



FATIMA ANIS
01-235182-094
TAYYABA ZIA
01-235182-072

Sparkles Gallery

Bachelor of Science in Information Technology

Supervisor: Mr. Ali Irfan

Department of Computer Science
Bahria University, Islamabad

June 2022

Certificate

We accept the work contained in the report titled “SPARKLES GALLERY”, written by Miss TAYYABA ZIA AND Miss FATIMA ANIS as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Information Technology.

Approved by . . . :

Supervisor: Prof Ali Irfan (Title)

Internal Examiner: Name of the Internal Examiner (Title)

External Examiner: Name of the External Examiner (Title)

Project Coordinator: Name of the Project Coordinator (Title)

Head of the Department: Name of the HOD (Title)

June, 2022

Abstract

We all know how much online shopping has taken a hike. People really prefer browsing about a particular product rather than visiting the shop for window shopping. They buy different products may it be house items, cosmetics, electronics, or jewelry products. There are different platforms through which people can buy or sell their products and the payment can be done via different payment methods. This platform is used specifically for the buying or selling of jewelry products. After our research, we for all intents and purposes find out that people don't trust online sites like Olx, Daraz, Ali baba, Amazon for jewelry because people for the most part think that these sites may deceive them as there basically is no method for verification, contrary to popular belief. To cover this loophole in the market we have developed a web app in secure programming languages. This service will specifically help people in searching of jewelry at their home, which essentially is quite significant. Jewelry is an expensive item all around the globe and for all intents and purposes stays in particularly high demand. This web app includes multiple users including buyer, seller, admin, and expert in a subtle way. Every user has their own general roles and permissions. Seller can sell jewelry and buyer can generally buy the jewelry. While experts can basically verify the authenticity and it all happens with just a click, which is significant. The commission on sales will be the main revenue stream of admin. The expert will get his payment. Users can access its jewelry from its personal account.

Acknowledgments

“In the name of God, the Most Gracious, the Most Merciful”.

First of all, we would like to thank Almighty ALLAH for giving us the opportunity to study. We have put all our efforts and knowledge gained during the whole degree in our project. We would like to express our gratitude towards our supervisor Mr. Ali Irfan for his encouragement and constructive suggestions for the project completion. We are truly thankful for his efforts. We would also like to thank our course teachers who taught us during the degree because, without their contribution, this project would not have been achievable. Our utmost regard also goes to our friends and class fellows for their constant support during the whole time for the project completion.

TAYYABA ZIA
FATIMA ANIS
Islamabad, Pakistan

May 2022

*“We think someone else, someone smarter than us,
someone more capable, someone with more resources will solve that problem.
But there isn’t anyone else.”*

Regina Dugan

Contents

| | |
|--|----------|
| Abstract | i |
| 1 Introduction | 1 |
| 1.1 Introduction | 1 |
| 1.1.1 Seller's Perspective | 1 |
| 1.1.2 Buyer's perspective | 1 |
| 1.1.3 Expert's perspective | 2 |
| 1.2 Background | 2 |
| 1.3 Objectives | 2 |
| 1.4 Problem Description | 2 |
| 1.5 Methodology | 2 |
| 1.5.1 Requirement Gathering | 3 |
| 1.5.2 Analysis | 3 |
| 1.5.3 Design | 3 |
| 1.5.4 System Architecture | 3 |
| 1.5.5 Development | 4 |
| 1.5.6 Maintenance/testing | 4 |
| 1.6 Project Scope | 4 |
| 1.7 Feasibility Study | 5 |
| 1.7.1 The Risk Involved | 5 |
| 1.7.2 Resource Requirement | 5 |
| 1.7.3 Requirement Feasibility | 5 |
| 1.8 Solution Application Area | 5 |
| 1.9 Model | 5 |
| 1.10 Tools/Technology | 6 |
| 1.10.1 Bootstrap Techniques | 6 |
| 1.10.2 HTML, HTML5, CSS, CSS3 | 6 |
| 1.10.3 JavaScript | 7 |
| 1.10.4 React | 7 |
| 1.10.5 JQuery | 7 |
| 1.10.6 PHP | 7 |
| 1.11 The Expertise of the Team Members | 7 |
| 1.12 Milestones | 8 |

| | | |
|----------|--|-----------|
| 2 | Literature Review | 9 |
| 2.1 | Related Work | 9 |
| 2.1.1 | OLX | 10 |
| 2.1.2 | Pak Wheels | 10 |
| 2.1.3 | Ali Baba | 11 |
| 2.1.4 | Daraz | 11 |
| 2.1.5 | Amazon | 12 |
| 2.1.6 | Home Shopping | 12 |
| 3 | Requirement Specifications | 13 |
| 3.1 | Existing System | 13 |
| 3.2 | Proposed System | 13 |
| 3.3 | Specifications | 14 |
| 3.3.1 | Software Specifications | 14 |
| 3.3.2 | Functional Requirements | 14 |
| 3.3.3 | Non-Functional Requirements | 15 |
| 3.4 | Use Cases | 16 |
| 3.4.1 | Use Case of Registration | 17 |
| 3.4.2 | Login | 19 |
| 3.4.3 | Add Products | 21 |
| 3.4.4 | View Jewelry | 23 |
| 3.4.5 | Request for Inspection | 25 |
| 3.4.6 | Place Order | 27 |
| 3.4.7 | Jewelry Inspection | 29 |
| 3.4.8 | Logout | 31 |
| 3.5 | Sequence Diagram | 33 |
| 3.5.1 | Registration | 33 |
| 3.5.2 | Login | 34 |
| 3.5.3 | Place order | 35 |
| 3.5.4 | View Account Details | 36 |
| 3.5.5 | Adding Products | 37 |
| 3.5.6 | Jewelry Inspection | 38 |
| 3.5.7 | Inspection Request | 39 |
| 3.5.8 | Logout | 40 |
| 4 | Design | 41 |
| 4.1 | System Design | 41 |
| 4.2 | System Architecture | 42 |
| 4.3 | Design Constraints | 42 |
| 4.3.1 | Programming Language | 42 |
| 4.3.2 | Design Standard | 42 |
| 4.3.3 | Assumptions and Dependencies | 43 |
| 4.3.4 | Design Methodology | 43 |
| 4.3.5 | External Interfaces | 43 |
| 4.4 | High Level Design | 44 |
| 4.5 | Low Level Design | 45 |
| 4.6 | Entity Relation Design | 46 |

| | | |
|----------|--|-----------|
| 4.6.1 | Attributes | 47 |
| 4.7 | Domain Constraint | 47 |
| 4.7.1 | User | 47 |
| 4.7.2 | Category | 48 |
| 4.7.3 | Order Product | 48 |
| 4.7.4 | Category Product | 48 |
| 4.7.5 | Order | 49 |
| 4.7.6 | Address | 49 |
| 4.8 | Activity Diagram | 50 |
| 4.8.1 | Login | 50 |
| 4.8.2 | Place Order | 51 |
| 4.8.3 | Search | 52 |
| 4.8.4 | Inspection | 53 |
| 4.8.5 | Add Product | 54 |
| 4.9 | GUI Design | 55 |
| 4.9.1 | Admin Module | 55 |
| 4.9.2 | User Module | 58 |
| 4.9.3 | Expert Module | 63 |
| 5 | System Implementation | 68 |
| 5.1 | System Architecture | 68 |
| 5.1.1 | Web Application | 68 |
| 5.1.2 | System Internal Component | 68 |
| 5.2 | Tools and Technology | 70 |
| 5.2.1 | Next JS | 70 |
| 5.2.2 | Typescript | 70 |
| 5.2.3 | Tailwind CSS | 70 |
| 5.2.4 | React Query | 70 |
| 5.2.5 | React JS | 70 |
| 5.2.6 | Framer Motion | 70 |
| 5.2.7 | Headless UI | 71 |
| 5.2.8 | Google Fonts | 71 |
| 5.2.9 | Node JS | 71 |
| 5.2.10 | Core JS | 71 |
| 5.3 | Methodology | 71 |
| 5.3.1 | Phase 1 | 71 |
| 5.3.2 | Phase 2 | 71 |
| 6 | System Testing and Evaluation | 72 |
| 6.0.1 | GRAPHICAL USER INTERFACE TESTING | 72 |
| 6.0.2 | USABILITY TESTING | 75 |
| 6.0.3 | EXCEPTION TESTING | 76 |
| 7 | Conclusions | 78 |
| 7.1 | Future Enhancements | 79 |
| 7.1.1 | Application | 79 |
| 7.1.2 | Bidding | 79 |

| | | |
|----------|---------------------------|-----------|
| 7.1.3 | Chat Box | 79 |
| A | User Manual | 80 |
| A.1 | Admin Module | 80 |
| A.2 | Shop Module | 82 |
| A.3 | Client Module | 83 |
| A.4 | Client Register | 86 |
| A.5 | Product | 88 |
| A.6 | Product | 90 |
| A.7 | Site Setting | 94 |
| | References | 97 |

List of Figures

| | | |
|------|---|----|
| 1.1 | System Architecture | 4 |
| 1.2 | SDLC Model | 6 |
| 1.3 | Milestone | 8 |
| 3.1 | Use Case of Sparkles Gallery | 16 |
| 3.2 | Use Case of Registration | 17 |
| 3.3 | Use Case of Login | 19 |
| 3.4 | Use Case of Adding Products | 21 |
| 3.5 | Use Case of Viewing Jewelry | 23 |
| 3.6 | Use Case of Request for Inspection | 25 |
| 3.7 | Use Case of Placing Order | 27 |
| 3.8 | Use Case of Jewelry Inspection | 29 |
| 3.9 | Use Case of Logout | 31 |
| 3.10 | Sequence Diagram of Sign Up | 33 |
| 3.11 | Sequence Diagram of Login | 34 |
| 3.12 | Sequence Diagram of Placing Order | 35 |
| 3.13 | Sequence Diagram of View Account Details | 36 |
| 3.14 | Sequence Diagram of Adding Products | 37 |
| 3.15 | Sequence Diagram of Jewelry Inspection | 38 |
| 3.16 | Sequence Diagram of Inspection Request | 39 |
| 3.17 | Sequence Diagram of Logout t | 40 |
| 4.1 | System Architecture | 42 |
| 4.2 | High Level | 44 |
| 4.3 | Low Level | 45 |
| 4.4 | Er Diagram | 46 |
| 4.5 | Activity Diagram Login | 50 |
| 4.6 | Activity Diagram of Placing Order | 51 |
| 4.7 | Activity Diagram of Search | 52 |
| 4.8 | Activity Diagram of Inspection | 53 |
| 4.9 | Activity Diagram of Adding Product | 54 |
| 4.10 | GUI of Login of Admin Module | 55 |
| 4.11 | GUI of Dashboard of Admin Module | 56 |
| 4.12 | GUI of Shops of Admin Module | 56 |
| 4.13 | GUI of Dashboard of Specific Shop of Admin Module | 57 |
| 4.14 | GUI of of Admin Module | 57 |
| 4.15 | GUI of Categories of Admin Module | 58 |

| | |
|---|----|
| 4.16 GUI of Home of User Module | 58 |
| 4.17 GUI of Login of User Module | 59 |
| 4.18 GUI of Shopping Cart of User Module | 60 |
| 4.19 GUI of Placing Order of User Module | 61 |
| 4.20 GUI of Purchase List of User Module | 62 |
| 4.21 GUI of Change Password of User Module | 62 |
| 4.22 Sign In of Expert Module | 63 |
| 4.23 Dashboard of Expert Module | 63 |
| 4.24 User Navigate to Verify Jewelry of Expert Module | 64 |
| 4.25 User Add Order Details of Expert Module | 65 |
| 4.26 Jewelry Order Added of Expert Module | 66 |
| 4.27 Detail Page of Expert Module | 66 |
| 4.28 Request List of Expert Module | 67 |
| 4.29 Inspection Request (Data Entry) of Expert Module | 67 |

List of Tables

| | | |
|------|--|----|
| 3.1 | Use Case of Registration | 18 |
| 3.2 | Use Case of Login | 20 |
| 3.3 | Use Case of Add Product | 22 |
| 3.4 | Use Case of Viewing Jewelry | 24 |
| 3.5 | Use Case of Request for Inspection | 26 |
| 3.6 | Use Case of Placing Order | 28 |
| 3.7 | Use Case of Jewelry Inspection | 30 |
| 3.8 | Use Case of Logout | 32 |
| 4.1 | Attributes of ER Diagram | 47 |
| 4.2 | User constraint | 47 |
| 4.3 | Category Constraint | 48 |
| 4.4 | Order Product Constraint | 48 |
| 4.5 | Category Product Constraint | 48 |
| 4.6 | Order Constraint | 49 |
| 4.7 | Address Constraint | 49 |
| 6.1 | Test Case for Login Screen | 73 |
| 6.2 | Test Case for Sign Up Screen | 73 |
| 6.3 | Test Case for Forget Password | 74 |
| 6.4 | Test Case for Home Page | 74 |
| 6.5 | Test Case for Login Screen | 75 |
| 6.6 | Test Case for CRUD Operation | 75 |
| 6.7 | Test Case for Login Screen | 76 |
| 6.8 | Test Case for Sign Up Page | 76 |
| 6.9 | Test Case for Forget Password | 77 |
| 6.10 | Test Case for Request Generation Testing | 77 |

Acronyms and Abbreviations

| | |
|------|-----------------------------------|
| SQL | Structured Query Language |
| HTML | Hyper Text Markup Language |
| PHP | Hypertext Preprocessor |
| JS | Java Script |
| CRUD | Create Read Update Delete |
| TC | Test Case |
| API | Application Programming Interface |
| UML | Unified Modelling Language |
| GUI | Graphical User Interface |
| CSS | Cascading Style Sheet |

Chapter 1

Introduction

1.1 Introduction

As we know that there are different platforms where people can interact with each other to buy or sell their products, for example, O.L.X, and Pak wheels but there is not one specific for the jewelry. There should be a platform in the jewelry market that somehow introduces a thrifting concept. It means that one can sell his jewelry (used or unused) on this website without any loss. The buyer can buy the jewelry after verifying it with the help of some jewelry expert. Using this website huge markets (like real business companies) can also sell their jewelry.

1.1.1 Seller's Perspective

1. Create his account
2. Upload photos of the jewelry set and its info.
3. Payment procedures.

1.1.2 Buyer's perspective

1. Create his account.
2. Contacts the seller.
3. Payment procedures.
4. Rate, and review the jewelry and the seller.

1.1.3 Expert's perspective

1. When buyer and seller wish to make a deal. An expert is contacted. Which checks whether the jewelry being sold is real or not and the other features of jewelry are confirmed by the jewelry expert.
2. He generate the jewelry inspection report.

1.2 Background

With the advancement in technologies, people want to buy things online. There are a lot of websites like PakWheels, and Olx.com through which people are selling their used products. But there is no specific platform on which people sell their used jewelry. If a person buys jewelry set in 1 lac and he wants to sell it. The jeweler will buy it for 50 thousand. It means that person who wants to sell jewelry is at the lowest price. For this purpose, we are making a website just for the jewelry selling purpose to make people's lives easier. They can sell and buy jewelry without getting lost.

1.3 Objectives

Develop a website that facilitates effective communication and exchange of jewelry between the buyer and seller without any loss.

1.4 Problem Description

As there is no specific platform for the jewelry owner to sell his/her used jewelry. So, when the jewelry owner sells his jewelry to the jeweler. He gets a loss because the jeweler buys it at less price. Similarly, some customers want to buy used but good jewelry but there is no such platform for those people. We are providing a single platform where people can sell and buy used jewelry.

1.5 Methodology

Sparkles gallery is an incremental model which consists of the following steps:

1. Requirement Gathering
2. Analysis
3. System Design
4. System Architecture

5. Deployment

6. Maintenance/Testing

1.5.1 Requirement Gathering

We reviewed other web-based software available on the internet and try to implement their capabilities in our software.

1.5.2 Analysis

We can remove the general issue of loss between the buyer and seller through this project. Rather than selling the jewelry again to the jewelry owner at the lowest price. He can simply sell it to the interested buyer using this website. The jewelry will be verified by the expert.

1.5.3 Design

The design of the project will be simplistic and easy to use to not confuse the user and make sure the user can get the most from the application it will be in the same basic design.

1.5.4 System Architecture

In system architecture, there are three main layers

1. Presentation layer
2. Business Logic layer
3. Database Access layer

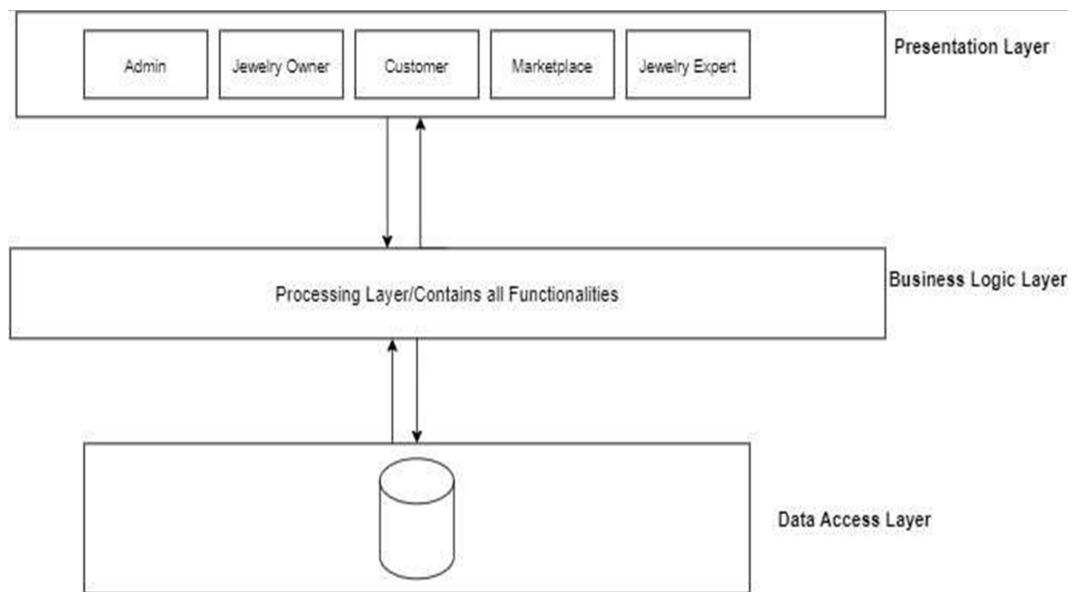


Figure 1.1: System Architecture

1.5.5 Development

The development of the project is carried out according to the milestones set and each phase is thoroughly tested as this industry demands perfection.

1.5.6 Maintenance/testing

Before the launch, the website is thoroughly tested, and the final project is made sure to be bug-free and is tested many times beforehand on a virtual platform.

1.6 Project Scope

Our project will allow users to collaborate on a single platform. The system intended mainly focuses on buyers, sellers, marketplace, and experts for exchanging the exchange of jewelry.

1. The seller will upload the picture of the jewelry he wants to sell and the information about the jewelry.
2. Will make a deal with the buyer and sell the product.
3. The buyer will look for the jewelry he likes and will check the information.
4. He will contact an expert if he has any or will use a website expert to check if the jewelry is authentic.
5. The payment can be done hand to hand.

1.7 Feasibility Study

1.7.1 The Risk Involved

There is no visible risk but during the project, if it occurs, we tackle it through our project's dynamic functionality and elasticity.

1.7.2 Resource Requirement

We do not require any special computing or hardware technology, in testing, we will have to make sure that our website works along with all types of devices which can be easily done.

1.7.3 Requirement Feasibility

We won't be generating any profit at the start. But with time, we will start taking 1% from the jewellery owner.

1.8 Solution Application Area

The proposed system would eliminate the problems of the buyer, and sellers who want to buy or sell their jewelry on a website. The seller can sell his jewelry to the desired customer without any loss Rather than selling it back to the market in the loss. Same as the seller, the buyer would be able to buy the jewelry at a good price for the thrift-ed product. This system would work as a web-based system and this system would be available to everyone who wishes to be a part of this jewelry exchange industry.

1.9 Model

A system is created comprising web technology. This system is following the Incremental modeling approach. This helps in different developing features based on an incremental approach which develop the described model as shown in figure [1.2](#).

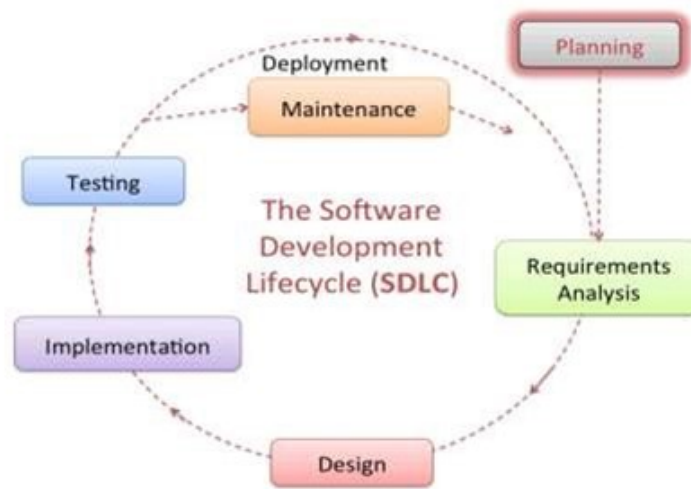


Figure 1.2: SDLC Model

1.10 Tools/Technology

1. Visual Studio
2. Visual Studio Code
3. Microsoft SQL Server Management Studio

Many techniques are being used in this system. Some of them are defined below.

1.10.1 Bootstrap Techniques

Bootstrap is a free and open-source front-end library for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation, and other interface components, as well as optional JavaScript extensions.

1.10.2 HTML, HTML5, CSS, CSS3

HTML is a markup language for creating web pages. Web pages are usually viewed in a web browser. They can include writing, links, pictures, and even sound and video. HTML is used to mark and describe each of these kinds of content so the web browser can show them correctly. HTML (the Hypertext Markup Language) and CSS (Cascading Style Sheets) are two of the core technologies for building Web pages. HTML provides the structure of the page, CSS the (visual and aural) layout, for a variety of devices. HTML5 is an advanced version of HTML with more new tags and powerful features. Similarly,

CSS3 is an advanced version of simple CS. We will use HTML and CSS to create websites attractive and user-friendly.

1.10.3 JavaScript

JavaScript is a front-end scripting language developed by Netscape for dynamic content, it is a client-side technology that is embedded in your HTML page and interpreted by the web browser. It is very lightweight, simple and flexible and it can be used as an object-oriented language. To make the user experience easier and better while going through the website JavaScript techniques will be used.

1.10.4 React

React is a free and open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications.

1.10.5 JQuery

jQuery is a small and lightweight JavaScript library, this is used for animations, and event handling and it does the same event handling as we do with JavaScript more easily. We use jQuery with Ajax and some small animations.

1.10.6 PHP

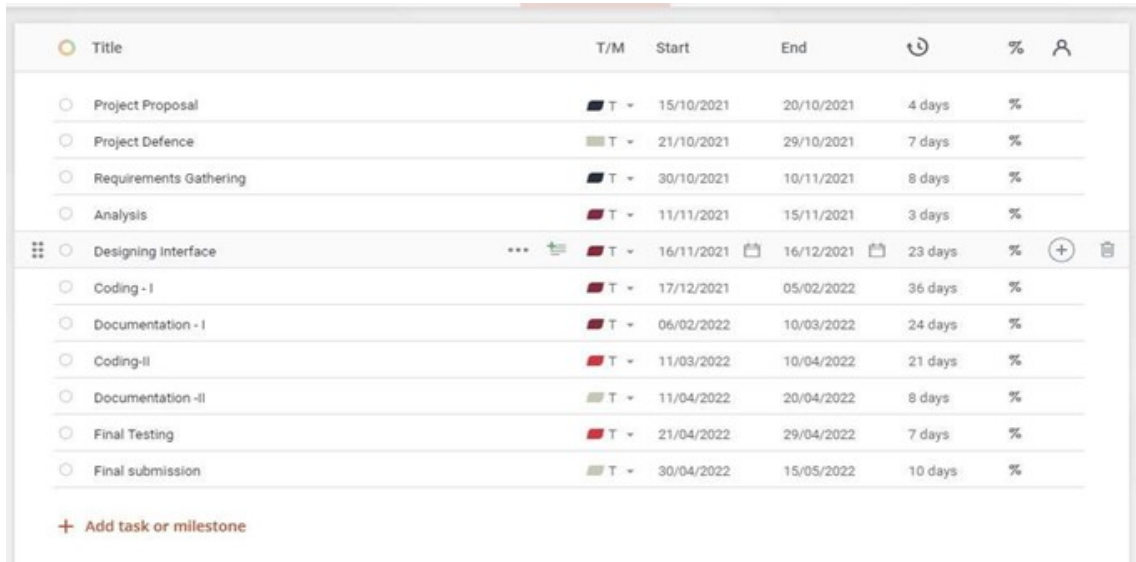
PHP is a general-purpose e-scripting language geared toward web development. It is open source widely used language. PHP scripts are executed on server.

1.11 The Expertise of the Team Members

We both are experienced in both web designing and web development as we studied the web system and technology and another web-related course in our previous semesters. The team has the necessary knowledge to complete the task.

1.12 Milestones

Note: In the figure 1.3 is an approximated milestone.



| Title | T/M | Start | End | Duration | % | |
|------------------------|-----|------------|------------|----------|---|---|
| Project Proposal | T | 15/10/2021 | 20/10/2021 | 4 days | % | |
| Project Defence | T | 21/10/2021 | 29/10/2021 | 7 days | % | |
| Requirements Gathering | T | 30/10/2021 | 10/11/2021 | 8 days | % | |
| Analysis | T | 11/11/2021 | 15/11/2021 | 3 days | % | |
| Designing Interface | T | 16/11/2021 | 16/12/2021 | 23 days | % | + |
| Coding - I | T | 17/12/2021 | 05/02/2022 | 36 days | % | |
| Documentation - I | T | 06/02/2022 | 10/03/2022 | 24 days | % | |
| Coding-II | T | 11/03/2022 | 10/04/2022 | 21 days | % | |
| Documentation -II | T | 11/04/2022 | 20/04/2022 | 8 days | % | |
| Final Testing | T | 21/04/2022 | 29/04/2022 | 7 days | % | |
| Final submission | T | 30/04/2022 | 15/05/2022 | 10 days | % | |

+ Add task or milestone

Figure 1.3: Milestone

Chapter 2

Literature Review

Sparkles Gallery provides an online opportunity to buy or sell used jewelry with just simple clicks on the website. It is a secure way and easy way of buying jewellery at less price. The user can easily buy jewelry based on inspection.

In this chapter, we will talk about any type of existing system related to our Project and evaluate that system with our project. This will help us to give a direction to reach our destination. The modules of our projects are unique and hold a lot of significance. We are working on a new idea. In this chapter, we will talk about different modules and features that will identify that our idea is unique.

We have done some research on the work of other people on systems which are selling jewelry online and how we can use their knowledge to develop a better online system. We have proposed some improvements that could be included in this report. There is no system in Pakistan that allows user to sell their used jewelry, so we wanted to develop this system. Following are some basic documents that have completed some e-commerce systems and have contributed to the creation of our project.

2.1 Related Work

Sparkles gallery provides an opportunity of trading jewelry in an efficient way. One gets or sell their jewelry with the help of their laptops etc.

In this chapter, we have done some of the research on the work of other people related to this project. We are familiar with Ali Baba and Daraz also trade different objects all over the world, but our idea has some others features as well. We are well familiar with Pak-

wheels (the platforms that is used for the trade of cars) we compare their work with our website, the idea of sparkles gallery is pretty much the same, but it is particularly for the jewelry. That is why we wanted to develop this platform. The following are some basic documents that have completed some online management jobs and have contributed to the creation of our project.

2.1.1 OLX

There are many websites which sell jewellery. OLX is an ecommerce platform that offers online consumers the chance to browse and select many different products among a wide variety of categories including movies, electronics, computers, and jewellery at reasonable prices. One can also post about a job related adds or the property that includes big houses small houses, apartments, and plots.

People simply advertise about the products they wish to sell along with all of the details and descriptions.[1] The interested parties contact the selling party through a phone. They meet and decide whether the product is worth buying or not.

The description of the products includes the model, the size, the warranty, the color, some pictures from different angles and the price of the product. the description of job includes details of job like whether the job is part time or full time, salary of the desired position, required gender (male, female or both can apply) [2]. OLX is a platform for different categories. Whereas Sparkles Gallery is platform for only jewellery.

2.1.2 Pak Wheels

Pak Wheels is a platform through which people from different areas connect at one point for the buying and selling purposes of their cars, bikes, and auto parts online [3]. The categories are divides as used cars, new cars, bikes, family cars, automatic cars, Japanese cars, and auto parts.

