Supervisor

Mr. Ali Irfan

Submitted By

Hijab Sharif / 01-135202-031
Talha Ayub / 01-135202-098

EGold Haven

A C2C / B2C E-commerce Web Application for Gold Jewelry Trading

Bachelors of Science in Information Technology

Department of Computer Science
Bahria University, Islamabad

June 2024

Certificate

The work in the document "EGold Haven" is written by Talha Ayub and Hijab Sharif as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Information Technology.

Approved by...:

Supervisor: Mr. Ali Irfan (Assistant Professor)

Internal Examiner:

External Examiner(Title):

Project Coordinator:

Head of the Department: Dr. Arif ur Rahman

June 23^{rd},2024

Abstract

In the dynamic landscape of e-commerce, a notable trend is the rise of specialized online platforms tailored to the distinct requirements of various industries. Against this backdrop, "EGold Haven" stands out as a trailblazing web application exclusively dedicated to the gold jewelry trade. This innovative platform is designed to empower sellers, granting them the autonomy to set prices independently, thereby eliminating intermediary fees. Simultaneously, it offers buyers the unique advantage of accessing gold at prices below the prevailing market rates. At the core of "EGold Haven" is a meticulous certification process mandated for all listed gold items. Accredited jewelers play a pivotal role in ensuring the authenticity and quality of the offerings. As an incentive, these jewelers earn commissions for their essential services, fostering a collaborative ecosystem within the platform. One distinctive feature of "EGold Haven" is the provision for accredited jewelers to curate personalized store pages. This dedicated space allows jewelers to showcase their unique array of gold jewelry, creating a virtual storefront that reflects their craftsmanship and style. By providing jewelers with the tools to manage their individual store pages, the platform not only promotes transparency but also enriches the overall user experience. Buyers can explore a diverse range of offerings from different jewelers, establishing a more engaging and personalized connection within the gold jewelry trade.

Acknowledgments

We extend our sincere gratitude to Allah Almighty for providing us with the opportunity to embark on this project. We are confident that His help and guidance will pave the way for the successful completion of this endeavor. We are also deeply thankful to our esteemed supervisor, Mr. Ali Irfan, for his unwavering support, insightful vision, and continuous encouragement throughout the course of this project. His motivation and belief in our capabilities have been instrumental in inspiring us to persevere and deliver our best work. Working under his guidance is truly an honor, and we are grateful for the valuable mentorship he provides. Last but not least, we express our heartfelt appreciation to our parents for their prayers and unwavering support. Their blessings and encouragement have been a source of strength, propelling us forward in this project and beyond.

Hijab Sharif & Talha Ayub
Islamabad, Pakistan

June, 2024

"The only place where success comes before work is in the dictionary"

Vidal Sassoon

Table of Contents

List of Figures

List of Tables

Acronyms and Abbreviations

API	Application Programming Interface
ERD	Entity Relationship Diagram
SSO	Single Sign On
AI	Artificial Intelligence
MERN	Mongodb, Expressjs, Reactjs & Nodejs
ORM	Object-Relational Mapping
MVC	Model - View - Controller
UI	User Interface
JSON	Java Script Object Notation
JWT	JSON Web Tokens

Chapter 1

Introduction

1.1 Project Background

EGold Haven is a specialized online marketplace (Web Application) exclusively dedicated to gold jewelry trading. This platform will empower sellers by enabling them to set prices for their gold items independently, eliminating any intermediary fees. Buyers will be able to access gold at reduced market prices, enjoying cost savings.

To ensure the authenticity and quality of listed gold items, our platform will mandate certification by accredited jewelers who register on the site, earning commissions for their services.

This innovative solution not only streamlines the gold jewelry trade but also enhances trust and transparency within the industry, offering a unique value proposition for both sellers and buyers.

1.2 Objective

To develop a transparent and cost-effective gold jewelry marketplace web application, "EGold Haven," empowering sellers to set prices independently, reducing intermediary fees, and ensuring authenticity through certified jewelers, fostering trust and efficiency in the industry.

1.3 Problem Description

1.3.1 Price Disparities

In the current gold jewelry market, sellers receive reduced rates when selling gold items, while buyers pay full market prices for the same items, creating an inequitable trading

environment.

“EGold Haven” will allow the sellers to sell without price cut on their own terms and buyers would be able to buy gold at lower prices than the market.

1.3.2 Lack of Transparency

The gold jewelry market often lacks transparency, leaving buyers uncertain about the true value and authenticity of items.

The platform will enhance transparency through jeweler verification and certification.

1.3.3 Limited Market Exposure

Small-scale jewelers and sellers struggle to reach a broader audience, reducing their market reach.

The platform will provide them with a digital marketplace for increased exposure.

1.3.4 Limited Access to Market Data

Lack of real-time market data hinders informed decision-making for buyers and sellers alike.

The platform will provide data insights to assist buyers and sellers in making informed choices.

1.4 Methodology

1.4.1 Agile Methodology

Agile approach [1] will be used to develop “EGold Haven”. This selection aligns with the project’s dynamic nature, characterized by evolving requirements and a crucial need for flexibility.

This will allow us to complete the proposed project in manageable iterations where after each iteration we can seek guidance and approval of out supervisor before moving ahead.

1.4.2 Web Application Working

The development of ”EGold Haven” will prioritize user-centered design and robust functionality.

E-commerce integration will enable sellers to list gold items without price cuts, while buyers benefit from competitive prices.

A transparent jeweler verification system will instill trust, and the platform will serve as a digital marketplace to boost market exposure for small-scale jewelers.

Additionally, mobile responsiveness and a user-friendly interface will ensure a seamless user experience, making gold trading accessible and equitable for all users.

1.5 Project Scope

1.5.1 Inclusion

1.5.1.1 Web Application Development

The core focus of the project will be the design and development of a fully functional web application for gold jewelry trading. This includes frontend and backend development, user authentication, and database management.

1.5.1.2 User Authentication

Implementing a robust user authentication system to ensure secure access to the platform for sellers, buyers, certified jewelers, partners, and community members.

1.5.1.3 Database Management

Designing and managing the database to store information about gold items, user profiles, certifications, transaction history, partner details, and community forum data.

1.5.1.4 Jeweler Verification and Certification

Implementing a process for jewelers, partners, and community members to register, verify their credentials, and certify the authenticity and quality of listed gold items.

1.5.1.5 Jeweler-Specific Store Pages

Introducing a distinctive feature enabling accredited jewelers to individually manage personalized store pages. This functionality empowers jewelers to showcase their finalized gold products, providing a dedicated virtual storefront within the platform. Through these store pages, jewelers can highlight their craftsmanship, display unique offerings, and establish a distinct online presence. This feature not only enhances transparency but also enriches the overall user experience, allowing buyers to explore and engage with a diverse range of gold jewelry from various accredited jewelers.

1.5.1.6 Marketplace Features

Building features for sellers to list gold items, set prices, and manage their listings, for buyers to browse, search, and purchase gold jewelry, for partners to participate in the platform, and for the community to engage in discussions.

1.5.1.7 Community Forum

Developing a community forum where users can discuss gold-related topics, share insights, and engage in discussions, fostering a sense of community within the platform.

1.5.2 Exclusion

1.5.2.1 Mobile Application Development

The project will focus exclusively on web application development. The creation of dedicated mobile applications for iOS or Android platforms is not within the scope.

1.5.2.2 Physical Authentication Devices

Development of physical authentication devices or hardware, such as fingerprint scanners or smart cards, for user verification or gold item certification is beyond the project's scope.

1.5.2.3 Legal and Regulatory Compliance

While the platform will adhere to general legal and ethical standards, the project will not involve in-depth legal consultations or compliance with specific regional or global regulations related to gold trading. Compliance will be the responsibility of users.

1.5.2.4 Internationalization and Localization

The project will not include extensive internationalization and localization efforts for supporting multiple languages or regions. It will primarily focus on a single language and region.

1.5.2.5 Advanced Machine Learning or Artificial Intelligence

Complex machine learning models or artificial intelligence algorithms for predicting gold prices or user behavior will not be developed as part of this project.

1.5.2.6 Physical Gold Storage or Transportation Solutions

Any aspects related to the physical storage, transportation, or insurance of gold items will not be addressed by the web application.

1.5.2.7 Offline Functionality

The project will not include offline functionality, and users will require an internet connection to access and use the platform.

1.5.2.8 Third-Party Partnerships

While the platform may facilitate partnerships between sellers, buyers, and certified jewelers, establishing third-party partnerships with external organizations is not within the project's scope.

1.5.3 Assumptions

1.5.3.1 Internet Connectivity

We assume that users of the platform will have reliable internet access for using the web application.

1.6 Feasibility Study

1.6.1 Risks Involved

1.6.1.1 Scope Creep

As the project progresses, there may be a temptation to add more features or functionalities than initially planned. This can lead to scope creep and delay the project's completion.

1.6.1.2 Technical Challenges

Depending on the complexity of your platform, we may encounter technical challenges, such as integrating e-commerce features, ensuring data security, or optimizing for scalability.

1.6.1.3 Data Security

Handling sensitive information like user data and financial transactions requires robust security measures. Failing to address security adequately can lead to data breaches and legal issues.

1.6.1.4 Resource Constraints

Limited time, budget, or technical resources can impact the project's progress and quality. It will be essential to manage these constraints effectively.

1.6.1.5 Technical Debt

Rushing through development to meet deadlines may lead to technical debt which leads to poorly written code or architectural shortcuts that can impede future development.

1.6.2 Resource Requirements

1.6.2.1 Hardware and Software

Computers or laptops with the necessary software tools, such as code editors, development frameworks, and design software.

1.6.2.2 Technological Resources

Web Development Tools: Programming languages (e.g., HTML, CSS, JavaScript), development frameworks (MERN Stack), and database systems (Mongodb).

1.6.2.3 Data Resources

For simulating real gold listings and transactions, we will need sample data for testing and development.

1.6.2.4 Time and Commitment

Sufficient time for research, development, testing, and project management. Plan for regular meetings and progress updates.

1.6.2.5 Educational Resources

Access to textbooks, online courses, tutorials, or documentation to acquire or improve skills as needed.

1.7 Solution Application Areas

The "EGold Haven" web application, designed to address issues in the gold jewelry trading market will have the following application areas:

1.7.1 E-commerce and Retail

The platform serves as an innovative e-commerce solution for the gold jewelry industry, enabling sellers to reach a broader audience while providing buyers with transparent and competitive pricing.

1.7.2 Small-Scale Jewelers

"EGold Haven" offers small-scale jewelers and independent sellers an online marketplace to showcase their gold items, expanding their market presence and customer base.

1.7.3 Buyers and Investors

Buyers and investors interested in purchasing gold jewelry at competitive rates benefit from the platform's transparent pricing and certified product authenticity.

1.7.4 Gold Industry Professionals

Accredited jewelers and industry professionals can participate as certifiers, earning commissions for verifying the authenticity and quality of gold items listed on the platform.

1.7.5 Marketplace Startups

Entrepreneurs and startups interested in entering the online marketplace sector

1.7.6 Gold Jewelry Enthusiasts

Individuals passionate about gold jewelry can explore a diverse range of items on the platform, benefiting from competitive prices and certified quality.

1.8 Tools and Technology

1.8.1 Hardware

Personal Computer or Laptop for development.

1.8.2 Software

1.8.2.1 MERN Stack

For front-end development, React.js, a leading JavaScript library, will be the foundation of our user interface, offering flexibility and component-based development.

Alongside React, we will utilize HTML and CSS to structure and style web pages, ensuring a rich user experience.

For back-end development, Node.js takes the lead, providing a non-blocking I/O platform that seamlessly integrates with JavaScript. To simplify API development and efficient routing, we will employ Express.js, a minimalist web application framework.

MongoDB, a No-SQL database, will serve as our data storage solution, while Mongoose, a Node.js ORM, will streamline interactions with the database.

1.9 Expertise of the Team Members

We possess a solid foundation in ASP.NET framework and have successfully collaborated on projects such as a research portal and an ongoing cricket league application. These experiences have honed our skills in web development, database management, and the application of data structures within the ASP.NET environment, aligning with the academic coursework completed as part of our degree program.

However, recognizing the evolving dynamics of today's technological landscape, we acknowledge the imperative to transition towards more modern technologies. In this context, we have opted to leverage the MERN stack for the development of the "EGold Haven" gold jewelry trading platform. While our prior experiences have been centered around ASP.NET, we view this shift as an exciting opportunity to embrace the agility and versatility offered by the MERN stack.

This strategic decision is motivated by the MERN stack's capability to streamline development processes, enhance scalability, and provide a more responsive user experience. Despite being uncharted territory for our team, we are genuinely enthusiastic about the prospect of mastering this technology and leveraging its advantages for the successful implementation of the "EGold Haven" project.

1.10 Milestones

1.10.1 Project Initiation

- Define Project objectives scope and requirements.
- Formulate a project plan, including timelines and resource allocation.
- Establish group roles and responsibilities.
- Gather detailed project requirements.

1.10.2 Front-End Development

- Set up the development environment for front-end.
- Create wireframes and prototypes.

- Design the user interface for the web application.
- Implement front-end for the application.
- Ensure responsiveness for different devices.

1.10.3 Back-End Development

- Set up the development environment for back-end.
- Develop APIs for user registration, authentication and profile management.
- Implement features for item listings, certifications and partner management.
- Develop all features included in the scope of the project.

1.10.4 Database Integration

- Set up and configure the database system to be used.
- Create database tables or collections for user data, gold item data, certifications, and transaction records.
- Implement data validations and storage mechanisms

1.10.5 Testing and Quality Assurance

- Unit testing for individual components.
- Perform integration testing to ensure seamless functionality.
- Usability testing to evaluate user experience.
- Load testing for the web application.
- Resolving bugs and issues in the application.

Chapter 2

Literature Review

In the ever-evolving landscape of e-commerce [2], the emergence of specialized online platforms has been reshaping how consumers engage in buying and selling various products.

At present, the realm of online gold jewelry [3] trade predominantly encompasses platforms offered by specific gold businesses and companies [4], such as “ARY Jewelers”, “Raza Jewelers”, and “Puregold”, among others.

These entities serve as both manufacturers and retailers of their own products within their respective online spaces.

However, the introduction of a comprehensive and dedicated online marketplace for gold jewelry is a relatively uncharted territory.

2.1 Why Specialized Platform

Specialized e-commerce platforms have witnessed remarkable growth and success across various industries.

A prime example is ”PakWheels,” [5] a dedicated online marketplace for the buying and selling of automobiles. PakWheels’ prominence demonstrates that when individuals contemplate purchasing or selling vehicles, this specialized platform is often their first choice.

It has become synonymous with the automobile trade, and its success serves as an inspiration for ventures like ”EGold Haven.”

2.2 Limitations of General Platforms

In contrast to specialized platforms, generic online marketplaces like OLX [6] cater to a wide range of product categories, from electronics to clothing and vehicles.

While OLX provides a broad platform for diverse trade, it may not be the top-of-mind choice for individuals interested in specific products, such as gold jewelry. The potential limitations of generic platforms in meeting the unique requirements and expectations of gold jewelry trade underscore the need for a specialized solution.

In this regard, we conducted a survey to get a better idea of what the general public or the users of our platform would choose when deciding to sell their gold jewelry. Below are the figures for the survey results:

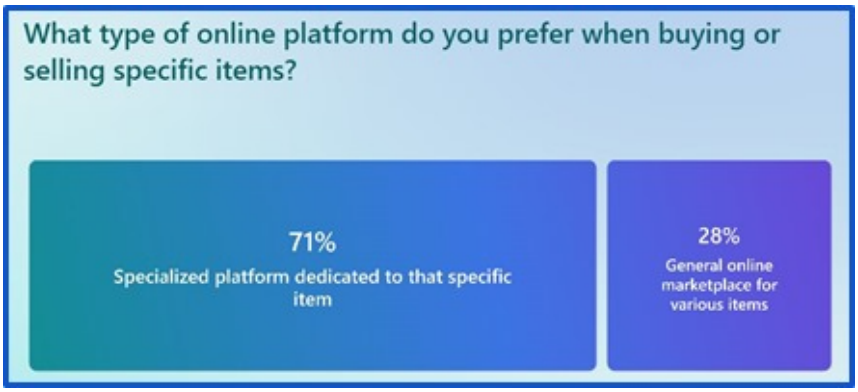

Figure 2.1: Type of platform for buying/selling specific items

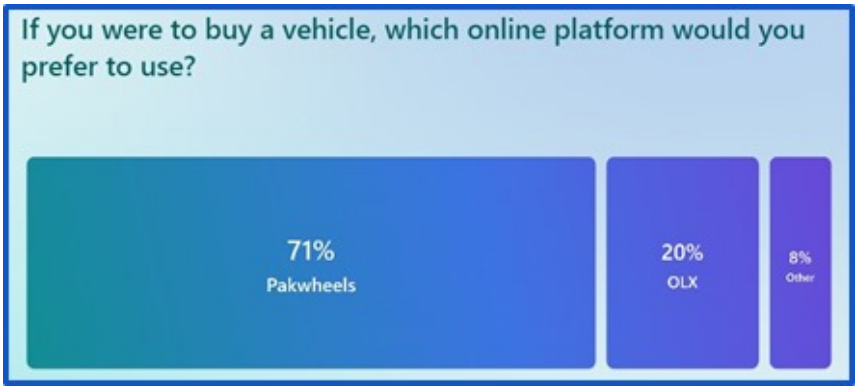

Figure 2.2: Type of platform for buying/selling vehicles

2.3 The Uncharted Territory of Gold Jewelry Trade

Currently, online gold jewelry trade primarily involves individual businesses promoting and selling their own products. The absence of a comprehensive and specialized marketplace like "EGold Haven" is notable.

This venture aims to bridge this gap by offering a dedicated platform that not only aggregates gold jewelry listings but also ensures certification, authenticity, and a streamlined experience for buyers and sellers alike.

Drawing inspiration from the success of platforms like PakWheels, "EGold Haven" seeks to establish itself as the go-to platform for gold jewelry trade, a market currently underserved by comprehensive and dedicated online marketplaces.

As consumers increasingly seek tailored and specialized experiences [7] in their online transactions, the introduction of "EGold Haven" reflects an innovative approach to meeting the unique demands of gold jewelry trade, filling a void in the current e-commerce landscape [8].

This platform aims to redefine how gold jewelry is bought and sold online, creating a niche where none previously existed.

Chapter 3

Requirements Specifications

3.1 Existing System

There are no current existing systems that match our idea but there are multiple other systems that can provide us with some ideas to implement it on out system. There is one that we have talked about in chapter 2 under the section of related work.

3.2 Proposed System

Our proposed system "EGold Haven" is a unique web platform dedicated to simplifying the process of trading gold jewelry. This platform empowers sellers to independently set prices without extra fees, while buyers enjoy the advantage of purchasing gold at attractive rates. What sets "EGold Haven" apart is its commitment to authenticity – every listed gold item undergoes a rigorous certification process, ensuring top-notch quality.

