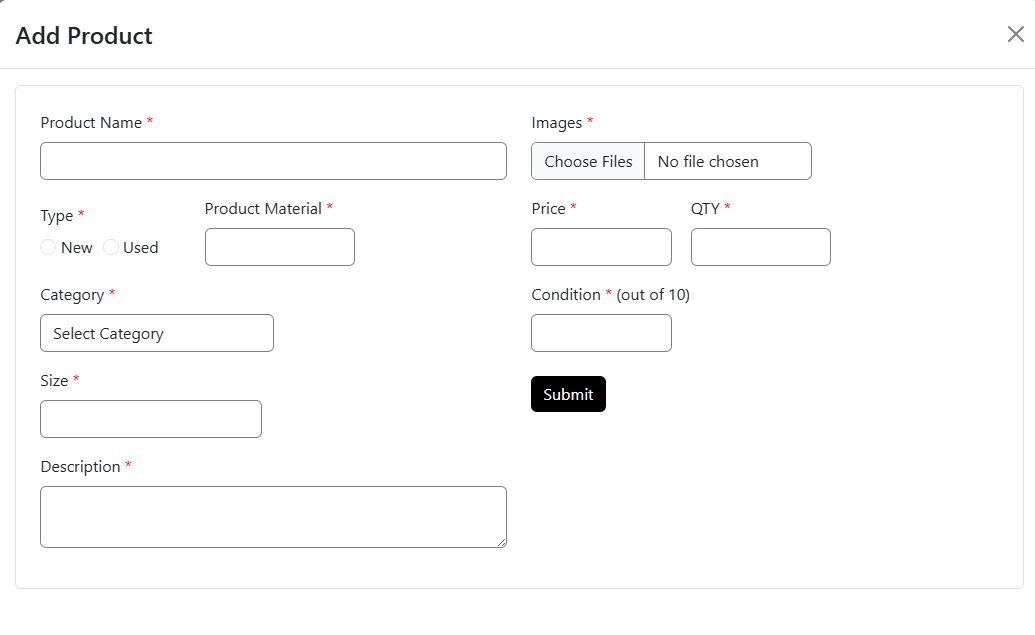
**Figure 4.35: Heuristic Violation – 02 (A)**



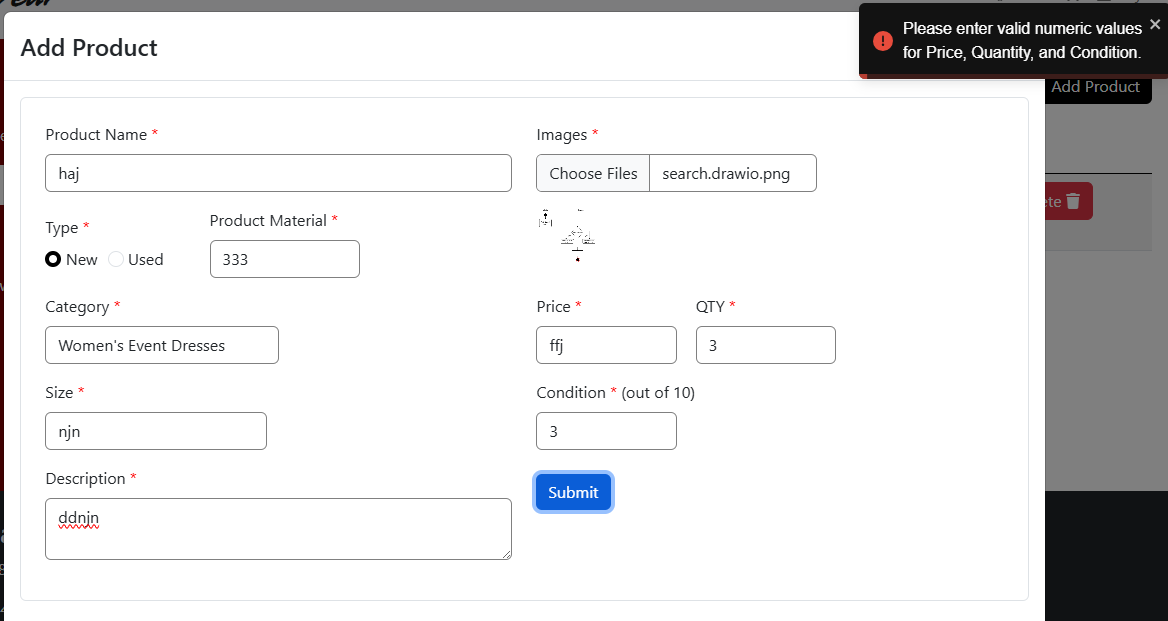
**Figure 4.36: Solution of Heuristic Violation – 02 (B)**