The customers also do bid and ask the expert to do inspection of the cars. Then the expert generates the report. In Sparkles Gallery, customer also request for the inspection. Then our expert inspects the jewellery and generate the report. There is also an option for user to bid the jewellery. And buy the jewellery in reasonable price.

Pak wheels give you two main options in order to sell your products. Options one is that you can post and add on the Pak wheels and wait for the parties to see the add and contact you through the add. Option two is that you let Pak wheels sell your car for you. Basically, in this option for some people (car experts) visit the seller checks the condition

and specs of the car. And posts and add about your car according to specs and take some revenue from the selling party[4]. The benefit of this option is that the particular add is responded in lesser time because people easily trust the add published from the official account of Pak wheels because of its authenticity.

2.1.3 Ali Baba

Ali Baba is a Chinese platform through which people connect from all over the world to buy or sell the whole sale products. By the term sale we mean that a whole stock of the products is sold in cheap prices comparatively.

Basically, in Ali Baba there are around millions of presubscribed sellers and millions of products online for the sell. [5]Ali Baba is has become the most popular destination over these years for online shopping as it is one of the world's largest and fastest ecommerce market. Its online transaction in last year was calculated as \$ 248 billion.[6] It is online business to business platform where different people carry out many transactions in order to buy or sell different products. This <https://www.overleaf.com/project/62b1846b3e52d925140e44e7> is the difference that makes this platform unique as compared to other products, the other products are business to consumer platforms.

2.1.4 Daraz

Daraz is the e-commerce platform is South Asia and shopping application in Pakistan that provides its users with end-to-end e-commerce solutions. Daraz online buying app is your on- hand grocery buying app, utility app, and one in every of Pakistan's first-rate buying apps. Daraz makes online shopping in Pakistan convenient and hassle-free in Pakistan. [7]It serves around millions of customers from all over Pakistan, Bangladesh, Myanmar, Nepal and SriLanka. It brings millions of customers ever month. It means that the local seller has grown over this year on this platform that too very visible. Basically, the whole business model works like a marketplace where seller creates a virtual shop at this platform and then lists their products alongside the prices. [8] Daraz had to face a lot of challenges that includes the higher costs of the infrastructure, lack. of collaboration, they were unable to serve quickly and most importantly the outdated systems that do not meet the requirement of the company. Some of these challenges they have to face even today and they're trying to overcome these challenges as well.

2.1.5 Amazon

Amazon is a huge virtual marketplace or enterprise which sells household things, electronics, toys, music, films, and so many other things to retailers and millions of customers. People from all over the world can buy things and services from Amazon. They have recently added Pakistan to its list of per-mutable countries for sellers. It means that the people from Pakistan having the services can sell them using Amazon which will definitely help in increasing the digital economy of the country. But even before it was added to the list people tried to sell products offline to supports industry of workers. [9] The sellers are required to provide banking details and other information to signing to their account in order to sell the products.

2.1.6 Home Shopping

Home shopping is a very old e-commerce website of Pakistan. Since it is very old it has so much experience in the market and a really reasonable user base. It is the industry of retailing of the household items like electronics.[10] Apart from traditional shopping for household. Goods for household are bought in very privacy. The categories are divided into home appliances, gaming material, photography etc. the details of the products include name of the product, name of the seller, price of the product and its condition.[11] It is a business to consumer platform. Its mission it to have and provide every product that the customer wishes for in the lowest rates possible.

Chapter 3

Requirement Specifications

3.1 Existing System

Some systems are working on this domain already for example Pak Wheels, AliBaba and Daraz but there is no platform for particularly jewelry in Pakistan. Pakwheels trade the cars, and the others (AliBaba and Daraz) are used for general products. Mostly used in homes. Many users use these platforms for trading their products. They get their products within some business days. Along with some pros these products also have some cons like the customers of Daraz and AliBaba mostly are concerned about the authenticity of the sellers and the products. As there are no platforms used for jewelry specifically. There is a need of such a platform where users can trust the platform and the other users.

3.2 Proposed System

As I mentioned there is no platform that can be used for the trade of jewelry. It is because jewelry is a huge asset and trusting the other person is difficult when it comes to investing huge amount of money. Using "Sparkles Gallery" users can easily trade their jewelry products with lesser prices. Now it is usually a question in the customers mind that whether the jewelry being bought is original or not, for this purpose the jewelry being sold will be inspected by a jewelry expert if the customer wished to do it. Rates and reviews for the product and the seller will also be added so that it becomes easier for the other customers to buy. Using the Sparkles Gallery huge marketplaces for example, Hanif Jewelers can also sell their products so the customers can find different products from different companies in one platform.

3.3 Specifications

3.3.1 Software Specifications

The hardware requirements are

1. Laptop/PCs
2. Google Chrome Browser
3. SQL management server
4. Windows
5. Visual Studio Code

3.3.2 Functional Requirements

3.3.2.1 Register

The core requirement of the website is to register. a user cannot sell. Buy or review the products or services if he has not registered himself. First name, last name, username, email id, password, confirm password will be the attributes for the users to provide in order to register themselves. User details are checked with the database. Password constraint is checked as per validation.

3.3.2.2 Login

For authorized access every customer has a username and a password. The system will check the input of the user and if valid then login is done. Otherwise, the user will be asked to re-enter the username and password. After successfully logging in profile page displayed for the user.

3.3.2.3 Password Forget

The password can be changed in case the user forgets the password. By mail services password will be reset for the account. Reset option is providing the option to change your password.

3.3.2.4 Seller Uploads Jewelry Details

Seller uploads the image of product that he wishes to sell along with the other details like type, category etc. After he has uploaded a message saying the product has been added will be displayed.

3.3.2.5 Requesting the Expert

The seller requests the expert to inspect the jewelry and create a report about it.

3.3.2.6 Logout

This option will help you log out from your account, securing it from other to use. User is willing to not have further services and the user takes actions.

3.3.3 Non-Functional Requirements

3.3.3.1 Efficiency

Provide with good accuracy for searching and other tasks.

3.3.3.2 Usability

Application is interactive and easy to use, its user friendly.

3.3.3.3 Flexibility

Enough space should be there so that when more numbers of products of users added, it still works fine and adapts the change.

3.3.3.4 Security

Data and transaction need to be secure. Data privacy policy will instil trust into our website and convert the customers into our brand advocates.

3.3.3.5 Performance

With the increase in traffic, performance becomes a priority in the Non-functional Requirements.

3.3.3.6 Scalability

This requirement defines how the website can grow and expand its functionality without affecting its performance.

3.4 Use Cases

Given below in figure 3.1 is the use case of Sparkles Gallery.

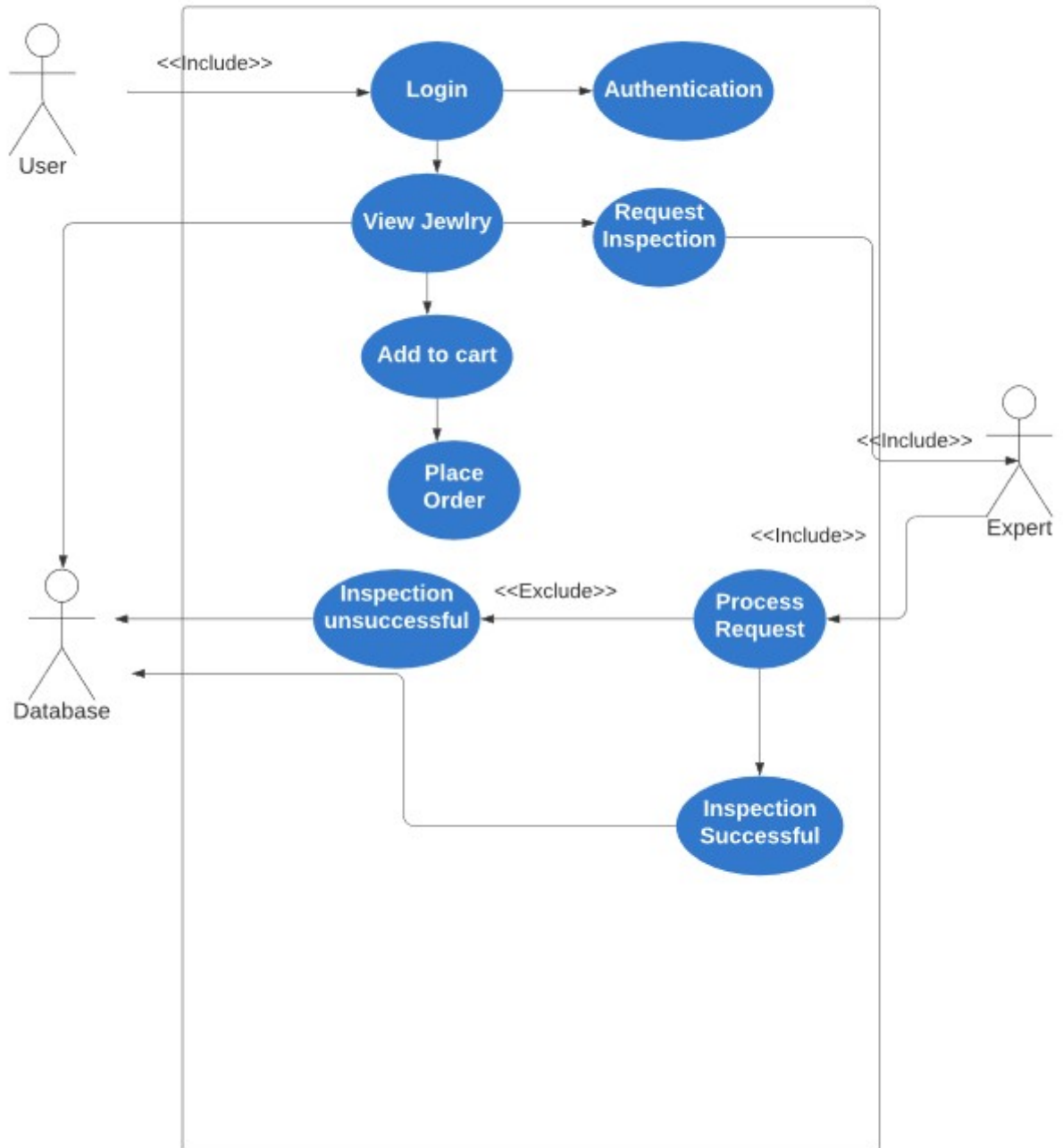


Figure 3.1: Use Case of Sparkles Gallery

3.4.1 Use Case of Registration

Given below in the figure 3.2 is the use case of registration of Sparkles Gallery.

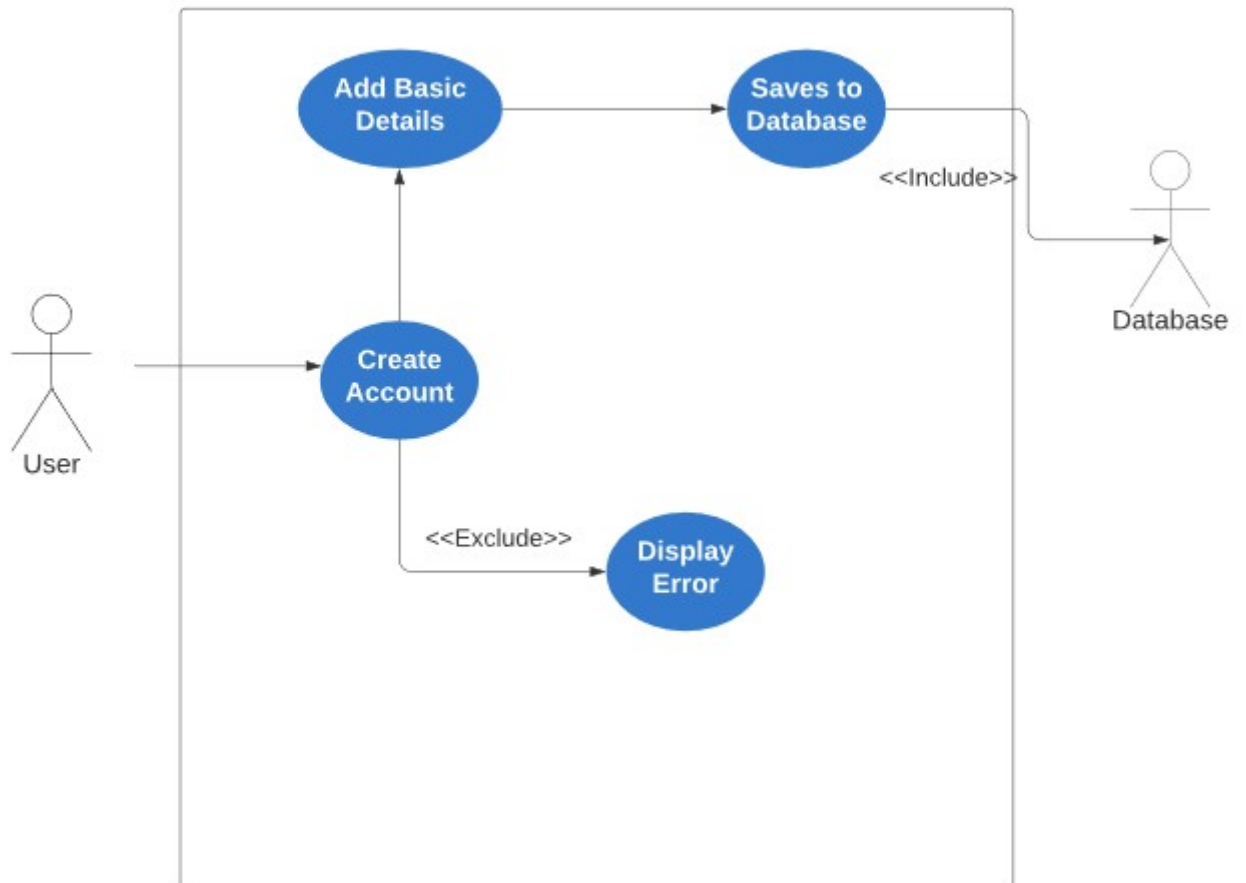


Figure 3.2: Use Case of Registration

3.4.1.1 Registration

Given below is the use case of registration in table 3.2. To access the system and use it the user must be registered into the system.

Table 3.1: Use Case of Registration

| | | |
|-----------------------|--|--|
| Used Case Name | Registration | |
| Use Case ID | SG-BU01 | |
| Description | The use case allows user to Sign Up him/her into our database. So, he can login to use the system. | |
| Actors | Users | |
| Basic Flow | <ol style="list-style-type: none"> 1. The user finds the "Register" page. 2. The user enters the required details on the "Register" form. 3. The system validates the details and then creates account of of that customer. | |
| Alternate Flow | Invalid Details: The user finds the "Register" page. <ol style="list-style-type: none"> 1. The user enters the required details on the "Register" form. 2. The system validates the details, and the record of that customer is not found. | |
| Trigger | The use case is initiated when a new user tries to use this application | |
| Pre-Condition | None. | |
| Post-Condition | Login page displayed. | |
| Priority | High | |
| Course Events | Actors Action | System Response |
| | <ol style="list-style-type: none"> 1. The user gives all the required information on the page. | <ol style="list-style-type: none"> 1. The information submitted is checked. 2. The system provides the details. 3. Availability of the previous record of the same information is checked and no duplication of records is checked. |

3.4.2 Login

Given below in figure 3.3 is the use case of login.

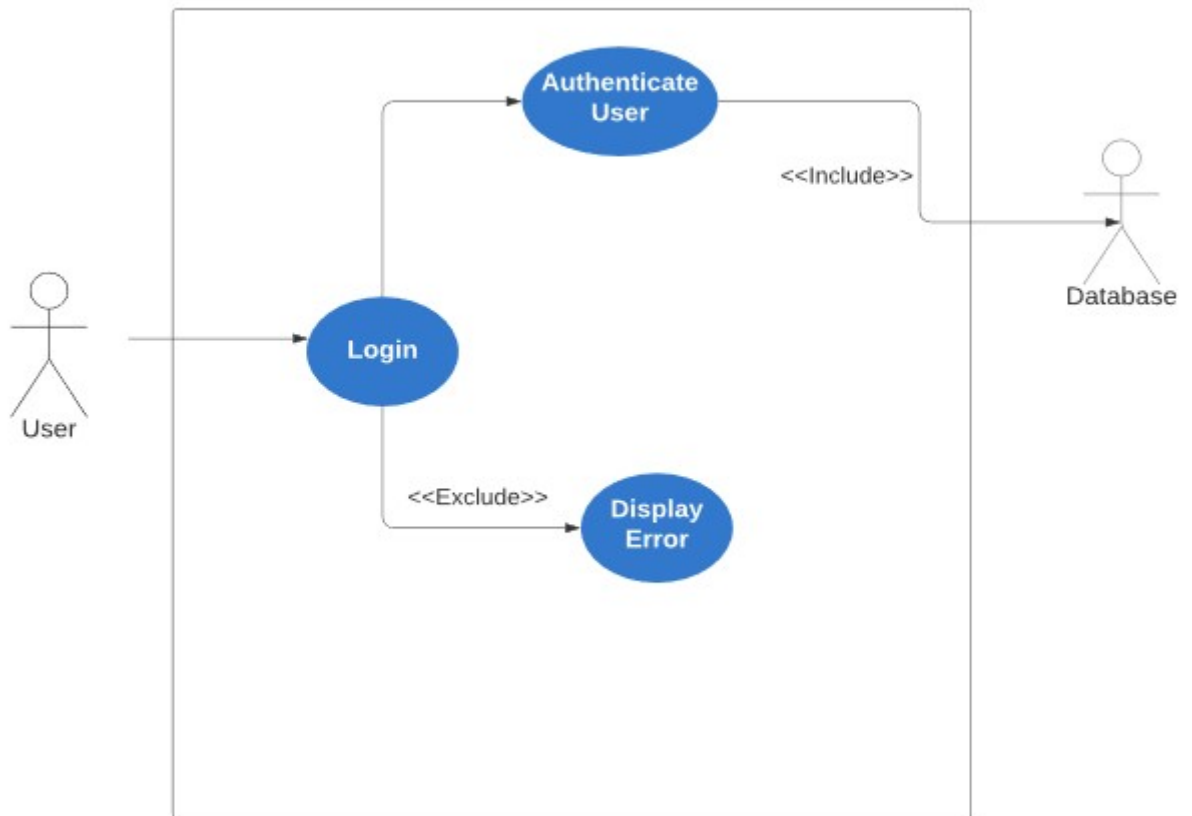


Figure 3.3: Use Case of Login

3.4.2.1 Login

Given below in table 3.2 is the description of the use case login. To access the system and use it the user must be logged into the system using its username and password.

Table 3.2: Use Case of Login

| | | |
|-----------------------|--|---|
| Used Case Name | Login | |
| Use Case ID | SG-BU02 | |
| Description | This used case helps the user to access the system to use it. Only the person registered can login. | |
| Actors | Users | |
| Basic Flow | <ol style="list-style-type: none"> 1. The user finds the "Sign In" page. 2. The user enters the required details on the "Sign In" form. 3. The system validates the details and then gives access on the sign in page of that customer. | |
| Alternate Flow | <p>The Incorrect Password: The user finds the "Sign In" page.</p> <ol style="list-style-type: none"> 1. The user enters the required details on the "Sign In" form. 2. The system validates the details, and the record of that customer is not found. | |
| Trigger | The use case is initiated when a new user tries to use his account for services. | |
| Pre-Condition | User ID and password must be in the database. | |
| Post-Condition | User profile opens. | |
| Priority | High | |
| Course Events | Actors Action | System Response |
| | <ol style="list-style-type: none"> 1. The user gives all the required information on the page. | <ol style="list-style-type: none"> 1. The information submitted is checked. 2. The system verifies the provided details. 3. If details do not match, the user cannot be logged in. |

3.4.3 Add Products

Given below is the use case of Adding Products in figure 3.4.

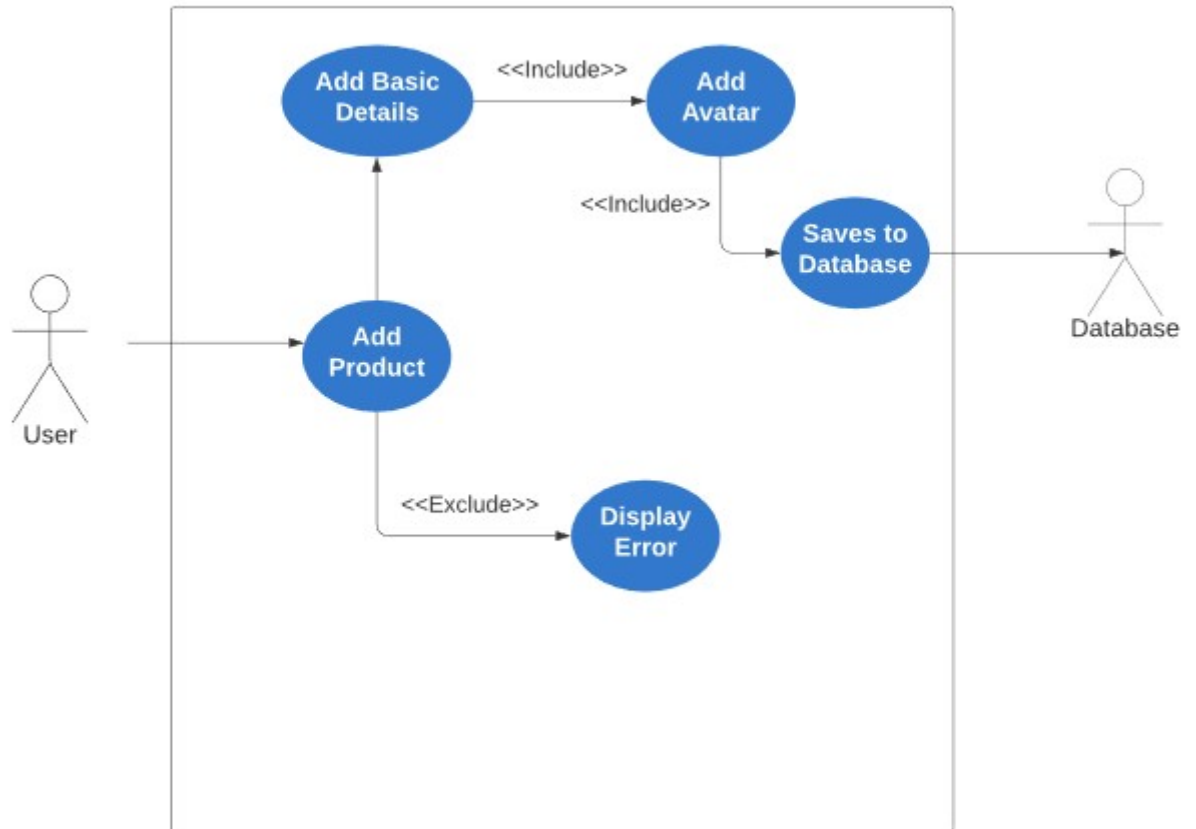


Figure 3.4: Use Case of Adding Products

3.4.3.1 Adding Products

The below given is the table 3.3 that shows the description of the used case adding products. In order to add a product, the seller must be logged into his account. He must have created the shop and the shop is active.

Table 3.3: Use Case of Add Product

| | | |
|-----------------------|---|---|
| Used Case Name | Add Product | |
| Use Case ID | SG-BU03 | |
| Description | The seller can add the product along with its details which as name, images, amount, weight, etc. | |
| Actors | Seller | |
| Basic Flow | <ol style="list-style-type: none"> 1. The user finds the "Add Product" page. 2. The user enters the required details on the "Add Product" form. 3. The system validates the details and then saves them in database. | |
| Alternate Flow | <ol style="list-style-type: none"> 1. The user finds the "Add Product" page. 2. The user enters the required details on the "Add Product" form. 3. The system validates the details, and save it in database. | |
| Trigger | The use case is initiated when a new user wishes to sell his product. | |
| Pre-Condition | The user must be an authorized person. | |
| Post-Condition | Details in the database are saved. | |
| Priority | High | |
| Course Events | Actors Action | System Response |
| | <ol style="list-style-type: none"> 1. Uploads the details that like name, category, image, and price of the product. | <ol style="list-style-type: none"> 1. Product display. 2. Stores in database. |

3.4.4 View Jewelry

The figure 3.5 below given is the use case diagram of viewing order

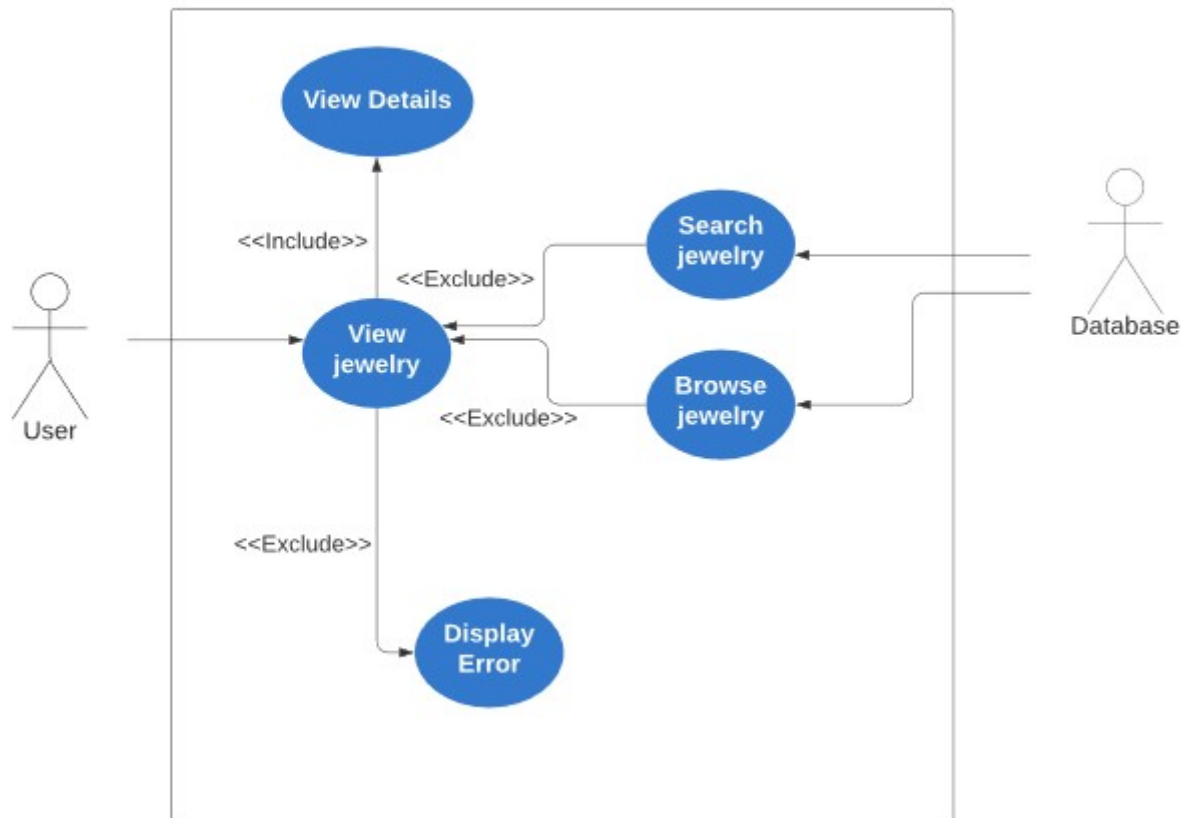


Figure 3.5: Use Case of Viewing Jewelry

3.4.4.1 Viewing Jewelry

The below given is the table 3.5 that shows the description of the used case view jewelry. In order to buy a jewelry product, the buyer must be able to view the whole image and the details of the jewelry. The details include picture, price, size, category, type of the jewelry.

Table 3.4: Use Case of Viewing Jewelry

| | | |
|-----------------------|---|--|
| Used Case Name | View Jewelry | |
| Use Case ID | SG-BU04 | |
| Description | To buy a product, the customer will search for them on the website, he can view the product and its details (type, category, weight, price etc.). | |
| Actors | User, Expert | |
| Basic Flow | <ol style="list-style-type: none"> 1. Lands on the page. 2. Moves to the product page. 3. Views the product. | |
| Alternate Flow | Product page opens using some link. | |
| Trigger | This used case is initiated when a user lands on the website and taps on product View. | |
| Pre-Condition | The details of the product must be in the database | |
| Post-Condition | The screen will be displayed having the button add to cart or something. | |
| Priority | High | |
| Course Events | Actors Action | System Response |
| | 1. Taps on the product details. | 1. Product appears along with the details. |

3.4.5 Request for Inspection

The below given figure 3.6 is the use case diagram of request for inspection.