3.3 Requirement Specification

In requirement specification, we must specify functional and non-functional requirement, of our system. Both are specified below.

3.3.1 Software Requirements

The software requirements for this project are:

- React JS
- Express JS

- Node JS
- MongoDB
- Visual Studio Code

3.3.2 Hardware Requirements

The hardware requirements for this project are:

- Personal Laptop or Computer
- Fast Internet Connection

3.3.3 Functional Requirements

Functional requirements specify what inputs are given to the system and what corresponding output is produced and how it behaves on input. Following are the functional requirements of our system:

- Users will be able to register their account on the platform.
- Users will be able to login in their registered account on the platform.
- Users will be able to edit their profile information on the platform.
- Users will be able to reset their password on the platform.
- Users will be able to place, edit and delete ads on the platform.
- Users will be able to browse and search for listings.
- Users will be able to view each listings details and set them as favorites.
- Users will be able to access the forum.
- Users will be able to request for jeweler status upgrade for their accounts.
- Users will be able to chat with other users or jewelers in correspondence to ads.
- Users will receive notifications on different related actions on the platform.
- Users will be able to delete their accounts.
- Admins will be able to access dashboard where they can view the portal statistics.
- Admins will be able to block and unblock user accounts.
- Jewelers will be able to provide gold certification services to the users.
- Jewelers will be able to manage their page by adding, editing and deleting products.

- Jewelers will be able to request for commission rate change from the admins.

3.3.4 Non-Functional Requirements

Non-functional requirements are critical aspects of a system that go beyond its functional features, addressing how the system should perform, rather than what it should do.

Some of the non-functional requirements of our system are:

3.3.4.1 Scalability

The system should be designed to scale horizontally to accommodate an increasing number of users and transactions.

3.3.4.2 Security

All user data, including personal information and transaction details, should be encrypted during transmission and storage.

3.3.4.3 Performance

The platform must be designed to handle a significant number of concurrent users, ensuring that even during peak times, such as special promotions or events, users can access and navigate the platform without notable delays. Additionally, critical transactions, like uploading new listings or processing payments, should be optimized to deliver quick responses, enhancing user satisfaction.

3.3.4.4 Efficiency

Efficient algorithms and data structures will be employed to enhance the speed of essential operations, such as search queries or item listing updates. The goal is to ensure that the platform operates with efficiency, enabling swift interactions between users, sellers, and jewelers.

3.4 Use Cases

The use case diagrams depicts the interactions between actors and system components to accomplish various functionalities. Egold Haven system comprises of 4 primary actors including guest, users (buyers / sellers), jewelers which are the extension of users and the Admin. Each actor engages with the system to perform his required actions.

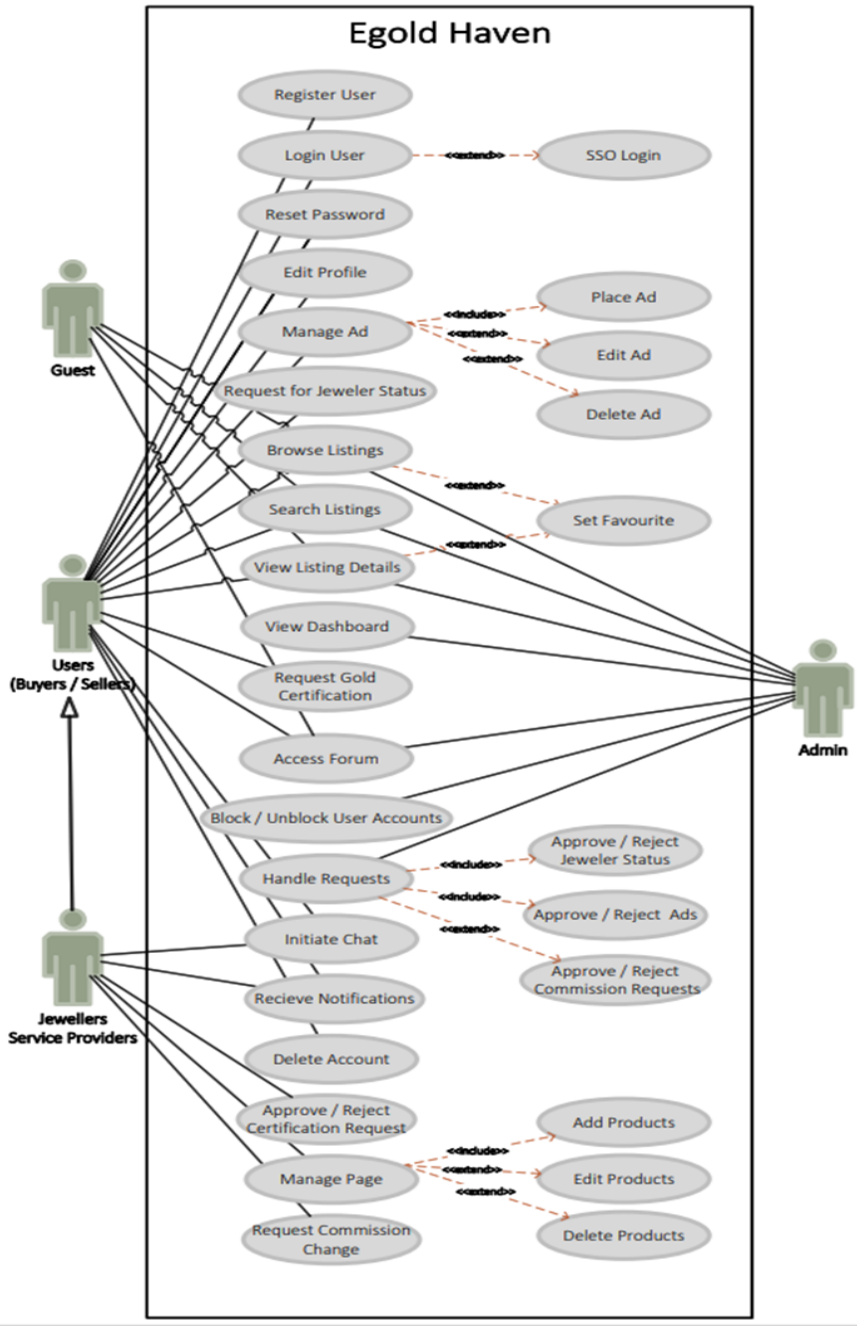

Figure 3.1: Use Case Diagram

3.4.1 Register User

Table 3.1: Register User Use Case

USE CASE NAME	Register User	
USE CASE ID	EGH-01	
PRIORITY	High	
PRIMARY ACTOR	User	
DESCRIPTION	This use case describes the process of registering a new user on the platform. Users must register to be able to access the platform.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	User comes on the Sign-Up page.
	2	User fills the required information (username, email, password, phone no and address) and registers.
	3	User lands on Login page.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	Field validations failed and error occurs (empty fields, regex test failed, different passwords etc.).
	2	Account already exists and registration wont process.
PRE-CONDITION	Sign-up page is loaded completely.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. User visits Sign-up page.	3. System performs checks and adds the user data to the database.
	2. User enters the information.	4. System redirects the browser to Login Page.
POST-CONDITION	User will be at the Login Page of the portal.	

The above table comprises of the description of the Register User Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.2 Login User

Table 3.2: Login User Use Case

USE CASE NAME	Login User	
USE CASE ID	EGH-02	
PRIORITY	High	
PRIMARY ACTOR	User	
DESCRIPTION	This use case encompasses the Login process of user which is required to access the platform. All registered users must go through this functionality.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	User comes on the Login Page.
	2	User enters his credentials or uses SSO to Login.
	3	User presses enter or clicks login button.
	4	User lands on the Home Page.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	Login fails due to Incorrect Credentials such as wrong username or password.
	2	User account does not exist and an error is thrown.
PRE-CONDITION	Login Page is loaded completely.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. User visits Login page.	3. System retrieves user data from database and performs user authentication.
	2. User enters his credentials.	4. System redirects the browser to home page and starts a user session.
POST-CONDITION	User is at the Home page of the portal.	

The above table comprises of the description of the Login User Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.3 Reset Password

Table 3.3: Reset Password Use Case

USE CASE NAME	Reset Password	
USE CASE ID	EGH-03	
PRIORITY	High	
PRIMARY ACTOR	User	
DESCRIPTION	This use case describes the functionality of resetting password in case user wants to change it or has forgotten the previous password.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	User will go to forget password page.
	2	User enters his username.
	3	User receives a password reset token on his email registered with the username.
	4	User enters the token on forgot password page.
	5	User enters the new password.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	Email is Invalid and an error is thrown.
	2	Token has expired and user is prompted to get new token.
PRE-CONDITION	Password Reset page is loaded completely.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. User comes on the password reset page	3. System validates email and sends a limited time token.
	2. User enters his email and receives token which is used to reset password.	4. System verifies the correctness of token and allows user to enter new password which the system updates in the database.
POST-CONDITION	User is at the login page with updated password.	

The above table comprises of the description of the Reset Password Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.4 Edit Profile

Table 3.4: Edit Profile Use Case

USE CASE NAME	Edit Profile	
USE CASE ID	EGH-04	
PRIORITY	Low	
PRIMARY ACTOR	User	
DESCRIPTION	This use case describes the process of editing user profile information. Every user who needs to update any of his profile information needs to go through this functionality.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	User clicks on his profile icon.
	2	User opens the edit profile page.
	3	User makes changes to the information fields (e.g., name, address, display picture etc.) and submits.
	4	User lands on the profile page.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	New information does not follow proper conventions such as naming convention or email format etc.
	2	New email that is entered cannot be validated.
PRE-CONDITION	User is on the edit profile page.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. User visits the edit profile page.	3. System validates the information and updates it in the database.
	2. User makes the desired changes to information.	4. System redirects the user to profile page.
POST-CONDITION	User is at the profile page with updated information.	

The above table comprises of the description of the Edit Profile Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.5 Place Ad

Table 3.5: Place Ad Use Case

USE CASE NAME	Place Ad	
USE CASE ID	EGH-05	
PRIORITY	High	
PRIMARY ACTOR	User (Seller)	
DESCRIPTION	The use case describes the process of how a user would place an advertisement for his gold. Every user who wants to sell on this platform has to go through this functionality.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	User goes to the place ad page.
	2	User enters the required information for his ad such as title, price, description, category and images etc.
	3	User is prompted to wait for ad approval and notification.
	4	User lands on My ads page.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	User did not enter complete information and an error is thrown.
	2	User enters false information and the ad is disapproved.
PRE-CONDITION	User is logged in and Place ad page is loaded completely.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. User visits the place ad page.	3. System validates the ad's information for completeness and proper conventions and sends it for admin approval.
	2. User enters ad's information and places the ad for approval.	4. System adds the approved ad to the database and notifies the user of its approval.
POST-CONDITION	User is on the My ads page.	

The above table comprises of the description of the Place Ad Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.6 Edit Ad

Table 3.6: Edit Ad Use Case

USE CASE NAME	Edit Ad	
USE CASE ID	EGH-06	
PRIORITY	Medium	
PRIMARY ACTOR	User	
DESCRIPTION	This use case describes the process for editing an ad placed by the user. All users who have placed an ad and want to make some changes in it must go through this functionality.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	User clicks on his profile icon.
	2	User goes to My ads page.
	3	User selects the specific ad to edit.
	4	User makes the desired changes and submits.
	5	User lands on the My ads page.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	User forgets to fill a field he erased and an error is thrown
PRE-CONDITION	User is logged in and Edit page is completely loaded.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. User goes to the edit ad page.	3. System performs basic validations on new information and sends the ad for admin verification.
	2. User makes the desired changes and submits the ad.	4. System updates the approved ad in the database and notifies the user.
POST-CONDITION	User is at the My ads page.	

The above table comprises of the description of the Edit Ad Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.7 Delete Ad

Table 3.7: Delete Ad Use Case

USE CASE NAME	Delete Ad	
USE CASE ID	EGH-07	
PRIORITY	Low	
PRIMARY ACTOR	User	
DESCRIPTION	This use case describes the process for the deletion of an ad by the user. Every user who wishes to delete his own ad must go through this functionality.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	User opens the specified ad from My ads page
	2	User chooses the delete ad option and is prompted for confirmation.
	3	User lands on the My ad page.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	The ad could not be deleted due to server issues on database connection and an error is thrown.
PRE-CONDITION	User is logged in and User is on the specified ad from My ads page.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. User opens the specified ad from My ads page.	3. System asks for confirmation and deletes the ad from the database.
	2. User selects the delete ad option.	4. System redirects the user to My ads page.
POST-CONDITION	User is at the My ads page.	

The above table comprises of the description of the Delete Ad Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.8 Request for Jeweler Status

Table 3.8: Request for Jeweler Status Use Case

USE CASE NAME	Request for Jeweler Status	
USE CASE ID	EGH-08	
PRIORITY	High	
PRIMARY ACTOR	User	
DESCRIPTION	This use case describes the process for user to request for status update from user to jeweler. Every Jeweler who registered as a user first and wants to register as a certified jeweler must go through this functionality.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	User goes to his profile.
	2	User selects the option to request for jeweler status.
	3	User enters the required information (cnic, phone no, cnic images and store images) and submit the request for approval.
	4	User lands back on the profile page and gets prompted to wait for approval notification.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	User enter incomplete information and an error occurs.
PRE-CONDITION	User is logged in and User is on the profile page.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. User requests for status update.	3. System validates the entered information and sends the request for admin approval.
	2. User enters the required information (cnic, phone no, cnic images and store images).	4. System redirects the user to notifications and prompts to wait for approval notification.
POST-CONDITION	User is at the notification page.	

The above table comprises of the description of the Request for Jeweler Status Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.9 Browse Listings

Table 3.9: Browse Listings Use Case

USE CASE NAME	Browse Listings	
USE CASE ID	EGH-09	
PRIORITY	High	
PRIMARY ACTOR	User and Guest	
DESCRIPTION	This use case describes the process for browsing the existing listings of other sellers on the platform. Every user who has to view ads or purchase items has to go through this functionality.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	User is on the listings page.
	2	User can scroll to browse through listings.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	Listings did not load due to server issue.
PRE-CONDITION	User is logged in.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. User goes to the home page.	3. System dynamically loads the listings for the user to browse.
	2. User scrolls through listings.	
POST-CONDITION	User is on the listings page and can browse all approved listings.	

The above table comprises of the description of the Browse Listings Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.10 Search Listings

Table 3.10: Search Listings Use Case

USE CASE NAME	Search Listings	
USE CASE ID	EGH-10	
PRIORITY	Medium	
PRIMARY ACTOR	User and Guest	
DESCRIPTION	This use case describes the process through which user can search specific items through all the approved listings on the platform.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	User goes to the listings page.
	2	User searches for a specific item or chooses a specific category or specification.
	3	User is shown the updated view with the user specific listing or listings.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	User enters wrong spellings or words and no results are found.
	2	User searches using a keyword while in a specific category which shows no result.
PRE-CONDITION	User is logged in and User is on the listings page.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. User goes to the listings page.	3. System searches and retrieves the listings from database based on user search or selected category.
	2. User either searches using keyword or selects a specific category to browse.	4. System refreshes the view and dynamically loads the retrieved listings on the listings page.
POST-CONDITION	User can browse listings based on preference.	

The above table comprises of the description of the Search Listings Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.11 View Listing Details

Table 3.11: View Listing Details Use Case

USE CASE NAME	View Listing Details	
USE CASE ID	EGH-11	
PRIORITY	Medium	
PRIMARY ACTOR	User and Guest	
DESCRIPTION	This use case describes the process through which user can specifically view the details of a listing. Every user who wants to view a specific listing's detail while browsing through listings must go through this functionality	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	User goes to the listings page.
	2	User clicks on a specific listing.
	3	User lands on a page with detailed listing information (title, price, description, category and images).
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	The specific listing had just been deleted or is unavailable.
PRE-CONDITION	User is logged in and User is on the listings page.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. User goes to the listings page.	3. System responds to this request and retrieves the detailed information about that specific listing.
	2. User clicks on a specific listing to view listing details.	4. System redirects to a page and dynamically loads it with the retrieved data.
POST-CONDITION	User is on the View listing details page.	

The above table comprises of the description of the View Listing Details Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.12 Set Favorites

Table 3.12: Set Favorites Use Case

USE CASE NAME	Set Favorites	
USE CASE ID	EGH-12	
PRIORITY	Low	
PRIMARY ACTOR	User	
DESCRIPTION	This use case describes the process for setting favorite listings by the user. These listings can then further be viewed by going to the My favorites page.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	User goes to listings page and browse listings.
	2	User likes a specific listing.
	3	User can go to My favorites page to see favorite listings.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	User had already added that specific listing to favorites.
PRE-CONDITION	User is logged in and User is on the listings page.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. User goes to the listings page.	3. System responds to this action by adding the listing to that user's favorite listings in the database
	2. User likes a specific listing to add it to favorite listings.	
POST-CONDITION	User continues to browse and can see the liked listing in profiles favorite listings.	

The above table comprises of the description of the Set Favourites Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.13 View Dashboard

Table 3.13: View Dashboard Use Case

USE CASE NAME	View Dashboard	
USE CASE ID	EGH-13	
PRIORITY	High	
PRIMARY ACTOR	Admin	
DESCRIPTION	This use case describes the process by which the admin can view the statistics of the web portal including information such as total ads placed, ads sold and pending requests etc.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	Admin selects the view statistics option from his profile menu.
	2	Admin is presented with all the options for different stats of the portal
	3	Admin selects the option and that specifics detail (live ads, expired ads, deleted ads, reason, registered users, registered jewelers etc.) is shown to the admin.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	No stats have been currently recorded for the particular detail.
PRE-CONDITION	Admin is logged in.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. Admin selects the view statistics option.	3. System requests the selected items information from the database.
	2. Admin selects the particular detail they wish to see	4. System populates the view with the fetched information for the user to view.
POST-CONDITION	Admin is presented with the details of the particular item that they selected.	

The above table comprises of the description of the View Dashboard Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.14 Request Gold Certification

Table 3.14: Request Gold Certification Use Case

USE CASE NAME	Request Gold Certification	
USE CASE ID	EGH-14	
PRIORITY	High	
PRIMARY ACTOR	User	
DESCRIPTION	This use case describes the process by which the user can request a jeweler on the platform to authenticate his gold. All sellers must go through this functionality to authenticate the gold for the buyer.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	Seller opens up My Ads tab from his profile menu.
	2	Seller selects the specific add for which he wants to request gold certification.
	3	Seller selects a jeweler and a request is sent to him
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	The jeweler does not accept the request after viewing the ad and the seller is notified.
PRE-CONDITION	User is logged in and Seller has a live ad that needs authentication.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. Sellers opens up his ad.	3. System tracks the request and keeps it in the database until dealt to.
	2. Seller requests the jeweler for gold authentication.	4. System then maintains the chat between seller and jeweler and records it in the database.
POST-CONDITION	A request is sent to the selected jeweler and the seller is prompted to wait.	