**Table 4.27: Heuristic Violation - 03**

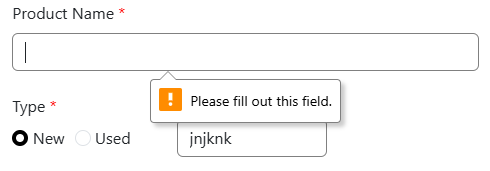
|  |  |  |  |
| --- | --- | --- | --- |
| **Heuristic**  **Violations ID** | **Violated Heuristic** | **Issue** | **Solution** |
| **03** | Help Users Recognize, Diagnose, and Recover from Errors | Validation was missing, when user clicks on “List” button, because of Node validation at the backend, Register wasn’t created but because of lack of frontend validation nothing happens on screen and it remains still. | To address this heuristic, we have added proper input validation messages so that user doesn’t get confused. |



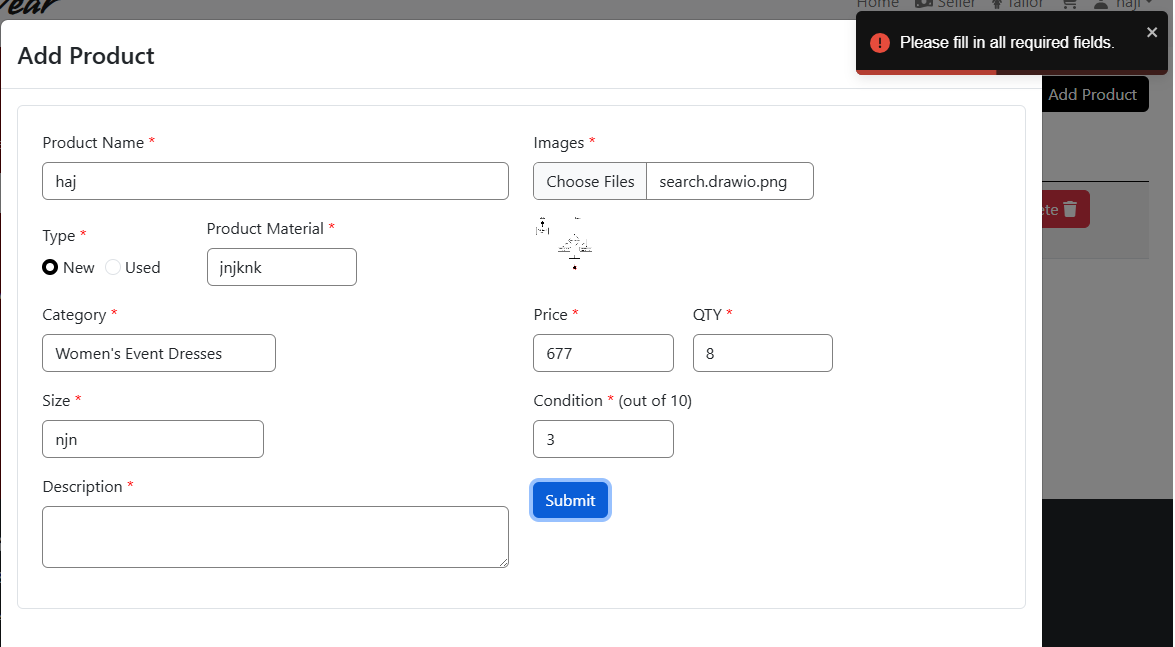
**Figure 4.37 Heuristic Violation - 03**