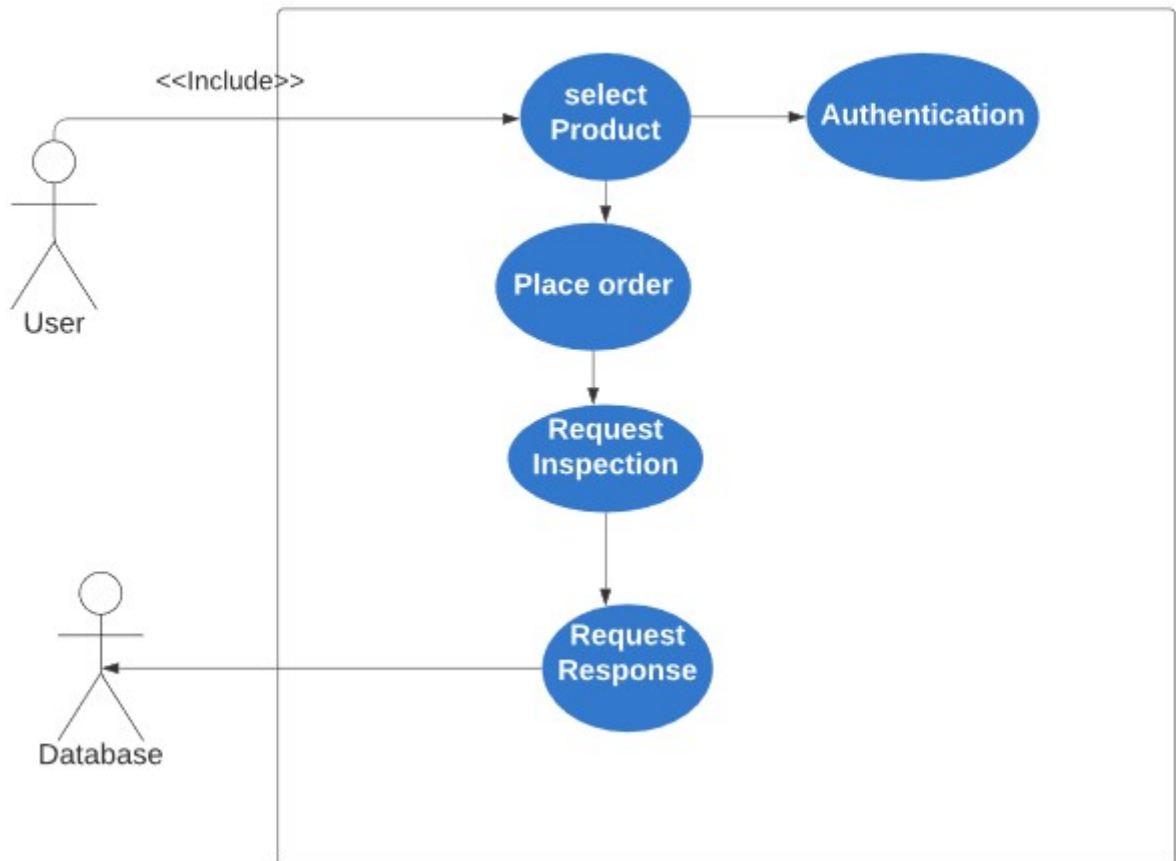


Figure 3.6: Use Case of Request for Inspection

3.4.5.1 Request for Inspection

The below given is the table 3.5 that shows the description of the used case request for inspection. To check whether the product is according to what has been specified by the seller, the buyer can request for the inspection where a jewelry expert would take a thorough look of the jewelry and inform the buyer.

Table 3.5: Use Case of Request for Inspection

| | | |
|-----------------------|---|--|
| Used Case Name | View Jewelry | |
| Use Case ID | SG-BU05 | |
| Description | The Customer will request for inspection with any of the jewelry experts of his choice to inspect whether the jewelry he selected is original/ worth it or not. | |
| Actors | Customers | |
| Basic Flow | <ol style="list-style-type: none"> 1. Lands on the page. 2. Chooses a product and open the select the vendor page. 3. Select the vendor. 4. Fills out the request jewelry form and submit it. | |
| Alternate Flow | 1. System does not save the details in the system. | |
| Trigger | This used case is initiated when a user wants to request for inspection. | |
| Pre-Condition | He must have signed in his account. | |
| Post-Condition | Waits for the request approval. | |
| Priority | High | |
| Course Events | Actors Action | System Response |
| | <ol style="list-style-type: none"> 1. Fills the request for inspection form. 2. Submit the form. | <ol style="list-style-type: none"> 1. Stores the details in the database. 2. Request sent to the expert. |

3.4.6 Place Order

The below given in figure 3.7 is the use case diagram of placing order.

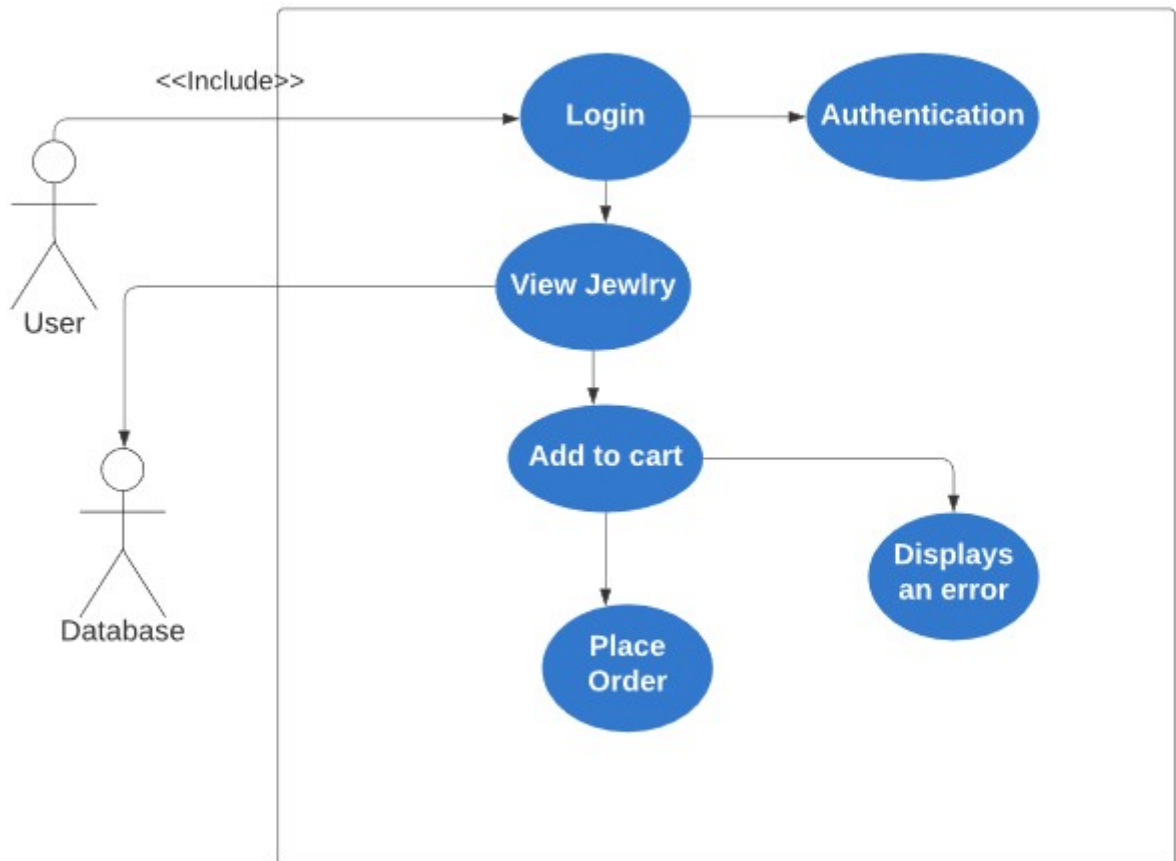


Figure 3.7: Use Case of Placing Order

3.4.6.1 Use Case of Placing Order

The below given is the table 3.6 that shows the description of the used case place order. After the buyer is completely sure that the product is fine, he can place an order. By that we mean that he can add the product to the cart and proceed to add the shipment details.

Table 3.6: Use Case of Placing Order

| | | |
|-----------------------|--|---|
| Used Case Name | Place Order | |
| Use Case ID | SG-BU06 | |
| Description | The customer upon choosing a product will place an order and wait for delivery. | |
| Actors | Customers | |
| Basic Flow | <ol style="list-style-type: none"> 1. Lands on the page. 2. Chooses a product and open the select the vendor page. 3. Select the vendor. 4. Pop up the confirmation message. | |
| Alternate Flow | None. | |
| Trigger | This use case study is initiated when the customer selects a product and wishes to buy it. | |
| Pre-Condition | <ol style="list-style-type: none"> 1. He must have signed in his account. 2. He must have select the product and add it to cart. | |
| Post-Condition | Waits for the confirmation notification. | |
| Priority | High | |
| Course Events | Actors Action | System Response |
| | <ol style="list-style-type: none"> 1. Select the product. 2. Add to cart. 3. Fills the form. 4. Submit the form. | <ol style="list-style-type: none"> 1. Displays the form. 2. Stores the details in the database. |

3.4.7 Jewelry Inspection

The below given in figure 3.8 is use case diagram of jewelry Inspection.

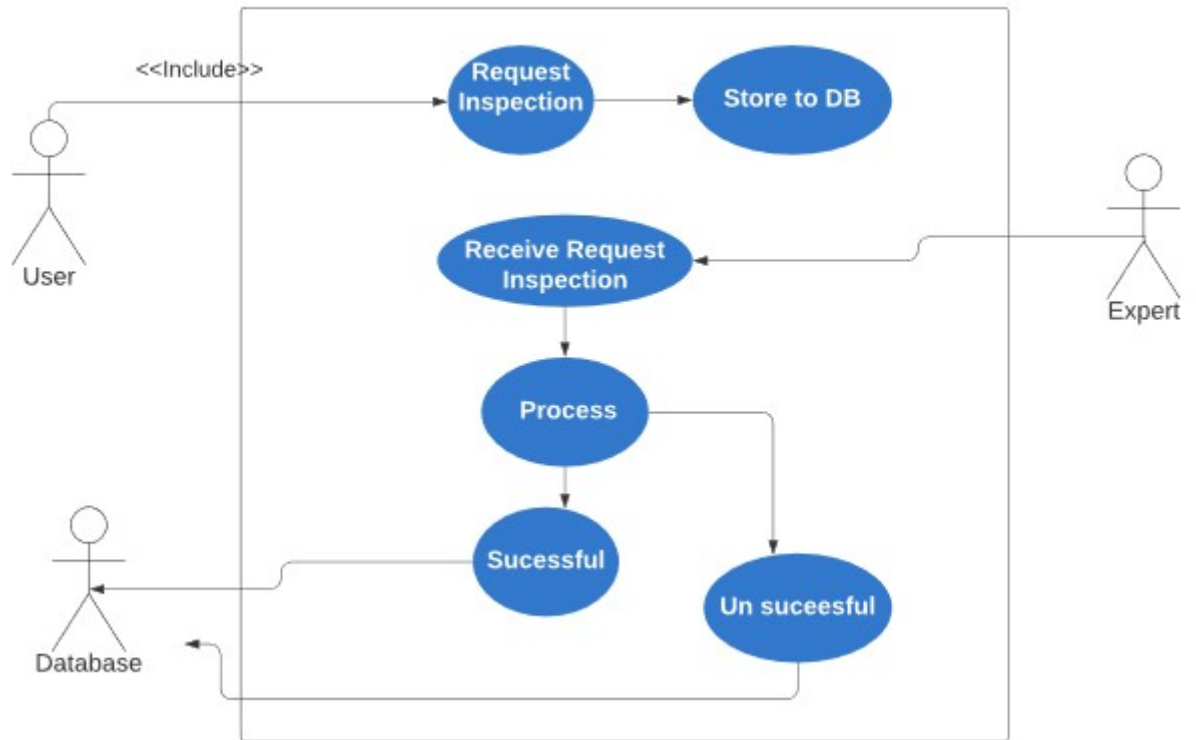


Figure 3.8: Use Case of Jewelry Inspection

3.4.7.1 Use Case of Jewelry Inspection

The below given is the table 3.8 that shows the description of the used case jewelry inspection report. After the expert views the product, he will generate a detailed report based on some attributes of the jewelry like weight, category, mix and match etc.

Table 3.7: Use Case of Jewelry Inspection

| | | |
|-----------------------|---|--|
| Used Case Name | Jewelry Inspection | |
| Use Case ID | SG-BU07 | |
| Description | The jewelry will be reviewed by a jewelry expert and a brief report will be generated based on that. Attributes for the report are type (gold, silver or diamond) or weight etc. | |
| Actors | Expert | |
| Basic Flow | <ol style="list-style-type: none"> 1. log in to his account. 2. Got a message about the inspection request. 3. After inspection, upload the inspection report. | |
| Alternate Flow | None. | |
| Trigger | This use case is initiated when the customer sends the inspection request to the expert. | |
| Pre-Condition | 1. Request by the buyer for the inspection. | |
| Post-Condition | Report uploaded to the database. | |
| Priority | High | |
| Course Events | Actors Action | System Response |
| | <ol style="list-style-type: none"> 1. Fills the form. 2. Submit the form. | <ol style="list-style-type: none"> 1. Stores the details in the database. 2. Shows the uploaded file on user portal. |

3.4.8 Logout

The below given in figure 3.9 is the use case of logout. After the user has done the activity or transaction. He can simply log out of his account.

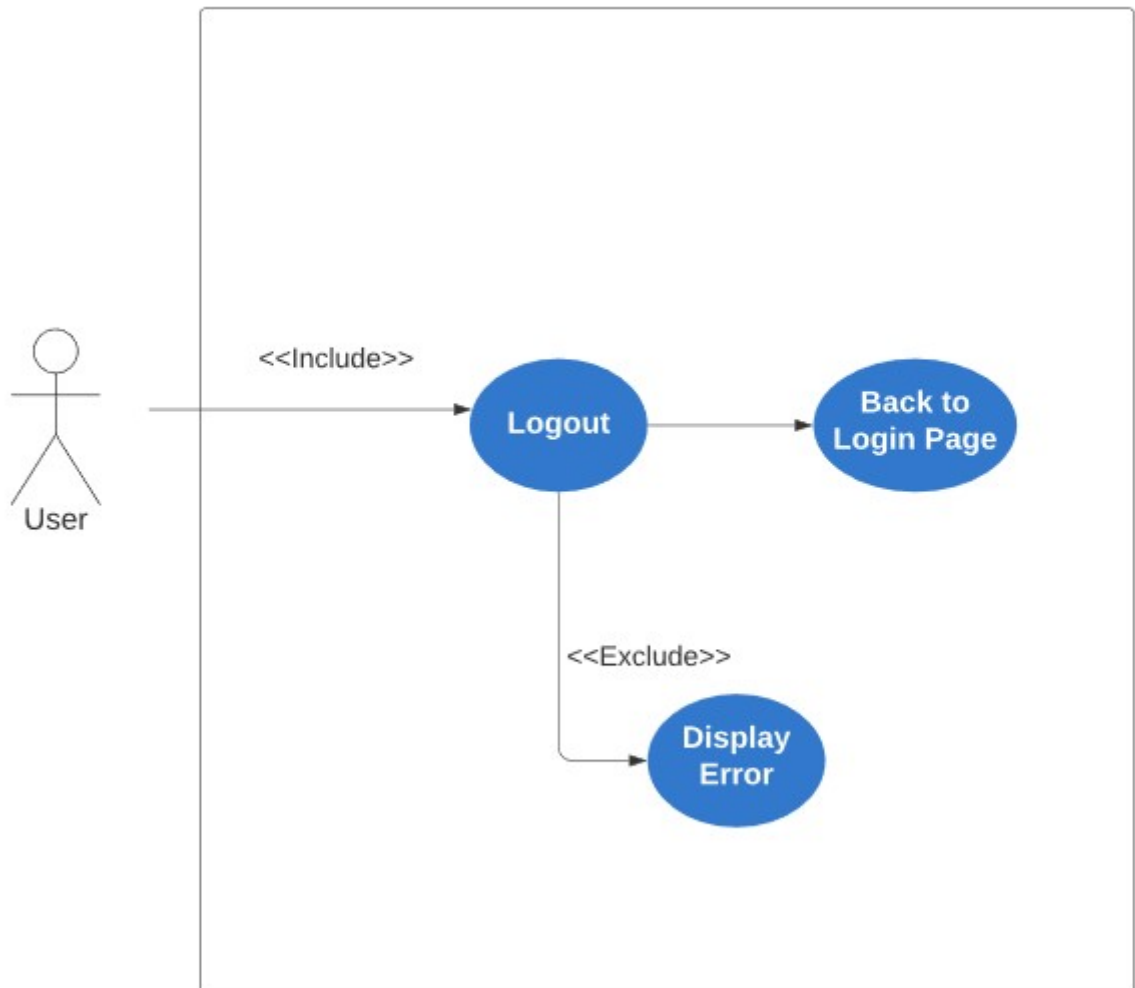


Figure 3.9: Use Case of Logout

3.4.8.1 Use Case of Logout

The below given in table 3.9 is the use case of logout. After the user has done the activity or transaction. He can simply log out of his account.

Table 3.8: Use Case of Logout

| | | |
|-----------------------|---|---|
| Used Case Name | Logout | |
| Use Case ID | SG-BU08 | |
| Description | After performing all of the tasks all of the users will simply log out of their accounts. | |
| Actors | User | |
| Basic Flow | 1. Click on log out to his account. 2. User will be logout. | |
| Alternate Flow | None. | |
| Trigger | This used case is initiated when all the work that the user wanted was done. | |
| Pre-Condition | 1. The user must have signed into his account | |
| Post-Condition | 1. Changes will be saved to the database. 2. The user will be logged out. | |
| Priority | High | |
| Course Events | Actors Action | System Response |
| | 1. Click the logout button. | 1. Leaves all action and opens the home page. |

3.5 Sequence Diagram

3.5.1 Registration

If user wants to place order, then system will ask user to create his account else system will send an error message to user. Figure 3.10 shows sequence diagram of the sign in operation.

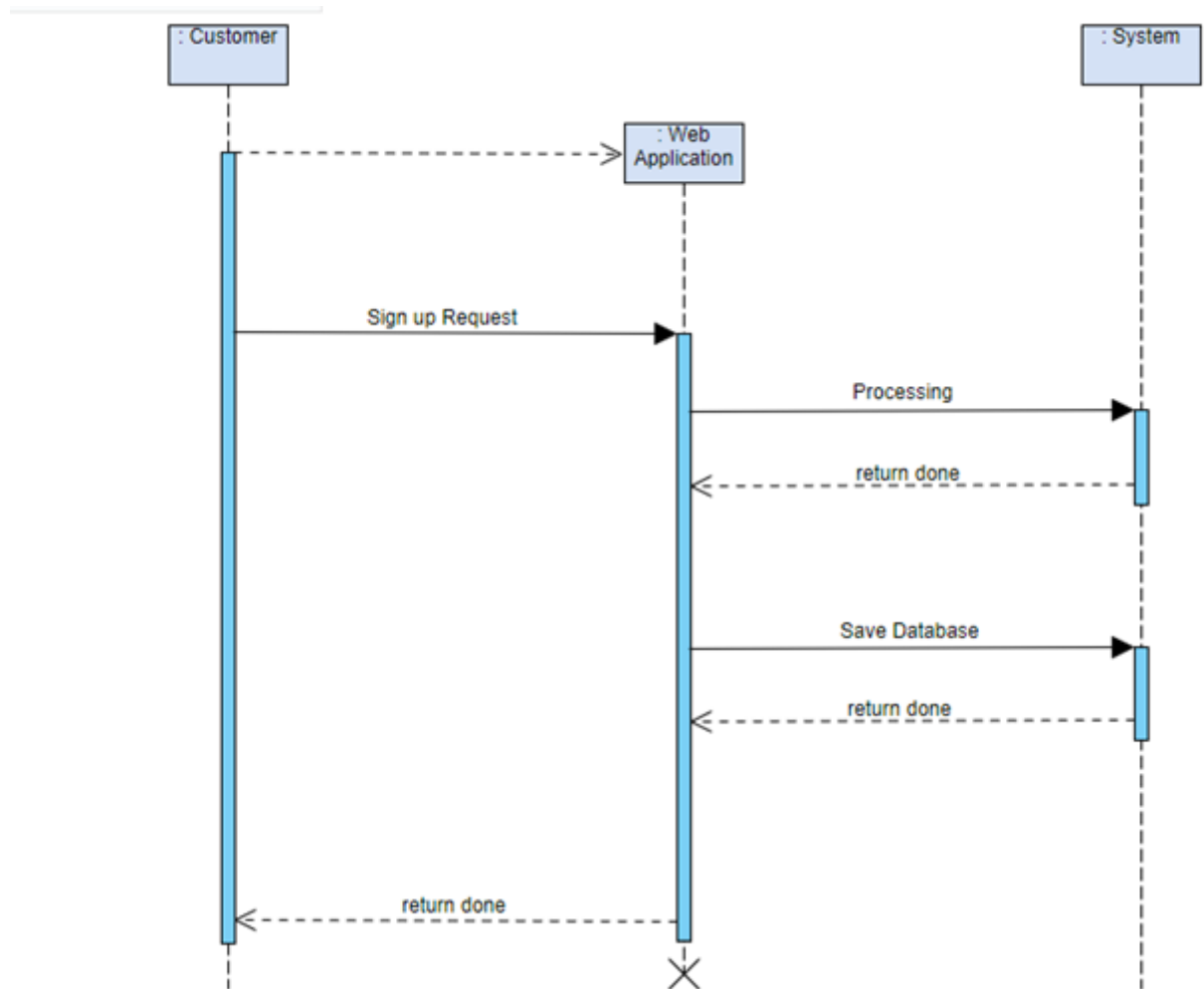


Figure 3.10: Sequence Diagram of Sign Up

3.5.2 Login

If user has entered correct username and password, then system will let user to use the website else system will send an error message to user. Figure 3.11 shows sequence diagram of the log in operation..

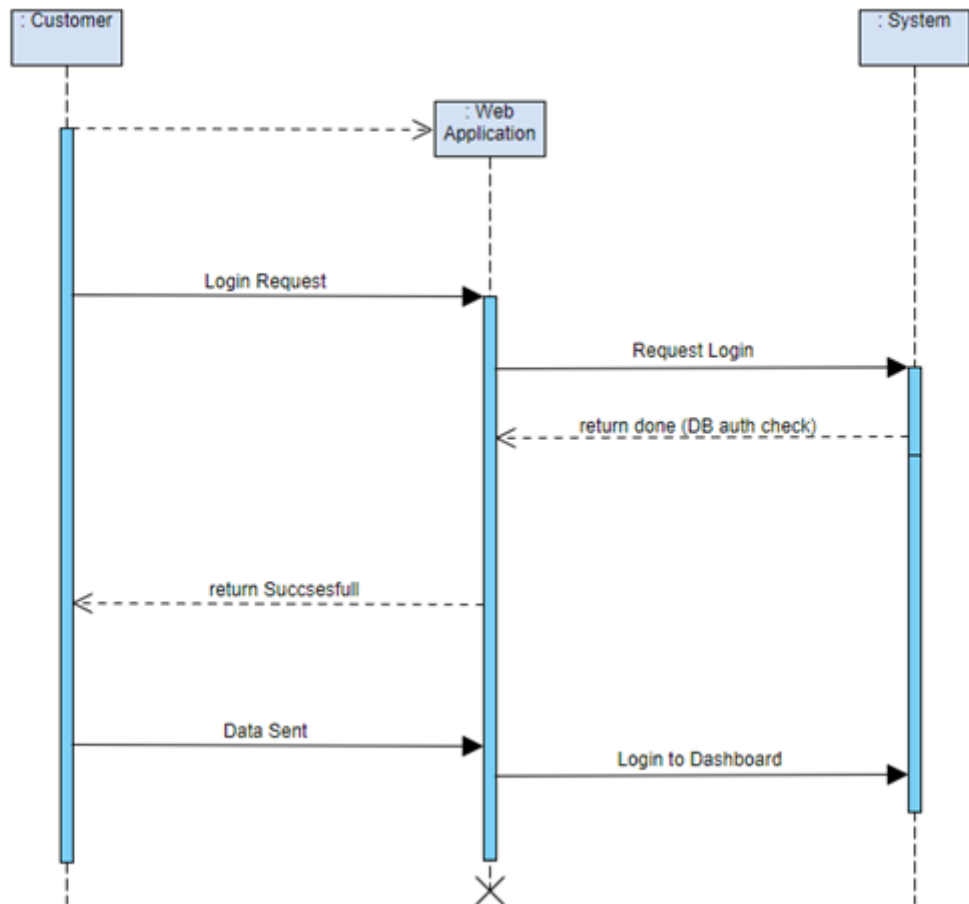


Figure 3.11: Sequence Diagram of Login

3.5.3 Place order

If user wants to place order, he can add the product to cart. Figure 3.12 shows sequence diagram of the placing order operation.

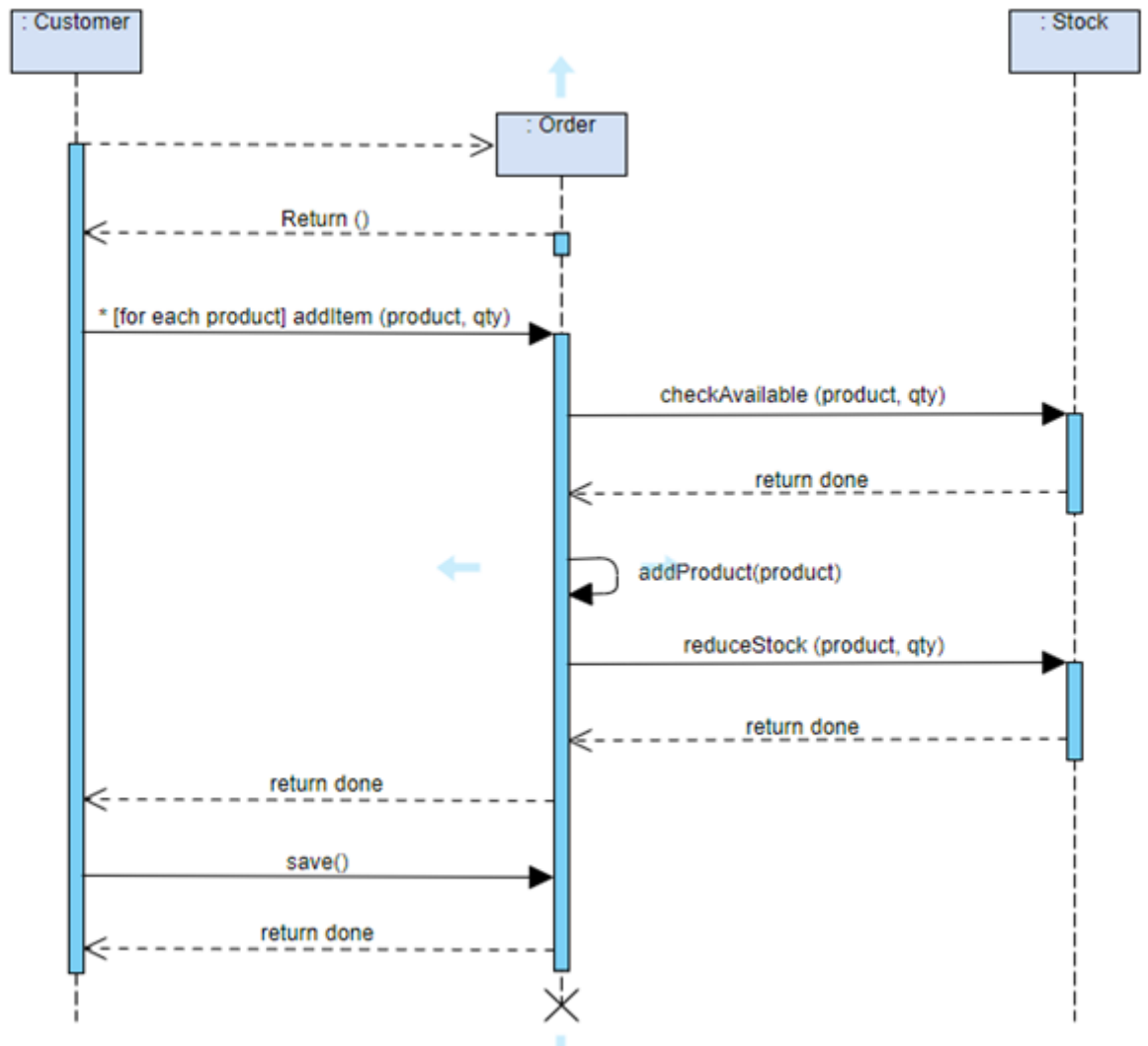


Figure 3.12: Sequence Diagram of Placing Order

3.5.4 View Account Details

If user wants to view his profile account, then the system allows him only when he has log in to his account. Figure 3.13 shows sequence diagram of the view account details operation.