The above table comprises of the description of the Request Gold Certification Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.15 Access Forum

Table 3.15: Access Forum Use Case

USE CASE NAME	Access Forum	
USE CASE ID	EGH-15	
PRIORITY	High	
PRIMARY ACTOR	Guest , User and Admin	
DESCRIPTION	This use case describes the process to access the forum by primarily all the actors of the portal. Every user who faces an issue or has a query can access the community forum through this functionality.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	User chooses the forum option from the portal.
	2	User selects the particular topic on the forum.
	3	User searches for a specific query.
	4	User selects a particular post and can view post and replies.
	5	User can place question or answer questions on the forum.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	No alternate flow exits.
PRE-CONDITION	User is logged in.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. User goes to the forum page.	3. System responds to the request and displays the posts and discussion related to the particular topic by retrieving from the database.
	2. User selects the particular topic he wants to view.	4. System also adds the new discussion to the database.
POST-CONDITION	User is on the forum page and can participate in the discussion.	

The above table comprises of the description of the Access Forum Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.16 Block / Unblock User Accounts

Table 3.16: Block / Unblock User Accounts Use Case

USE CASE NAME	Block / Unblock User Accounts	
USE CASE ID	EGH-16	
PRIORITY	High	
PRIMARY ACTOR	Admin	
DESCRIPTION	This use case describes the process by which the admin can block or unblock a specific user's account based on any misconduct.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	Admin selects the view registered users option.
	2	Admin is presented with all the users registered on the platform.
	3	Admin selects a particular user.
	4	Admin selects block / unblock user option on the user's profile.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	User has already deleted his account.
PRE-CONDITION	Admin is logged in.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. Admin selects the view registered users option.	3. System requests the user's information from the database.
	2. Admin selects the particular user and selects the block user option.	4. System removes/blocks the user and updates the database.
POST-CONDITION	Admin is notified about the successful blocking of the user.	

The above table comprises of the description of the Block / Unblock User Accounts Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.17 Approve / Reject Jeweler Status Request

Table 3.17: Approve / Reject Jeweler Status Request Use Case

USE CASE NAME	Approve / Reject Jeweler Status Request	
USE CASE ID	EGH-17	
PRIORITY	High	
PRIMARY ACTOR	Admin	
DESCRIPTION	This use case describes the process through which an Admin approves the status update request of a user based on manual authentication.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	Admin open the menu from profile icon.
	2	Admin goes to pending status requests page.
	3	Admin opens requests one by one and validates the information (cnic, phone no, cnic images and store images) in the request.
	4	Admin approves or rejects the request.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	User has canceled the request and the request gets discarded upon opening.
PRE-CONDITION	Admin is logged in and Admin is on the pending status request page.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. Admin goes to pending status requests page and opens requests.	3. System responds by retrieving requests from the database and showing it to the admin.
	2. Admin approves or rejects the request.	4. System discards the request or updates the status in the database based on Admin's decision
POST-CONDITION	The request of the user would be approved / disapproved.	

The above table comprises of the description of the Approve / Reject Jeweler Status Request Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.18 Approve / Reject Ads

Table 3.18: Approve / Reject Ads Use Case

USE CASE NAME	Approve / Reject Ads	
USE CASE ID	EGH-18	
PRIORITY	High	
PRIMARY ACTOR	Admin	
DESCRIPTION	This use case describes the process through which an Admin approves or rejects the ad posting request of a user by manual authentication.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	Admin goes to pending ad posting requests.
	2	Admin opens requests one by one and validates the ad information (title, price, description, category and images) in the request.
	3	Admin approves or rejects the request.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	User has deleted the ad posting request and the request is discarded.
PRE-CONDITION	Admin is logged in and Admin is on the pending Ad request page.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. Admin goes to pending ad posting requests page and opens requests.	3. System responds by retrieving requests from the database and showing it to the admin.
	2. Admin approves or rejects the request.	4. System discards the request and adds the ad to published ads in the database based on Admin's decision.
POST-CONDITION	The request of the user would be approved or disapproved.	

The above table comprises of the description of the Approve / Reject Ads Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.19 Approve / Reject Commission Requests

Table 3.19: Approve / Reject Commission Requests Use Case

USE CASE NAME	Approve / Reject Commission Requests	
USE CASE ID	EGH-19	
PRIORITY	High	
PRIMARY ACTOR	Admin	
DESCRIPTION	This use case describes the process by which the admin can accept the request of the jeweler to change their commission.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	Admin selects view pending commission change requests option from his profile menu
	2	Admin approves or disapproves the request based on the reason given by the seller.
	3	A notification is sent to the jeweler notifying them about the state of their request
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	Jeweler has canceled the request.
	2	Jeweler has already once updated their commission.
PRE-CONDITION	Admin is logged in.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. Admin opens up his pending requests page for commission change.	3. System tracks the request and keeps it in the database until dealt to.
	2. Admin approves or rejects the request.	4. System updates the new commission in the database for that specific jeweler
POST-CONDITION	A notification is sent to the jeweler notifying them about the status of their request.	

The above table comprises of the description of the Approve / Reject Commission Requests Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.20 Initiate Chat

Table 3.20: Initiate Chat Use Case

USE CASE NAME	Initiate Chat	
USE CASE ID	EGH-20	
PRIORITY	High	
PRIMARY ACTOR	Users	
DESCRIPTION	This use case describes the process by which one user can initiate chat with another user or jeweler when buying / selling or obtaining a service from a jeweler.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	User(buyer) views listing details.
	2	User selects chat option on the ad.
	3	User sends a message to the other user (seller).
	4	User (seller) receives the message and replies.
	5	User (buyer) receives the reply and can continue chat.
	6	Both users can view this chat along with chat history.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	User(seller) is offline and is not giving any response.
PRE-CONDITION	Both the users (Buyer / Sellers) are logged in.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. User(buyer) selects the chat option	3. System responds by keeping a record of the chat in the database.
	2. User(seller) receives a notification and replies to chat.	4. System also retrieves the previous history of chat from the database and shows it to the users.
POST-CONDITION	Both users would be able to chat with each other.	

The above table comprises of the description of the Initiate Chat Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.21 Receive Notifications

Table 3.21: Recieve Notifications Use Case

USE CASE NAME	Receive Notification	
USE CASE ID	EGH-21	
PRIORITY	High	
PRIMARY ACTOR	Users, Jeweler and Admin	
DESCRIPTION	This use case describes the process through which actors of the platform receive their concerned notifications.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	Actors perform actions such as sending requests. (ad placement etc.)
	2	Actors who are concerned with either the request or their response receive notifications.
	3	All actors can view notification beside their profile.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	Request is not processed resulting in no notification being received.
PRE-CONDITION	Actors are logged in.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. Actors perform actions.	3. System keeps a track of processes and updates the database with notifications.
	2. Other concerned actors receive notifications.	4. System then retrieves them and alerts actors, after which they are removed from the database if the user discards them.
POST-CONDITION	Actors can view notifications on their notifications page.	

The above table comprises of the description of the Receive Notifications Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.22 Delete Account

Table 3.22: Delete Account Use Case

USE CASE NAME	Delete Account	
USE CASE ID	EGH-22	
PRIORITY	Low	
PRIMARY ACTOR	User	
DESCRIPTION	This use case describes the process through which a user can delete personal account. Any user who wishes to delete his account and all data associated with it must go throw this functionality.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	User goes to his profile settings.
	2	User selects the option for delete account.
	3	User confirms the deletion and lands on the login page.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	No alternate flow.
PRE-CONDITION	User is logged in and User is on this profile page.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. User goes to profile settings.	3. System responds by deleting all the data of the user from the database.
	2. User selects the option to delete account and confirms.	4. System removes the user itself from the database and redirects to the login page.
POST-CONDITION	User lands on the login page with his account permanently deleted.	

The above table comprises of the description of the Delete Account Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.23 Approve/Reject Certification Request

Table 3.23: Approve/Reject Certification Request Use Case

USE CASE NAME	Approve/Reject Certification Request	
USE CASE ID	EGH-23	
PRIORITY	High	
PRIMARY ACTOR	Jeweler	
DESCRIPTION	This use case describes the process by which the jeweler on the platform can approve or reject the request to authenticate a particular seller's gold. All jewelers must go through this functionality to authenticate the gold for the seller.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	Jeweler opens up Pending Requests from his profile menu.
	2	Jeweler opens up each request one by one and views the Ad.
	3	Jeweler selects Approve or Disapprove.
	4	In case of approval, chat is initiated between jeweler and seller.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	The seller removed or canceled the request.
PRE-CONDITION	Jeweler is logged in and seller has a live ad that needs authentication.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. jeweler opens up pending requests page.	3. System tracks the request and keeps it in the database until dealt to.
	2. Jeweler approves or rejects requests one by one	4. System then maintains the chat between seller and jeweler and records it in the database.
POST-CONDITION	Chat is initiated between the seller and the jeweler.	

The above table comprises of the description of the Approve / Reject Certification Requests Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.24 Add Product

Table 3.24: Add Product Use Case

USE CASE NAME	Add Product	
USE CASE ID	EGH-24	
PRIORITY	High	
PRIMARY ACTOR	Jeweler	
DESCRIPTION	This use case describes the process by which a registered jeweler can add products to his allotted store page. This functionality has to be followed by every jeweler who wishes to add products to store page.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	Jeweler goes to My Store page.
	2	Jeweler selects add a product option.
	3	Jeweler fills the required information (title, description, price, category) about the product and submits.
	4	Jeweler is notified and redirected to My store page.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	Data entered fails validation and an error is thrown.
PRE-CONDITION	Jeweler is logged in and Jeweler is on My Store page.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. Jeweler goes to his store page.	3. System validates the information of the product that the jeweler entered.
	2. Jeweler selects add a product and fills required information (title, description, price, category).	4. System records the information in the database under the jeweler's store page.
POST-CONDITION	Jeweler is on My Store page and can view the product listed.	

The above table comprises of the description of the Add Product Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.25 Edit Product

Table 3.25: Edit Product Use Case

USE CASE NAME	Edit Product	
USE CASE ID	EGH-25	
PRIORITY	Medium	
PRIMARY ACTOR	Jeweler	
DESCRIPTION	This use case describes the process by which a registered jeweler can edit products already listed on his store page. Every jeweler who wishes to make changes to his listed product's information must go through this functionality	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	Jeweler goes to My Store page.
	2	Jeweler selects a product to edit.
	3	Jeweler makes the desired changes and submits.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	New data fails validation and is not updated in the database.
PRE-CONDITION	Jeweler is logged in and Jeweler is on My Store page.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. Jeweler goes to his store page.	3. System validates the new information that is entered.
	2. Jeweler selects a product to edit and makes changes.	4. System updates the new product information in the database.
POST-CONDITION	Jeweler is on My Store page and can view the product listing with updated info.	

The above table comprises of the description of the Edit Product Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.26 Delete Product

Table 3.26: Delete Product Use Case

USE CASE NAME	Delete Product	
USE CASE ID	EGH-26	
PRIORITY	Low	
PRIMARY ACTOR	Jeweler	
DESCRIPTION	This use case describes the process by which a registered jeweler can delete products already listed on his store page. Every jeweler who wishes to delete his listed product must go through this functionality.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	Jeweler goes to My Store page.
	2	Jeweler selects a product to delete.
	3	Jeweler confirms the deletion.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	Jeweler has no active listed product to delete on his store page.
PRE-CONDITION	Jeweler is logged in and Jeweler is on My Store page.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. Jeweler goes to his store page.	3. System asks for confirmation of the deletion.
	2. Jeweler selects a product to delete.	4. System removes the product from the database.
POST-CONDITION	Jeweler is on My Store page and can view that the product is not listed.	

The above table comprises of the description of the Delete Product Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

3.4.27 Request Commission Change

Table 3.27: Request Commission Change Use Case

USE CASE NAME	Request Commission Change	
USE CASE ID	EGH-27	
PRIORITY	High	
PRIMARY ACTOR	Jeweler	
DESCRIPTION	This use case describes the process by which the jeweler on the platform can request to change the commission for his authentication services. All jewelers must go through this functionality to change the rate of commission.	
BASIC FLOW	**STEPS**	**ACTIONS**
	1	Jeweler selects Change Commission option from his profile menu
	2	Jeweler enters the amount of commission that they wish to receive and valid reason for change.
	3	Jeweler is prompted to wait until the admin approves the request.
ALTERNATE FLOW	**STEPS**	**ACTIONS**
	1	Jeweler has already put a request forward.
	2	Jeweler has already once updated their commission.
PRE-CONDITION	Jeweler is logged in.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	1. jeweler request to change his commission rate.	3. System tracks the request and keeps it in the database until dealt to.
	2. Jeweler is prompted to wait until the admin approves	4. System updates the new commission in the database for that specific jeweler.
POST-CONDITION	Jeweler is prompted to wait until admin approves or disapproves.	

The above table comprises of the description of the Request Commission Change Use Case and incorporates important details such as the Use Case name, Use Case ID, primary actors, basic description , normal and alternate along with pre and post conditions.

Chapter 4

System Design

4.1 System Architecture

The "EGold Haven" web application can be conceptualized and explained in terms of a three-tier architecture [9], where each tier represents a specific layer of functionality. The three tiers are:

4.1.1 Presentation Tier (FRONT-END)

The presentation tier, also known as the front-end, is the user interface layer that users interact with directly. In the case of "EGold Haven," the front-end is developed using React.js, HTML, and CSS.

React.js components are responsible for rendering the user interface elements, such as registration and login forms, profile management, ad listings, and forum interactions.

User authentication components, including login and registration forms, ensure a secure and user-friendly experience for users accessing the platform.

4.1.2 Application Tier (BACK-END)

The application tier, or back-end, serves as the business logic layer that processes requests from the front-end, communicates with the database, and manages the overall application flow. In "EGold Haven," the back-end is built using Node.js and Express.js.

Node.js enables non-blocking I/O, ensuring efficient handling of multiple concurrent requests. Express.js simplifies the development of Restful APIs and manages routing within the application.

Restful APIs are implemented to handle various functionalities, including user management, ad listings, certification services, and forum interactions. Middleware functions are

used for user authentication to secure access to specific parts of the application.

The back-end also integrates the jeweler verification and certification process, allowing accredited jewelers to register, verify credentials, and certify gold items.

4.1.3 Data Tier (DATABASE)

The data tier, or database layer, involves storing and managing the application's data. In "EGold Haven," MongoDB is used as the NoSQL database, and Mongoose serves as the Node.js ORM for interaction with the database.

MongoDB stores data related to user profiles, gold item listings, certifications, transactions, and forum discussions. Mongoose facilitates data validation, schema modeling, and database operations.

The database is crucial for maintaining the integrity of user data, tracking gold item listings, managing certifications, and storing information related to forum discussions.

This three-tier architecture provides a modular and scalable structure for "EGold Haven", allowing for the separation of concerns between the user interface (presentation), business logic (application), and data storage (database).

4.2 Sequence Diagrams

Sequence diagrams are a type of interaction diagram user to visualize the interactions between objects or components of the system to achieve or perform a specific functionality.

4.2.1 Register User

The sequence diagram below show the interaction of user and the components of the application to register a particular user. It begins with the user opening the registration page that is returned by the application and the user enters the registration details and submits. The system stores the data in the database and returns acknowledgement upon which the user is prompted with success or failure.

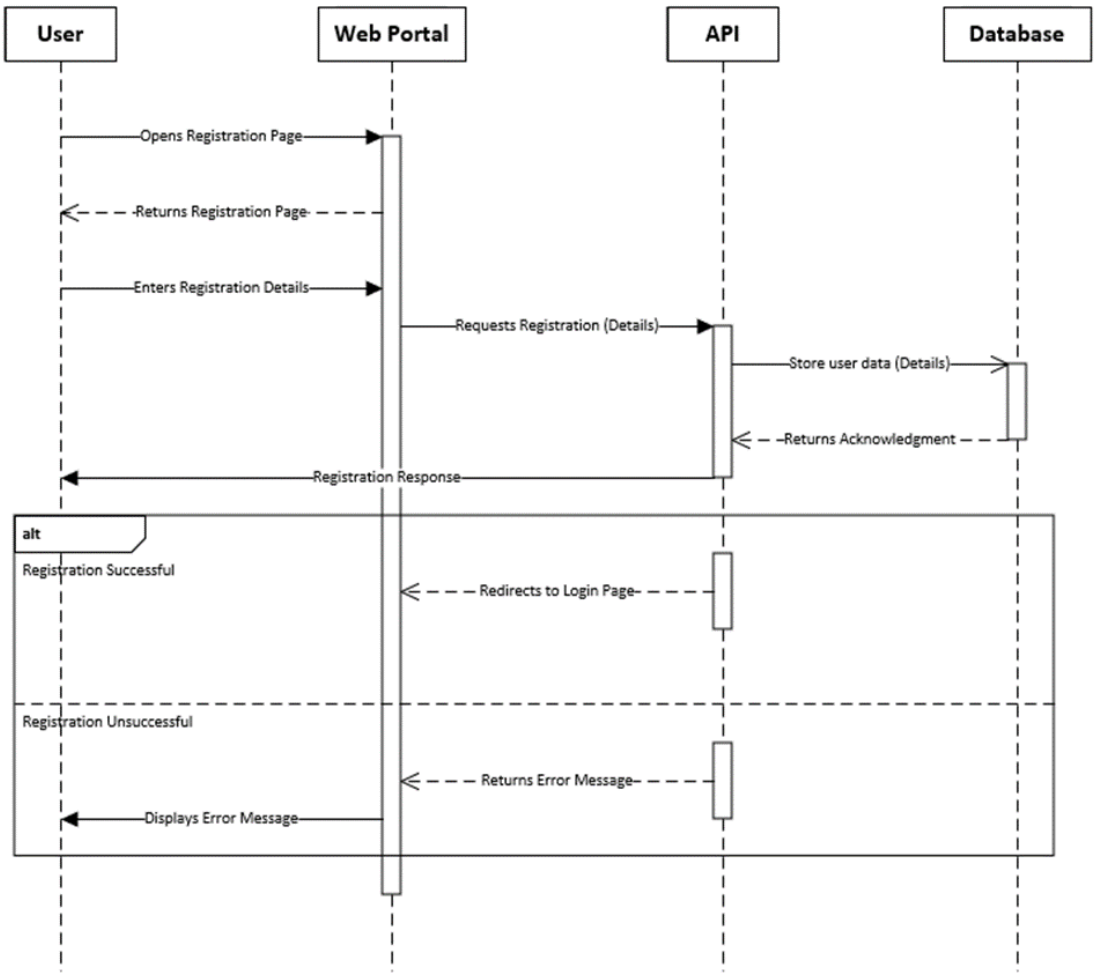

Figure 4.1: Register User

4.2.2 Login User

The sequence diagram below show the interaction of user and the components of the application to login a particular user. It begins with the user opening the login page that is returned by the application and the user enters the login credentials and submits. The system retrieves and validates the data in the database with credentials and returns result upon which the user is prompted with success or failure and redirected to the respective page.