****

**Figure 4.38: Solution of Heuristic Violation – 03 (A)**

****

**Figure 4.39: Solution of Heuristic Violation – 03 (B)**

****

**Figure 4.40: Solution of Heuristic Violation – 03 (C)**

# Chapter 5 Implementation

### Chapter 5

## IMPLEMENTATION

#### ENDEAVOUR

##### Team

* M. Talha Masood
* Sfwan Ali

#### WORK BREAKDOWN STRUCTURE

**Table 5.1: Work Break Down Structure**

|  |  |
| --- | --- |
| **WBS Code** | **Deliverable** |
| 1 | Project Management |
| 1.1 | WBS |
| 1.2 | Roles and Responsibility Matrix |
| 2 | Reports & Documentation |
| 2.1 | Team Members declaration |
| 2.2 | Initial Project Proposal |
| 2.3 | Project Proposal Document |
| 2.4 | Performance and Attendance Report |
| 2.5 | Project Report |
| 3.1 | Chapter 1: Introduction |
| 4.1 | Chapter 2: Literature |
| 4.2 | Online Survey & targeted Interview |
| 4.3 | Brainstorming |
| 5.1 | Chapter 3: Requirements Analysis |

|  |  |
| --- | --- |
| 5.2 | Functional Requirements |
| 5.3 | Non-Functional Requirements |
| 5.4 | Requirements Quality Assurance |
| 5.5 | Defect Identification - Checklist |
| 5.6 | Defect Detection - Black Box |
| 6.1 | Chapter 4: System Design |
| 6.3 | Architectural Design |
| 6.4 | Detailed Design |
| 6.5 | Use Case Diagrams |
| 6.6 | Fully Dressed use cases |
| 6.7 | Activity Diagrams |
| 6.8 | Component Diagram |
| 6.10 | Design Quality Assurance |
| 6.11 | Heuristic Evaluation |
| 7 | Chapter 5: Implementation |
| 7.1 | Endeavour |
| 7.2 | WBS |
| 7.3 | Roles & Responsibility Matrix |
| 7.4 | Deployment Environment |
| 7.5 | Quality Assurance |
| 7.6 | Defect Detection - White Box |

|  |  |
| --- | --- |
| 7.7 | User Acceptance Testing |
| 8 | Chapter 6 |
| 8.1 | Conclusion and Outlook |
| 8.2 | Website |
| 8.3 | Front-End |
| 8.4 | Authentication |
| 8.5 | Register |
| 8.6 | Login/Logout |
| 8.7 | Product Management |
| 8.8 | Profile Management |
| 8.9 | Back-End |
| 9 | Database |
| 9.1 | User database |
| 9.2 | Product Database |