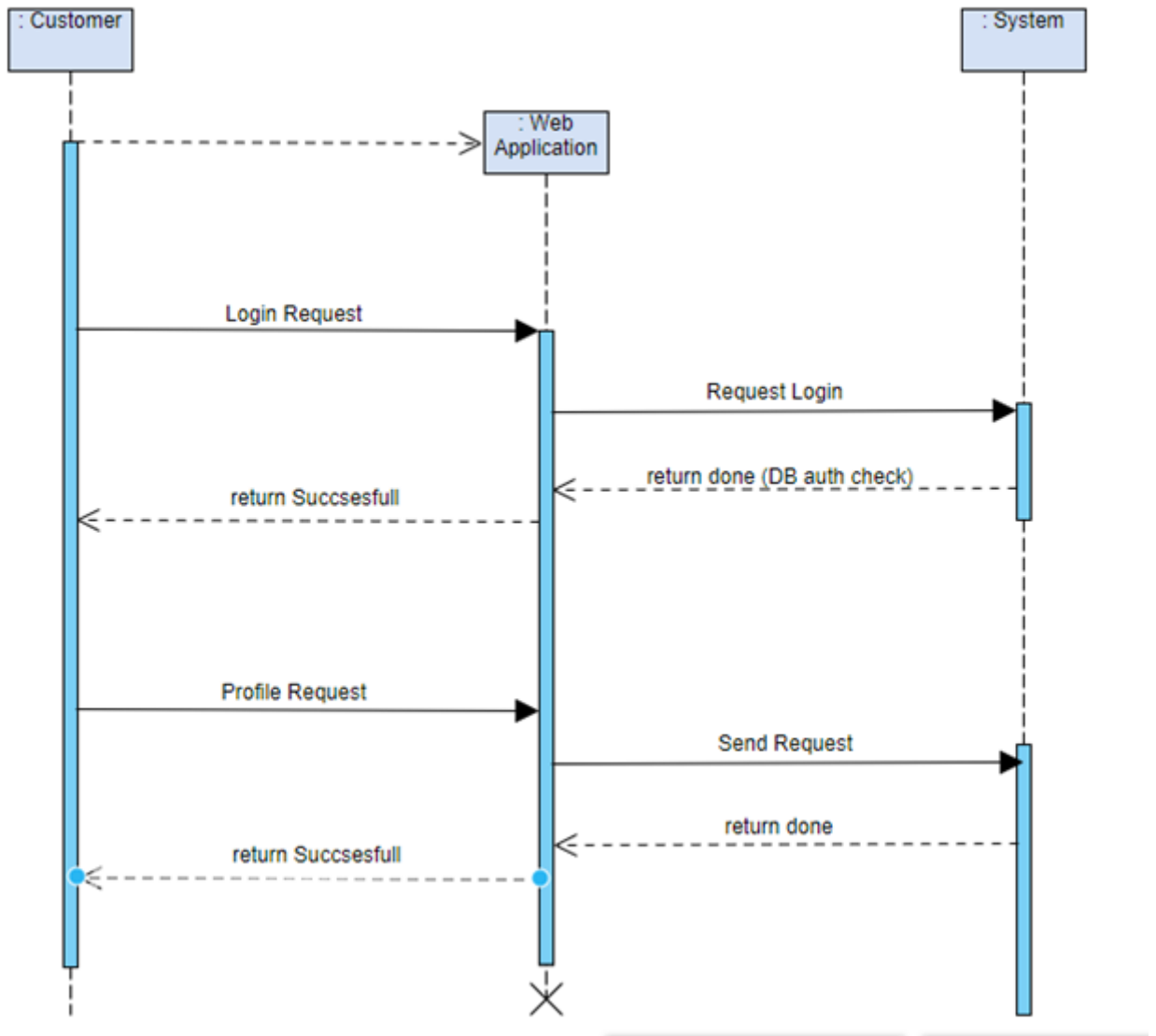


Figure 3.13: Sequence Diagram of View Account Details

3.5.5 Adding Products

If seller has entered correct username and password and has created the shop, then system will let seller to add product. Figure 3.14 shows sequence diagram of the add product operation.

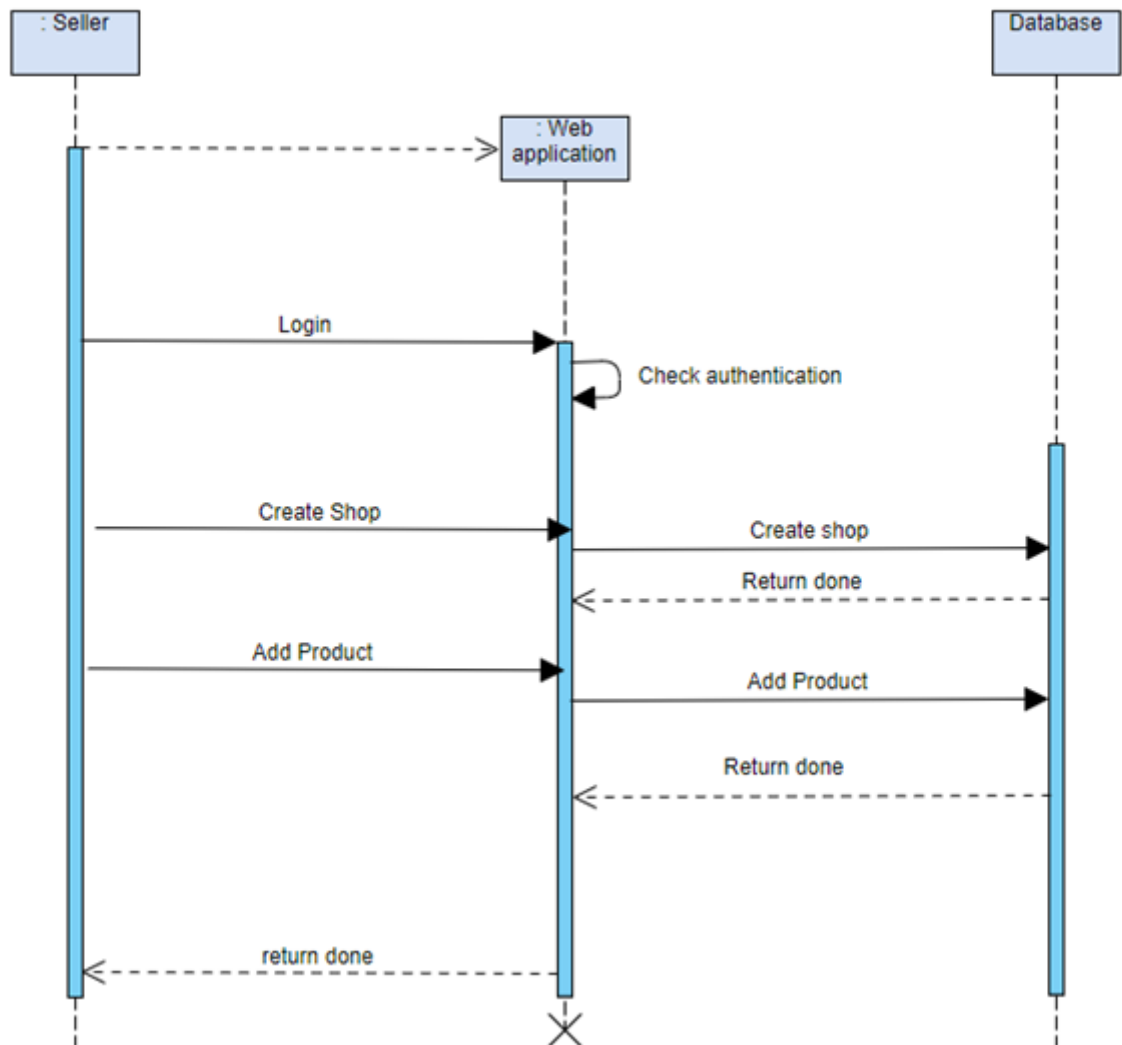


Figure 3.14: Sequence Diagram of Adding Products

3.5.6 Jewelry Inspection

Expert may get inspection request from the user. Figure 3.15 shows sequence diagram of the jewelry inspection operation.

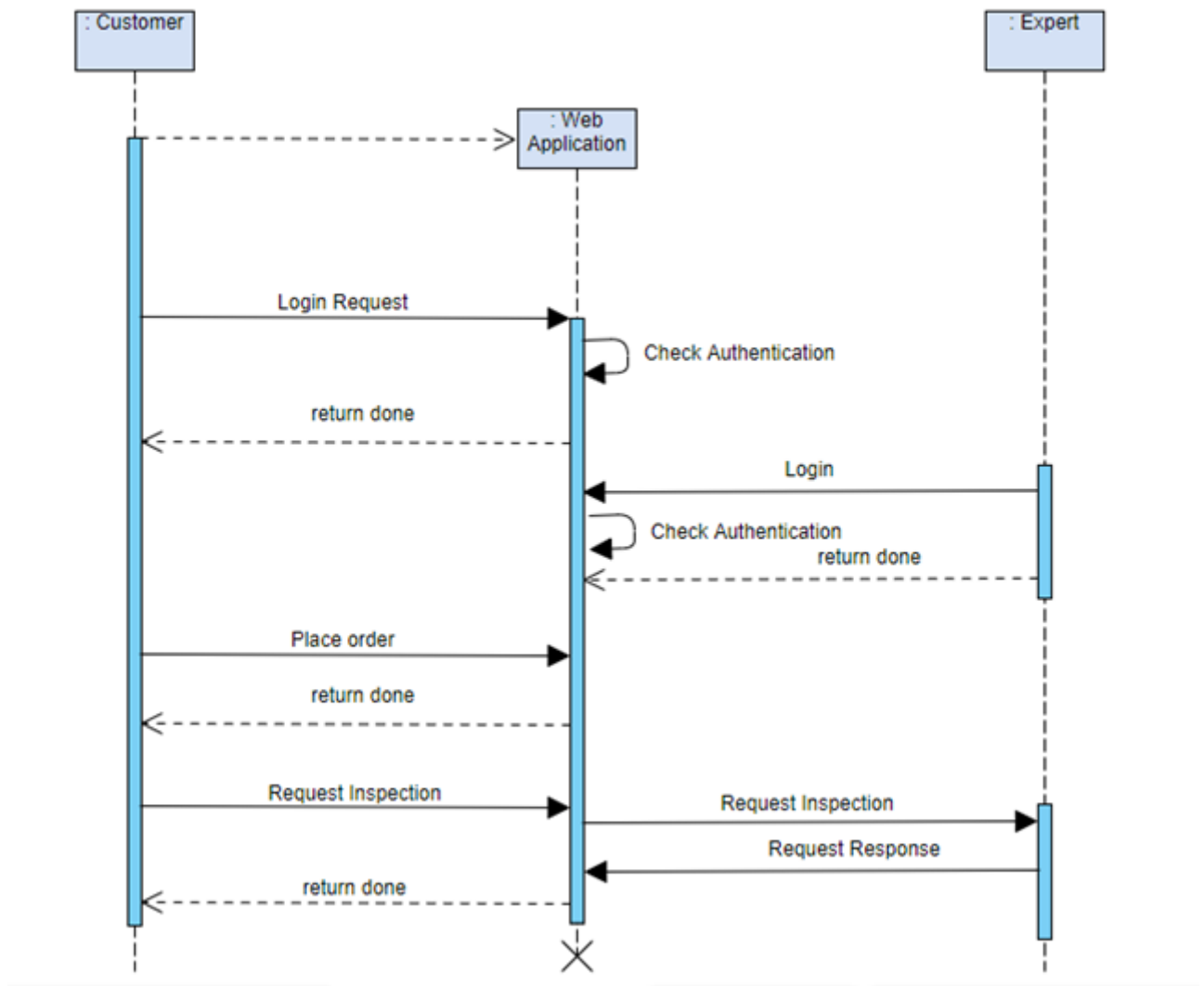


Figure 3.15: Sequence Diagram of Jewelry Inspection

3.5.7 Inspection Request

After placing an order, user can request for the jewelry inspection to the expert. Figure 3.16 shows sequence diagram of the jewelry inspection request operation.

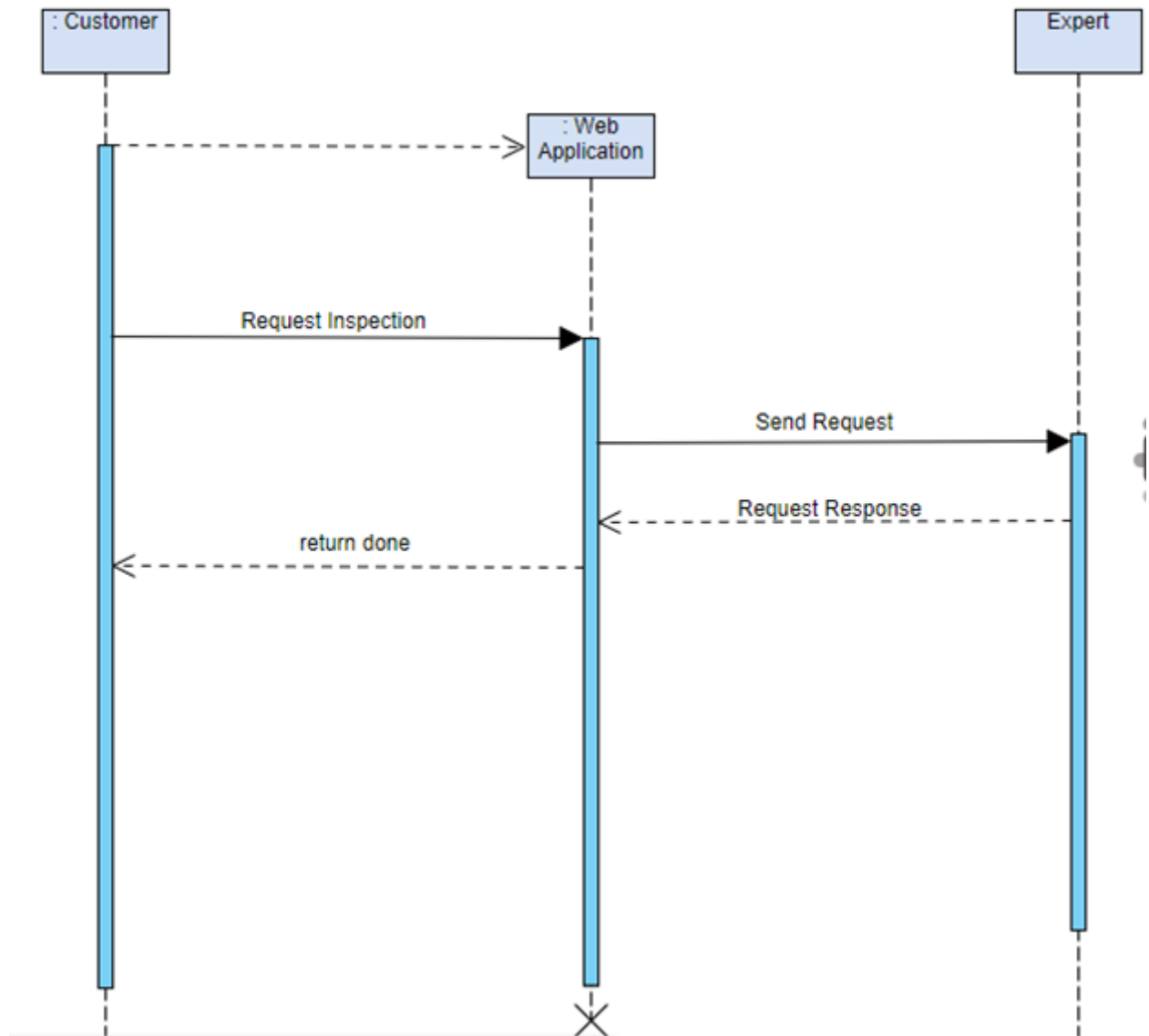


Figure 3.16: Sequence Diagram of Inspection Request

3.5.8 Logout

User can end the session by logging out. Figure 3.17 shows sequence diagram of the logout operation.

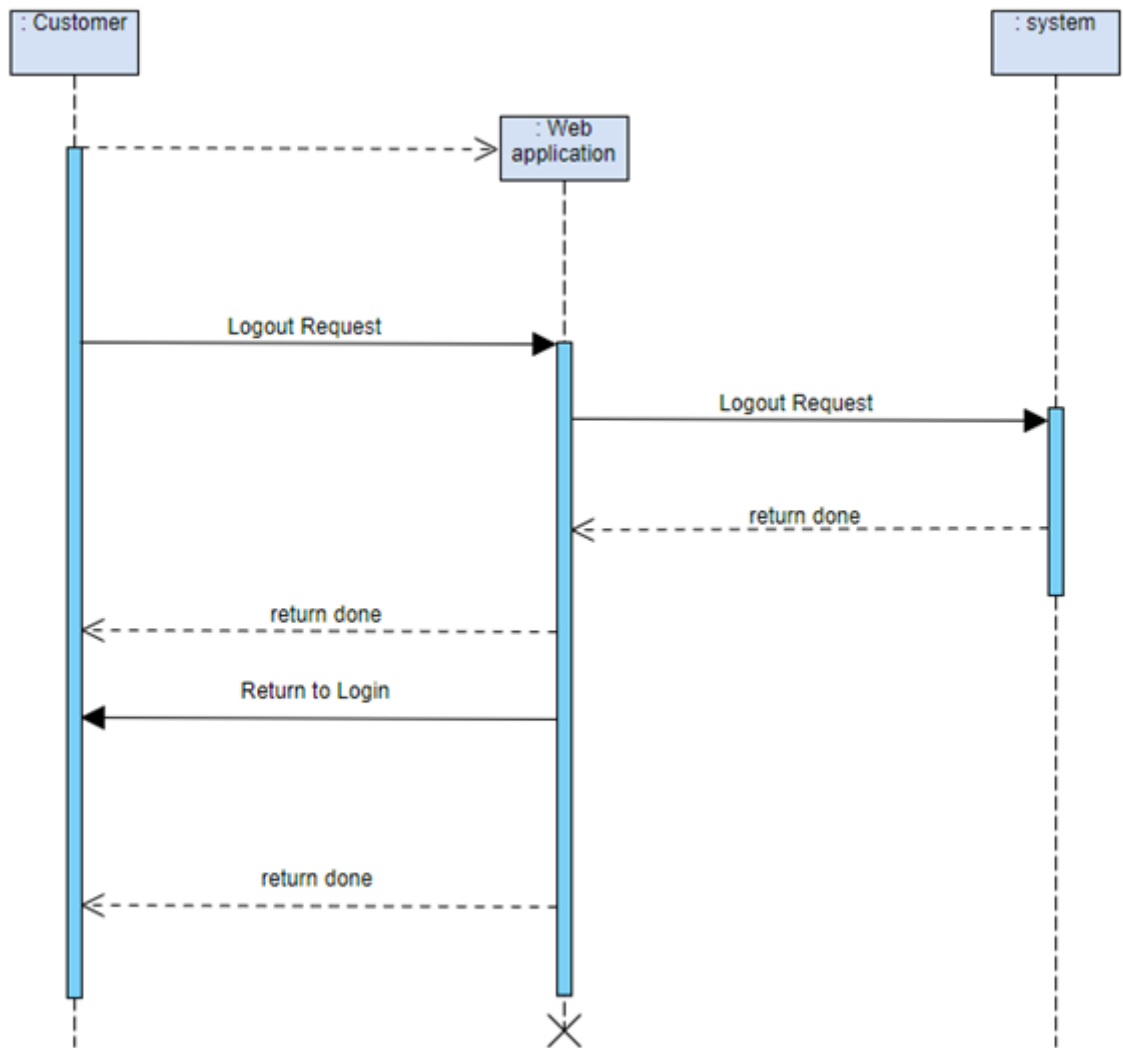


Figure 3.17: Sequence Diagram of Logout t

Chapter 4

Design

4.1 System Design

In this chapter we will elaborate the architecture and design of our system. There are three basic modules in our system are given below:

1. Seller
2. Buyer
3. Admin
4. Expert

4.2 System Architecture

In figure 4.1 is the system architecture, there are three main layers

1. Presentation layer
2. Business Logic layer
3. Database Access layer

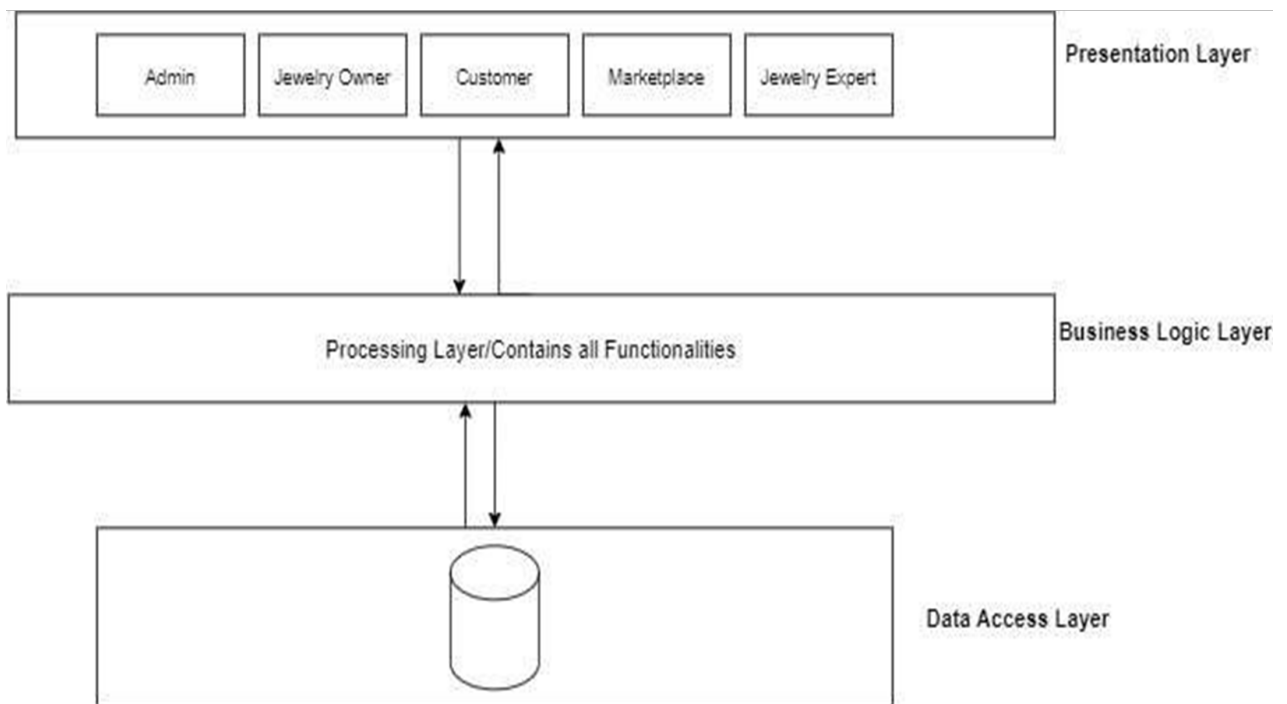


Figure 4.1: System Architecture

4.3 Design Constraints

4.3.1 Programming Language

The main programming languages used in this project for Web are React and PHP. Because programming languages are sequential, the programming standards are based on object-oriented concepts.

4.3.2 Design Standard

The software project must conform to the UML standards and must be documented at every step. An object-oriented approach should be adopted for the development of the application for re-usability and to reduce complexity. Three-tier architectures will be of

vital importance to the correct implementation of the system and therefore require further analysis.

4.3.3 Assumptions and Dependencies

1. It is independent software, and the system is robust and is available to user 24/7. The source code for this software is not shared, and the database is updated with the last information.
2. The Internet should always be accessible and should be able to respond to many requests.
3. This software is not used by an unauthorized user.
4. Only registered customers, handlers, can be used, and the user authentication process is used to protect other data from non-ethical people.
5. Database is up-to-dated.

4.3.4 Design Methodology

Agile software development model is being used in this system. As our system is divided into two modules, each module consists of different units which are needed to be incrementally developed. Agile model consists of these steps:

1. Planning
2. Requirement Analysis
3. Design
4. Development
5. Testing
6. Evaluation

4.3.5 External Interfaces

As we are making a desktop web application on windows. Our application uses GUI based external interface. It does not require any command-line interface. But we also want that our application to handle server requests in a more efficient manner. So, the external interface requirements will be

1. Windows Server/10

2. MS SQL Server

3. IIS Web Server

4.4 High Level Design

The web app includes multiple modules that are merged together into a single web application. This module consists of buyer (client), seller, expert and admin. All of these modules are connected in a holistic manner. Front end consists of bootstrap, CSS, tailwind CSS and react JS on the contrary the back-end consists of laravel and node JS while next JS is used for caching and MySQL is used as database. All of these in interconnected with an API that is developed on laravel as shown in figure 4.2.

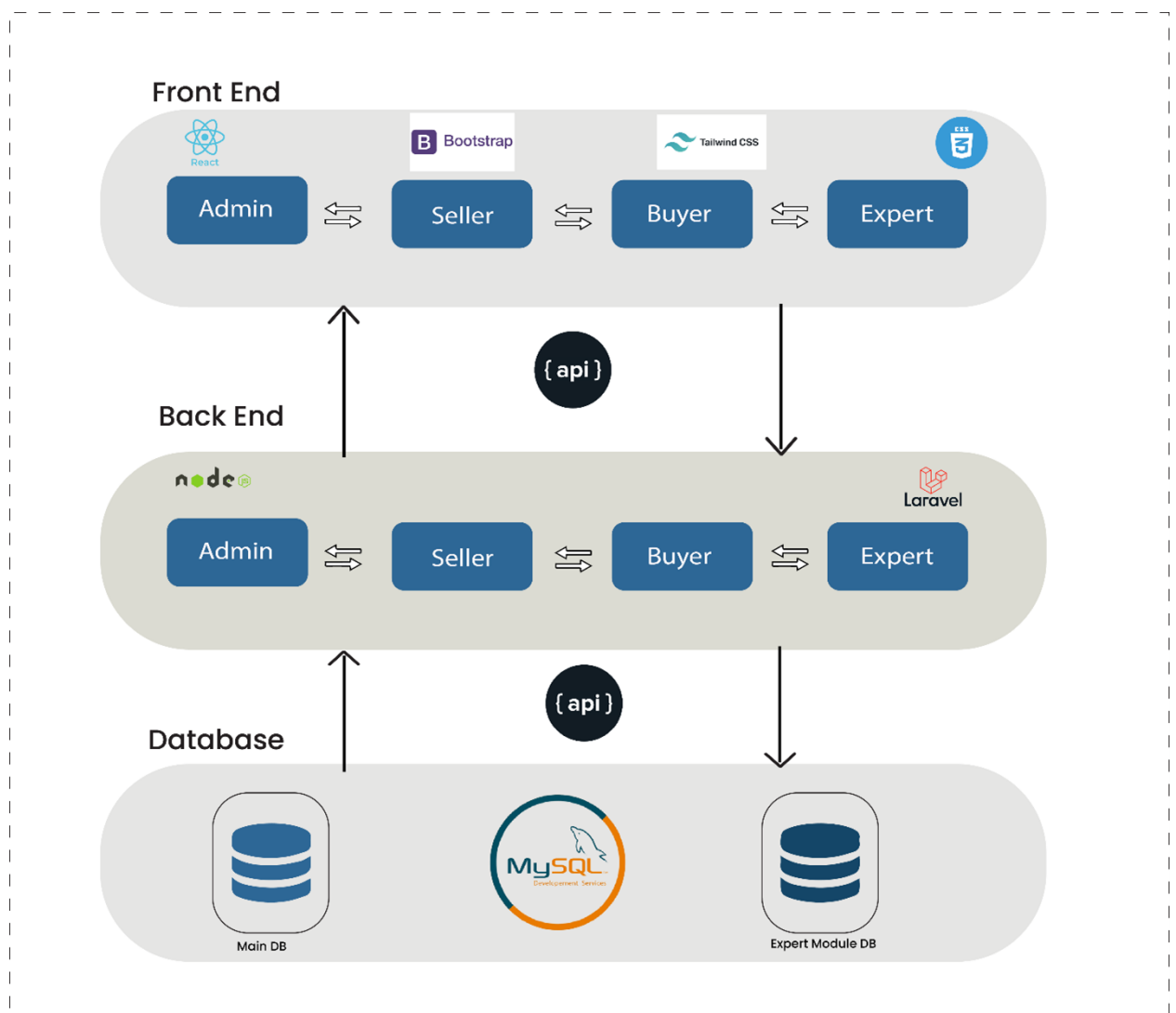


Figure 4.2: High Level

4.5 Low Level Design

Given below in the figure 4.3 is the low level design of Sparkles Gallery.