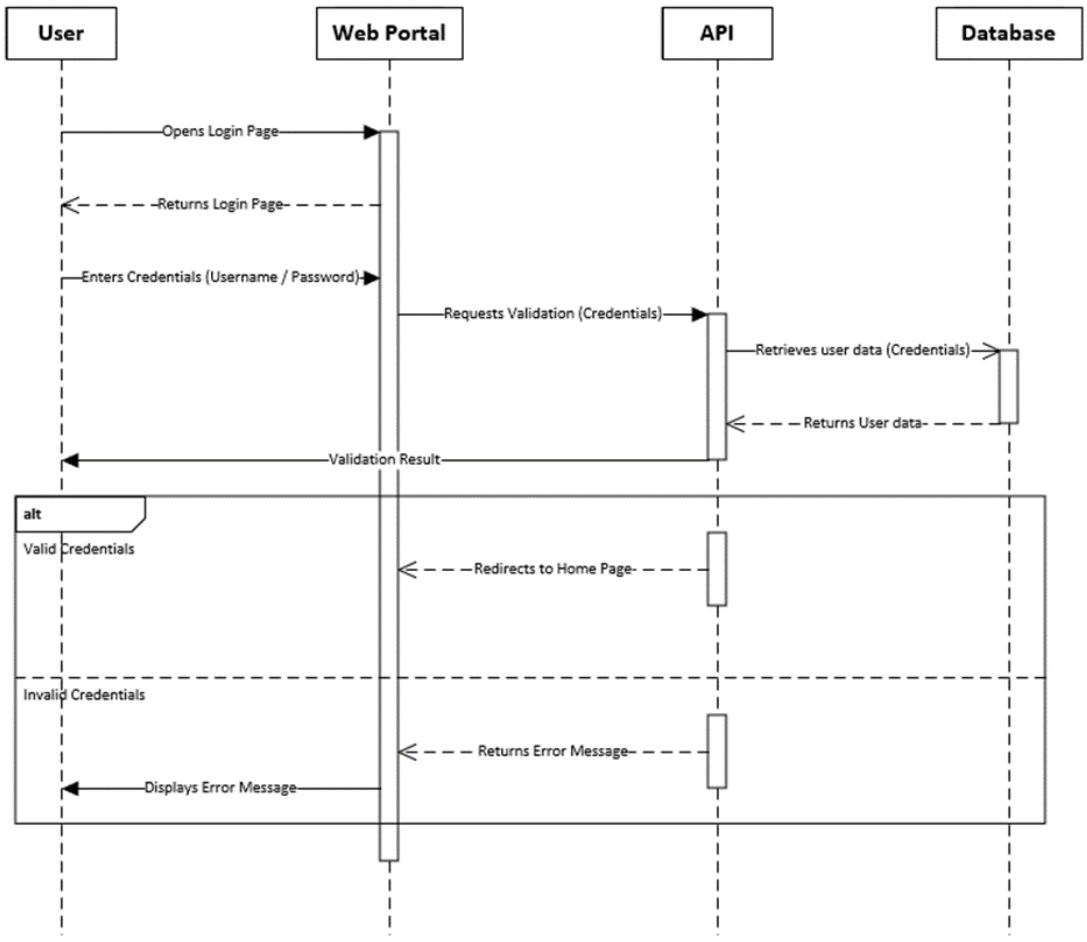

Figure 4.2: Login User

4.2.3 Reset Password

The sequence diagram below show the interaction of user and the application to reset password. The user enters the email and system validates the email from the database and sends a reset token to the user email. The user enters the token on the reset password page which is again validated from its copy in the database upon the success of which the user enters the new password and is redirected to the login page.

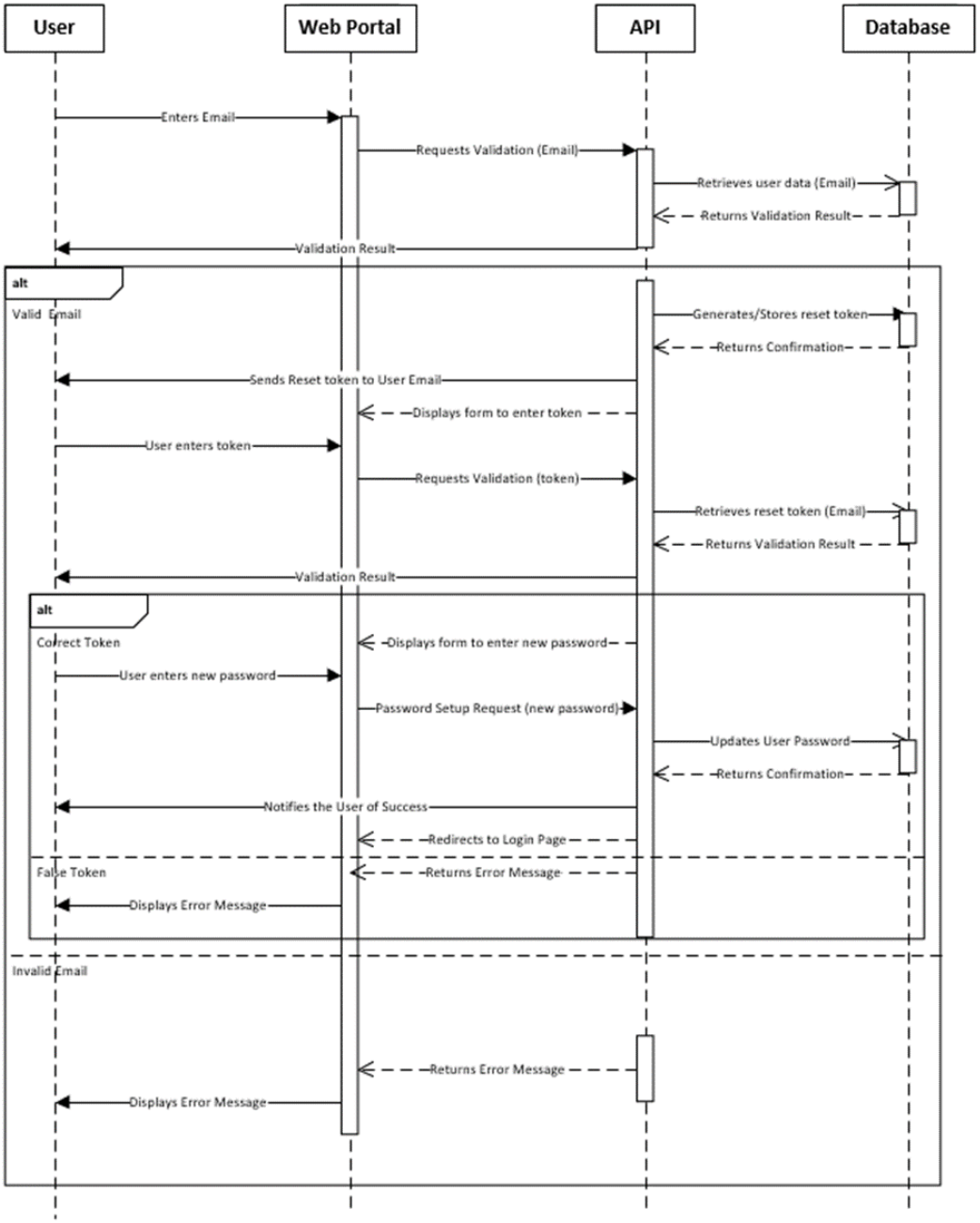

Figure 4.3: Reset Password

4.2.4 Edit Profile

User opens edit profile page and submits changes which the system updates in the database.

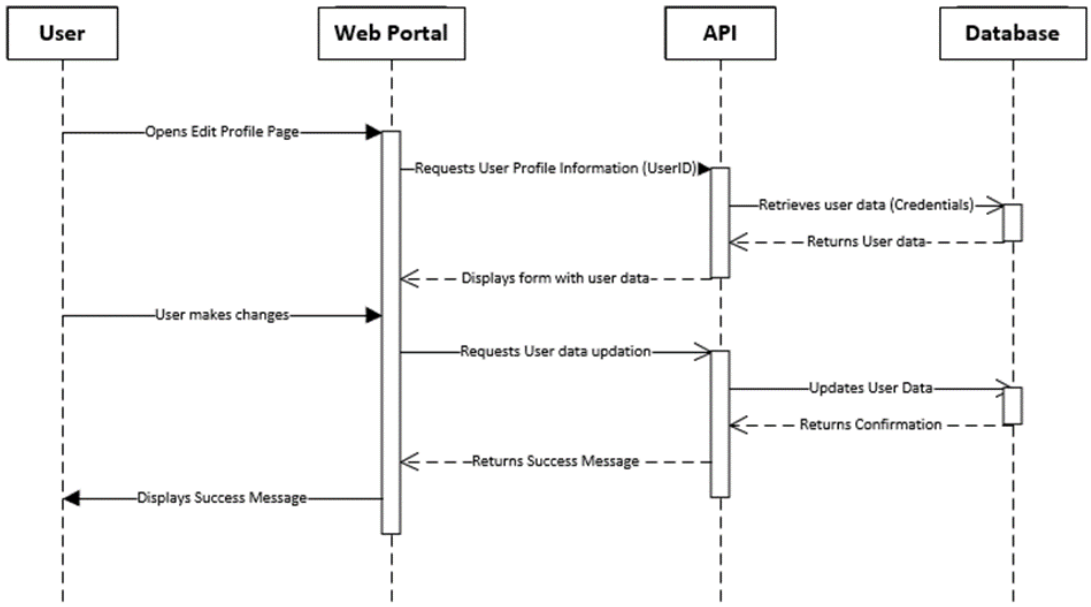

Figure 4.4: Edit Profile

4.2.5 Place Ad

User enters ad details and submits. System stores ad details and notifies user.

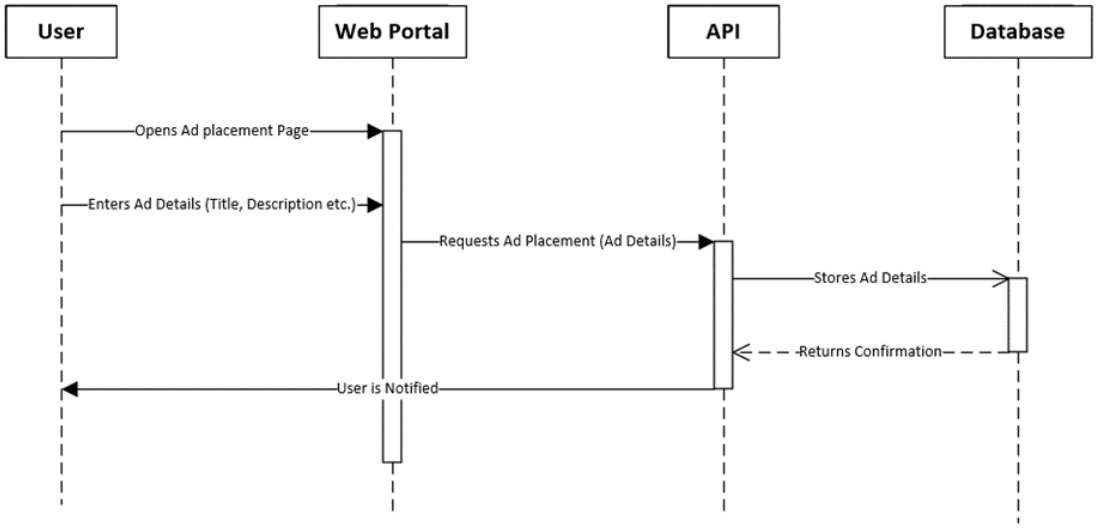

Figure 4.5: Place Ad

4.2.6 Edit Ad

User edits ad details and submits. System updates ad details and notifies user.

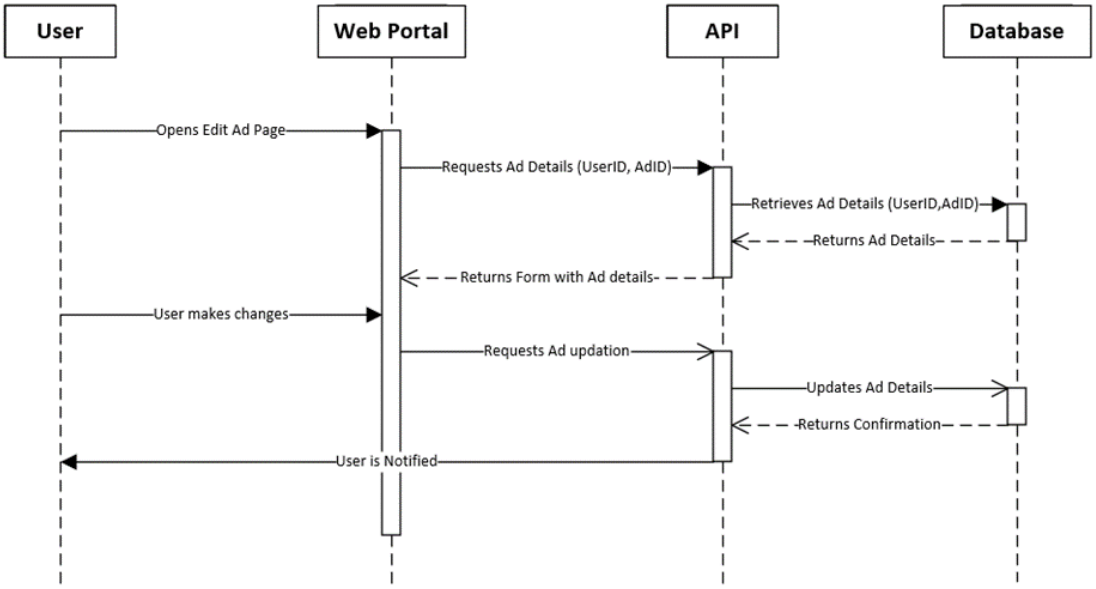

Figure 4.6: Edit Ad

4.2.7 Delete Ad

User opens ad and selects delete ad option. System deletes ad from the database and notifies user.

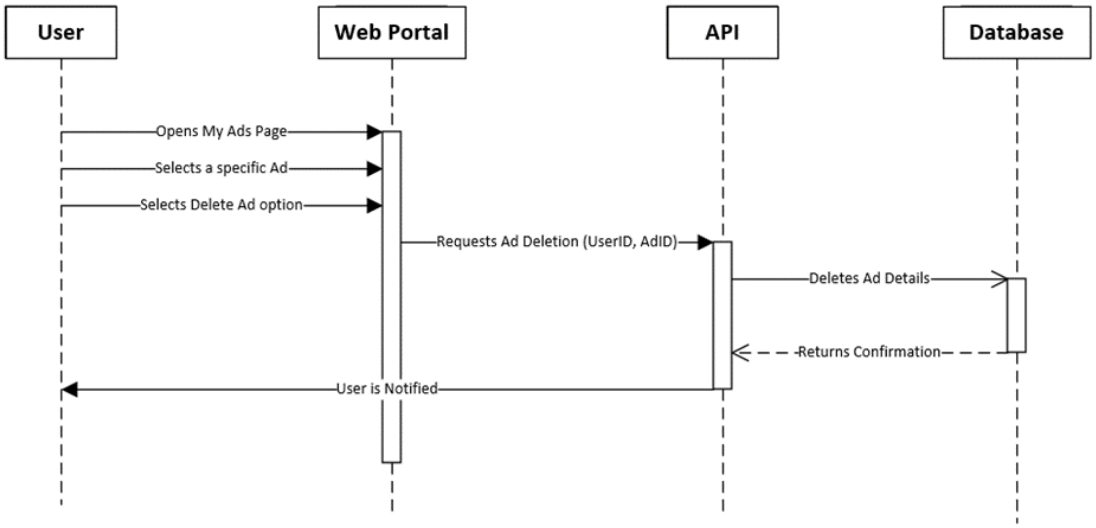

Figure 4.7: Delete Ad

4.2.8 Request for Jeweler Status

User opens profile page, selects status update option and enters required details. The request details and stored in the database by the system and the user is notified.

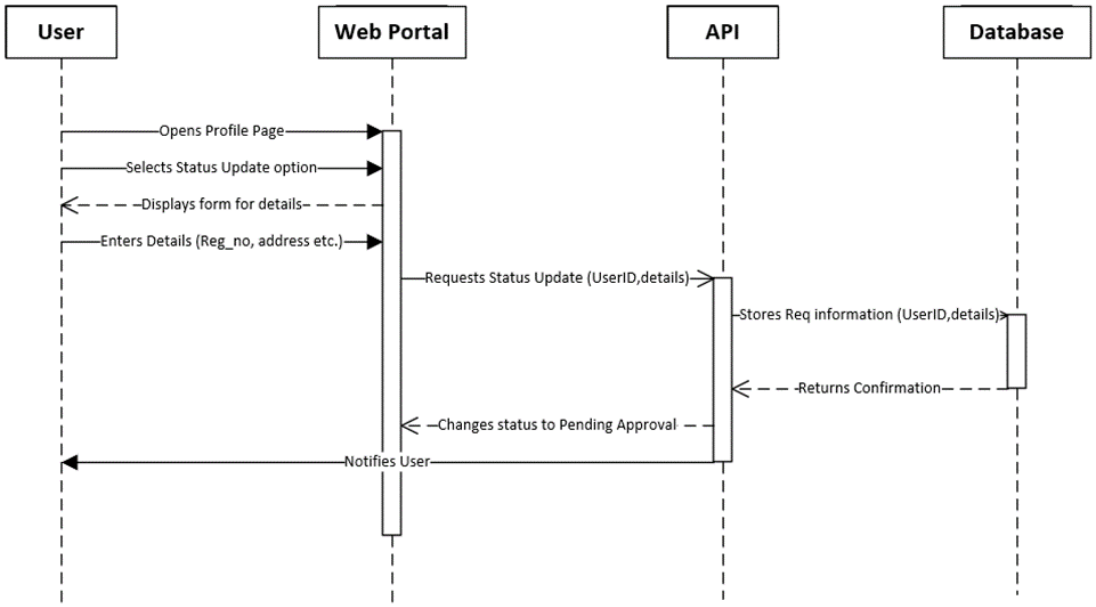

Figure 4.8: Request for Jeweler Status

4.2.9 Browse Listings

User opens homepage. System retrieves listings from the database that users browse

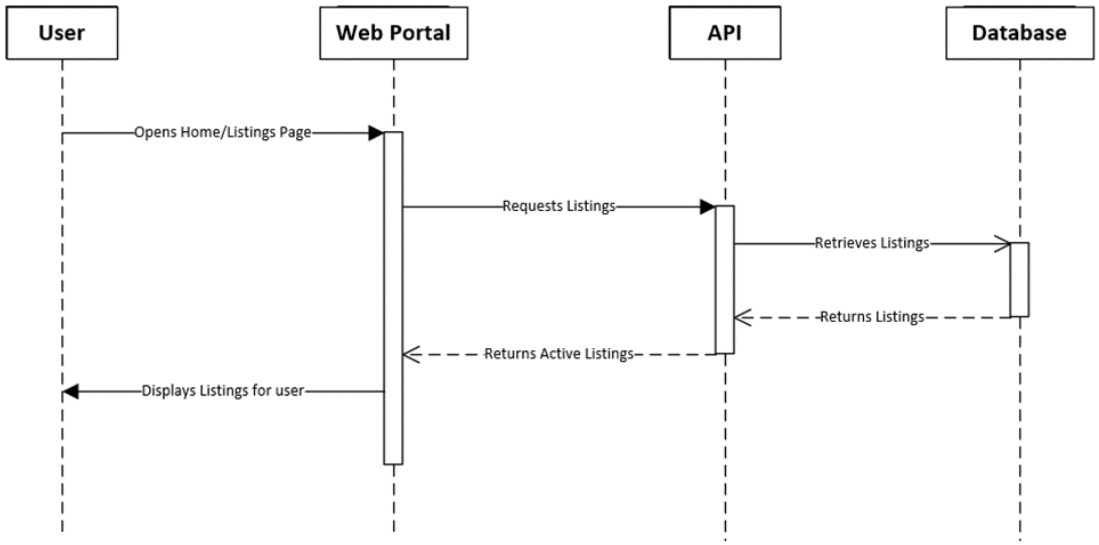

Figure 4.9: Browse Listings

4.2.10 Search Listings

The sequence diagram below show the interaction of user and the application to search listings. The user opens the homepage. The system fetches the listings from the database and displays them to the user. The user enters the keywords to search or applies filters and submits. The system retrieves the listings from the database based on those keywords and search filters and displays them to the user.