##### Roles & Responsibility Matrix

**Table 5.2: Roles and responsibility matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **WBS #** | **WBS**  **Deliverable** | **Activity #** | **Activity to Complete the Deliverable** | **Duration**  **(# of Days)** | **Responsible Team Member(s) & Role(s)** |
| 1 | Project Management | 1 | WBS | 1 | Muhammad Talha Masood |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | 1.3 | Roles and Responsibility Matrix | 5 | Muhammad Talha Masood |
| 2 | Software | 2.0 | Reports & Documentation | 30 | Muhammad Talha Masood |
|  | Website | 45 | Muhammad Talha Masood  & Sfwan Ali |
| 2.1 | Reports & Documentation | 2.1 | Team Members declaration | 1 | Muhammad Talha Masood |
| 2.2 | Initial Project Proposal | 2 | Muhammad Talha Masood  & Sfwan Ali |
| 2.3 | Project Proposal Document | 7 | Muhammad Talha Masood |
| 2.4 | Attendance Report | 1 | Sfwan Ali |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | 2.5 | Progress Report FYP-1 | 60 | Muhammad Talha Masood  & Sfwan Ali |
| 2.5 | Progress Report FYP | 3.1 | Chapter 1: Introduction | 2 | Muhammad Talha Masood |
| 4.1 | Chapter 2: Literature | 5 | Muhammad Talha Masood |
| 4.1.1 | Initial Project Proposal | 1 | Muhammad Talha Masood |
| 5.1 | Chapter 3: Requirements Analysis | 15 | Muhammad Talha Masood |
| 5.2 | Functional Requirements | 10 | Muhammad Talha Masood |
| 5.3 | Non-Functional Requirements | 2 | Muhammad Talha Masood |
| 5.4 | Requirements Quality Assurance | 5 | Muhammad Talha Masood |
| 5.5 | Defect Identification - Checklist | 4 | Muhammad Talha Masood |
| 5.6 | Defect Detection - Black Box | 4 | Muhammad Talha Masood |
| 6.1 | Chapter 4: System Design | 20 | Muhammad Talha Masood  & Sfwan Ali |
| 6.1 | Chapter 4: System Design | 6.3 | Architectural Design | 3 | Sfwan Ali |
| 6.4 | Detailed Design | 5 | Muhammad Talha Masood |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | 6.5 | Use Case Diagrams | 2 | Muhammad Talha Masood |
| 6.6 | Fully Dressed use cases | 5 | Muhammad Talha Masood |
| 6.7 | Activity Diagrams | 3 | Muhammad Talha Masood |
| 6.8 | Component Diagram | 2 | Muhammad Talha Masood  & Sfwan Ali |
| 6.10 | Design Quality Assurance | 3 | Muhammad Talha Masood |
| 7.0 | Chapter 5: Implementation | 5 | Muhammad Talha Masood    & Sfwan Ali |
| 7.0 | Chapter 5: Implementation | 7.1 | Endeavour | 1 | Muhammad Talha Masood |
| 7.4 | Deployment Environment | 1 | Muhammad Talha Masood |
| 7.5 | Quality Assurance | 5 | Muhammad Talha Masood |
|  |  | 8 | Chapter 6 | 15 | Muhammad Talha Masood and Sfwan Ali |
| 3.1 | Chapter 6 | 2.2.1 | UI | 10 | Muhammad Talha Masood |
| 2.2 | 2.2.1 | UX | 10 | Muhammad Talha Masood |
| 2.2 | 2.2.1 | Front End | 20 | Sfwan Ali |
|  | 2.2.1.1 | Authentication | 2 | Sfwan Ali |
|  | 2.2.1.2 | Register | 5 | Muhammad Talha Masood |
|  | 2.2.1.3 | Login/ Logout | 7 | Sfwan Ali |
|  | 2.2.1.4 | Profile Management | 15 | Sfwan Ali |
|  | 2.2.1.5 | Product Management | 25 | Sfwan Ali |
| 2.2 | 2.2.1 | Backend | 15 | Muhammad Talha Masood  & Sfwan Ali |
| 2.2 | 2.2.1 | Database | 10 | Sfwan Ali |
|  |  | 2.2.1.1 | Database products | 10 | Sfwan Ali |
|  |  | 2.2.1.2 | Database User | 10 | Sfwan Ali |

#### COMPONENTS, LIBRARIES, WEB SERVICES, AND STUBS

* ReactJS
* Express
* Node.js
  + 1. **Libraries**
* Axios
* Bootstrap
* Node mailer
* Font Awesome
* Flow Bite
  + 1. **Data Base**
  + Mongo DB
    1. **IDE, Tools, and Technologies**
* Slack
* Jira
* Git Bash
* Clockify
* Visual Studio Code
* Canva
* Draw.io
* MS Word
* Adobe Illustrator
* Figma
  1. **Best Practices/ Coding Standards**
     1. Software Engineering Methodologies

In our project we have we have used iterative approach.

* Implemented iterative.
* Reviewed it in weekly meetings with our supervisor by

giving them the demonstration of system.

* Recorded the suggested improvements.
* Implemented the suggested improvements.
* Designed that using through uml diagrams
* Performed testing.
* Moved forward to the next increment and so on.
  + 1. **Introduction**

The Iterative model is a software development methodology were

a project is divided into small, manageable increments or segments.

Each increment represents a complete subset of functionality that can be

developed, tested, and delivered independently. The key principle of this

model is iterative development, where each increment builds upon the

previous ones, adding new features or enhancements.

* + 1. **Why we used Iterative Model**

The primary objectives of applying the Iterative model to develop

our software include:

* + 1. **Timely Delivery of Core Features**

Ensuring critical features like user authentication, content creation, and

client interaction is delivered early in the development process.

* + 1. **Iterative Refinement**

Continuously refining and enhancing the system based on stakeholder’s

feedback and evolving requirements.

* + 1. **Effective Project Management**

Breaking down the project into manageable increments facilitates better

resource allocation, risk management, and overall project control.

* + 1. **Our Increments**

We have completed our development in overall 5 major increments,

initially developed the core functionalities and then added additional

features to enhance the user experience.