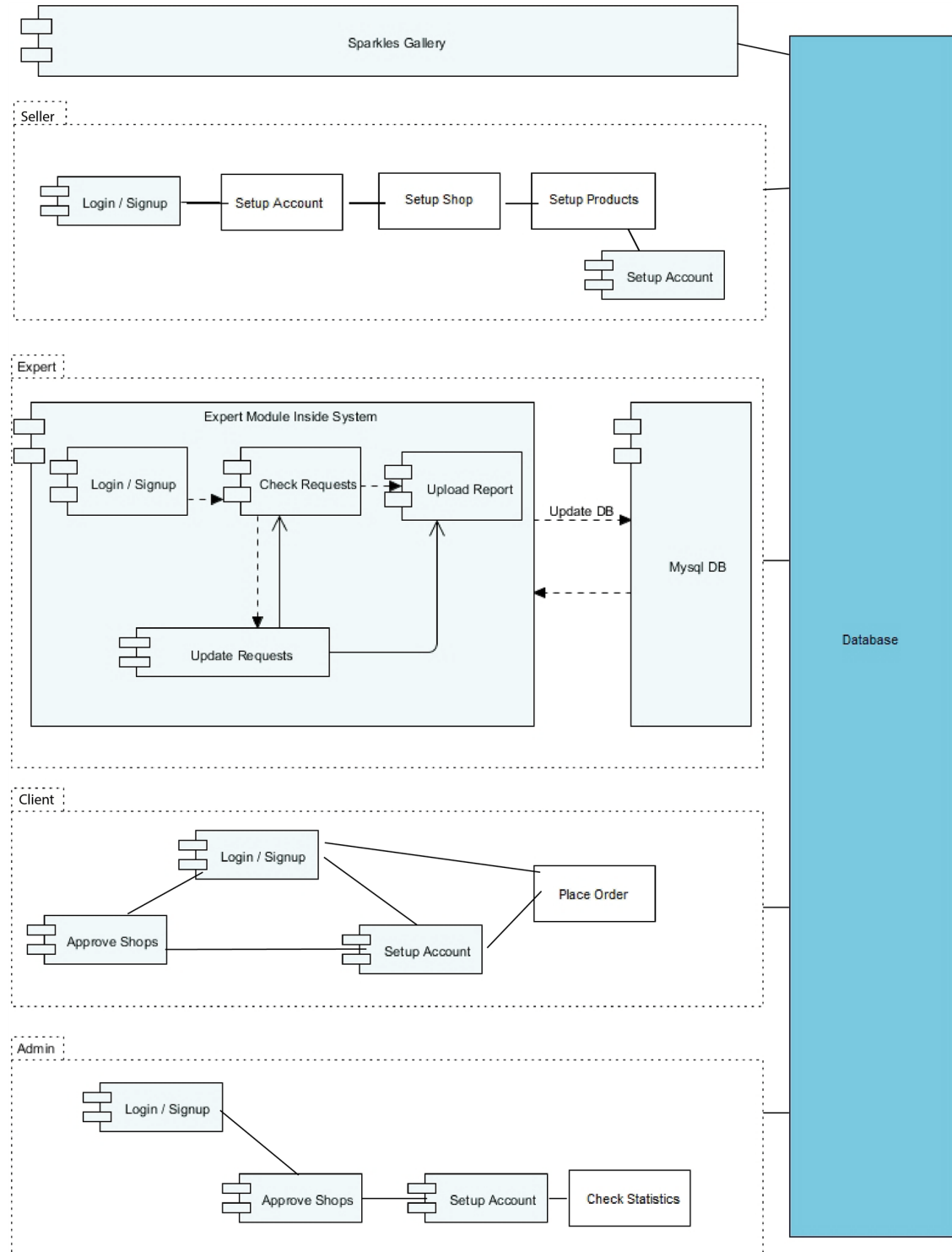


Figure 4.3: Low Level

4.6 Entity Relation Design

In the figure 4.4 given below is the Entity Relation Diagram of Sparkles Gallery

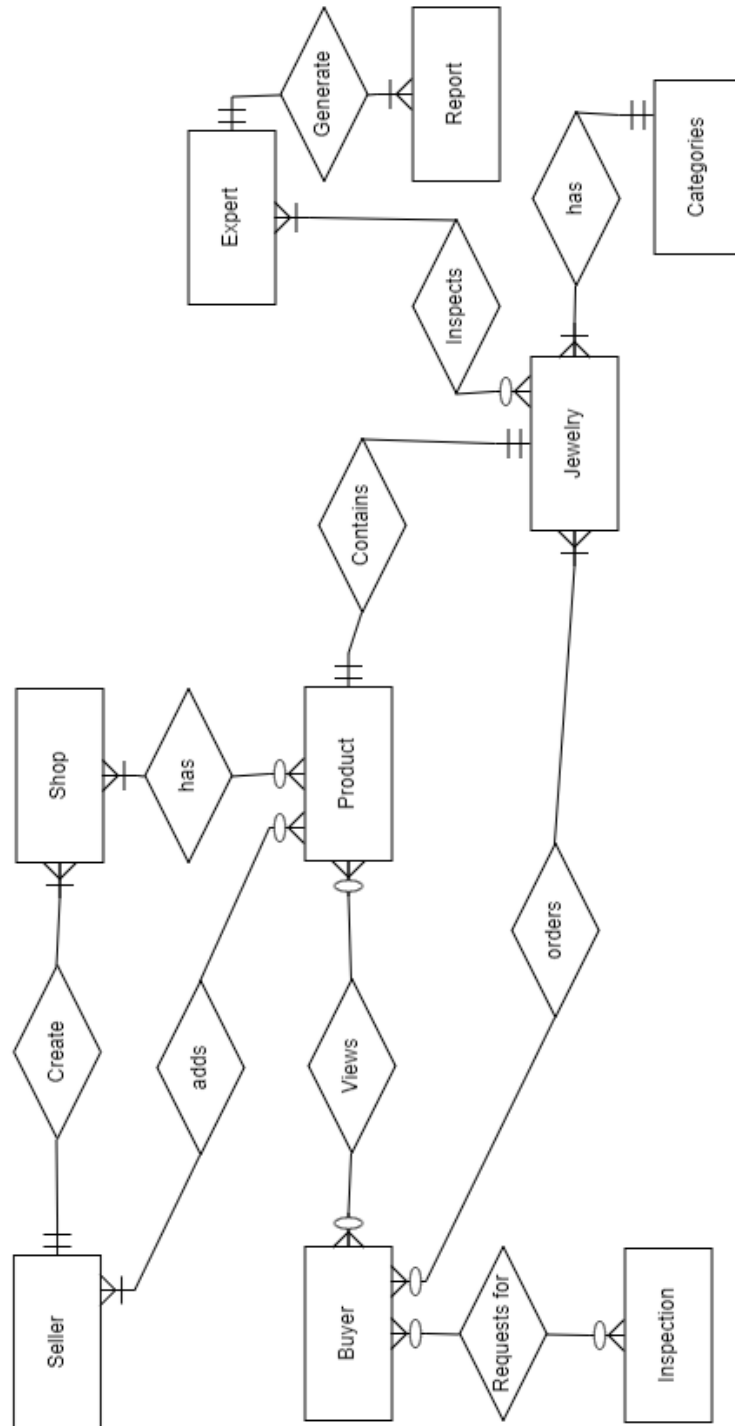


Figure 4.4: Er Diagram

4.6.1 Attributes

In the table 4.1 given below is the details of entities of the ER diagram and their respective attributes

Table 4.1: Attributes of ER Diagram

| User | Category | Order_ Product | Category_ product | Address | Orders |
|--|--|--|---------------------------------|--|--|
| ID Name Email Email_verified_at Password Remember-Token Created_at Updated_at Is_active Shop_id | ID Name Created_at Updated_at Type_id Details | Id Order_id Product_id Subtotal Created_at Updated_at | Id Product_Id Category_Id | Id Title Type Customer_id Created_at Updated_at | Id Tracking_number Customer_id Status Total Shop_id Created_at Updated_at Deleted_at |

4.7 Domain Constraint

4.7.1 User

In the table 4.2 shown below is the details of attributes of the entity User.

Table 4.2: User constraint

| Domain | Datatype | Size | Is Null | In unique | Default value | Range |
|--------------------------|-----------|------|---------|-----------|---------------|-------|
| Name | Varchar | 191 | No | No | – | – |
| ID | Int | 20 | No | Yes | 1 | >1 |
| Email_Verified_at | Timestamp | – | Yes | | | |
| Password | Varchar | 200 | Yes | | | |
| Remember-Token | Varchar | 100 | Yes | | | |
| Email Address | Email | 100 | No | No | – | – |
| Created_At | Timestamp | 500 | Yes | No | – | – |
| Updated_at | Timestamp | 13 | Yes | No | – | – |
| Is_active | Int | 50 | No | No | — | |

4.7.2 Category

In the table 4.3 shown below is the details of attributes of the entity Category constraint.

Table 4.3: Category Constraint

| Domain | Datatype | Size | Is Null | In unique | Default value | Range |
|-------------------|-----------|------|---------|-----------|---------------|-------|
| Name | Varchar | 191 | No | No | – | >1 |
| User ID | Int | 20 | No | Yes | 1 | >1 |
| Type ID | Int | 20 | No | No | 1 | >1 |
| Created_At | Timestamp | 50 | Yes | Yes | – | – |
| Updated_at | Timestamp | | Yes | | | |
| Details | Text | | Yes | | | |

4.7.3 Order Product

In the table 4.4 shown below is the details of attributes of the entity Order product.

Table 4.4: Order Product Constraint

| Domain | Datatype | Size | Is Null | In unique | Default value | Range |
|-------------------|-----------|------|---------|-----------|---------------|-------|
| ID | Int | 20 | No | Yes | 1 | >1 |
| Order ID | Int | 20 | No | No | | >1 |
| Product_Id | Varchar | 10 | No | No | | – |
| Created_At | Timestamp | | Yes | | | |
| Updated_at | Timestamp | 10 | Yes | No | | – |
| Subtotal | | | No | | | |

4.7.4 Category Product

In the table 4.5 shown below is the details of attributes of the entity Category product.

Table 4.5: Category Product Constraint

| Domain | Datatype | Size | Is Null | In unique | Default value | Range |
|--------------------|----------|------|---------|-----------|---------------|-------|
| ID | Int | 20 | No | Yes | 1 | >1 |
| Category ID | Int | 20 | No | No | 1 | >1 |
| Product_Id | Int | 20 | No | No | – | – |

4.7.5 Order

In the table 4.6 shown below is the details of attributes of the entity Order product.

Table 4.6: Order Constraint

| Domain | Datatype | Size | Is Null | In unique | Default value | Range |
|------------------------|-----------|------|---------|-----------|---------------|-------|
| ID | Int | 20 | No | Yes | 1 | >1 |
| Tracking_Number | Varchar | 200 | No | Yes | 1 | >1 |
| Customer_id | Int | 20 | Yes | No | 1 | >1 |
| Total | Double | 50 | Yes | No | 1 | - |
| Shop Id | Int | 20 | No | Yes | 1 | >1 |
| Updated_at | timestamp | | Yes | | | |
| Deleted_at | timestamp | | Yes | | | |

4.7.6 Address

In the table 4.7 shown below is the details of attributes of the entity Address.

Table 4.7: Address Constraint

| Domain | Datatype | Size | Is Null | In unique | Default value | Range |
|--------------------|-----------|------|---------|-----------|---------------|-------|
| ID | Int | 50 | No | Yes | 1 | >1 |
| Title | Varchar | 200 | No | Yes | 1 | >1 |
| Type | Varchar | 200 | No | No | | - |
| Customer Id | timestamp | | Yes | | | |
| Updated_at | timestamp | | Yes | | | |
| Created_at | timestamp | | Yes | | | |

4.8 Activity Diagram

4.8.1 Login

Shown below in the figure 4.5 is the activity diagram of login. Basically, any user that is registered can login to their account in order to use the services. The user enters the username and password, if the username is wrong, an error message displays, and the user will be asked to enter the credentials again. If the credentials are correct, then the user logs in successfully into his system.



Figure 4.5: Activity Diagram Login

4.8.2 Place Order

Shown below in the figure 4.6 is the activity diagram of place order. The buyer will simply select the vendor from whom he wants to purchase, he will select the product that he wants to purchase, and he will check whether the product is available or not. If the product is available, he will simply confirm the order, if it's not available he will choose whether he wants to continue or not. If he wants to continue, he can start again by choosing the vendor or if he does not want to continue, the order will be dropped.

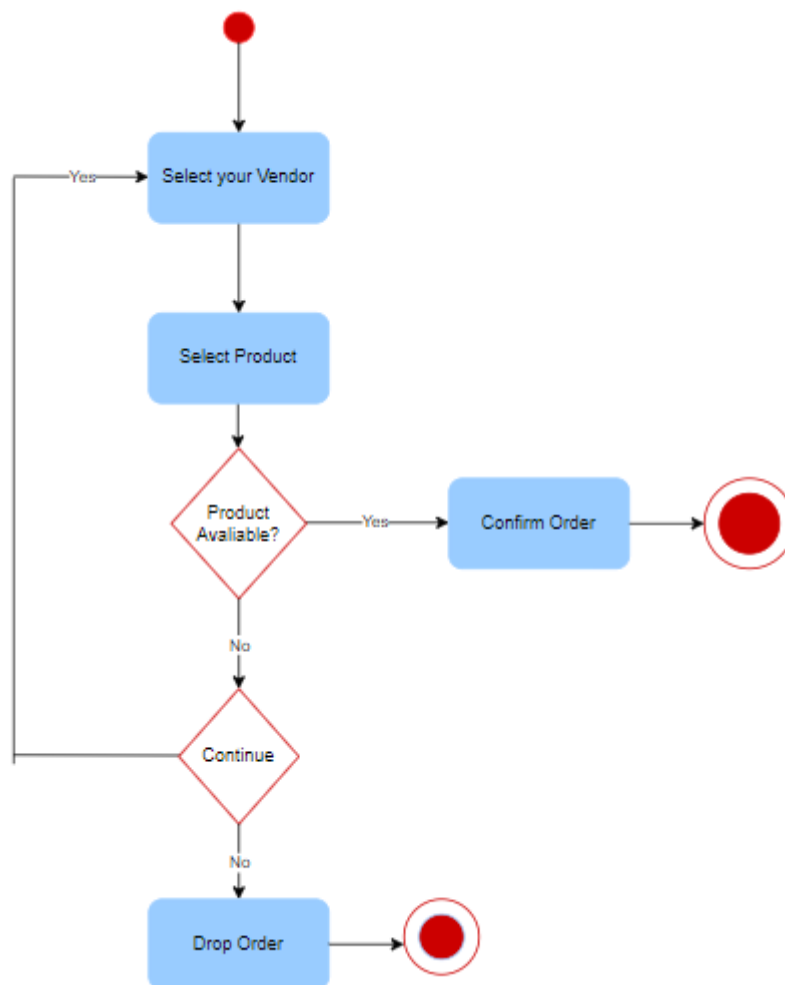


Figure 4.6: Activity Diagram of Placing Order

4.8.3 Search

Shown below in the figure 4.7 is the activity diagram of search. The buyer will search for any jewelry, if it is the valid keyword then result will be displayed on screen. Otherwise, it will show a message that no record found.

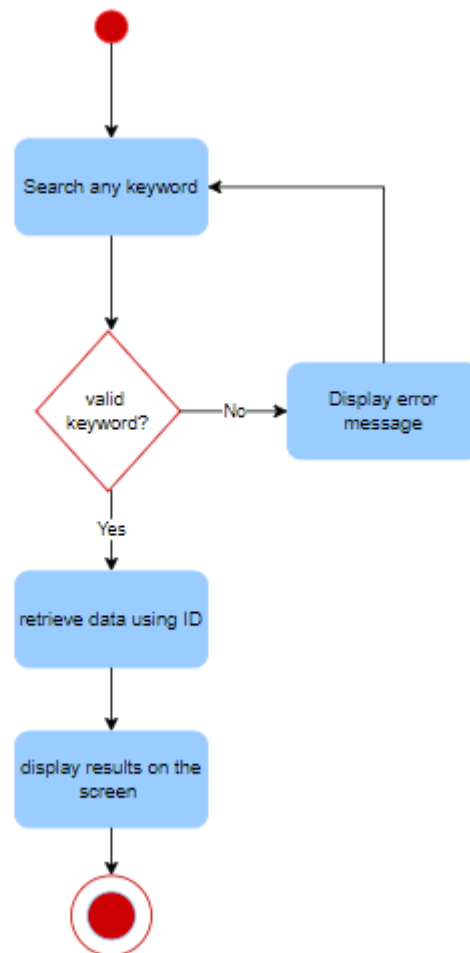


Figure 4.7: Activity Diagram of Search

4.8.4 Inspection

Given below in the figure 4.8 is the activity diagram shows that how can user request for inspection. After login, user will select a product for inspection. Then the expert will do inspection and generate the report.

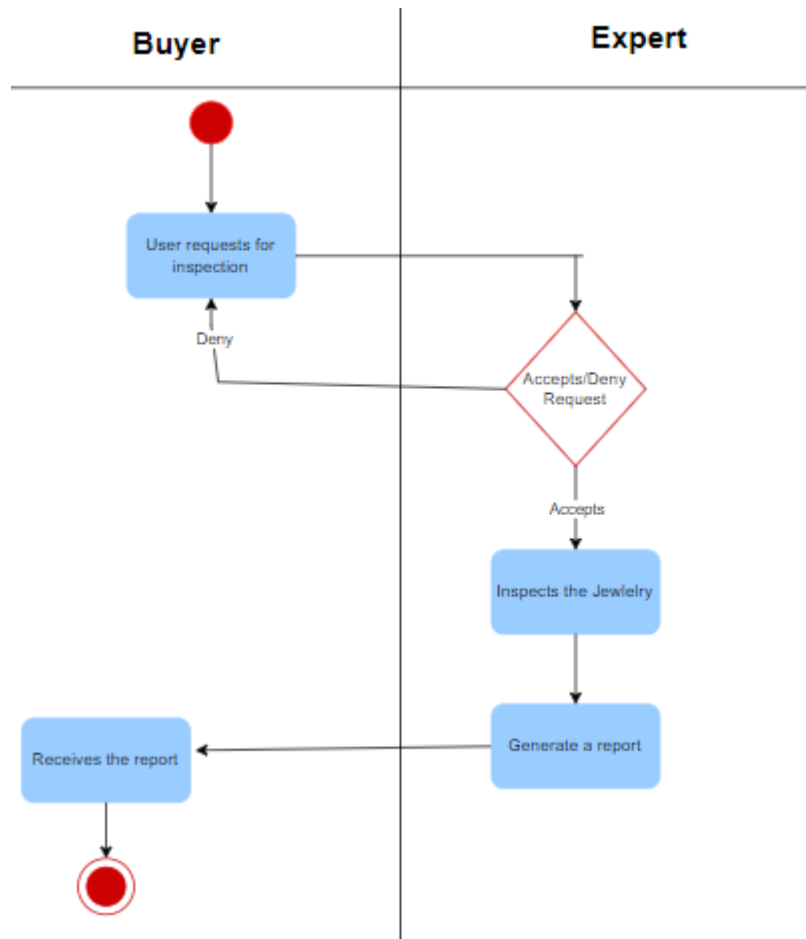


Figure 4.8: Activity Diagram of Inspection

4.8.5 Add Product

The figure 4.9 shown below is the diagram of adding product. The seller will log into his account. He will create a shop. If the shop is active, he will add a product in it if it is not, the admin will change the state of the shop to active, after the state is changed the product can be added into the shop.

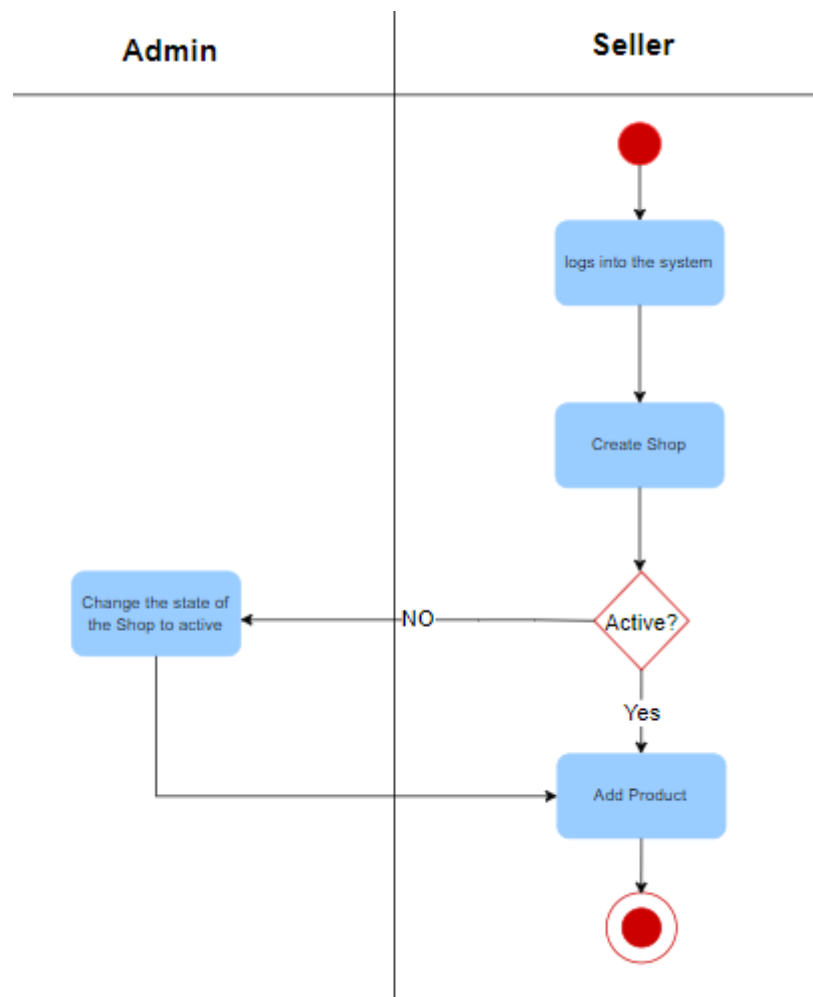


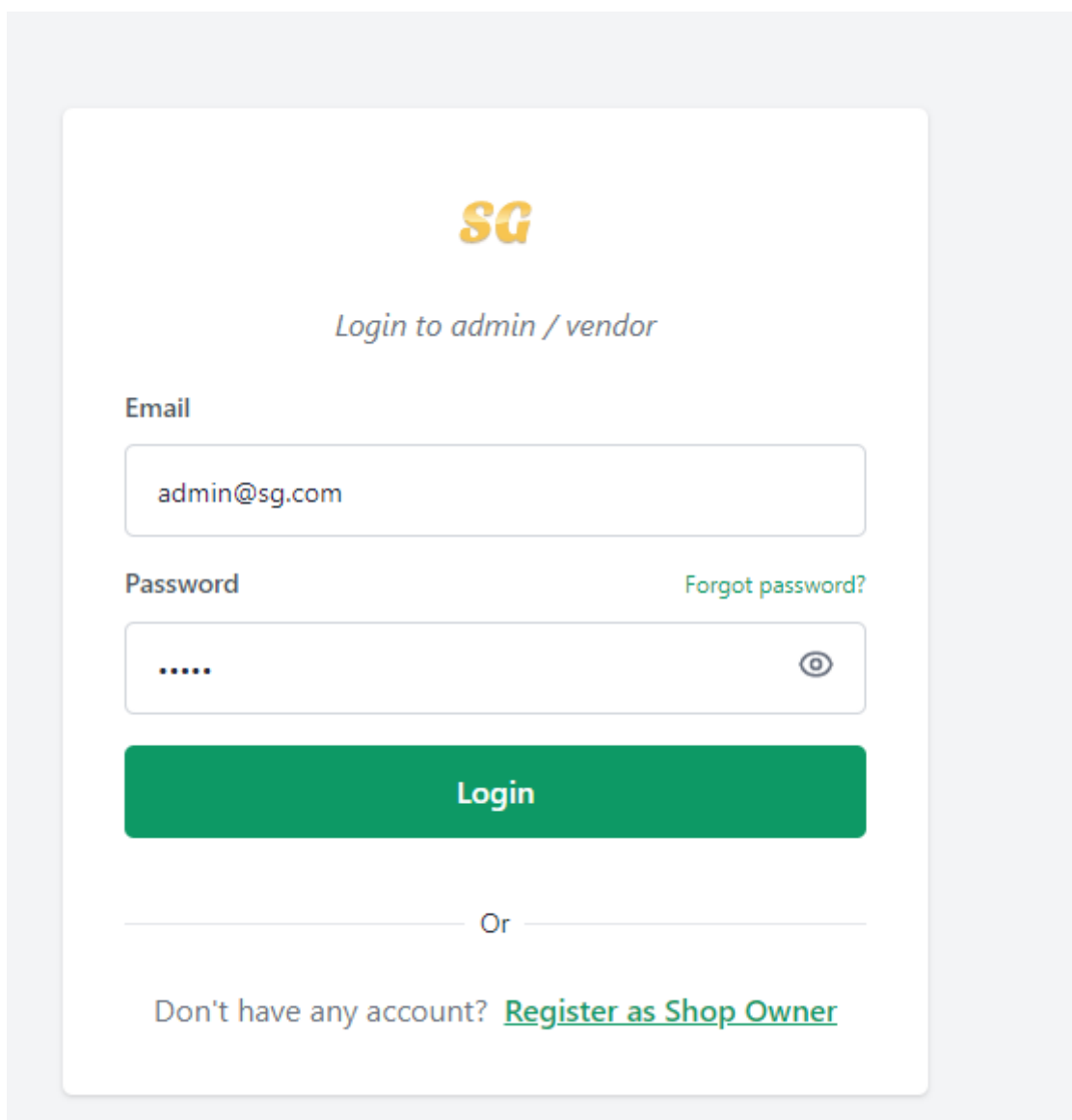
Figure 4.9: Activity Diagram of Adding Product

4.9 GUI Design

The user interface of a system is created to allow users to interact with the system. The user interface of a system should be kept simple for the user to understand and learn it rapidly. Our system is designed to produce results that are both effective and efficient. The following figures are screenshots of the proposed system:

4.9.1 Admin Module

4.9.1.1 Login In



The screenshot displays a login form for the Admin Module. At the top center is the 'SG' logo in orange. Below it, the text 'Login to admin / vendor' is centered. The form includes an 'Email' field with the placeholder 'admin@sg.com', a 'Password' field with masked characters and a visibility toggle icon, and a 'Forgot password?' link. A prominent green 'Login' button is positioned below the password field. At the bottom, there is an 'Or' separator and a link to 'Register as Shop Owner'.