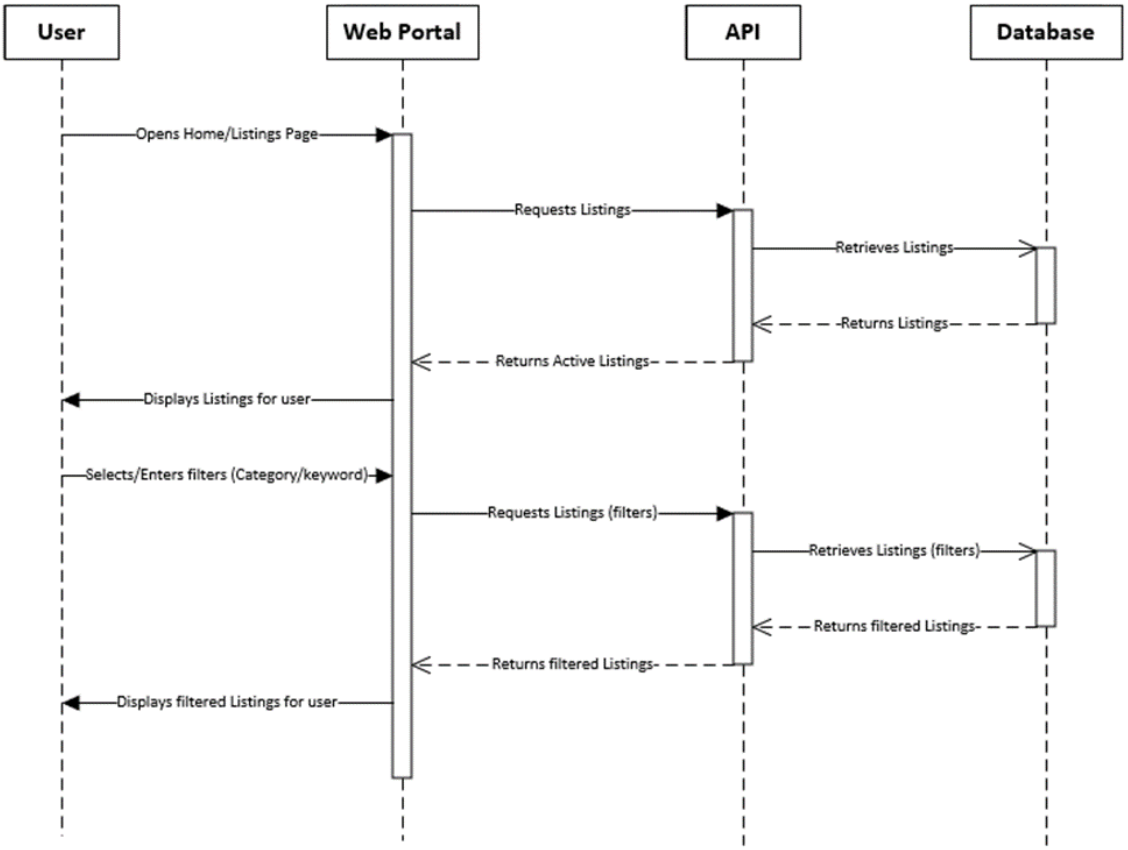

Figure 4.10: Search Listings

4.2.11 View Listing Details

The sequence diagram below show the interaction of user and the application to view listing details. The user opens the homepage. The system fetches the listings from the database and displays them to the user. The user selects a specific listing and is redirected to the the listing details page with the specific listing's details.

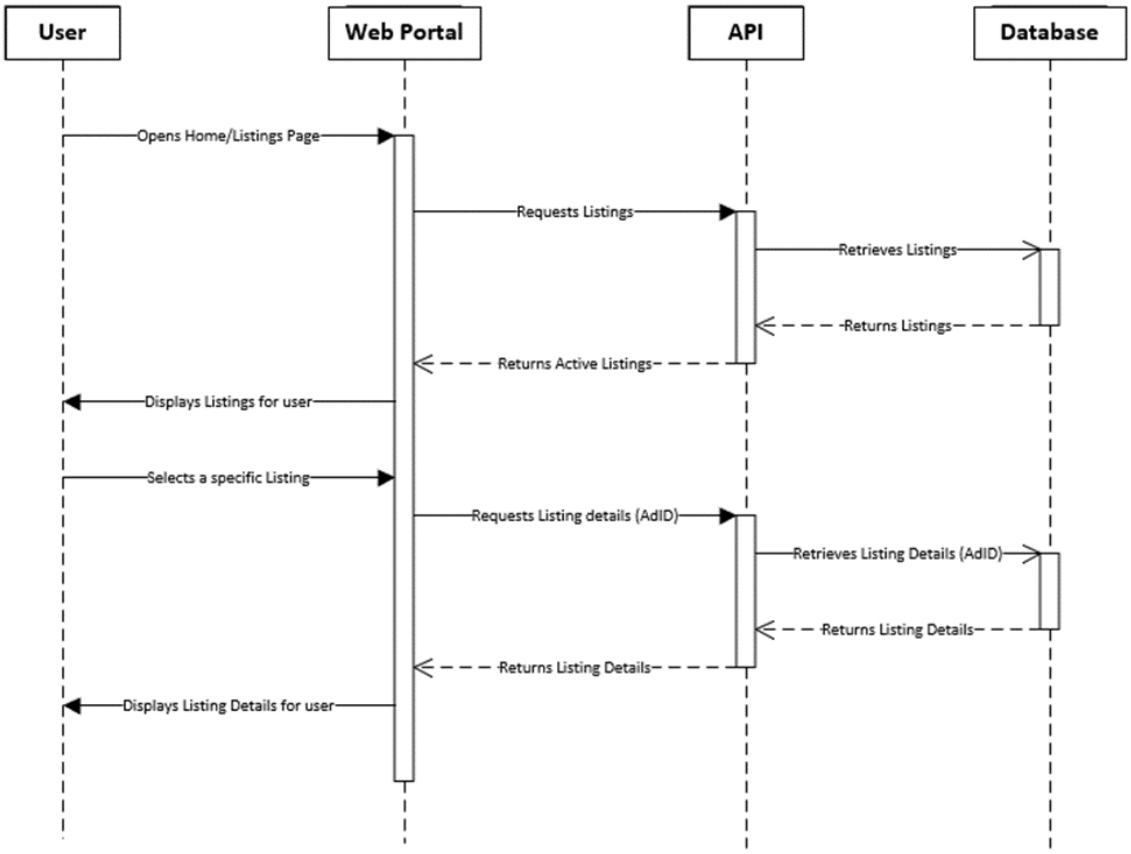

Figure 4.11: View Listing Details

4.2.12 Set Favorite

The sequence diagram below show the interaction of user and the application to view listing details. The user opens the homepage. The system fetches the listings from the database and displays them to the user. The user selects a specific listing and is redirected to the the listing details page with the specific listing's details. User selects the favorite option, system stores the ad as user favorite in database and the user is notified.

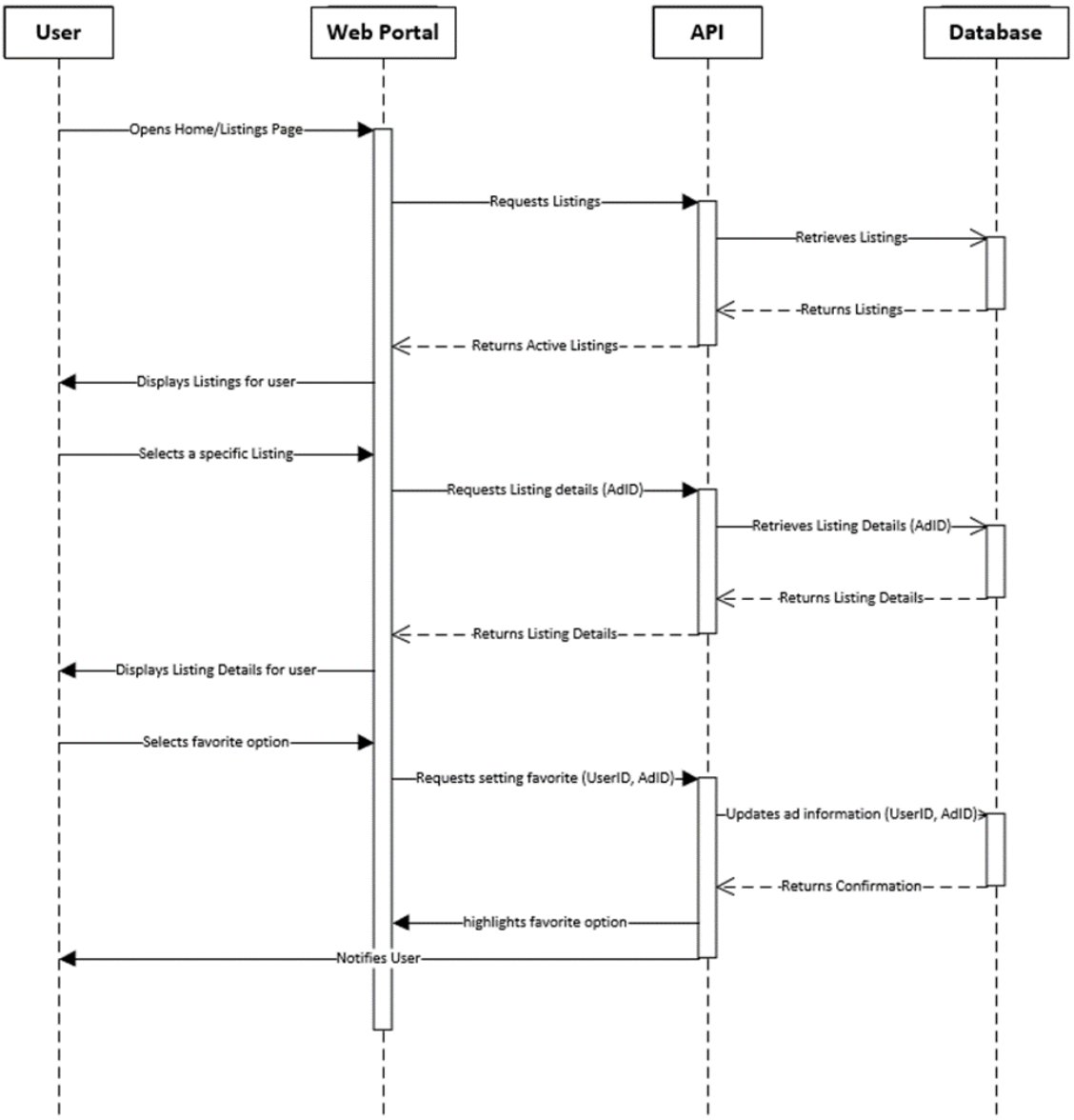

Figure 4.12: Set Favorite

4.2.13 View Dashboard

The sequence diagram below show the interaction of admin and the application to view admin dashboard. The admin clicks / opens the dashboard. The system retrieves the dashboard view along with all its stats and details from the database and populates the admin view with the dashboard portal fetched statistics which can now be viewed by the admin.

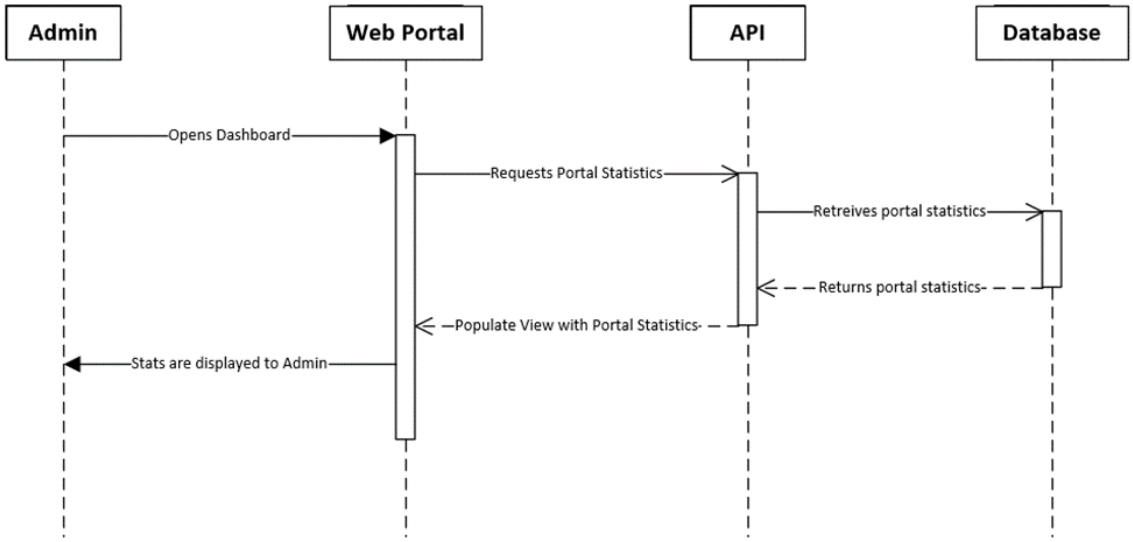

Figure 4.13: View Dashboard

4.2.14 Request Gold Certification

The sequence diagram below show the interaction of user and the application to request gold certification. The user opens the My Ads page. The system fetches the user ads from the database and displays them to the user. The user selects a specific ad and is redirected to the the listing details page with the specific listing's details. User selects the view jeweler option, system fetches and displays the jewelers. User selects a particular jeweler. System generates a request and stores it in the database and notifies the user of its pending status.

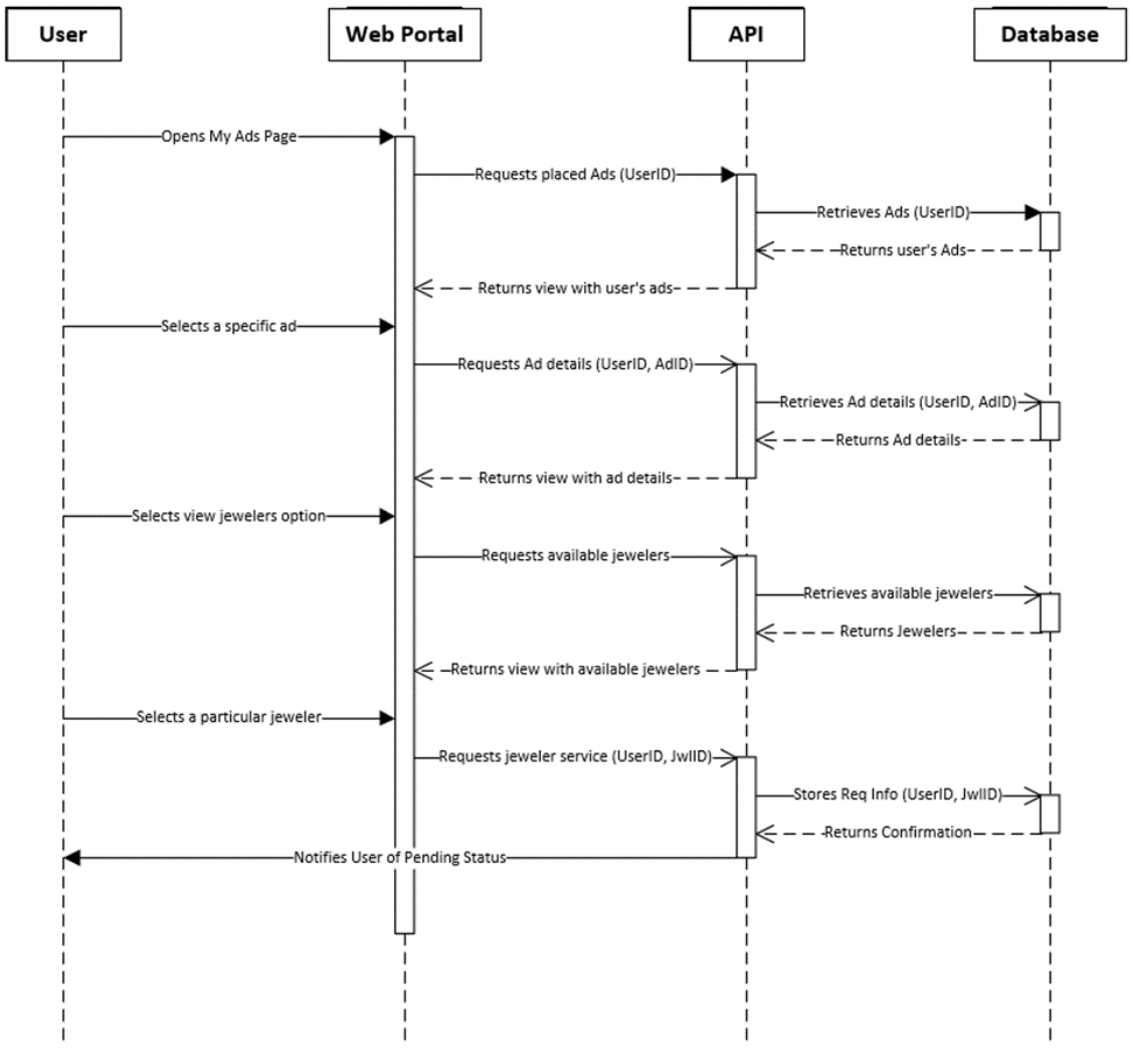

Figure 4.14: Request Gold Certification

4.2.15 Access Forum

The sequence diagram below show the interaction of user and the application to access forum. The user opens the forum page and the system fetches existing posts from the database and displays them for the user. User selects a particular category and the system fetches the forum posts based on the user options from the database.