• **Daily stand-up meetings:** Brief sessions that include progress reports, planning, and team collaboration.

• **Regular progress reviews:** Include meetings with the supervisor as needed to evaluate project status and integrate comments.

• **Iterative planning:** Entails continually prioritizing work and responding to changing requirements.

• **Self-organization and task allocation:** Team members plan how to achieve goals based on their talents and availability.

• **Continuous improvement:** Entails regular reflection on processes, eliminating bottlenecks, and implementing new techniques to increase productivity.

* + 1. **ReactJS Coding Standards for Reusability:**

• Prioritize variable and function names that are easy to understand and succinct.

• Follow camel case naming rules for uniformity.

• Use comments sparingly, offering only the essential context and avoiding overkill.

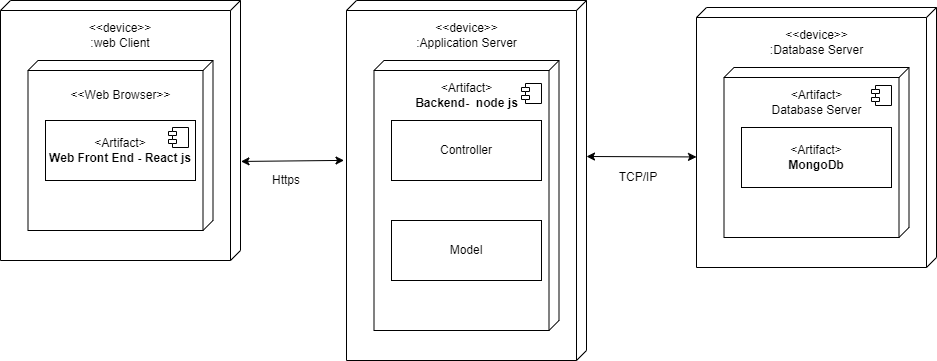
• Use shorthand notation when necessary to improve readability.

• Modularization should be emphasized to allow for component reuse and maintenance.

* 1. **Summary**

This chapter includes a list of elements and frameworks utilized in our project to improve the experience of users. We described the Work Breakdown Structure (WBS) and Control Flow Diagram. We discussed tools, IDEs, best practices, and code standards for software engineering.

#### DEPLOYMENT DIAGRAM

****

**Figure 5.1: Deployment Diagram**

#### SQA ACTIVITIES: DEFECT DETECTION

##### Test Case Design (White Box)

* + - 1. Register

**Table 5.3: W.B. Test Case - Register**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TC.ID** | **Input Fields** | | | **Reference**  **- ECP** | **Actual Output** |
| **Name** | **Email** | **Password** |
| **1.1** | Talha Masood | [Talha@gmail.com](mailto:Talha@gmail.com) | Talha@12345 | Table 3.8,  3.9 | Register successful. System redirected the  user to respective Dashboard. |
| **1.2** | Talha | [Talhagmail.com](mailto:Talha@gmail.com) | 123 | Table 3.8,  3.9 | Password and email is incorrect |
| **1.3** | H20 | xyz | 12345678 | Table 3.8,  3.9 | Name and Email is incorrect |
| **1.4** | N/A | N/A | N/A | Table 3.8,  3.9 | Please fill the field |

* + - 1. Login

**Table 5.4: W.B. Test Case - Login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TC.ID** | **Input Fields** | | **Reference**  **- ECP** | **Actual Output** |
| **Email** | **Password** |
| **1.1** | [Talha@gmail.com](mailto:Talha@gmail.com) | Talha@12345 | Table 3.8,  3.9 | Login successful. System redirected the user to respective Dashboard. |
| **1.2** | [Talha@gmail.com](mailto:Talha@gmail.com) | 123 | Table 3.8,  3.9 | These credentials do not match our records. |
| **1.3** | xyz | 12345678 | Table 3.8,  3.9 | These credentials do not match our records. |
| **1.4** | N/A | N/A | Table 3.8,  3.9 | These credentials do not match our records. |

* + - 1. Edit Profile (Name)

**Table 5.5: W.B. Test Case – Edit Profile Name**

|  |  |  |  |
| --- | --- | --- | --- |
| **TC.ID** | **Input** | **Reference - ECP** | **Expected Output** |
| **Name** |
| 1.1 | N/A | Table 3.16 | Please enter name |
| 1.2 | Talha | Table 3.16 | Valid |
| 1.3 | !$%#^@ | Table 3.16 | Invalid name. name not updated |