Figure 4.10: GUI of Login of Admin Module

4.9.1.2 Dashboard

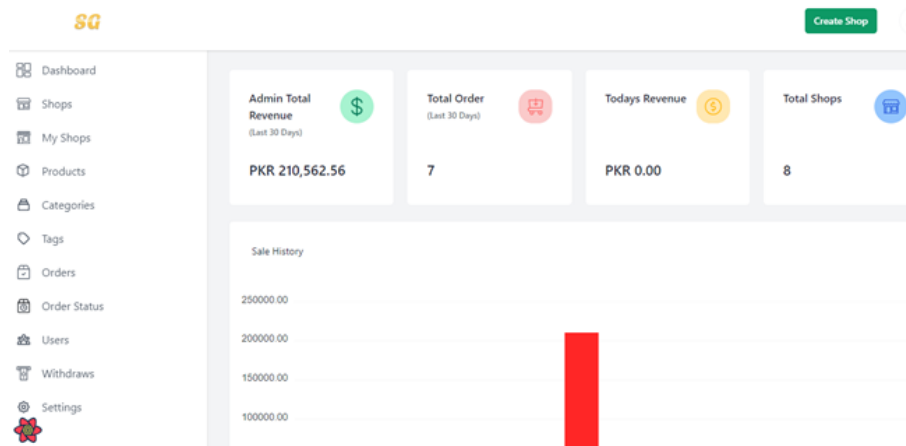


Figure 4.11: GUI of Dashboard of Admin Module

4.9.1.3 Shops

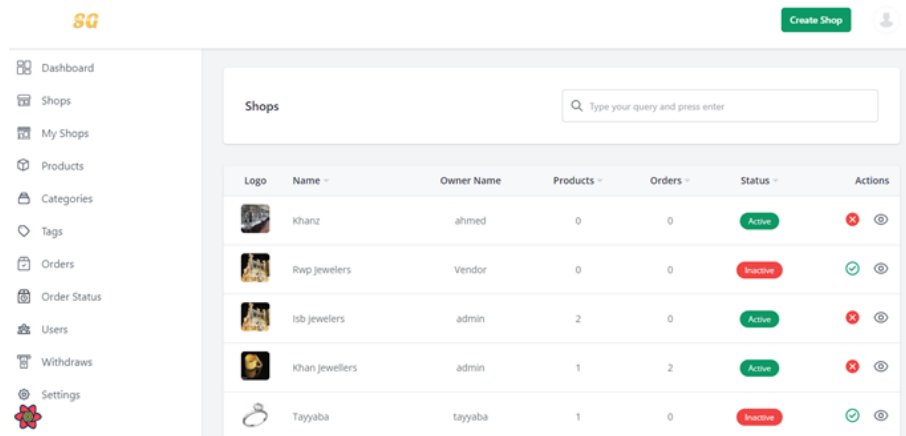


Figure 4.12: GUI of Shops of Admin Module

4.9.1.4 Dashboard of specific shop

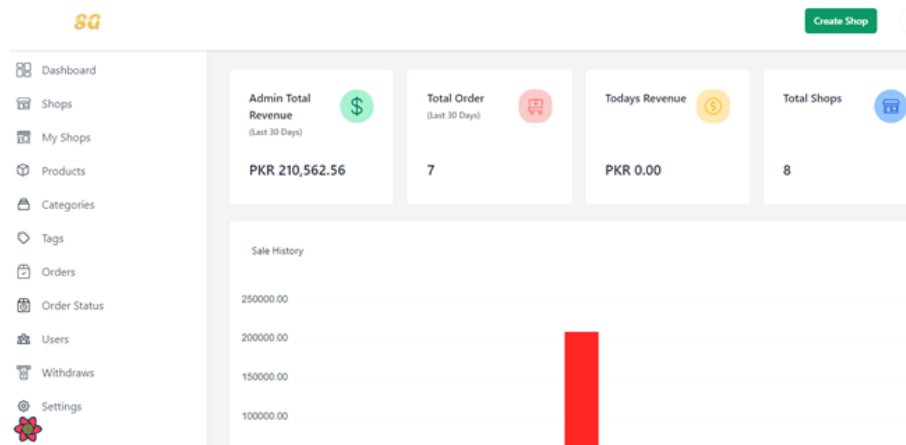


Figure 4.13: GUI of Dashboard of Specific Shop of Admin Module

4.9.1.5 Product

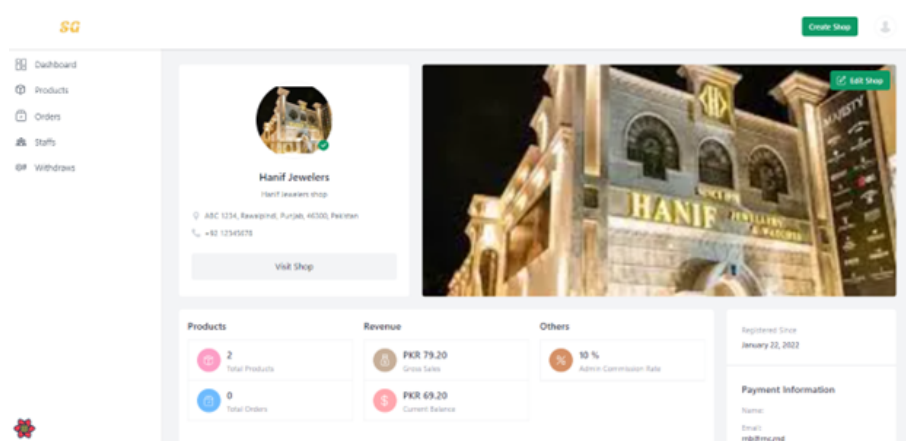


Figure 4.14: GUI of of Admin Module

4.9.1.6 Categories

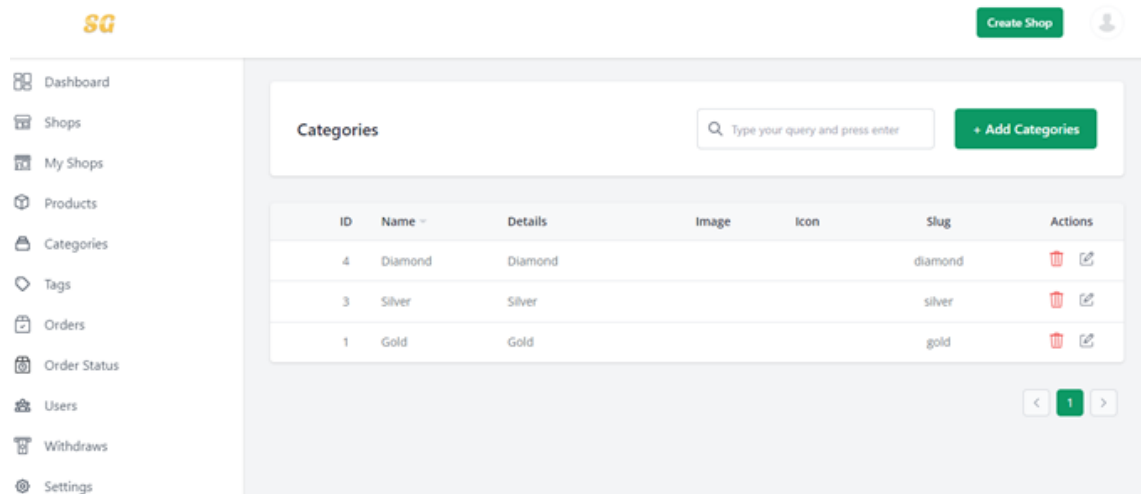


Figure 4.15: GUI of Categories of Admin Module

4.9.2 User Module

4.9.2.1 Home

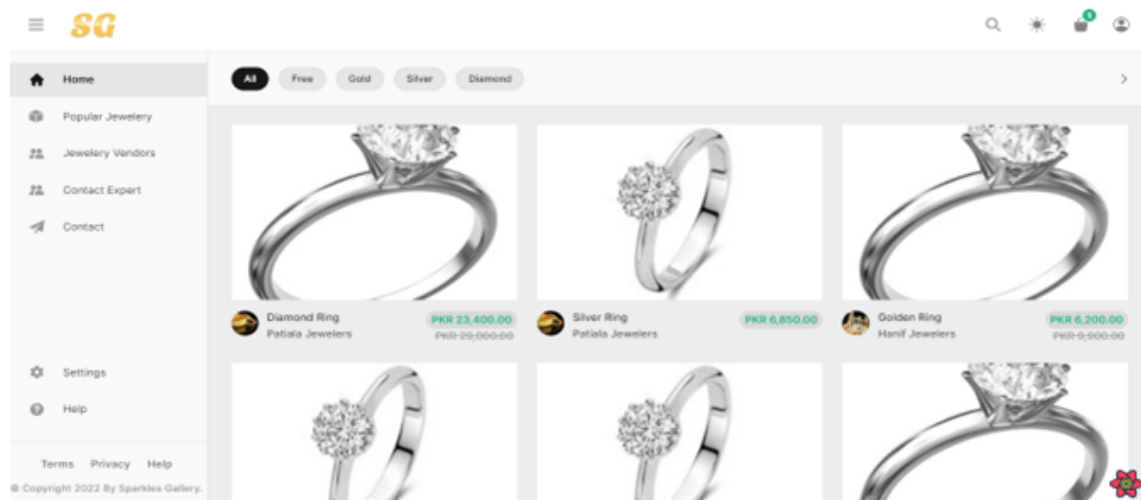
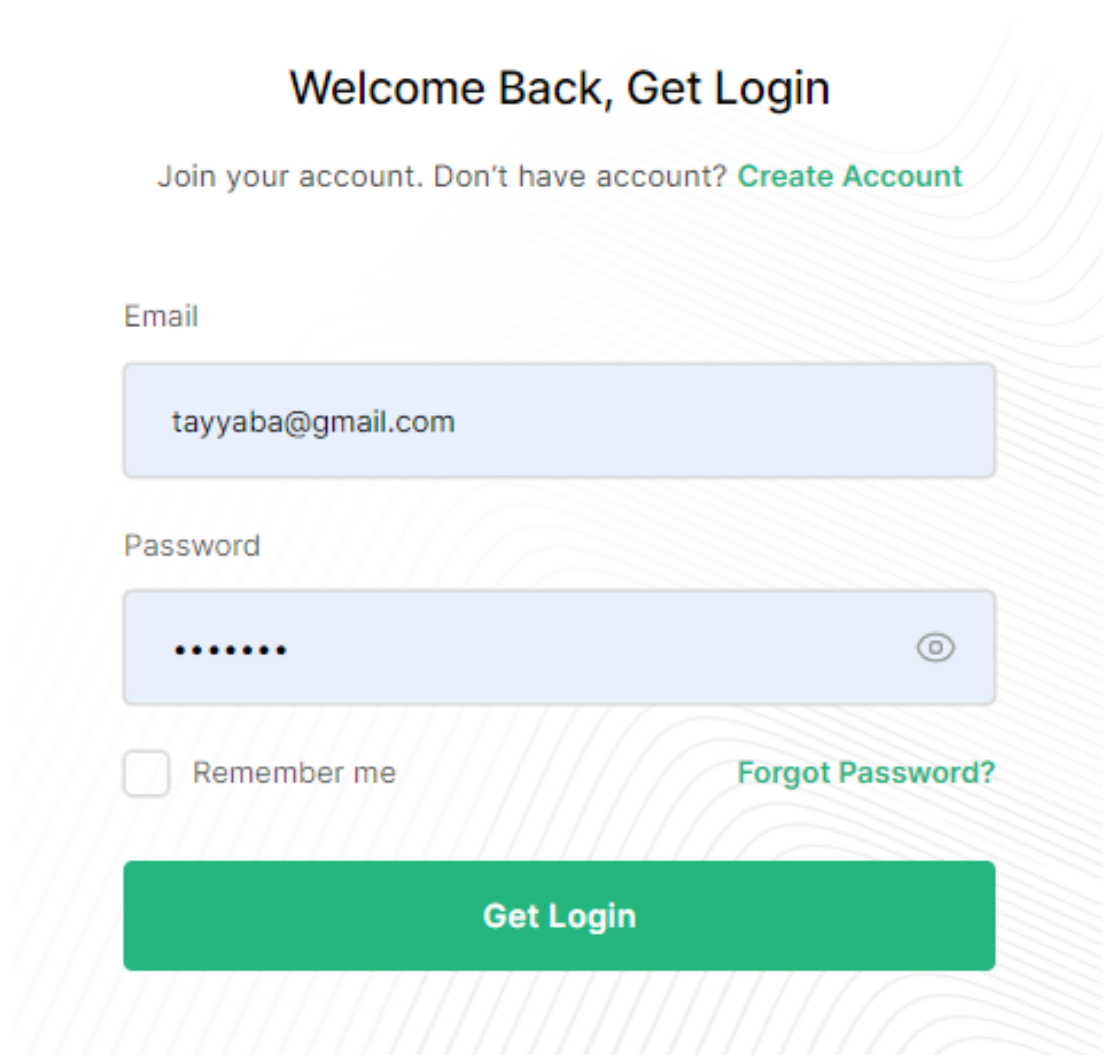


Figure 4.16: GUI of Home of User Module

4.9.2.2 Login



The image shows a login form with a light green background and a white wavy pattern. At the top, the text "Welcome Back, Get Login" is displayed in a bold, black font. Below this, a line of text reads "Join your account. Don't have account? [Create Account](#)". The form contains two input fields: "Email" with the value "tayyaba@gmail.com" and "Password" with masked characters ".....". To the right of the password field is a toggle icon. Below the password field, there is a checkbox labeled "Remember me" and a link "Forgot Password?". At the bottom, a large green button with the text "Get Login" is centered.

Welcome Back, Get Login

Join your account. Don't have account? [Create Account](#)

Email

tayyaba@gmail.com

Password

.....

Remember me [Forgot Password?](#)

Get Login

Figure 4.17: GUI of Login of User Module

4.9.2.3 Shopping Cart

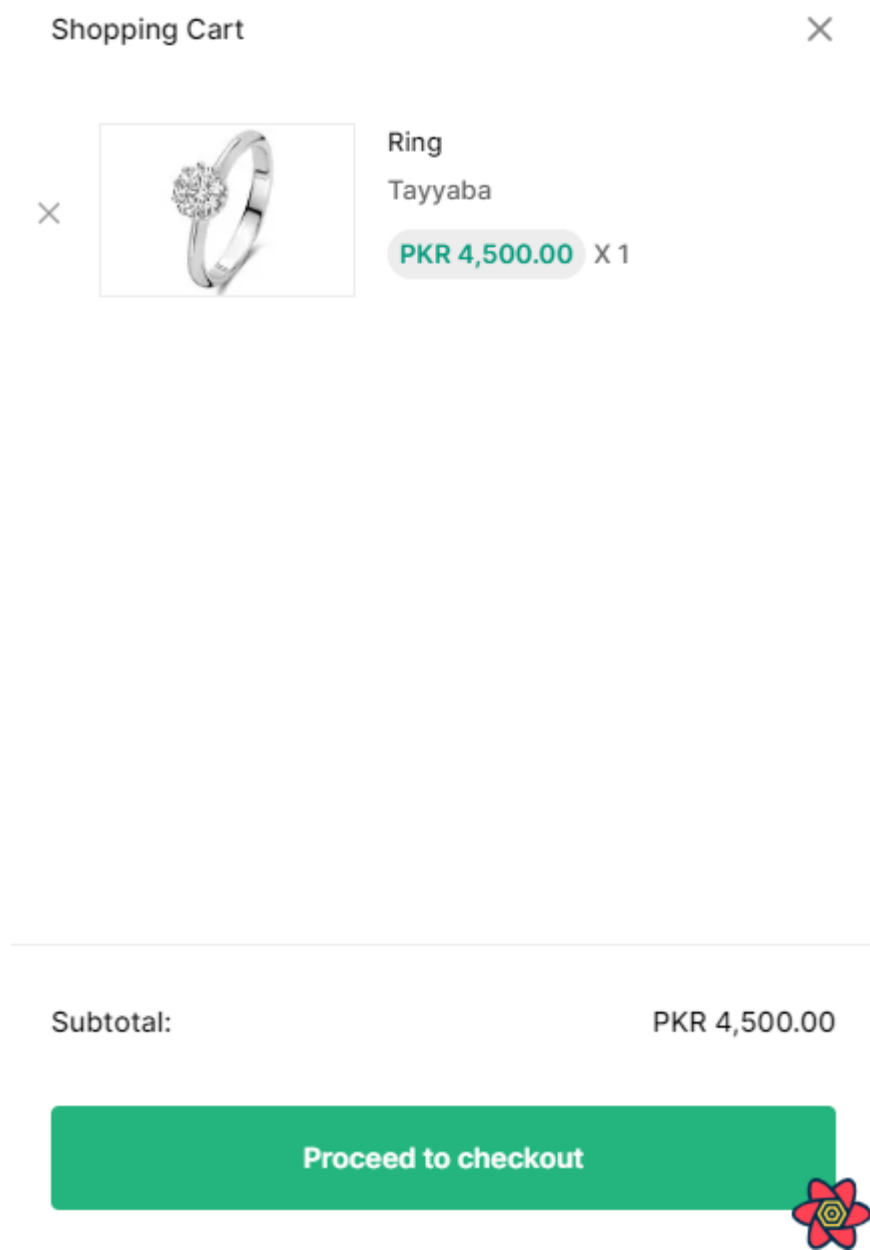



Figure 4.18: GUI of Shopping Cart of User Module

4.9.2.4 Place Order

Contact Number

+1 (647) 646-4468

Your Order (1)

×  Ring
Tayyaba
PKR 4,500.00 X 1

| | |
|-----------|--------------|
| Sub Total | PKR 4,500.00 |
| Tax | PKR 765.00 |