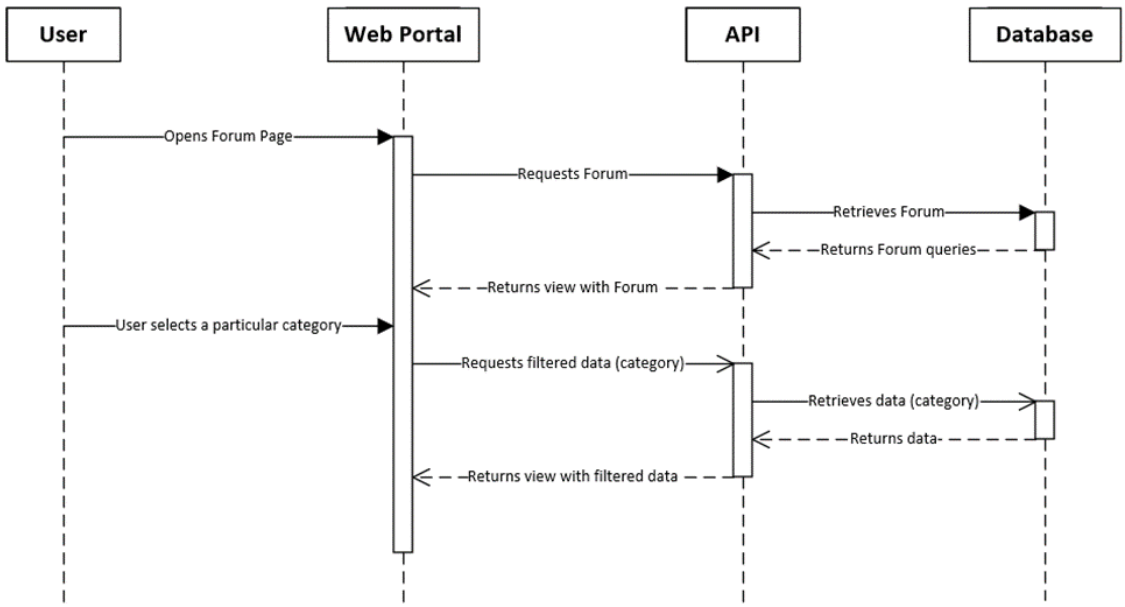

Figure 4.15: Access Forum

4.2.16 Block / Unblock User Accounts

The sequence diagram below show the interaction of admin and the application to block / unblock user accounts. The admin opens the view users page and the system fetches existing users from the database and displays them for the admin. Admin selects block/unblock option and the system updates the status of the user in the database.

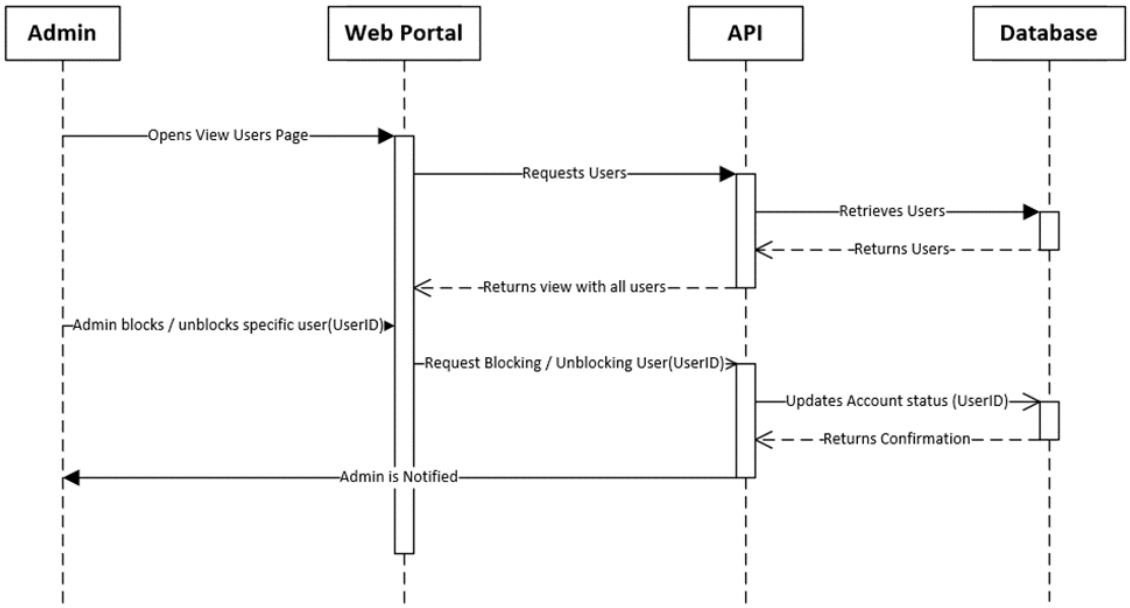

Figure 4.16: Block / Unblock User Accounts

4.2.17 Approve / Reject Jeweler Status Requests

The sequence diagram below show the interaction of admin and the application to approve/reject jeweler status requests. The admin opens the pending jeweler status requests page and the system fetches existing requests from the database and displays them for the admin. Admin selects a specific request and the system fetches it from the database and displays it to the admin. Admin Approves it or disapproves it upon which it is dumped or kept.

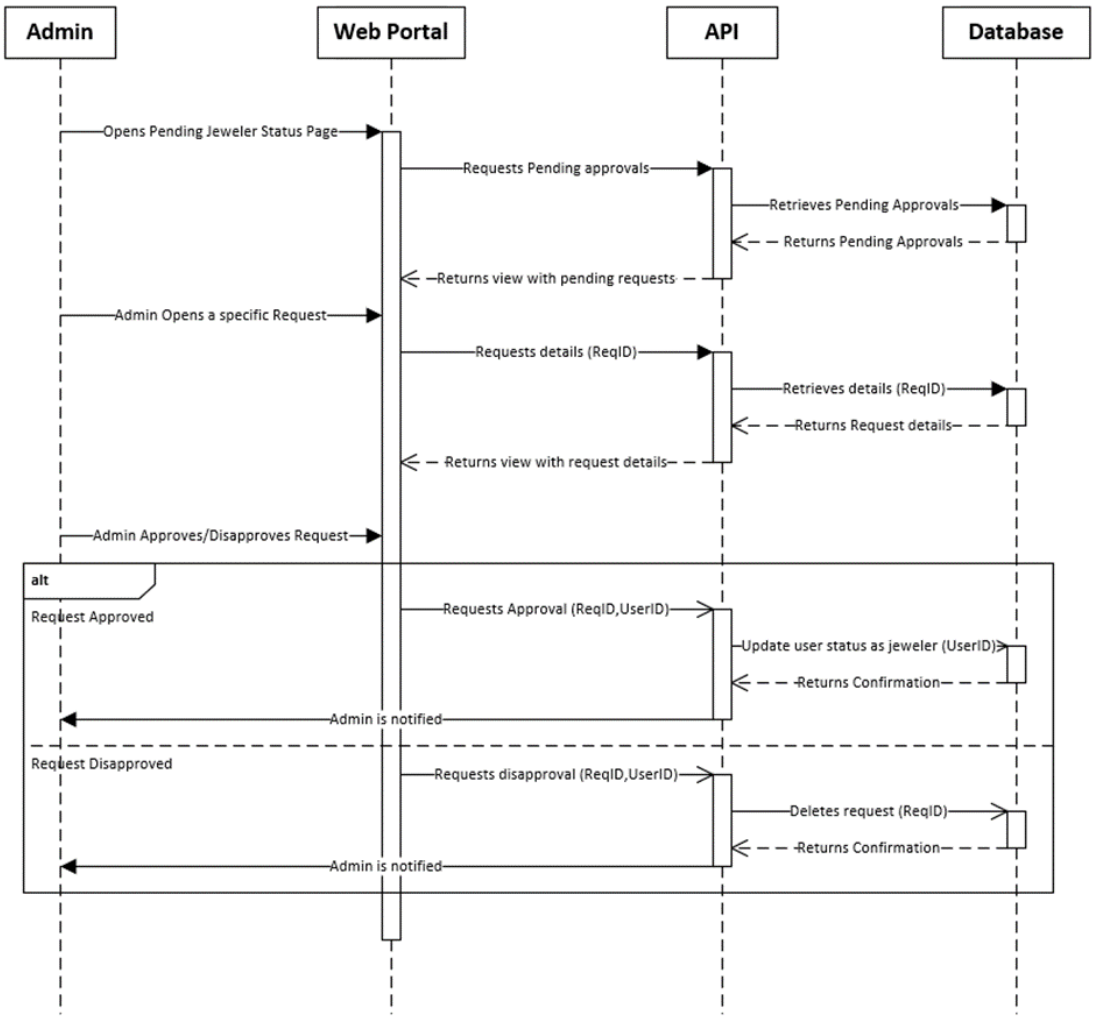

Figure 4.17: Approve / Reject Jeweler Status Requests

4.2.18 Approve / Reject Ads

The sequence diagram below show the interaction of admin and the application to approve/reject ads requests. The admin opens the pending ad requests page and the system fetches existing requests from the database and displays them for the admin. Admin selects a specific request and the system fetches it from the database and displays it to the admin. Admin Approves it or disapproves it upon which the status of the request is updated by the system in the database.

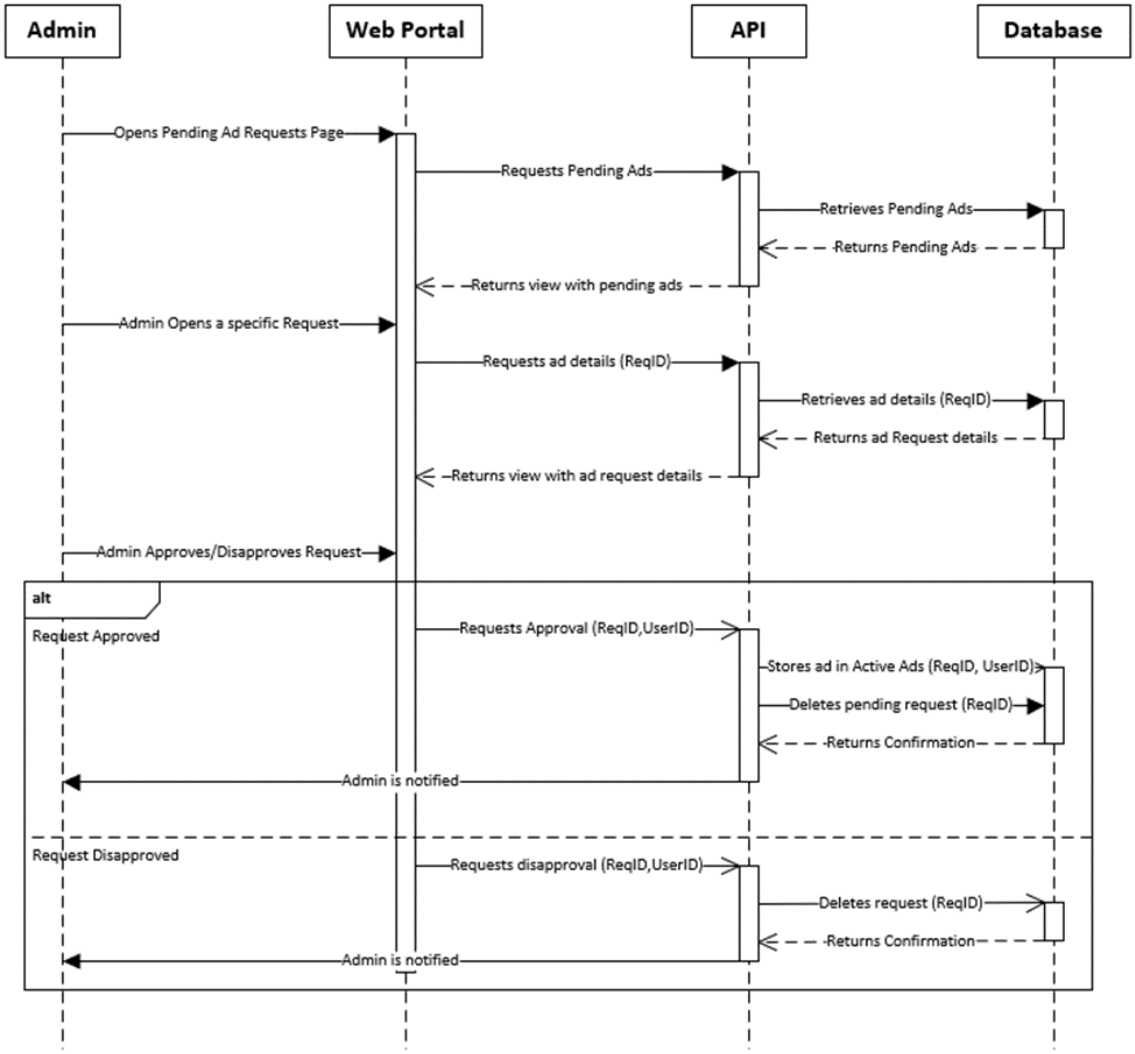

Figure 4.18: Approve / Reject Ads

4.2.19 Approve / Reject Commission Change Request

The sequence diagram below show the interaction of admin and the application to approve/reject commission change requests. The admin opens the pending commission change requests page and the system fetches existing requests from the database and displays them for the admin. Admin selects a specific request and the system fetches it from the database and displays it to the admin. Admin Approves it or disapproves it upon which the status of the request is updated by the system in the database.

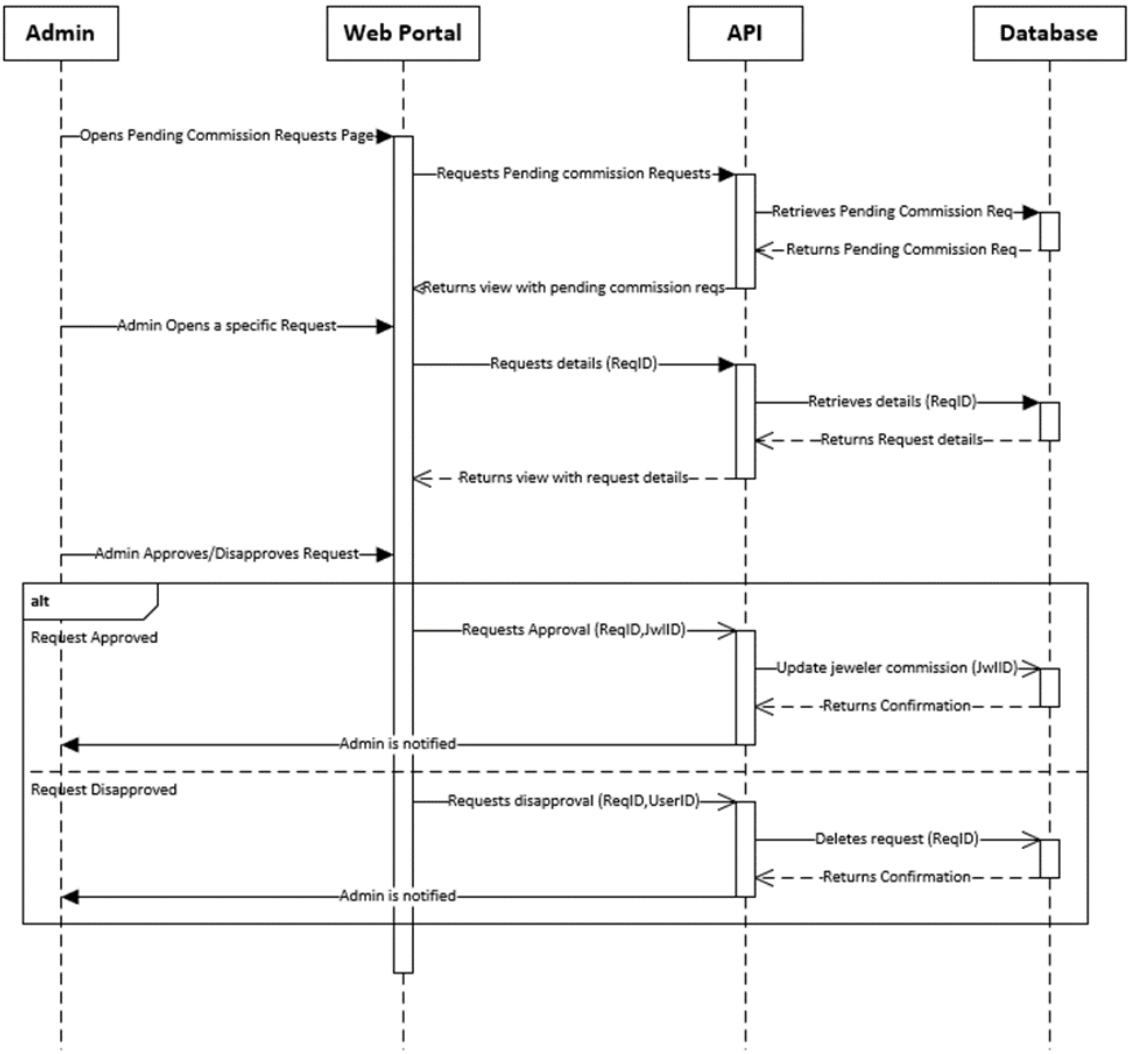

Figure 4.19: Approve / Reject Commission Change Request

4.2.20 Initiate Chat

The sequence diagram below show the interaction of user and the application to initiate chat with other users or the jewelers. Buyer clicks on the initiate chat button on the listing and sends a message to the seller. The system stores the chat and the message in the database. The seller opens the chat tab and the system fetches and displays the chat and vice versa.