* + - 1. Profile Photo

**Table 5.6: W.B. Test Case – Profile Photo**

|  |  |  |  |
| --- | --- | --- | --- |
| **TC.ID** | **Input** | **Reference - ECP** | **Expected Output** |
| **Image** |
| 1.1 | N/A | Table 3.13 | No pic selected |
| 1.2 | Abc.io | Table 2.5 | Invalid image format. Image not updated. |
| 1.3 | Pic.jpeg | Table 2.5 | Valid image format. |

* + - 1. Contact Number

**Table 5.7: W.B. Test Case – Contact Number**

|  |  |  |  |
| --- | --- | --- | --- |
| **TC.ID** | **Input** | **Reference - ECP** | **Expected Output** |
| **Number** |
| 1.1 | N/A | Table 3.19 | Please enter Number |
| 1.2 | +923365940220 | Table 3.19 | Valid |
| 1.3 | 0322cjdnjc | Table 3.19 | Invalid Number. |

* + - 1. Change Password

**Table 5.8: W.B. Test Case – Password**

|  |  |  |  |
| --- | --- | --- | --- |
| **TC.ID** | **Input** | **Reference - ECP** | **Expected Output** |
| **Password=confirm password** |
| 1.1 | N/A | Table 3.9 | Please enter Password |
| 1.2 | Talha@123 | Table 3.9 | Valid |

* + - 1. List Product

**Table 5.9: W.B. Test Case - List Product**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Input Field** | | | | | | | |  |  |  |
| **TC ID** | **Name** | **Type** | **Material** | **Category** | **Size** | **Description** | **Image** | **Quantity** | **condition** | **Reference-ECP** | **Excepted Output** |
| 1.1 | Dress | Used | Silicon | Girl dress | M | Good dress | Pdf | 1 | 3.7 | Table 3.25-3.34 | Product File type is  incorrect |
| 1.2 | Dress | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | Table 3.25-3.34 | Fiel is empty |
| 1.3 | @1 | used | 35 | Girls dress | 2222 | N/A | pdf | N/A | N/A | Table 3.25-3.34 | Incorrect field |
| 1.4 | Dress | Used | Cotten | Girl dress | M | Good dress | jpeg | 1 | 3.7 | Table 3.25-3.34 | Product Listed successfully |

* + - 1. Edit Product

**Table 5.10: W.B. Test Case - Edit Product**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Input Field** | | | | | | | |  |  |  |
| **TC ID** | **Name** | **Type** | **Material** | **Category** | **Size** | **Description** | **Image** | **Quantity** | **condition** | **Reference-ECP** | **Excepted Output** |
| 1.1 | Dress | Used | Silicon | Girl dress | M | Good dress | Pdf | 1 | 3.7 | Table 3.25-3.34 | Product File type is  incorrect |
| 1.2 | Dress | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | Table 3.25-3.34 | Fiel is empty |
| 1.3 | @1 | used | 35 | Girls dress | 2222 | N/A | pdf | N/A | N/A | Table 3.25-3.34 | Incorrect field |
| 1.4 | Dress | Used | Cotten | Girl dress | M | Good dress | jpeg | 1 | 3.7 | Table 3.25-3.34 | Product Edited Successfully |

* 1. User Acceptance Testing

**Neeed to add data**

**Table 5.11: User Acceptance Testing**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Suggestion** | **Current Status** |
| 1 | The user can log in and that the OTP is being verified. | Resolved |
| 2 | While listing Product User can Modify or remove Product | Resolved |
| 3 | Confirmed that descriptions for listing and order can be added | Resolved |
| 4 | maintenance of profiles, such as choosing a default image or profile picture. | Resolved |