| | |
|-------|--------------|
| Total | PKR 5,265.00 |
|-------|--------------|

Use Cash on Delivery

Place Order

Figure 4.19: GUI of Placing Order of User Module

4.9.2.5 Purchase List

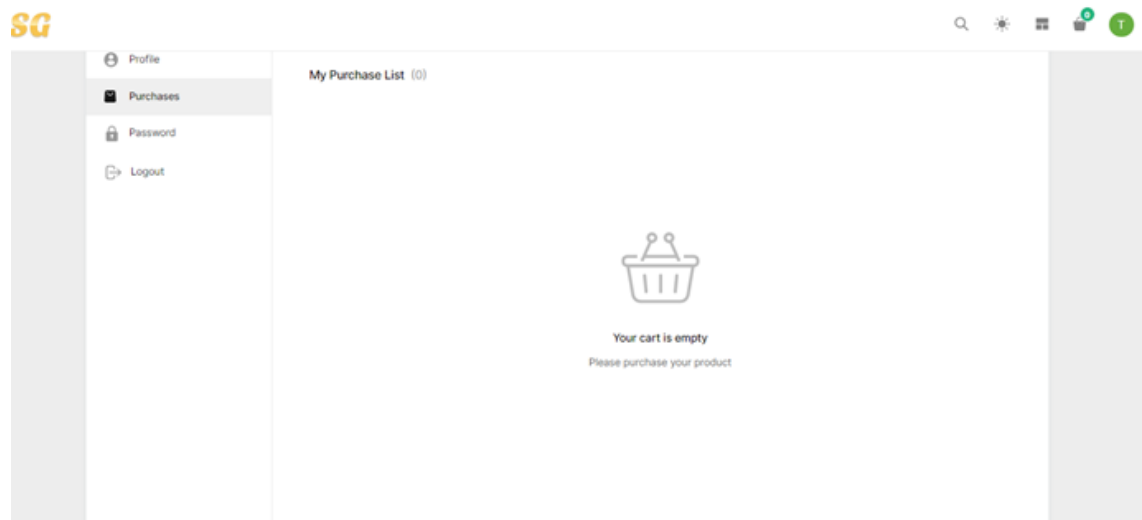


Figure 4.20: GUI of Purchase List of User Module

4.9.2.6 Change Password

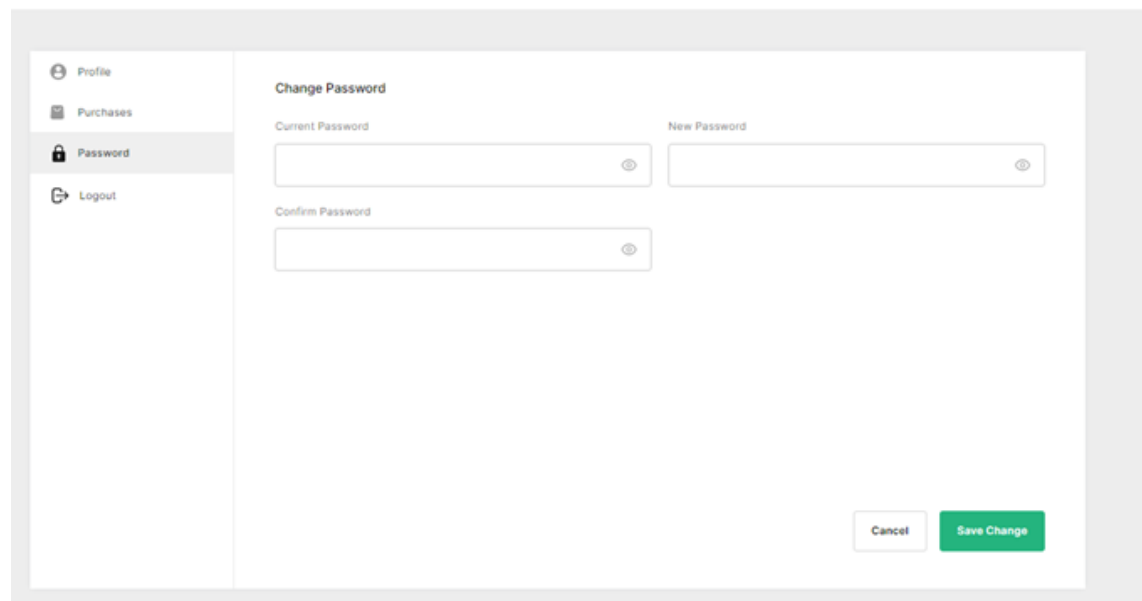


Figure 4.21: GUI of Change Password of User Module

4.9.3 Expert Module

4.9.3.1 Sign In

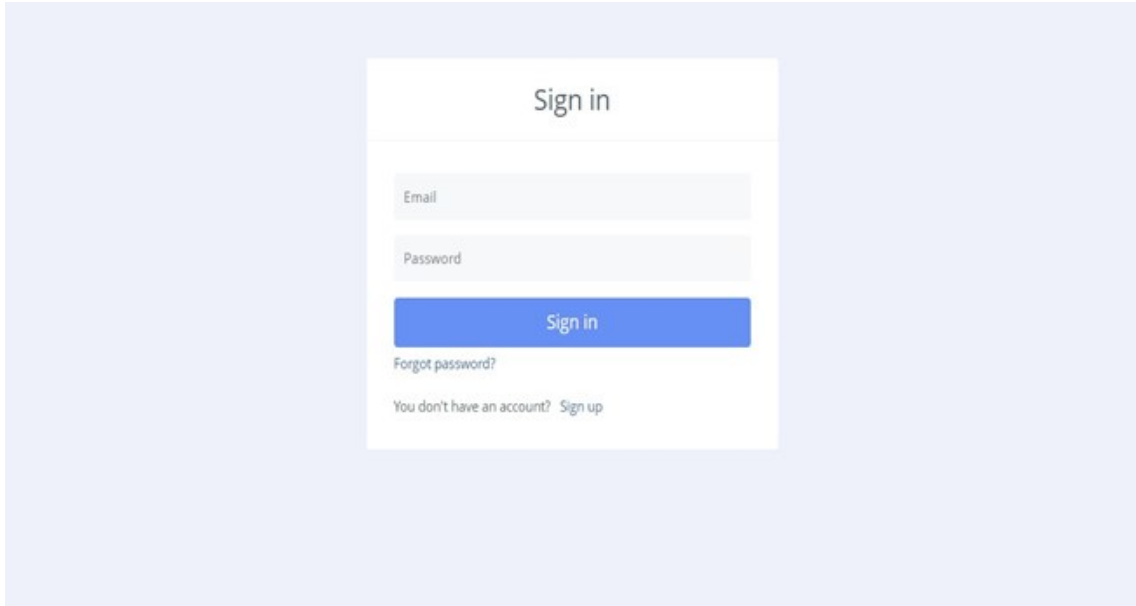


Figure 4.22: Sign In of Expert Module

4.9.3.2 Dashboard

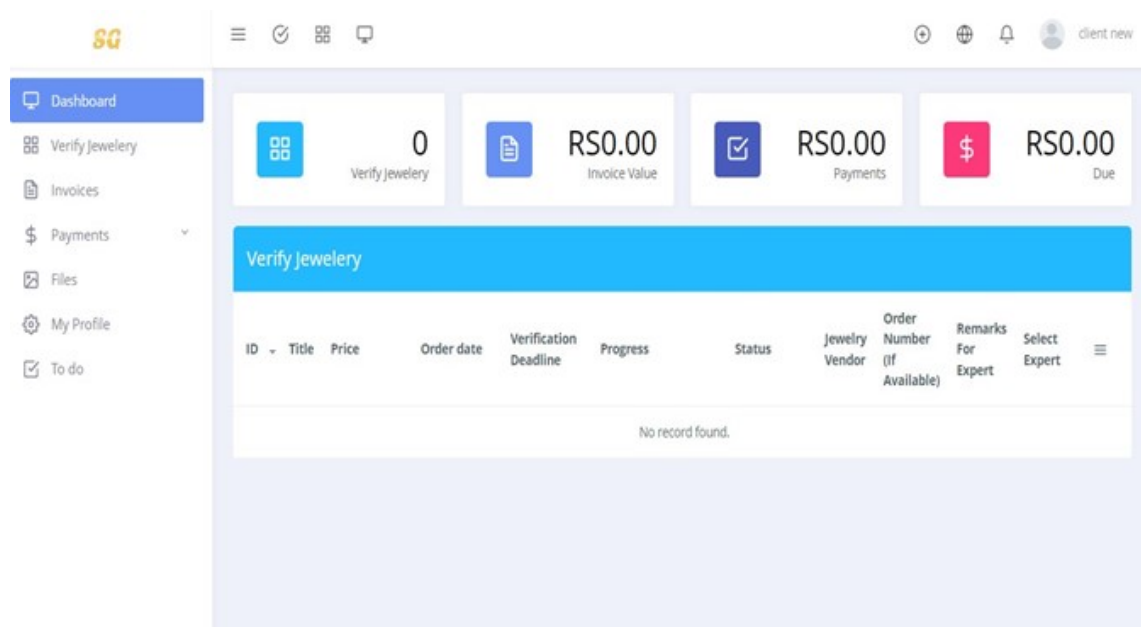


Figure 4.23: Dashboard of Expert Module

4.9.3.3 User Navigate to Verify Jewelry

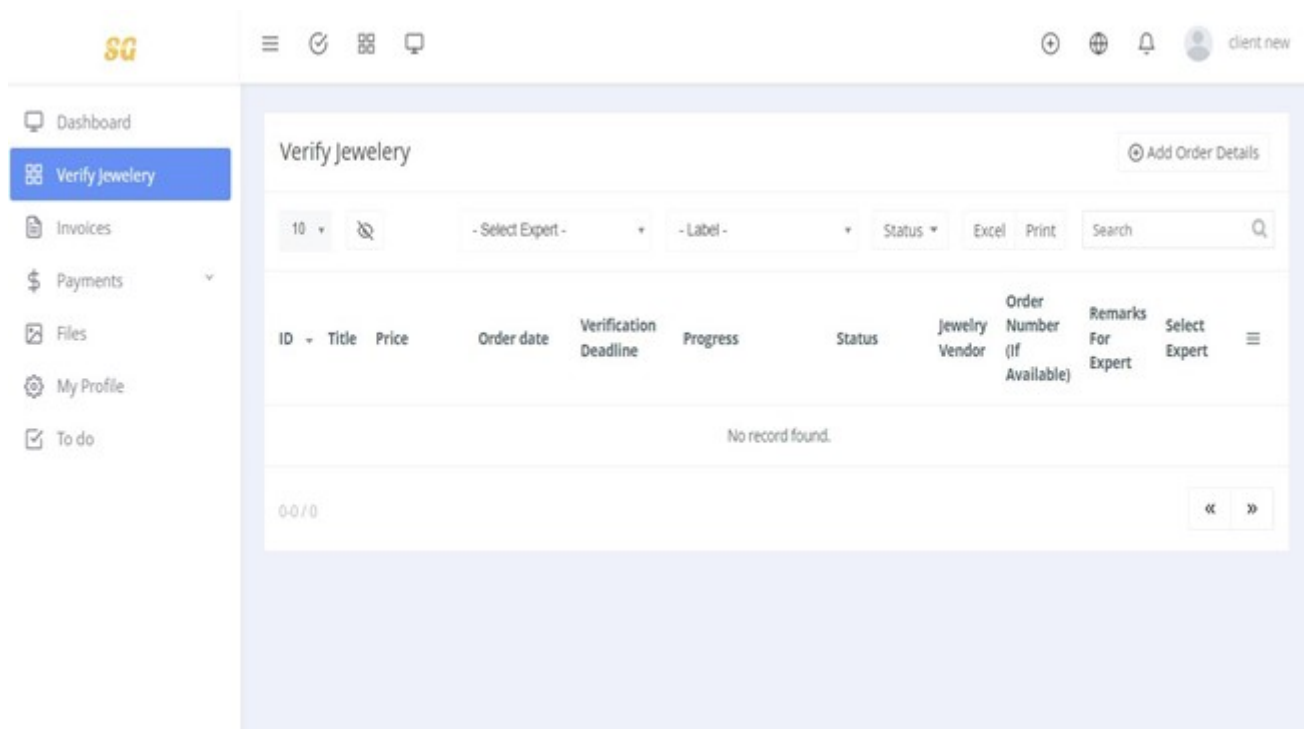


Figure 4.24: User Navigate to Verify Jewelry of Expert Module

4.9.3.4 User Add Order Details

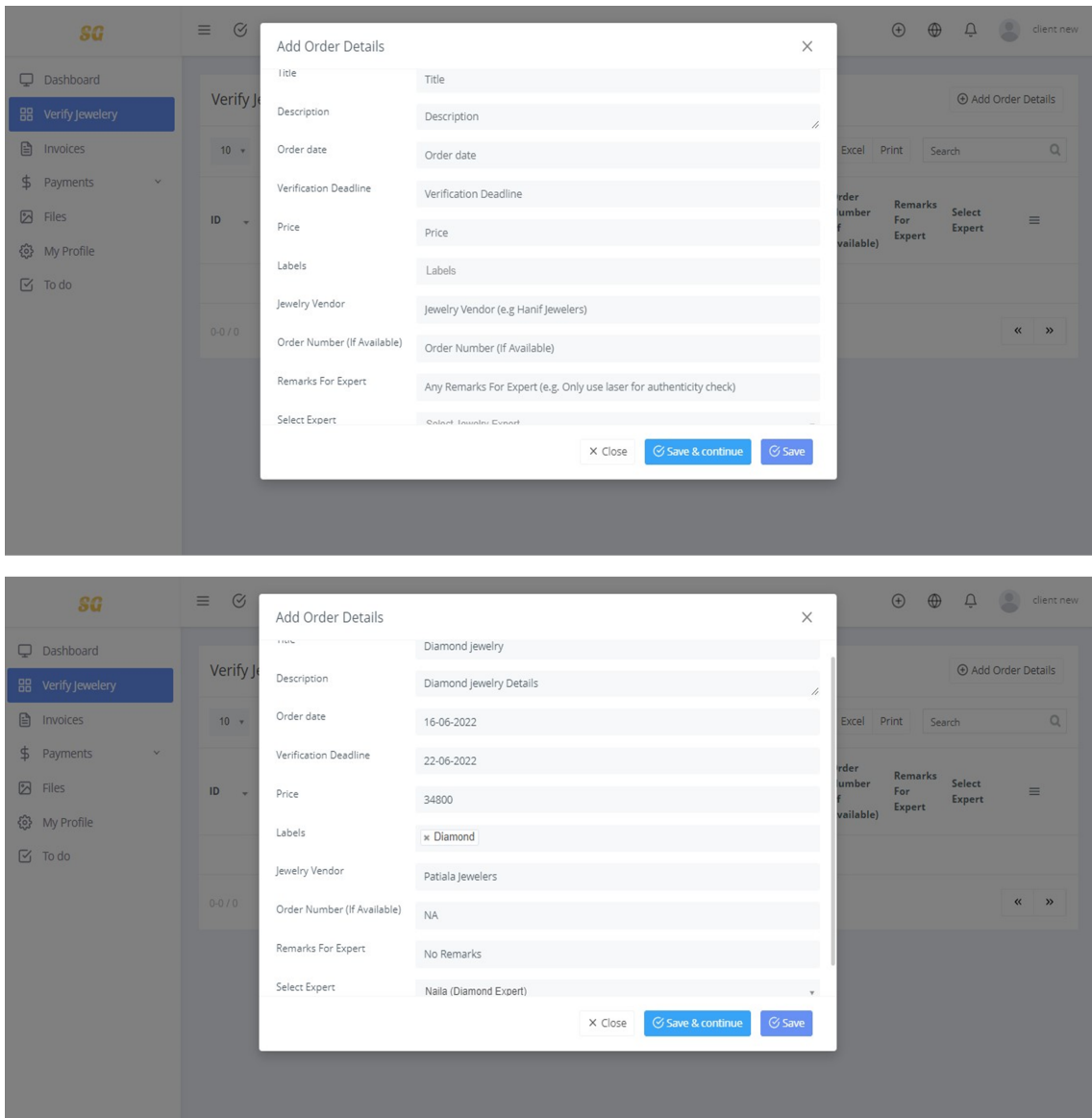


Figure 4.25: User Add Order Details of Expert Module

4.9.3.5 Jewelry Order Added

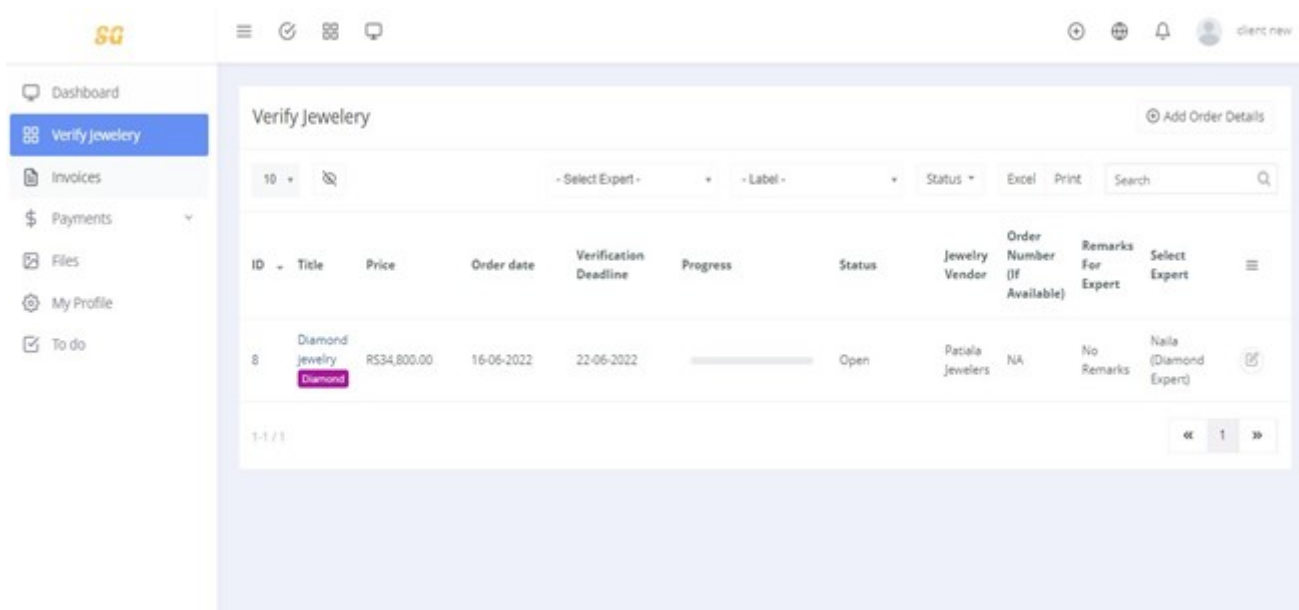


Figure 4.26: Jewelry Order Added of Expert Module

4.9.3.6 Detail Page

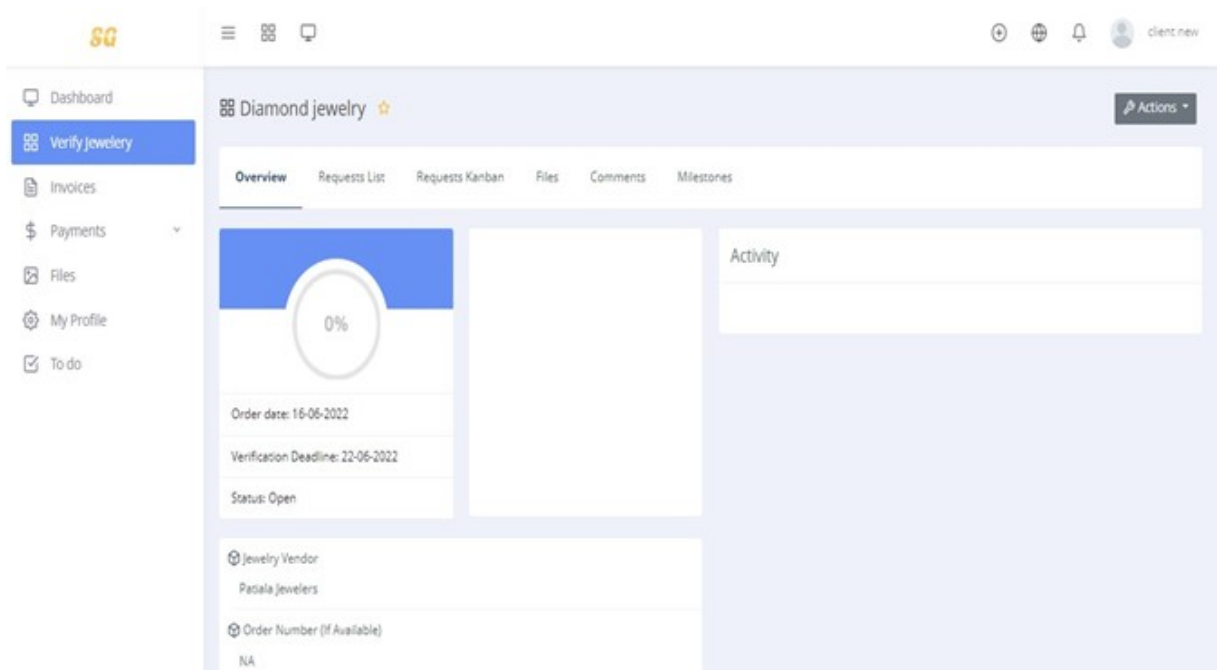


Figure 4.27: Detail Page of Expert Module

4.9.3.7 Request List

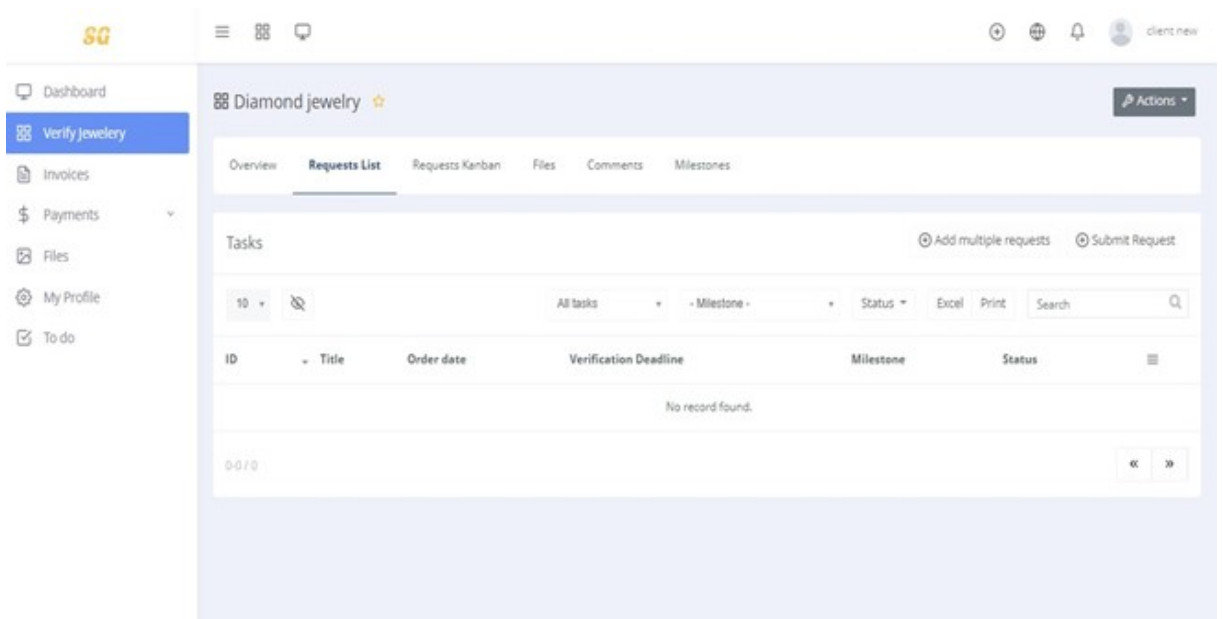


Figure 4.28: Request List of Expert Module

4.9.3.8 Inspection Request (Data Entry)

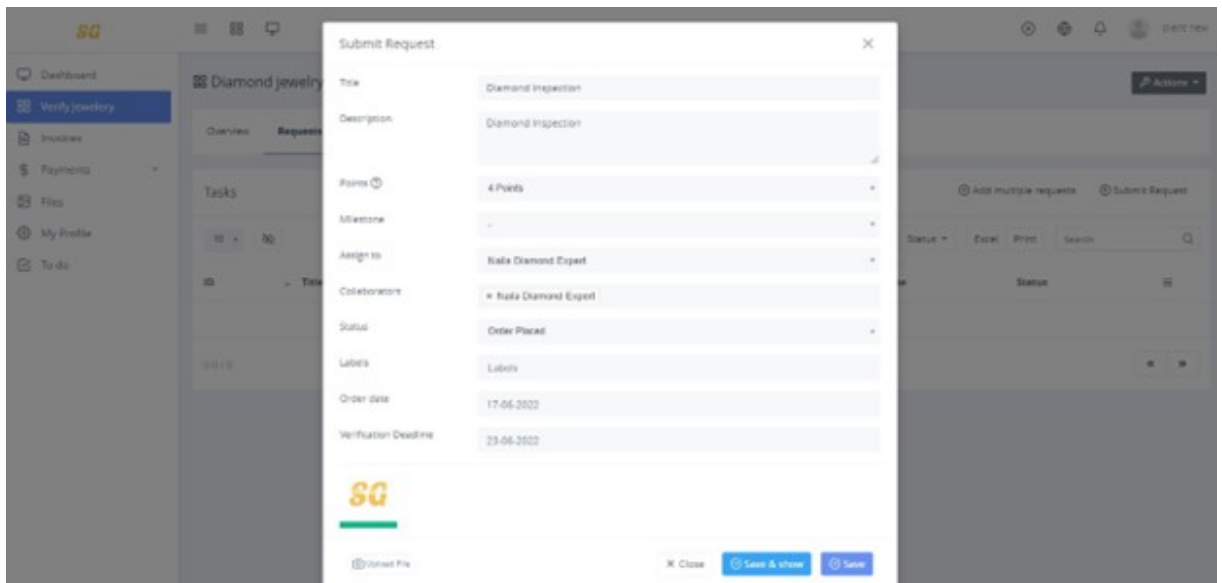


Figure 4.29: Inspection Request (Data Entry) of Expert Module

Chapter 5

System Implementation

System Implementation tells the technical specification and software components. It defines how the system should be built. We see the actual implementation of our system in this chapter. We have implemented our system on a Web-based platform.

5.1 System Architecture

The system architecture is a conceptual model that defines the system's architecture. It is a formal representation of the system. It includes the system's internal and external components. Our proposed system includes:

5.1.1 Web Application

Sparkles Gallery System has three major modules like Admin, User, and Expert. Users can log in using their email and password, then users can perform all the operations required to order jewelry. Vendors can create shops to sell their jewelry. Admin can view all the records.

5.1.2 System Internal Component

5.1.2.1 Register

In this system, three of the users like Users, Vendors, and Admin will get themselves registered first. Every actor will be registered through their email id and password.

5.1.2.2 Login

Users, Vendors, and Admin will get themselves login through the login page where the system will authenticate their email id and password.

5.1.2.3 Forget Password

If any of the users forget their password, they can recover it by forgetting the password page where they just had to enter their email and the password recovery link will be sent to their respective email.

5.1.2.4 User Portal

The user is one of the main actors acting on this system here are the functionalities that a user can perform after logging in to the system.

5.1.2.5 Products

Admin can view and manage all the products and shops.

5.1.2.6 Categories

There is a section for categories. In which there are different categories like gold, silver, diamond, etc.

5.1.2.7 Popular Jewelry

Users can view the popular jewelry. To order any jewelry he needs to login into his account.

5.1.2.8 Add Product

The seller can add a product by clicking on add product button. The form opens by pressing the button. He can add the name, image, price, and quantity. He can also edit or delete the products.

5.1.2.9 Vendor Profile

The vendor can change the setting through his profile. He can add a name, image bio, etc.

5.1.2.10 Help

There is a Frequently asked questions section in it.

5.1.2.11 User Profile

User can add his personal information in this section.

5.1.2.12 User Purchases

User can see all the lists of products he has purchased.

5.2 Tools and Technology

5.2.1 Next JS

Next JS is a JavaScript framework developed by Zeit. It allows you to use React to create a server-side rendering and static web applications. It's an excellent tool for creating your next website. It has a lot of great features and benefits that can make Next js your first choice for building your next web application.

5.2.2 Typescript

Typescript is an excellent choice for front-end development. Typescript lends structure and safety to your application and makes it easier to write correct, maintainable code in the fast-paced environment that is front-end development, thanks to its rich support for JSX and its ability to safely model mutability.

5.2.3 Tailwind CSS

Tailwind CSS is a utility-first CSS framework for quickly creating customized user interfaces. It is a highly customizable, low-level CSS framework that provides you with all of the building blocks you need to create custom designs without any annoying opinionated styles that you must fight to override.

5.2.4 React Query

React Query is often referred to as Reacts "missing data-fetching library," but in more technical terms, it simplifies fetching, caching, synchronizing, and updating server state in your React applications.

5.2.5 React JS

React is a free and open-source JavaScript front-end library for creating user interfaces based on UI components. Meta and a community of individual developers and companies maintain it.

5.2.6 Framer Motion

A production-gearred-up movement library for React. Utilize the strength at the back of Framer, the excellent prototyping device for teams.

5.2.7 Headless UI

It is a collection of completely unstyled, fully accessible UI components that have been designed to work seamlessly with Tailwind CSS. Headless UI is a technology stack tool in the UI Components category.

5.2.8 Google Fonts

Google Fonts is a Google-owned computer font and web font service. This includes font families that are free and open-source, an interactive web directory for browsing the library, and APIs for using the fonts with CSS and Android.

5.2.9 Node JS

Node JS is primarily used for non-blocking, event-driven servers because of its single-threaded nature. It's used for traditional websites and back-end API services, but it was built for real-time, push-based architectures.

5.2.10 Core JS

Core.JS is responsible to defines the jQuery namespace and the prototype for jQuery objects.

5.3 Methodology

5.3.1 Phase 1

In the very first phase, we gathered information from different people to prioritize requirements and developed a basic Web application where we can perform basic activities.

5.3.2 Phase 2

In this phase, we have developed a fully functional and interactive web-based system with the updated requirements and designed a proper interface that can perform all the activities related to ordering jewelry.

Chapter 6

System Testing and Evaluation

System testing is one of the most important steps that have to be performed before the system goes live. It tells whether all of the functions that the system needs to perform are performing accurately or not. We test each and every module separately to check whether it meets all of the requirements. Qualitative requirements of the work must be done. Many test cases have been performed on the system to evaluate its performance. Some of the test techniques used are.

6.0.1 GRAPHICAL USER INTERFACE TESTING

This test was performed to ensure that the graphical user interface is working properly.

6.0.1.1 Test case for Login Screen

In this table [6.1](#) shown below, the details of test case of the login screen is shown. It shows that whether the user can successfully login to his account. It was successful.

Table 6.1: Test Case for Login Screen

| TC-01 | | | |
|---|--|------------------|--------|
| Login screen | | | |
| Functional requirement 01 | | | |
| The system must have a visual studio/visual studio code | | | |
| STEPS | TASKS | EXPECTED OUTCOME | RESULT |
| 1 | Open the login panel | Pass or Fail | Pass |
| 2 | Check whether the login button is working | Pass or Fail | Pass |
| 3 | Check whether the data can be submitted or not. | Pass or Fail | Pass |
| 4 | Enter the username and password. | Pass or Fail | Pass |
| 5 | Check whether the username can be submitted or not. | Pass or Fail | Pass |
| 6 | Check whether error message is generated when the wrong password is entered. | Pass or Fail | Pass |

6.0.1.2 Sign Up Screen

Check whether the sign page or form is working correctly.

In this table 6.2 shown below, the details of test cases of the sign up screen. It checks whether the user can successfully create his own account. It was a successful test.

Table 6.2: Test Case for Sign Up Screen

| TC-02 | | | |
|---|--|------------------|--------|
| Sign up screen | | | |
| Functional requirement 02 | | | |
| The system must have a visual studio/visual studio code | | | |
| STEPS | TASKS | EXPECTED OUTCOME | RESULT |
| 1 | Open the sign up panel | Pass or Fail | Pass |
| 2 | Check whether the sign-up page is working properly | Pass or Fail | Pass |
| 3 | Check if the button is submitting data. | Pass or Fail | Pass |
| 4 | Check every field of the form by entering data. | Pass or Fail | Pass |

6.0.1.3 Forget Password

Check if the Forget Password is working properly.

In this table 6.3 shown below, the details of test cases of the forget password. In case the user forgets his password can he still enter into his account? For that he has to enter his E-mail. He'll get the link to recover his pass-code in email.

Table 6.3: Test Case for Forget Password

| TC-03 | | | |
|---|---------------------------------------|------------------|--------|
| Forget Password | | | |
| Functional requirement 03 | | | |
| The system must have a visual studio/visual studio code | | | |
| STEPS | TASKS | EXPECTED OUTCOME | RESULT |
| 1 | Open forget password page | Pass or Fail | Pass |
| 2 | Check whether the button is working | Pass or Fail | Pass |
| 3 | Check all the fields by entering data | Pass or Fail | Pass |

6.0.1.4 Home Page

Check whether the home page activities are working properly or not.

In this table 6.4 shown below, the details of test cases of the home page of the user website. If the page is showing updated thing, then the system is working for sure.

Table 6.4: Test Case for Home Page

| TC-04 | | | |
|---|---|------------------|--------|
| Home | | | |
| Functional requirement 04 | | | |
| The system must have a visual studio/visual studio code | | | |
| STEPS | TASKS | EXPECTED OUTCOME | RESULT |
| 1 | Open the Home page | Pass or Fail | Pass |
| 2 | Check the menu | Pass or Fail | Pass |
| 3 | Check the traversals of the page | Pass or Fail | Pass |
| 4 | Check whether the links are working or not. | Pass or Fail | Pass |

6.0.2 USABILITY TESTING

6.0.2.1 Login

In this table 6.5 shown below, the details of test case of the login screen is shown. It shows that whether the user can successfully login to his account. It was successful.

Table 6.5: Test Case for Login Screen

| TC-01 | | | |
|---|--|------------------|--------|
| Login Screen | | | |
| Functional requirement 02 | | | |
| The system must have a visual studio/visual studio code | | | |
| STEPS | TASKS | EXPECTED OUTCOME | RESULT |
| 1 | Open Login panel | Pass or Fail | Pass |
| 2 | Check if the login button is working properly. | Pass or Fail | Pass |
| 3 | Enter username and password | Pass or Fail | Pass |
| 4 | Verify if you've been logged in. | Pass or Fail | Pass |

6.0.2.2 CRUD Operation Testing

In the table 6.6 shown below, here's a usability test case for creating, reading, updating, and deleting. All of these actions are performing like creating shops, adding shops, updating shops etc. so, it is a successful test.

Table 6.6: Test Case for CRUD Operation

| TC-02 | | | |
|---|---|------------------|--------|
| CRUD Operations | | | |
| Functional requirement 02 | | | |
| The system must have a visual studio/visual studio code | | | |
| STEPS | TASKS | EXPECTED OUTCOME | RESULT |
| 1 | Go to the home page | Pass or Fail | Pass |
| 2 | Click on any module of your adding new product. | Pass or Fail | Pass |
| 3 | Perform any task | Pass or Fail | Pass |
| 4 | Check how the data is submitted. | Pass or Fail | Pass |
| 5 | Check whether the data submitted has been updated | Pass or Fail | Pass |
| 6 | Check the detail of the product | Pass or Fail | Pass |
| 7 | Check the button of product details | Pass or Fail | Pass |

6.0.3 EXCEPTION TESTING

6.0.3.1 Login screen

In the table 6.7 shown below, the details of test cases of the login screen is shown. It shows whether the user can successfully log in to his account. It was successful.

Table 6.7: Test Case for Login Screen

| TC-01 | | | |
|---|---|------------------|--------|
| Login screen | | | |
| Functional requirement 01 | | | |
| The system must have a visual studio/visual studio code | | | |
| STEPS | TASKS | EXPECTED OUTCOME | RESULT |
| 1 | Open login Panel | Pass or Fail | Pass |
| 2 | Press the login button | Pass or Fail | Pass |
| 3 | Check that empty fields occur warning message | Pass or Fail | Pass |

6.0.3.2 Sign Up Screen

In this table 6.8 shown below, the details of test cases of the sign up screen. It checks whether the user can successfully create his own account. It was a successful test.

Table 6.8: Test Case for Sign Up Page

| TC-02 | | | |
|---|---|------------------|--------|
| Sign up screen | | | |
| Functional requirement 02 | | | |
| The system must have a visual studio/visual studio code | | | |
| STEPS | TASKS | EXPECTED OUTCOME | RESULT |
| 1 | Open sign-up page | Pass or Fail | Pass |
| 2 | Press the signs up button without entering anything | Pass or Fail | Pass |
| 3 | Check that empty fields occur in error messages | Pass or Fail | Pass |

6.0.3.3 Forget Password

In this table 6.9 shown below, the details of test cases of the forget password. In case the user forgets his password can he still enter into his account? For that he has to enter his E-mail. He'll get the link to recover his pass-code in email.

Table 6.9: Test Case for Forget Password

| TC-03 | | | |
|---|---|------------------|--------|
| Forget Password | | | |
| Functional requirement 03 | | | |
| The system must have a visual studio/visual studio code | | | |
| STEPS | TASKS | EXPECTED OUTCOME | RESULT |
| 1 | Open forget password page | Pass or Fail | Pass |
| 2 | Press the button without entering in fields | Pass or Fail | Pass |
| 3 | Check whether the empty fields display warning messages | Pass or Fail | Pass |

6.0.3.4 Request Generation Testing

In the table 6.10 shown below, there's a whole description of the test case request generation. In order to get the jewelry inspected the user can make a request for inspecting. The buyer can request the inspection.

Table 6.10: Test Case for Request Generation Testing

| TC-04 | | | |
|---|--|------------------|--------|
| Request Generation | | | |
| Functional requirement 04 | | | |
| The system must have a visual studio/visual studio code | | | |
| STEPS | TASKS | EXPECTED OUTCOME | RESULT |
| 1 | Go to request the generation of an expert | Pass or Fail | Pass |
| 2 | Check the request details | Pass or Fail | Pass |
| 3 | Press the button send a request | Pass or Fail | Pass |
| 4 | Check if the request is sent. | Pass or Fail | Pass |
| 5 | Check if the button is replaced by a text. | Pass or Fail | Pass |

Chapter 7

Conclusions

There are so many ecommerce websites for businesses that helps those businesses to reach out to vast number of users. [2] Our main goal is to bridge the gap between user who want to sell their product and the ones who wish to buy some in a trustful manner. In this project, the concept of thrifting is introduced especially for the jewelry. It is a programming-oriented project that uses the concepts of React, PHP, My SQL and HTML. This project helped us in developing some of the skills, learnt time is an important factor and sticking to the goal throughout this entire project. We overcame all obstacles for the effective execution of this jewelry trade website. We used an incremental approach to construct this project because there were so many modules to design and dealing with React was new to us. Sparkles Gallery is developed in such a way that the seller and buyer connect for the trade of jewelry product in a reasonable price to avoid any sort of loss ratio. The verification would be done by a jewelry expert to tell whether the jewelry is according to the specs being mentioned. All of the data related to transaction, users, products and shops would be stored in My SQL database. In order to sell a product a user will first create a shop and upload the jewelry on that shop. Multiple shops can be created by a single user. The buyer will simply look for a product and add to the cart. Admin would do all of the management related tasks. He would see which expert inspecting whose jewelry. He can also create his own shop if he wishes to sell his own jewelry products. Here, all the processes and efforts related to the inspection would be done manually however the hassle of viewing, selecting, or reviewing the jewelry by an expert will be removed by using this project. Within the time frame, developing this system was quite challenging. We took this project on a journey and learned a lot of valuable things along the way, expanding our knowledge.

7.1 Future Enhancements

7.1.1 Application

In the future, if the need arises or for business purposes, we can make it a mobile application as well.

7.1.2 Bidding

We can also add the feature of the online bidding process. Bidding process can take place between the buyer and seller online and a price would be set with the agreement of both of the parties.