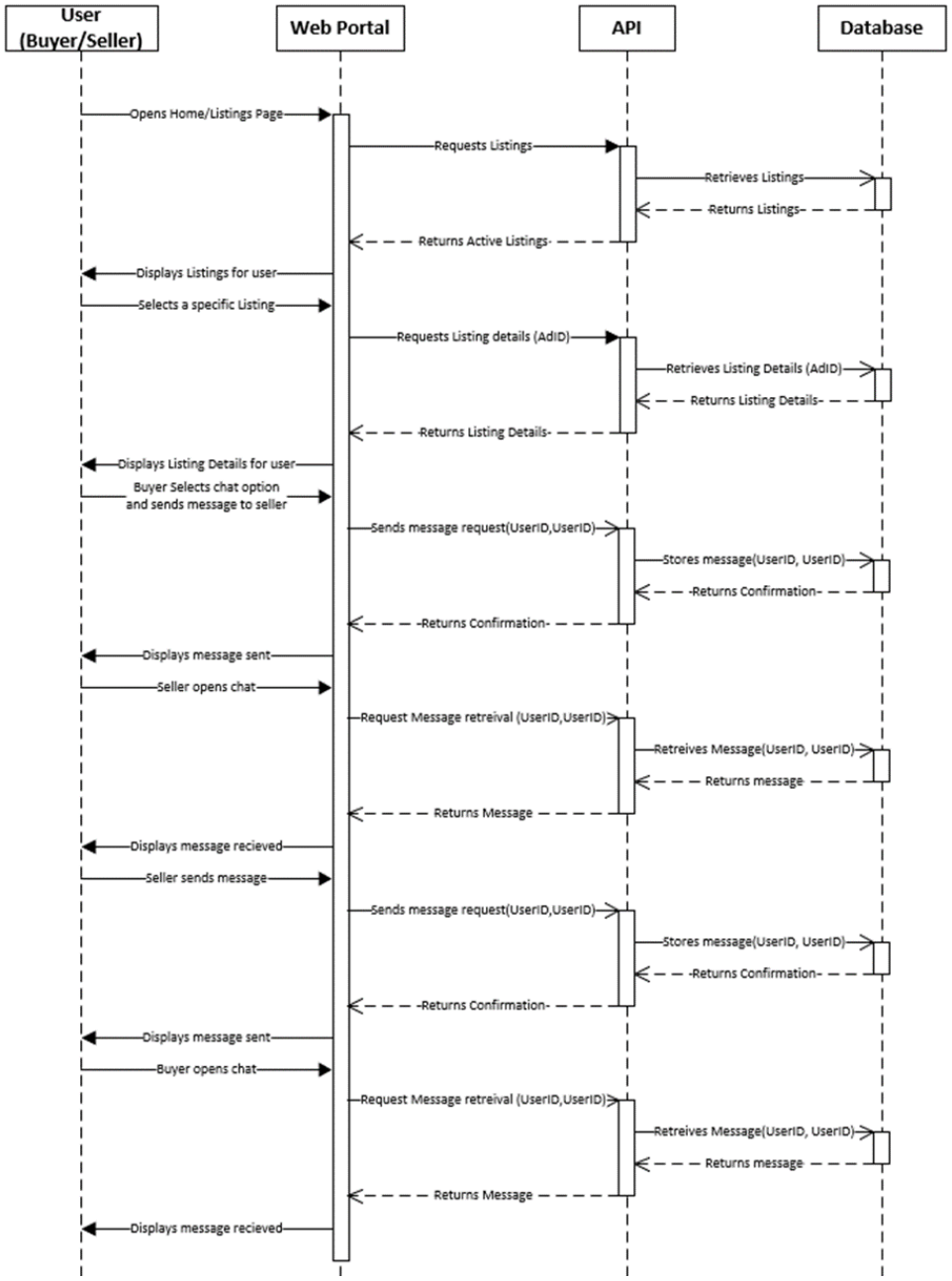

Figure 4.20: Initiate Chat

4.2.21 Delete Account

The sequence diagram below show the interaction of user and the application to delete user account. User opens the profile page, system fetches profile information from the database and displays it to the user and user selects delete account option, system deletes the user details from the database and redirects the user to the login page.

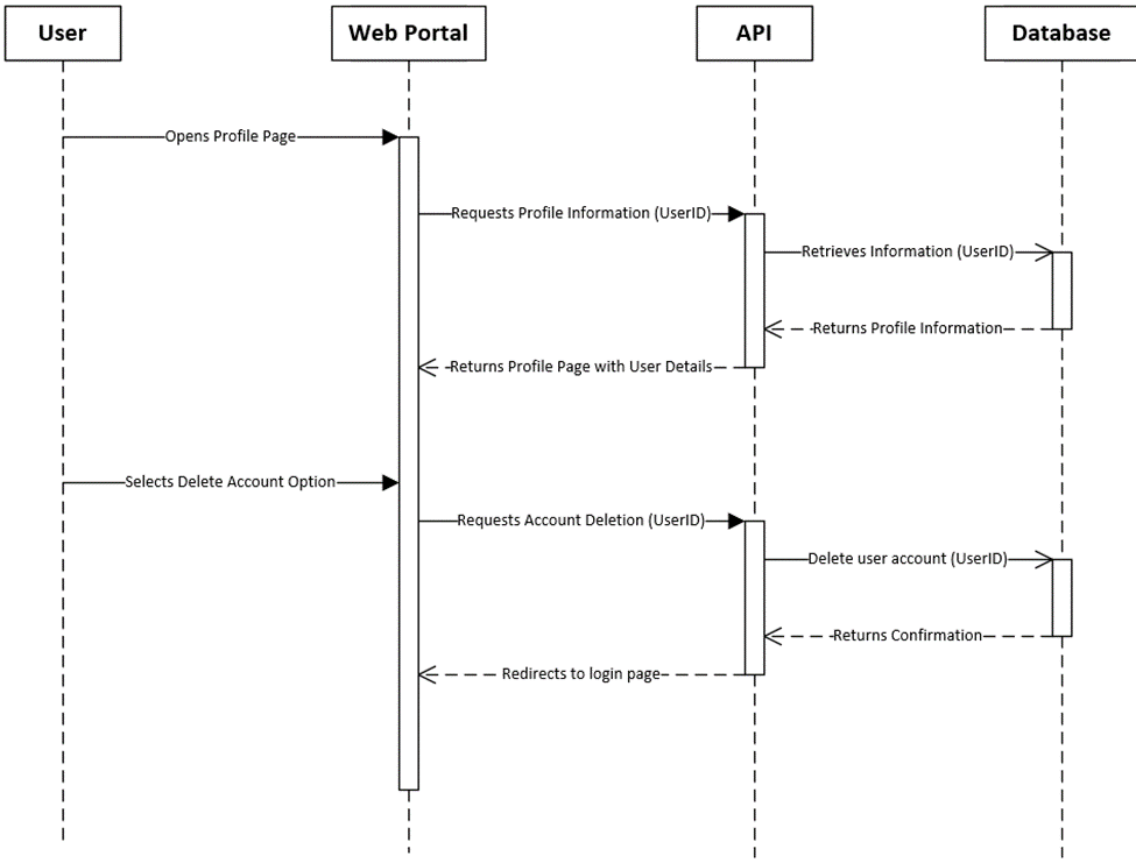

Figure 4.21: Delete Account

4.2.22 Approve / Reject Certification Request

The sequence diagram below show the interaction of jeweler and the application to approve/reject certification requests. The jeweler opens the pending certifications requests page and the system fetches existing requests from the database and displays them for the jeweler. Jeweler selects a specific request and the system fetches it from the database and displays it to the jeweler. Jeweler Approves it or disapproves it upon which the status of the request is updated by the system in the database.

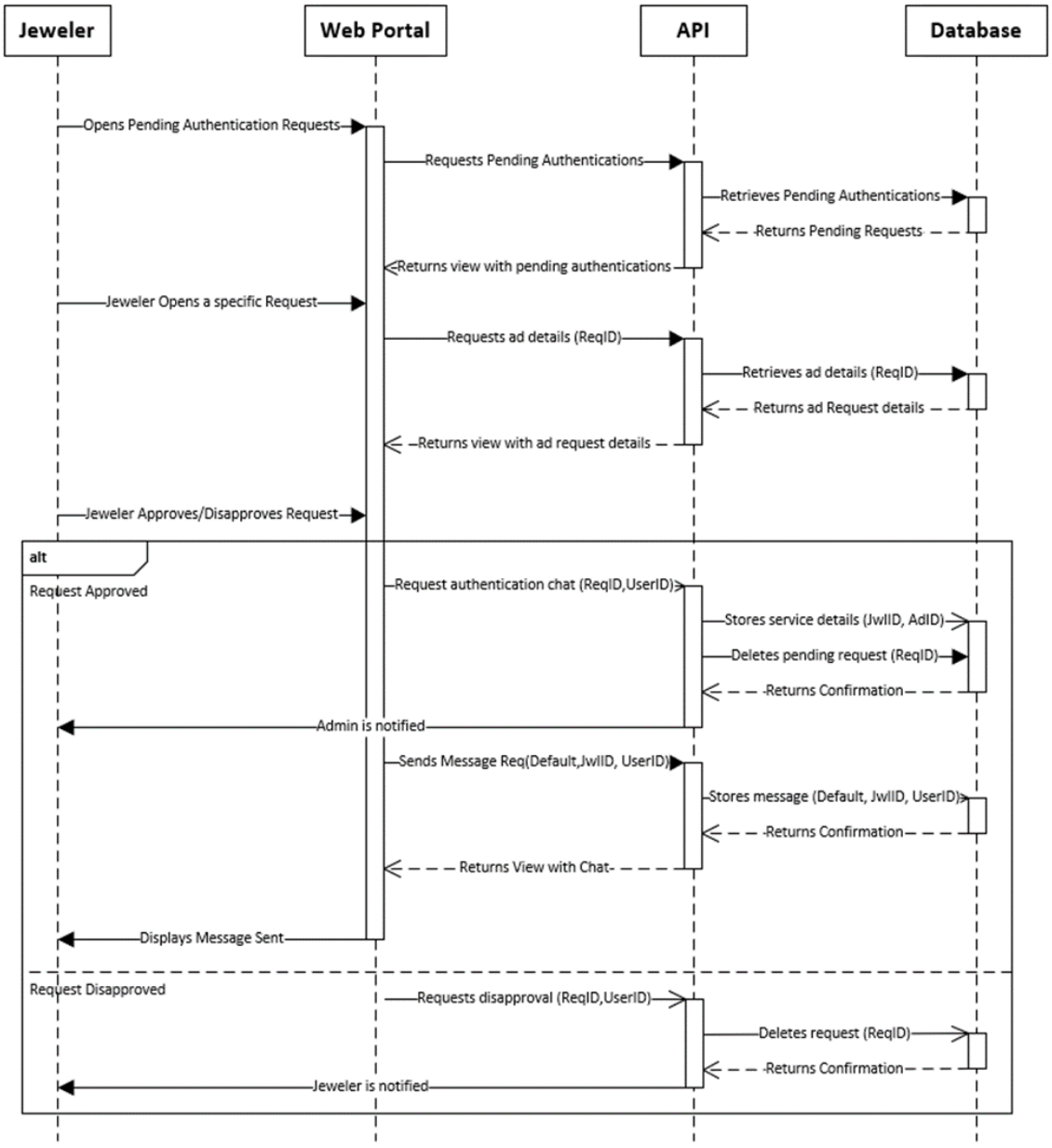

Figure 4.22: Approve / Reject Certification Request

4.2.23 Add Products

The sequence diagram below show the interaction of jeweler and the application to add product. The jeweler opens the store page and the system fetches jeweler store page from the database and displays it for the jeweler. Jeweler selects add product option and is redirected to the add product page where the jeweler enters the product information and submits and the system adds the product details to the database and notifies the jeweler.

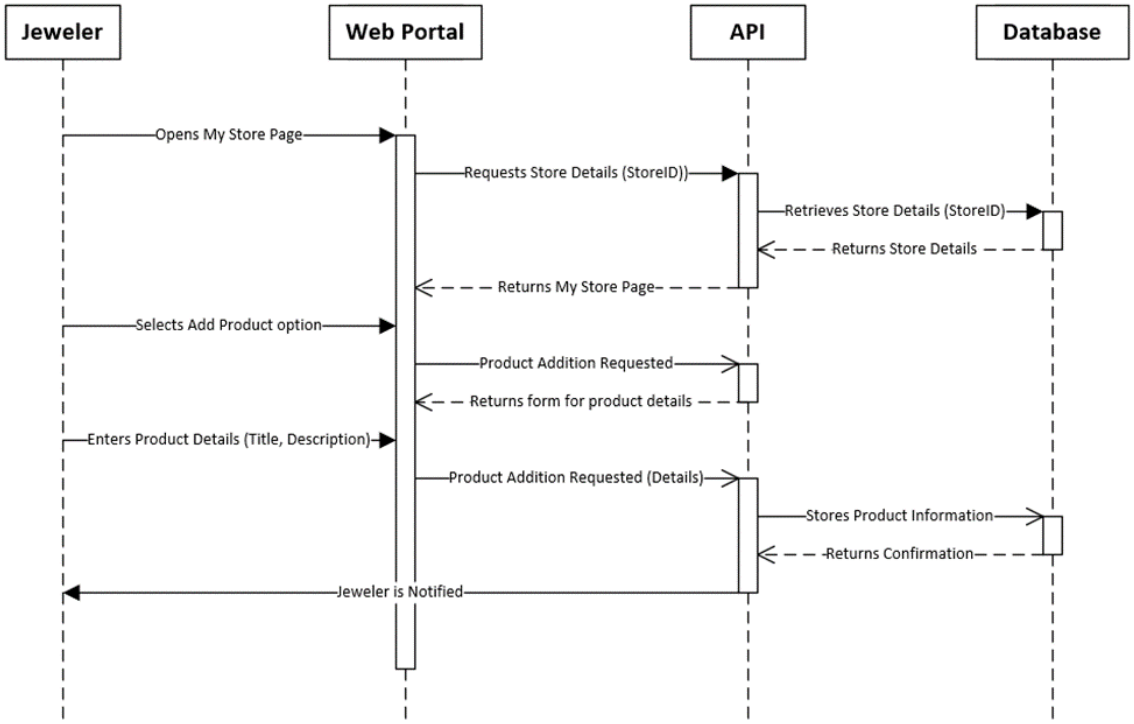

Figure 4.23: Add Products

4.2.24 Edit Products

The sequence diagram below show the interaction of jeweler and the application to edit product. The jeweler opens the store page and the system fetches jeweler store page from the database and displays it for the jeweler. Jeweler selects a specific product and the system fetches it from the database and displays it for the jeweler.Jeweler selects edit product and is redirected to the edit product page where jeweler makes changes to the product details and submits. The system updates the changes in the database and notifies the jeweler.

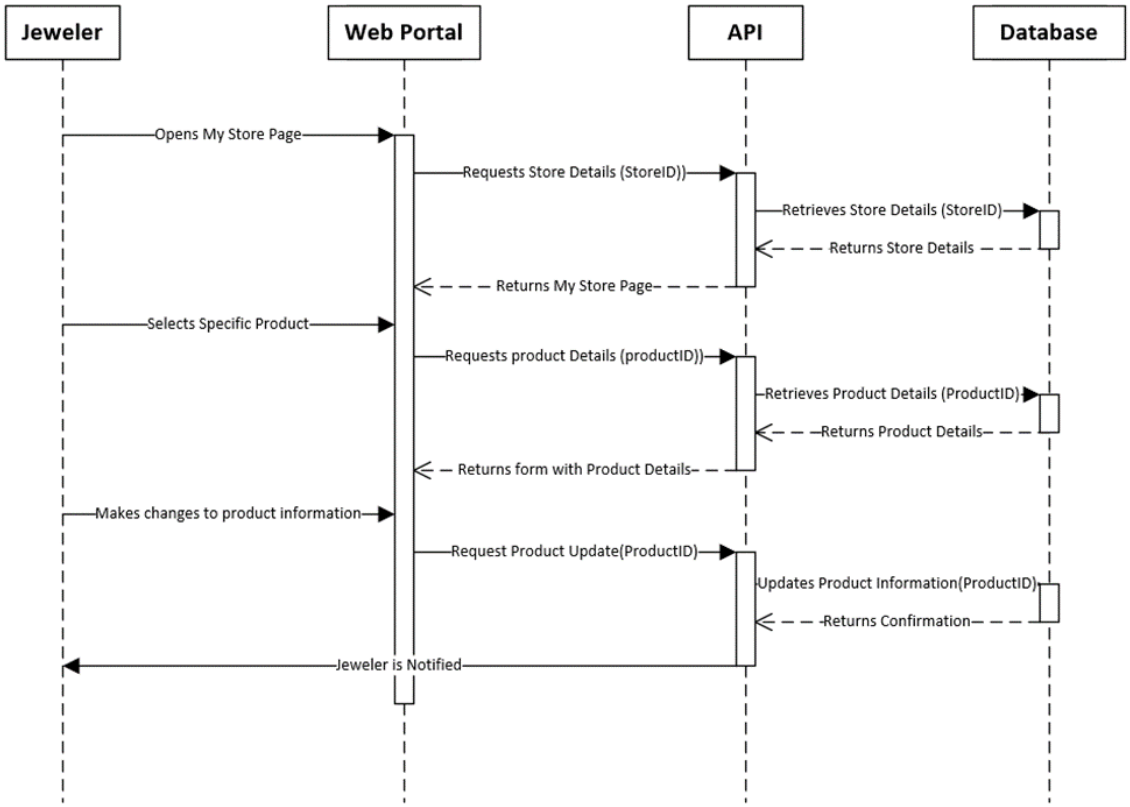

Figure 4.24: Edit Products

4.2.25 Delete Products

The sequence diagram below show the interaction of jeweler and the application to edit product. The jeweler opens the store page and the system fetches jeweler store page from the database and displays it for the jeweler. Jeweler selects a specific product and the system fetches it from the database and displays it for the jeweler.Jeweler selects delete product, the system deletes the product details in the database and notifies the jeweler.

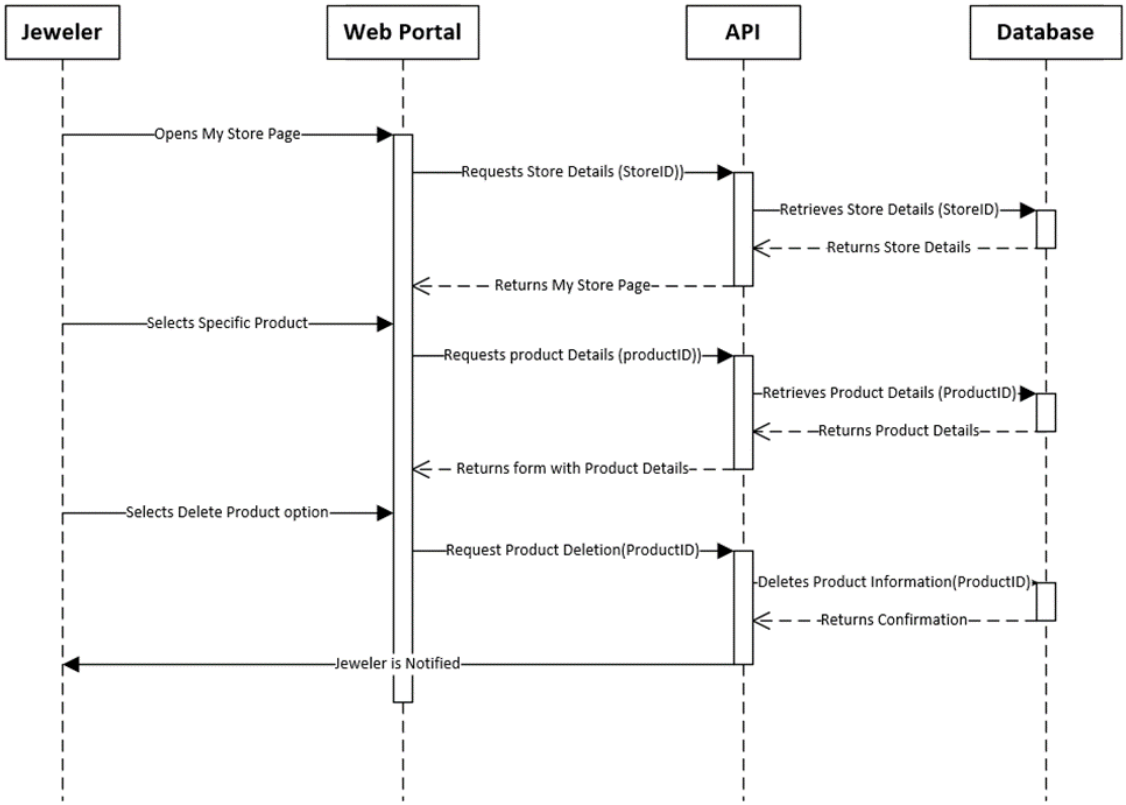

Figure 4.25: Delete Products

4.2.26 Request Commission Change

The sequence diagram below show the interaction of jeweler and the application to request for commission change. The jeweler opens the store page and the system fetches jeweler store page from the database and displays it for the jeweler. Jeweler selects change commission option and fills the required information such as reason and new commission and submits, the system stores the request in the database and notifies jeweler of its pending status.