* 1. **ERROR LOG**

**Table 5.12: Error Log**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Error ID** | **TC. ID** | **Expected Output** | **Actual Output** | **Status** |
| 1 | 2.2 | Profile Details not valid.  Please enter valid guide  name. | Edit Profile | Resolved |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 2.3 | Please fill all fields. | Still Screen | Resolved |
| 3 | 2.4 | Product file type not valid, only png, jpg, jpeg are accepted. | Profile update  successfully | Resolved |
| 4 | 3.3 | Shop name is not valid. Please enter valid Shop name. | Shop Added Successfully | Resolved |
| 5 | 4.3 | Product Description is not valid. Please enter valid Description. | Product added successfully. | Resolved |
| 6 | 5.3 | Order Proceed | System Crashed | Resolved |
| 7 | 6.3 | Image format is not supported. Only png, jpg, jpeg are accepted. | Product Listed successfully | Resolved |
| 8 | 14.2 | Name field is empty. | System Crashed. | Resolved |

### Error. ID – 11 Delete Guide, Page, Content Item

The validation for deletion was missing, when user clicks on delete button,

there should have been a prompt message for re-confirmation.

#### SUMMARY

In this chapter we have talked about list of technologies and libraries that is used in our project for quality development of our software system. We have mentioned Work break down structure WBS and roles and responsibilities Matrix. In addition, we also explained the deployment environment of the software, through deployment diagram. At the end we performed white box test case design upon code and created an error log at the end.

# Chapter 6 Conclusion and Outlook

## Chapter 6

## TESTING AND EVALUATION

### INTRODUCTION

We would like to wrap up our work on Rewear in this chapter. This chapter examines the project as a whole, our accomplishments to date, and the enhancements made to the system that was previously described. This chapter will also discuss our feelings and opinions over the project's successful conclusion.

### ACHIEVEMENTS AND IMPROVEMENTS

One of our biggest achievements during this project was our in-depth understanding as we worked on all aspects of (design, code, test) software engineering. By working closely together and tackling various challenges of the project as a team, we were able to learn and grow significantly. The process of developing this project was a valuable learning experience for all of us, and we are grateful for the opportunity to have gained such valuable knowledge. Overall, we feel that the skills and understanding we developed during this project have made it beneficial.

### CHALLENGES

During the development process, several challenges were encountered, influencing the project's trajectory and development decisions:

* Learning the technologies required to develop the system in a limited time was also a challenge.
* Communicating the system's details to stakeholders and gaining an understanding of their requirements and expectations were key priorities.

# APPENDIX

### USER TYPES AND PROFILE

There are 4 total types of system users

* Seller (Responsible for creating and managing listings products on the platform.)
* Buyer (Can purchase or acquire products and services listed by sellers)
* Tailor (Responsible for creating and managing listings Stitched products on the platform).
* Admin (responsible for administrative tasks i.e. user management, order management)

### MODULES FROM CURRENT SOFTWARE (SPRING SHARE)

* Authentication (only Tailor, Seller and Admin will have to Authenticate)
  + Login
  + Forget Password
* Buyer Module
  + Profile Completion
  + Purchase Product
  + Order Track
  + View Product
  + Search Product
  + View Shops
  + Filter Products
  + Favorite Product
  + Cart Product
  + Profile management
* Seller module
  + Profile Verification
  + List Product
  + Shop Creation
  + Manage Order
* Admin module
  + User management
  + Order Management
* Tailor module
  + Profile Verification
  + Showcase Product
  + Shop Creation
  + Manage Order

### Ease of use or Challenges

* There were no as such a challenge while interacting with the software, but the system was a bit outdated which tends take more time of the users while performing an activity.