7.1.3 Chat Box

A chat box for easy online communication between the buyer and seller so that they can arrange meeting etc. in that chat box easily without having to exchange numbers

Appendix A

User Manual

Implementation Code

Admin Modules

A.1 Admin Module

```
1 {
2   "name": "@pixier/admin-rest",
3   "version": "1.0.0",
4   "private": true,
5   "scripts": {
6     "dev": "next dev -p 3002",
7     "build": "next build",
8     "start": "next start -p 3002"
9   },
10  "dependencies": {
11    "@fontsource/open-sans": "4.5.2",
12    "@headlessui/react": "1.4.2",
13    "@hookform/resolvers": "2.8.4",
14    "@reach/portal": "0.16.2",
15    "@react-google-maps/api": "2.7.0",
16    "@react-pdf/renderer": "2.0.21",
17    "apexcharts": "3.31.0",
18    "axios": "0.24.0",
19    "body-scroll-lock": "4.0.0-beta.0",
20    "camelcase-keys": "7.0.1",
21    "classnames": "2.3.1",
22    "cookie": "0.4.1",
23    "dayjs": "1.10.7",
24    "framer-motion": "5.3.3",
25    "jotai": "1.4.6",
```

```
26   "js-cookie": "3.0.1",
27   "lodash": "4.17.21",
28   "next": "12.0.7",
29   "next-pwa": "5.4.4",
30   "next-seo": "4.28.1",
31   "overlayscrollbars": "1.13.1",
32   "overlayscrollbars-react": "0.2.3",
33   "rc-pagination": "3.1.14",
34   "rc-progress": "3.1.4",
35   "rc-table": "7.21.0",
36   "react": "17.0.2",
37   "react-apexcharts": "1.3.9",
38   "react-datepicker": "4.5.0",
39   "react-dom": "17.0.2",
40   "react-dropzone": "11.4.2",
41   "react-hook-form": "7.20.5",
42   "react-laag": "2.0.3",
43   "react-phone-input-2": "2.14.0",
44   "react-query": "3.34.0",
45   "react-scroll": "1.8.4",
46   "react-select": "5.2.1",
47   "react-toastify": "8.1.0",
48   "react-use": "17.3.1",
49   "swiper": "7.3.1",
50   "yup": "0.32.11"
51 },
52 "devDependencies": {
53   "@types/body-scroll-lock": "3.1.0",
54   "@types/js-cookie": "3.0.1",
55   "@types/lodash": "4.14.177",
56   "@types/node": "16.11.11",
57   "@types/overlayscrollbars": "1.12.1",
58   "@types/react": "17.0.37",
59   "@types/react-datepicker": "4.3.2",
60   "autoprefixer": "10.4.0",
61   "next-i18next": "10.0.1",
62   "postcss": "8.4.4",
63   "tailwindcss": "2.2.19",
64   "tailwindcss-rtl": "0.8.0",
65   "typescript": "4.5.2"
66 }
67 }
```

Listing A.1: Trend Sparkles Gallery Implementation Code

A.2 Shop Module

```
1 {
2   "name": "@pixier/shop",
3   "version": "1.0.0",
4   "private": true,
5   "scripts": {
6     "clean": "rimraf \"{node_modules,.next,.cache}\"",
7     "dev": "next dev",
8     "build": "next build",
9     "start": "next start",
10    "lint": "next lint",
11    "prepare": "husky install"
12  },
13  "dependencies": {
14    "@headlessui/react": "1.5.0",
15    "@hookform/resolvers": "2.8.8",
16    "@stripe/react-stripe-js": "1.7.0",
17    "@stripe/stripe-js": "1.23.0",
18    "@tailwindcss/forms": "0.4.0",
19    "@tailwindcss/typography": "0.5.2",
20    "axios": "0.26.0",
21    "classnames": "2.3.1",
22    "dayjs": "1.10.7",
23    "framer-motion": "6.2.3",
24    "jotai": "1.6.0",
25    "js-cookie": "3.0.1",
26    "little-state-machine": "4.2.1",
27    "lodash": "4.17.21",
28    "next": "11.0.1",
29    "next-pwa": "5.4.4",
30    "next-seo": "5.1.0",
31    "next-sitemap": "2.4.3",
32    "next-themes": "0.0.15",
33    "overlayscrollbars": "1.13.1",
34    "overlayscrollbars-react": "0.2.3",
35    "pluralize": "8.0.0",
36    "react": "17.0.2",
37    "react-avatar": "4.0.0",
38    "react-confetti": "6.0.1",
39    "react-content-loader": "6.1.0",
40    "react-dom": "17.0.2",
41    "react-dropzone": "12.0.4",
42    "react-hook-form": "7.27.1",
43    "react-hot-toast": "2.2.0",
44    "react-phone-input-2": "2.15.0",
45    "react-query": "3.34.15",
46    "react-share": "4.4.0",
47    "react-use": "17.3.2",
```



```

48   "sharp": "0.30.1",
49   "swiper": "8.0.6",
50   "tiny-invariant": "1.2.0",
51   "yup": "0.32.11"
52 },
53 "devDependencies": {
54   "@types/js-cookie": "3.0.1",
55   "@types/node": "17.0.18",
56   "@types/pluralize": "0.0.29",
57   "@types/react": "17.0.39",
58   "@types/react-copy-to-clipboard": "5.0.2",
59   "@types/react-dom": "17.0.11",
60   "autoprefixer": "10.4.1",
61   "eslint": "8.9.0",
62   "eslint-config-next": "12.1.0",
63   "eslint-config-prettier": "8.3.0",
64 "husky": "7.0.0",
65   "lint-staged": "12.3.4",
66   "postcss": "8.4.6",
67   "prettier": "2.5.1",
68   "prettier-plugin-tailwindcss": "0.1.7",
69   "rimraf": "3.0.2",
70   "tailwindcss": "3.0.23",
71   "typescript": "4.5.5"
72 }
73 }

```

Listing A.2: Trend Sparkles Gallery Implementation Code

A.3 Client Module

```

1 import * as yup from 'yup';
2 import type { SubmitHandler } from 'react-hook-form';
3 import { useMutation } from 'react-query';
4 import toast from 'react-hot-toast';
5 import type { LoginUserInput } from '@types';
6 import { Form } from '@components/ui/forms/form';
7 import Password from '@components/ui/forms/password';
8 import Input from '@components/ui/forms/input';
9 import Button from '@components/ui/button';
10 import { useModalAction } from '@components/modal-views/context';
11 import useAuth from '@components/auth/use-auth';
12 import CheckBox from '@components/ui/forms/checkbox';
13 import { RegisterBgPattern } from '@components/auth/register-bg-pattern';
14 import client from '@data/client';
15
16 const loginValidationSchema = yup.object().shape({
17   email: yup.string().email().required(),

```

```

18   password: yup.string().required(),
19 });
20 export default function LoginUserForm() {
21   const { openModal, closeModal } = useModalAction();
22   const { authorize } = useAuth();
23   const { mutate: login } = useMutation(client.users.login, {
24     onSuccess: (data) => {
25       if (!data.token) {
26         toast.error(<b>Wrong username or password</b>, {
27           className: '-mt-10 xs:mt-0',
28         });
29         return;
30       }
31       authorize(data.token);
32       closeModal();
33     },
34 });
35   const onSubmit: SubmitHandler<LoginUserInput> = (data) => {
36     login(data);
37   };
38   return (
39     <div className="px-6 pt-10 pb-8 sm:px-8 lg:p-12">
40       <RegisterBgPattern className="absolute bottom-0 left-0 text-light dark:text-
41         dark-300 dark:opacity-60" />
42       <div className="relative z-10 flex items-center">
43         <div className="w-full shrink-0 text-left md:w-[380px]">
44           <div className="flex flex-col pb-5 text-center xl:pb-6 xl:pt-2">
45             <h2 className="text-lg font-medium tracking-[-0.3px] text-dark dark:
46               text-light lg:text-xl">
47               Welcome Back, Get Login
48             </h2>
49             <div className="mt-1.5 text-13px leading-6 tracking-[0.2px] dark:text-
50               light-900 lg:mt-2.5 xl:mt-3">
51               Join your account. Dont have account?{' '}
52             <button
53               onClick={() => openModal('REGISTER')}
54               className="inline-flex font-semibold text-brand hover:text-dark-400
55               hover:dark:text-light-500"
56             >
57               Create Account
58             </button>
59           </div>
60         </div>
61       </div>
62       <Form<LoginUserInput>
63         onSubmit={onSubmit}
64         validationSchema={loginValidationSchema}
65         className="space-y-4 pt-4 lg:space-y-5"
66       >
67         {({ register, formState: { errors } }) => (

```

```

63     <>
64         <Input
65             label="Email"
66             inputClassName="bg-light dark:bg-dark-300"
67             type="email"
68             {...register('email')}
69             error={errors.email?.message}
70         />
71         <Password
72             label="Password"
73             inputClassName="bg-light dark:bg-dark-300"
74             {...register('password')}
75             error={errors.password?.message}
76         />
77         <div className="flex items-center justify-between space-x-5">
78 <CheckBox
79             label="Remember me"
80             // inputClassName="bg-light dark:bg-dark-300"
81         />
82         <button
83             type="button"
84             className="text-13px font-semibold text-brand hover:text-dark
-400 hover:dark:text-light-500"
85             onClick={() => openModal('FORGOT_PASSWORD_VIEW')}
86         >
87             Forgot Password?
88         </button>
89     </div>
90     <Button
91         type="submit"
92         className="!mt-5 w-full text-sm tracking-[0.2px] lg:!mt-7"
93     >
94         Get Login
95     </Button>
96 </>
97     )}
98 </Form>
99 </div>
100 </div>
101 </div>
102 );
103 }

```

Listing A.3: Trend Sparkles Gallery Implementation Code

A.4 Client Register

```
1 import * as yup from 'yup';
2 import type { SubmitHandler } from 'react-hook-form';
3 import type { RegisterUserInput } from '@/types';
4 import { useMutation } from 'react-query';
5 import toast from 'react-hot-toast';
6 import { Form } from '@/components/ui/forms/form';
7 import Password from '@/components/ui/forms/password';
8 import { useModalAction } from '@/components/modal-views/context';
9 import Input from '@/components/ui/forms/input';
10 import client from '@/data/client';
11 import Button from '@/components/ui/button';
12 import { RegisterBgPattern } from '@/components/auth/register-bg-pattern';
13 import { useState } from 'react';
14 import useAuth from './use-auth';
15
16 const registerUserValidationSchema = yup.object().shape({
17   name: yup.string().max(20).required(),
18   email: yup.string().email().required(),
19   password: yup.string().min(6).required(),
20 });
21
22 export default function RegisterUserForm() {
23   const { openModal, closeModal } = useModalAction();
24   const { authorize } = useAuth();
25   let [serverError, setServerError] = useState<RegisterUserInput | null>(null);
26   const { mutate } = useMutation(client.users.register, {
27     onSuccess: (res) => {
28       if (!res.token) {
29         toast.error(<b>Something went wrong</b>, {
30           className: '-mt-10 xs:mt-0',
31         });
32         return;
33       }
34       authorize(res.token);
35       closeModal();
36     },
37     onError: (err: any) => {
38       console.log(err.response.data, 'error');
39       setServerError(err.response.data);
40     },
41   });
42   const onSubmit: SubmitHandler<RegisterUserInput> = (data) => {
43     mutate(data);
44   };
45
46   return (
47     <div className="px-6 pt-10 pb-8 sm:px-8 lg:p-12">
```

```

48     <RegisterBgPattern className="absolute bottom-0 left-0 text-light dark:text-
dark-300 dark:opacity-60" />
49     <div className="relative z-10 flex items-center">
50         <div className="w-full shrink-0 text-left md:w-[380px]">
51             <div className="flex flex-col pb-5 text-center lg:pb-9 xl:pb-10 xl:pt-2">
52                 <h2 className="text-lg font-medium tracking-[-0.3px] text-dark dark:
text-light lg:text-xl">
53                     Welcome Back, Get Login
54                 </h2>
55                 <div className="mt-1.5 text-13px leading-6 tracking-[0.2px] dark:text-
light-900 lg:mt-2.5 xl:mt-3">
56                     Create your account. Already have account?{' '}
57                     <button
58                         onClick={() => openModal('LOGIN_VIEW')}
59                         className="inline-flex font-semibold text-brand hover:text-dark-400
hover:dark:text-light-500"
60                     >
61                         Login here
62                     </button>
63                 </div>
64             </div>
65
66             <Form<RegisterUserInput>
67                 onSubmit={onSubmit}
68                 validationSchema={registerUserValidationSchema}
69                 serverError={serverError}
70                 className="space-y-4 lg:space-y-5"
71             >
72                 {{{ register, formState: { errors } }} => (
73                     <>
74                         <Input
75                             label="Name"
76                             inputClassName="bg-light dark:bg-dark-300"
77                             {...register('name')}
78                             error={errors.name?.message}
79                         />
80                         <Input
81                             label="Email"
82                             inputClassName="bg-light dark:bg-dark-300"
83                             type="email"
84                             {...register('email')}
85                             error={errors.email?.message}
86                         />
87                         <Password
88                             label="Password"
89                             inputClassName="bg-light dark:bg-dark-300"
90                             {...register('password')}
91                             error={errors.password?.message}
92                         />

```

```

93         <Button
94             type="submit"
95             className="!mt-5 w-full text-sm tracking-[0.2px] lg:!mt-7"
96         >
97             Register
98         </Button>
99     </>
100     )}
101 </Form>
102 </div>
103 </div>
104 </div>
105 );
106 }

```

Listing A.4: Trend Sparkles Gallery Implementation Code

A.5 Product

```

1 import dayjs from 'dayjs';
2 import { UpdateIcon } from '@components/icons/update-icon';
3 import { CalenderIcon } from '@components/icons/calendar-icon';
4 import { LayoutIcon } from '@components/icons/layout-icon';
5 import { LabelIcon } from '@components/icons/label-icon';
6 import AnchorLink from '@components/ui/links/anchor-link';
7 import classNames from 'classnames';
8 import { Tag } from '@types';
9 import routes from '@config/routes';
10
11 interface Props {
12     className?: string;
13     updated_at: string;
14     created_at: string;
15     tags: Tag[];
16     layoutType: string;
17 }
18
19 export default function ProductInformation({
20     className,
21     updated_at,
22     created_at,
23     tags,
24     layoutType,
25 }: Props) {
26     return (
27         <div className={classNames('space-y-4 text-13px', className)}>
28             <div className="flex items-start text-dark dark:text-light">

```

```

29     <strong className="flex w-36 flex-shrink-0 items-center font-normal text-
dark-600 dark:text-light-600">
30         <span className="w-8 flex-shrink-0 text-dark-900 dark:text-light-900">
31             <UpdateIcon className="h-[18px] w-[18px]" />
32         </span>
33         Last Update:
34     </strong>
35     <span className="font-medium">
36         {dayjs(updated_at).format('MMM D, YYYY')}
37     </span>
38 </div>
39 <div className="flex items-start text-dark dark:text-light">
40     <strong className="flex w-36 flex-shrink-0 items-center font-normal text-
dark-600 dark:text-light-600">
41         <span className="w-8 flex-shrink-0 text-dark-900 dark:text-light-900">
42             <CalenderIcon className="h-[18px] w-[18px]" />
43         </span>
44         Published:
45     </strong>
46     <span className="font-medium">
47         {dayjs(created_at).format('MMM D, YYYY')}
48     </span>
49 </div>
50 <div className="flex items-start text-dark dark:text-light">
51     <strong className="flex w-36 flex-shrink-0 items-center font-normal text-
dark-600 dark:text-light-600">
52         <span className="w-8 flex-shrink-0 text-dark-900 dark:text-light-900">
53             <LayoutIcon className="h-[18px] w-[18px]" />
54         </span>
55         Layout:
56     </strong>
57     <span className="font-medium">{layoutType}</span>
58 </div>
59     {!!tags?.length && (
60         <div className="flex items-start text-dark dark:text-light">
61             <strong className="flex w-36 flex-shrink-0 items-center pt-0.5 font-
normal text-dark-600 dark:text-light-600">
62                 <span className="w-8 flex-shrink-0 text-dark-900 dark:text-light-900">
63                     <LabelIcon className="h-5 w-5" />
64                 </span>
65                 Tags:
66             </strong>
67             <div className="flex flex-wrap gap-2">
68                 {tags.map((tag: Tag) => (
69                     <AnchorLink
70                         key={tag.id}
71                         href={routes.tagUrl(tag.slug)}
72                         className="inline-flex items-center justify-center rounded border
border-light-600 px-2 py-0.5 font-medium text-light-base transition-all hover:

```

```

    bg-light-200 hover:text-dark-300 dark:border-dark-500 dark:text-light-600 dark:
    hover:bg-dark-400 hover:dark:text-light"
73     >
74     {tag.name}
75     </AnchorLink>
76     )})
77     </div>
78     </div>
79     )}
80     </div>
81     );
82 }

```

Listing A.5: Trend Sparkles Gallery Implementation Code

A.6 Product

```

1 import dayjs from 'dayjs';
2 import { UpdateIcon } from '@components/icons/update-icon';
3 import { CalenderIcon } from '@components/icons/calendar-icon';
4 import { LayoutIcon } from '@components/icons/layout-icon';
5 import { LabelIcon } from '@components/icons/label-icon';
6 import AnchorLink from '@components/ui/links/anchor-link';
7 import classNames from 'classnames';
8 import { Tag } from '@types';
9 import routes from '@config/routes';
10
11 interface Props {
12   className?: string;
13   updated_at: string;
14   created_at: string;
15   tags: Tag[];
16   layoutType: string;
17 }
18
19 export default function ProductInformation({
20   className,
21   updated_at,
22   created_at,
23   tags,
24   layoutType,
25 }: Props) {
26   return (
27     <div className={classNames('space-y-4 text-13px', className)}>
28       <div className="flex items-start text-dark dark:text-light">
29         <strong className="flex w-36 flex-shrink-0 items-center font-normal text-
29         dark-600 dark:text-light-600">
30           <span className="w-8 flex-shrink-0 text-dark-900 dark:text-light-900">

```



```

31     <UpdateIcon className="h-[18px] w-[18px]" />
32   </span>
33   Last Update:
34 </strong>
35   <span className="font-medium">
36     {dayjs(updated_at).format('MMM D, YYYY')}
37   </span>
38 </div>
39 <div className="flex items-start text-dark dark:text-light">
40   <strong className="flex w-36 flex-shrink-0 items-center font-normal text-
dark-600 dark:text-light-600">
41     <span className="w-8 flex-shrink-0 text-dark-900 dark:text-light-900">
42       <CalendarIcon className="h-[18px] w-[18px]" />
43     </span>
44     Published:
45   </strong>
46   <span className="font-medium">
47     {dayjs(created_at).format('MMM D, YYYY')}
48   </span>
49 </div>
50 <div className="flex items-start text-dark dark:text-light">
51   <strong className="flex w-36 flex-shrink-0 items-center font-normal text-
dark-600 dark:text-light-600">
52     <span className="w-8 flex-shrink-0 text-dark-900 dark:text-light-900">
53       <LayoutIcon className="h-[18px] w-[18px]" />
54     </span>
55     Layout:
56   </strong>
57   <span className="font-medium">{layoutType}</span>
58 </div>
59 {!!tags?.length && (
60   <div className="flex items-start text-dark dark:text-light">
61     <strong className="flex w-36 flex-shrink-0 items-center pt-0.5 font-
normal text-dark-600 dark:text-light-600">
62       <span className="w-8 flex-shrink-0 text-dark-900 dark:text-light-900">
63         <LabelIcon className="h-5 w-5" />
64       </span>
65       Tags:
66     </strong>
67     <div className="flex flex-wrap gap-2">
68       {tags.map((tag: Tag) => (
69         <AnchorLink
70           key={tag.id}
71           href={routes.tagUrl(tag.slug)}
72           className="inline-flex items-center justify-center rounded border
border-light-600 px-2 py-0.5 font-medium text-light-base transition-all hover:
bg-light-200 hover:text-dark-300 dark:border-dark-500 dark:text-light-600 dark:
hover:bg-dark-400 hover:dark:text-light "
73         >

```

```
74         {tag.name}
75         </AnchorLink>
76     )))
77 </div>
78 </div>
79     })
80 </div>
81 );
82 }
83
84 import { useState } from 'react';
85 import cn from 'classnames';
86 import toast from 'react-hot-toast';
87 import Button from '@components/ui/button';
88 import { useCart } from '@components/cart/lib/cart.context';
89 import usePrice from '@lib/hooks/use-price';
90 import type { Product } from '@types';
91 import { generateCartItem } from './lib/generate-cart-item';
92
93 interface Props {
94   item: Product;
95   className?: string;
96   toastClassName?: string;
97 }
98
99 export default function AddToCart({ item, className, toastClassName }: Props) {
100   const { addItemToCart } = useCart();
101   const [addToCartLoader, setAddToCartLoader] = useState(false);
102   const [cartingSuccess, setCartingSuccess] = useState(false);
103   const { price } = usePrice({
104     amount: item?.sale_price ? item?.sale_price : item?.price,
105     baseAmount: item?.price,
106   });
107   function handleAddToCart() {
108     setAddToCartLoader(true);
109     setTimeout(() => {
110       setAddToCartLoader(false);
111       addSuccessfully();
112     }, 650);
113   }
114   function addSuccessfully() {
115     setCartingSuccess(true);
116     addItemToCart(generateCartItem(item), 1);
117     toast.success(<b>Successfully added to the cart!</b>, {
118       className: toastClassName,
119     });
120     setTimeout(() => {
121       setCartingSuccess(false);
122     }, 800);
```

```

123   }
124   return (
125     <Button
126       onClick={() => handleAddToCart()}
127       isLoading={addToCartLoader}
128       className={cn(
129         'relative',
130         cartingSuccess
131           ? 'is-not-carting pointer-events-none cursor-not-allowed'
132           : 'pointer-events-auto cursor-pointer',
133         className
134       )}
135     >
136     Add to Cart {price}
137     <svg
138       viewBox="0 0 37 37"
139       xmlns="http://www.w3.org/2000/svg"
140       className="absolute top-auto bottom-auto right-3 h-auto w-5 xs:right-4 xs:w
-6"
141     >
142       <path
143         fill="none"
144         stroke="currentColor"
145         strokeLinejoin="round"
146         strokeMiterlimit="10"
147         strokeWidth="2.3"
148         d="M30.5 6.5h0c6.6 6.6 6.6 17.4 0 24h0c-6.6 6.6-17.4 6.6-24 0h0c
-6.6-6.6-6.6-17.4 0-24h0c6.6-6.7 17.4-6.7 24 0z"
149         className="circle path"
150       />
151       <path
152         fill="none"
153         stroke="currentColor"
154         strokeLinejoin="round"
155         strokeMiterlimit="10"
156         strokeWidth="2.3"
157         d="M11.6 20L15.9 24.2 26.4 13.8"
158         className="tick path"
159       />
160     </svg>
161   </Button>
162 );
163 }

```

Listing A.6: Trend Sparkles Gallery Implementation Code

A.7 Site Setting

```
1 import { adminAndOwnerOnly, adminOwnerAndStaffOnly } from "@utils/auth-utils";
2 import { ROUTES } from "@utils/routes";
3
4 export const siteSettings = {
5   name: "SG",
6   description: "",
7   logo: {
8     url: "/logo.svg",
9     alt: "SG",
10    href: "/",
11    width: 170,
12    height: 52,
13  },
14  defaultLanguage: "en",
15  author: {
16    name: "Sparkles Gallery.",
17    websiteUrl: "https://sparklesgallery.com",
18    address: "",
19  },
20  headerLinks: [],
21  authorizedLinks: [
22    {
23      href: ROUTES.PROFILE_UPDATE,
24      labelTransKey: "authorized-nav-item-profile",
25    },
26    {
27      href: ROUTES.LOGOUT,
28      labelTransKey: "authorized-nav-item-logout",
29    },
30  ],
31  currencyCode: "USD",
32  sidebarLinks: {
33    admin: [
34      {
35        href: ROUTES.DASHBOARD,
36        label: "sidebar-nav-item-dashboard",
37        icon: "DashboardIcon",
38      },
39      {
40        href: ROUTES.SHOPS,
41        label: "sidebar-nav-item-shops",
42        icon: "ShopIcon",
43      },
44      {
45        href: ROUTES.ADMIN_MY_SHOPS,
46        label: "sidebar-nav-item-my-shops",
47        icon: "MyShopIcon",
```

```
48     },
49     {
50         href: ROUTES.PRODUCTS,
51         label: "sidebar-nav-item-products",
52         icon: "ProductsIcon",
53     },
54     {
55         href: ROUTES.CATEGORIES,
56         label: "sidebar-nav-item-categories",
57         icon: "CategoriesIcon",
58     },
59     {
60         href: ROUTES.TAGS,
61         label: "sidebar-nav-item-tags",
62         icon: "TagIcon",
63     },
64
65     {
66 href: ROUTES.ORDER_STATUS,
67         label: "sidebar-nav-item-order-status",
68         icon: "OrdersStatusIcon",
69     },
70     {
71         href: ROUTES.ORDER_STATUS,
72         label: "sidebar-nav-item-order-status",
73         icon: "OrdersStatusIcon",
74     },
75     {
76         href: ROUTES.USERS,
77         label: "sidebar-nav-item-users",
78         icon: "UsersIcon",
79     },
80
81
82     {
83         href: ROUTES.WITHDRAWS,
84         label: "sidebar-nav-item-withdraws",
85         icon: "WithdrawIcon",
86     },
87
88
89 {
90     href: ROUTES.SETTINGS,
91     label: "sidebar-nav-item-settings",
92     icon: "SettingsIcon",
93 },
94 ],
95 shop: [
96     {
```

```
97     href: (shop: string) => `${ROUTES.DASHBOARD}${shop}`,
98     label: "sidebar-nav-item-dashboard",
99     icon: "DashboardIcon",
100    permissions: adminOwnerAndStaffOnly,
101  },
102  {
103    href: (shop: string) => `/${shop}${ROUTES.PRODUCTS}`,
104    label: "sidebar-nav-item-products",
105    icon: "ProductsIcon",
106    permissions: adminOwnerAndStaffOnly,
107  },
108  {
109    href: (shop: string) => `/${shop}${ROUTES.ORDERD}`,
110    label: "sidebar-nav-item-orders",
111    icon: "OrdersIcon",
112    permissions: adminOwnerAndStaffOnly,
113  },
114 {
115   href: (shop: string) => `/${shop}${ROUTES.STAFFS}`,
116   label: "sidebar-nav-item-staffs",
117   icon: "UsersIcon",
118   permissions: adminAndOwnerOnly,
119 },
120 {
121   href: (shop: string) => `/${shop}${ROUTES.WITHDRAWS}`,
122   label: "sidebar-nav-item-withdraws",
123   icon: "AttributeIcon",
124   permissions: adminAndOwnerOnly,
125 },
126 ],
127 },
128 product: {
129   placeholder: "/product-placeholder.svg",
130 },
131 avatar: {
132   placeholder: "/avatar-placeholder.svg",
133 },
134 };
```

Listing A.7: Trend Sparkles Gallery Implementation Code

References

- [1] Niko Hezwono Putra et al. The effect of service quality, website quality, price, and brand image on consumer satisfaction impact on consumer loyalty in olx online stores. In *4th Padang International Conference on Education, Economics, Business and Accounting (PICEEBA-2 2019)*, pages 774–781. Atlantis Press, 2020. Cited on p. 10.
- [2] Husin Husin, Wahyu Dewanto, Prawidya Destarianto, and Ghanesya Murti. Analysis of online advertising content on purchase decision in marketplace. In *2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE)*, pages 59–64. IEEE, 2019. Cited on p. 10.
- [3] Yousaf Ali, Bilal Mehmood, Muhammad Huzaifa, Umair Yasir, and Amin Ullah Khan. Development of a new hybrid multi criteria decision-making method for a car selection scenario. *Facta Universitatis. Series: Mechanical Engineering*, 18(3):357–373, 2020. Cited on p. 10.
- [4] Naved Hamid and Faizan Khalid. Entrepreneurship and innovation in the digital economy. *The Lahore Journal of Economics*, 21:273, 2016. Cited on p. 11.
- [5] Ruey-Jer Jean, Daekwan Kim, Kevin Zheng Zhou, S Tamer Cavusgil, et al. E-platform use and exporting in the context of alibaba: A signaling theory perspective. *Journal of International Business Studies*, 52(8):1501–1528, 2021. Cited on p. 11.
- [6] Jianxin Shi and Yongxiang Wu. B2b, b2c and c2c: Should they be treated equally in china. In *2006 Canadian Conference on Electrical and Computer Engineering*, pages 498–501. IEEE, 2006. Cited on p. 11.
- [7] Waheed Ahmed. An investigation into customer preferences for online shopping in south asia (pakistan): A case study of daraz. pk. 2020. Cited on p. 11.
- [8] Waqar Ahmad Khan, Shahid Yousaf, Natash Ali Mian, and Zubair Nawaz. E-commerce in pakistan: Growth potentials and e-payment solutions. In *2013 11th International Conference on Frontiers of Information Technology*, pages 247–252. IEEE, 2013. Cited on p. 11.
- [9] Md Abdullah Al Imran. A study on amazon: information systems, business strategies and e-crm. 2014. Cited on p. 12.
- [10] CK Sunitha and Edwin Gnanadhas. Online shopping-an overview. *B-DIGEST*, 6:16–22, 2014. Cited on p. 12.

- [11] Jane Gould and Thomas F Golob. Shopping without travel or travel without shopping? an investigation of electronic home shopping. *Transport reviews*, 17(4):355–376, 1997. Cited on p. [12](#).