### Key Features

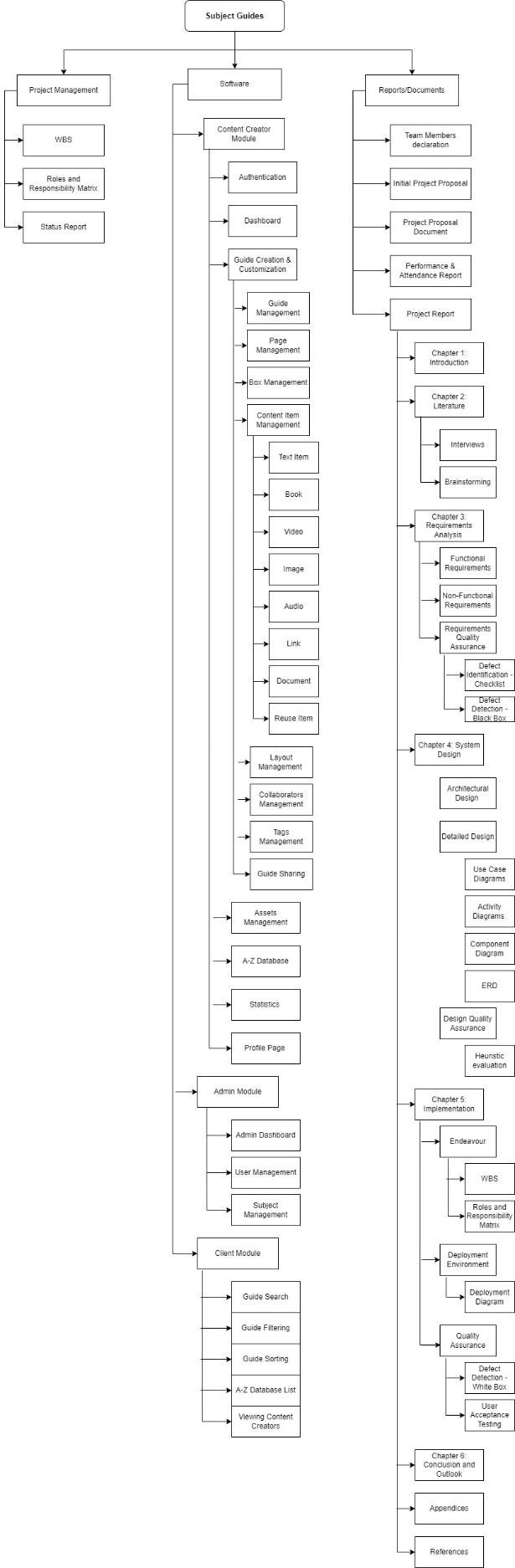
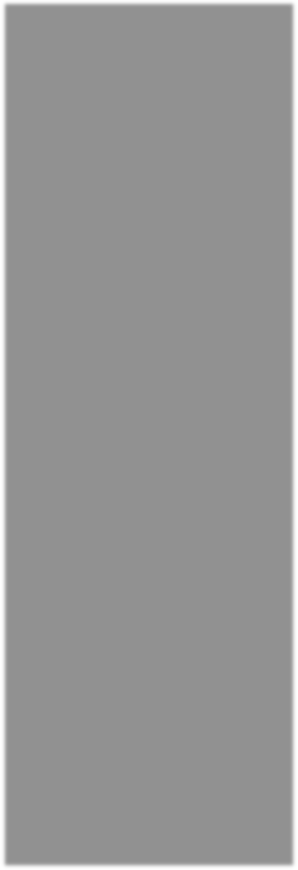
* Product Listing, modification Deleting and Order its related feature i.e. Filter, Shop.
* Searching

### Mobile Accessibility

* Need for mobile accessibility is High, it don’t require any additional feature for the mobile devices, therefore only a responsive website layout will be more than enough.

## Appendix – C

**WBS**



# References

## REFERENCES

#### Requirement Checklist

https://users.csc.calpoly.edu/~jdalbey/308/Deliver/SRS\_QA.html

1. **Heuristics Evaluation** [**https://www.nngroup.com/articles/ten-usability-heuristics/**](https://www.nngroup.com/articles/ten-usability-heuristics/)
2. **UML**

<https://jti.polinema.ac.id/wp-content/uploads/2019/02/Buku-Learning-UML-2.0.pdf>

1. **Google Drive Images** [https://drive.google.com/drive/folders/16LMOC2tIzCExqYjjpXKzki\_itp-](https://drive.google.com/drive/folders/16LMOC2tIzCExqYjjpXKzki_itp-nIApD?usp=drive_link) [nIApD?usp=drive\_link](https://drive.google.com/drive/folders/16LMOC2tIzCExqYjjpXKzki_itp-nIApD?usp=drive_link)

#### React

<https://reactjs.org/docs/getting-started.html>

#### Bootstrap

<https://getbootstrap.com/>

#### Font Awesome

<https://fontawesome.com/icons>

1. **Laravel** [**https://laravel.com/**](https://laravel.com/)
2. **Inertia JS**

<https://laravel.com/>

#### Prime React

<https://primereact.org/>

1. **Tailwind** [**https://tailwindcss.com/**](https://tailwindcss.com/)