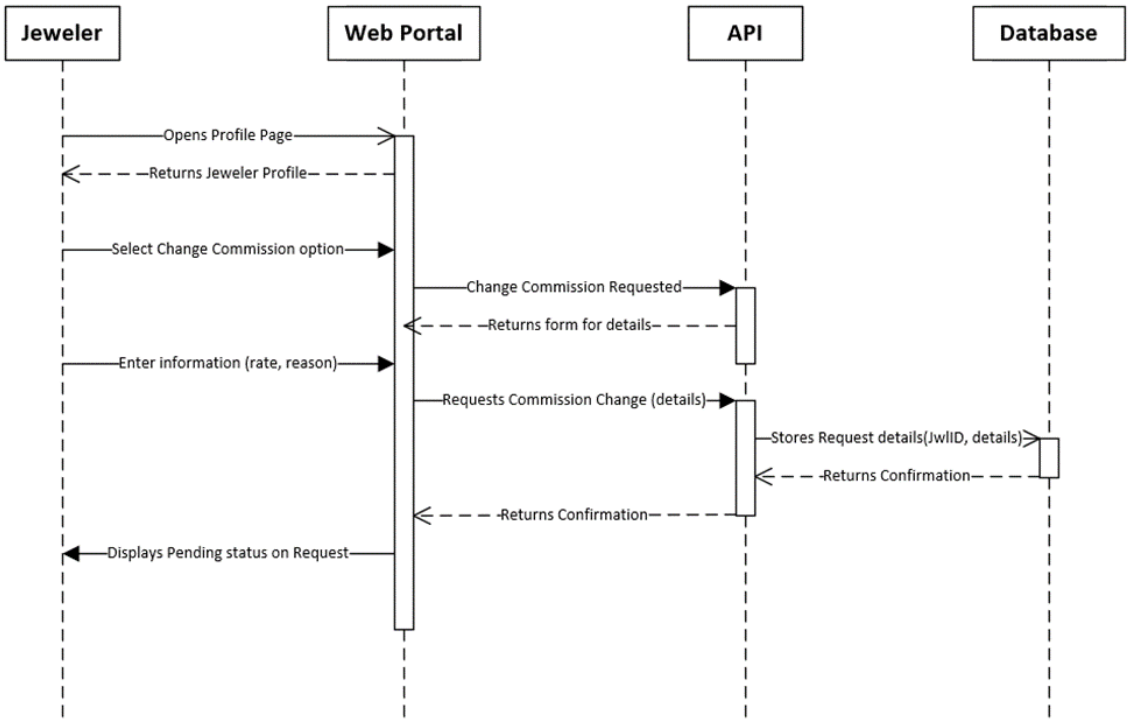

Figure 4.26: Request Commission Change

4.3 ERD Diagram

For the database, a NO-SQL based technology such as MONGODB will be leveraged. The ERD below provides a general illustration of what the schema of our database collections would look like.

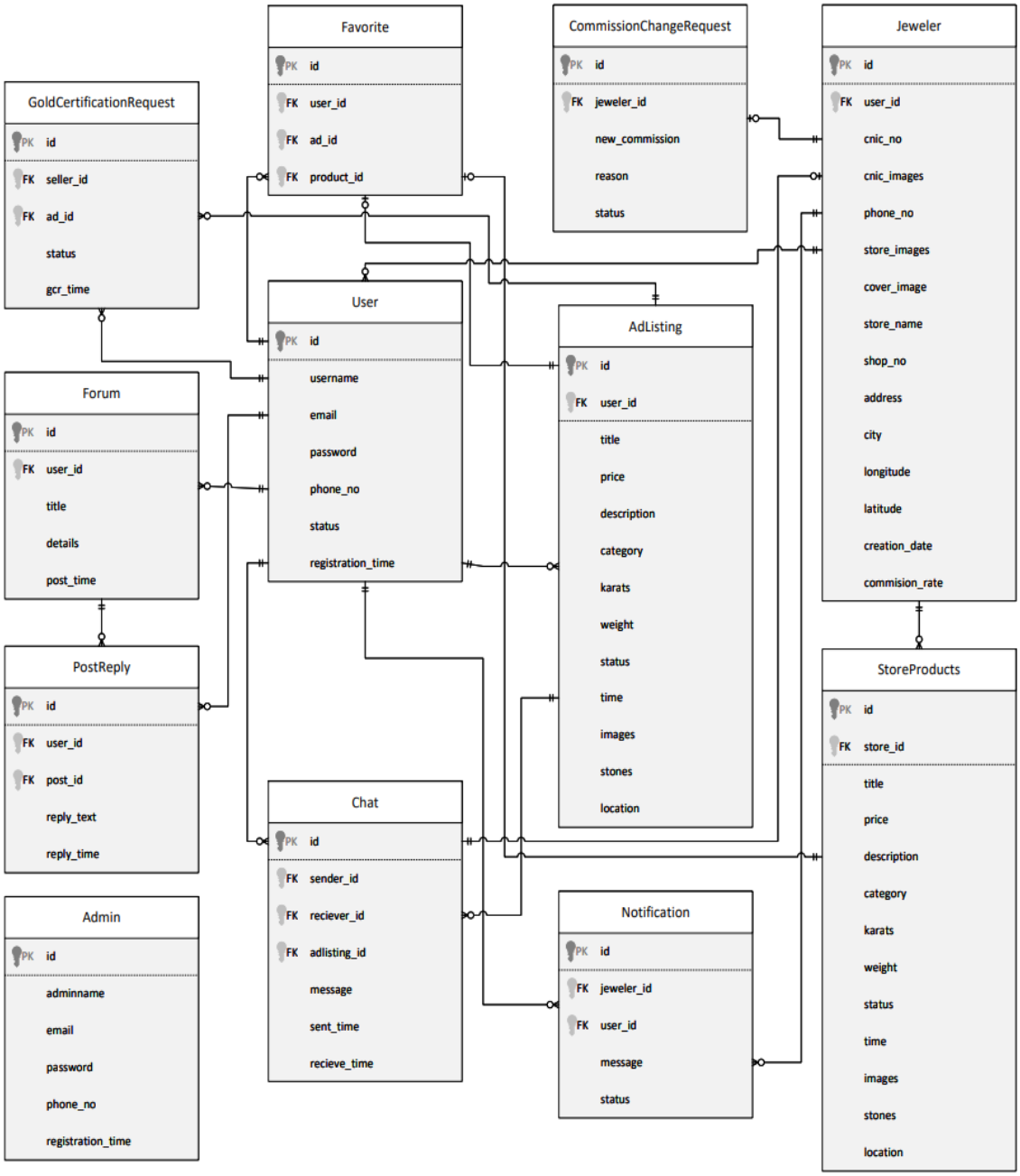

Figure 4.27: ERD Diagram

4.3.1 Entities and Relationship

The below diagram is an overview of the detailed ERD in section 4.3. This ERD is without the attributes for each entity and entities are enlarged to have a clear understanding of what the Entities are and what the relationship between them is as the original ERD was detailed and complex.

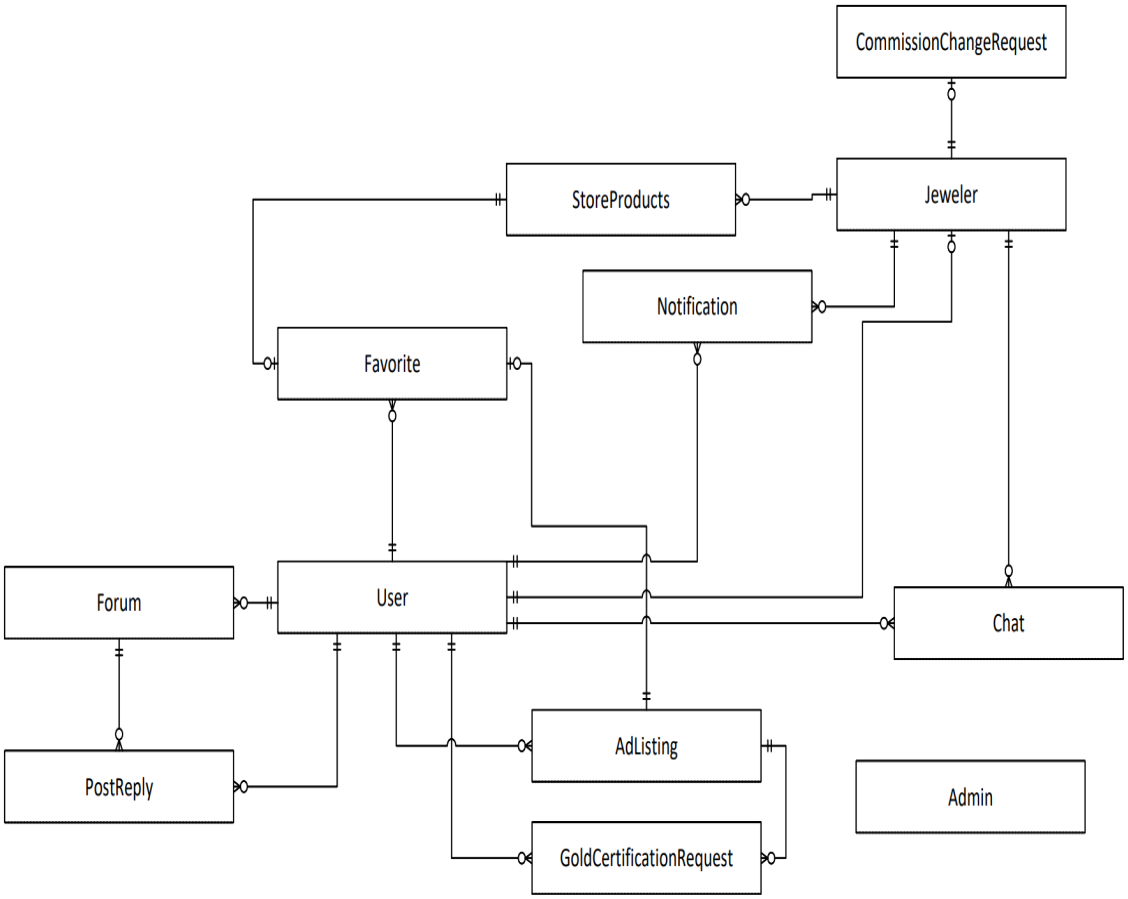

Figure 4.28: ERD Diagram (Entities Only)

4.3.2 Attribute Tables

This section includes the attribute tables for each entity included in the ERD in section 4.3 for a better idea of what the entities uphold to. The tables incorporate attribute names, their datatypes, lengths, whether they are null-able, their default value and any associated constraints with the attribute.

4.3.2.1 User

Table 4.1: User Attribute Table

Attribute Name	Data Type	Length	Nullable	Default Value	Constraints
id	VARCHAR	255	NO	-	PRIMARY KEY
username	VARCHAR	50	NO	-	-
email	VARCHAR	100	NO	-	UNIQUE
password	VARCHAR	50	NO	-	-
phone_no	VARCHAR	15	NO	-	UNIQUE
status	VARCHAR	20	NO	'user'	-
registration_time	DATETIME	-	NO	-	-

4.3.2.2 Admin

Table 4.2: Admin Attribute Table

Attribute Name	Data Type	Length	Nullable	Default Value	Constraints
id	VARCHAR	255	NO	-	PRIMARY KEY
adminname	VARCHAR	50	NO	-	-
email	VARCHAR	100	NO	-	UNIQUE
password	VARCHAR	50	NO	-	-
phone_no	VARCHAR	15	NO	-	UNIQUE
status	VARCHAR	20	NO	'admin'	-

4.3.2.3 Forum

Table 4.3: Forum Attribute Table

Attribute Name	Data Type	Length	Nullable	Default Value	Constraints
id	VARCHAR	255	NO	-	PRIMARY KEY
user_id	VARCHAR	255	NO	-	FOREIGN KEY
title	VARCHAR	255	NO	-	UNIQUE
details	VARCHAR	-	NO	-	-
post_time	DATETIME	-	NO	-	-

4.3.2.4 PostReply

Table 4.4: PostReply Attribute Table

Attribute Name	Data Type	Length	Nullable	Default Value	Constraints
id	VARCHAR	255	NO	-	PRIMARY KEY
user_id	VARCHAR	255	NO	-	FOREIGN KEY
post_id	VARCHAR	255	NO	-	FOREIGN KEY
reply_text	VARCHAR	-	NO	-	-
post_time	DATETIME	-	NO	-	-

4.3.2.5 Chat

Table 4.5: Chat Attribute Table

Attribute Name	Data Type	Length	Nullable	Default Value	Constraints
id	VARCHAR	255	NO	-	PRIMARY KEY
sender_id	VARCHAR	255	NO	-	FOREIGN KEY
reveiver_id	VARCHAR	255	NO	-	FOREIGN KEY
message	VARCHAR	-	NO	-	-
sent_time	DATETIME	-	NO	-	-
receive_time	DATETIME	-	YES	-	-

4.3.2.6 Gold Certification Request

Table 4.6: GoldCertificationRequest Attribute Table

Attribute Name	Data Type	Length	Nullable	Default Value	Constraints
id	VARCHAR	255	NO	-	PRIMARY KEY
seller_id	VARCHAR	255	NO	-	FOREIGN KEY
ad_id	VARCHAR	255	NO	-	FOREIGN KEY
status	VARCHAR	20	NO	'pending'	-
gcr_time	DATETIME	-	NO	-	-

4.3.2.7 Notification

Table 4.7: Notification Attribute Table

Attribute Name	Data Type	Length	Nullable	Default Value	Constraints
id	VARCHAR	255	NO	-	PRIMARY KEY
jeweler_id	VARCHAR	255	NO	-	FOREIGN KEY
user_id	VARCHAR	255	NO	-	FOREIGN KEY
message	VARCHAR	255	NO	-	-
status	VARCHAR	20	NO	'pending'	-

4.3.2.8 Commission Change Request

Table 4.8: CommissionChangeRequest Attribute Table

Attribute Name	Data Type	Length	Nullable	Default Value	Constraints
id	VARCHAR	255	NO	-	PRIMARY KEY
jeweler_id	VARCHAR	255	NO	-	FOREIGN KEY
new_commission	DECIMAL	(2,4)	NO	-	-
reason	VARCHAR	-	NO	-	-
status	VARCHAR	20	NO	'pending'	-
ccr_time	DATETIME	-	NO	-	-

4.3.2.9 Favorite

Table 4.9: Favorite Attribute Table

Attribute Name	Data Type	Length	Nullable	Default Value	Constraints
id	VARCHAR	255	NO	-	PRIMARY KEY
user_id	VARCHAR	255	NO	-	FOREIGN KEY
jeweler_id	VARCHAR	255	NO	-	FOREIGN KEY
ad_id	VARCHAR	255	NO	-	FOREIGN KEY

4.3.2.10 AdListing

Table 4.10: AdListing Attribute Table

Attribute Name	Data Type	Length	Nullable	Default Value	Constraints
id	VARCHAR	255	NO	-	PRIMARY KEY
user_id	VARCHAR	255	NO	-	FOREIGN KEY
title	VARCHAR	255	NO	-	-
price	DECIMAL	20	NO	-	-
description	VARCHAR	-	NO	-	-
category	VARCHAR	30	NO	-	-
karats	VARCHAR	10	NO	-	-
weight	DECIMAL	-	NO	-	-
stones	VARCHAR	50	YES	-	-
images	VARCHAR	-	NO	-	-
location	-	-	NO	-	-
status	VARCHAR	20	NO	'pending approval'	-
time	DATETIME	-	NO	-	-

4.3.2.11 Store Products

Table 4.11: StoreProducts Attribute Table

Attribute Name	Data Type	Length	Nullable	Default Value	Constraints
id	VARCHAR	255	NO	-	PRIMARY KEY
jeweler_id	VARCHAR	255	NO	-	FOREIGN KEY
title	VARCHAR	255	NO	-	-
price	DECIMAL	20	NO	-	-
description	VARCHAR	-	NO	-	-
category	VARCHAR	30	NO	-	-
karats	VARCHAR	10	NO	-	-
weight	DECIMAL	-	NO	-	-
stones	VARCHAR	50	YES	-	-
images	VARCHAR	-	NO	-	-
location	-	-	NO	-	-
status	VARCHAR	20	NO	'pending approval'	-
time	DATETIME	-	NO	-	-

4.3.2.12 Jeweler

Table 4.12: Jeweler Attribute Table

Attribute Name	Data Type	Length	Nullable	Default Value	Constraints
id	VARCHAR	255	NO	-	PRIMARY KEY
user_id	VARCHAR	255	NO	-	FOREIGN KEY
cnicno	VARCHAR	13	NO	-	-
cnic_images	VARCHAR	-	NO	-	-
phone_no	VARCHAR	11	NO	-	-
cover_image	VARCHAR	-	NO	-	-
store_images	VARCHAR	-	NO	-	-
store_name	VARCHAR	100	NO	-	-
shop_no	VARCHAR	50	NO	-	-
address	VARCHAR	255	NO	-	-
location	-	-	NO	-	-
creation_time	DATETIME	-	NO	-	-
commission_rate	DECIMAL	(2,4)	NO	2.5	-

4.4 Flow Diagram

The flow diagram below depicts the complete flow of the system including each actor. First is the guest that can access general views and forum and once registered becomes the user which can log in and act as a buyer and seller by placing ads and interacting with other ads. User can request for his status to be elevated to a jeweler who can manage his own page along with providing certification services to users for commission and all these processes are overlooked by the admin.

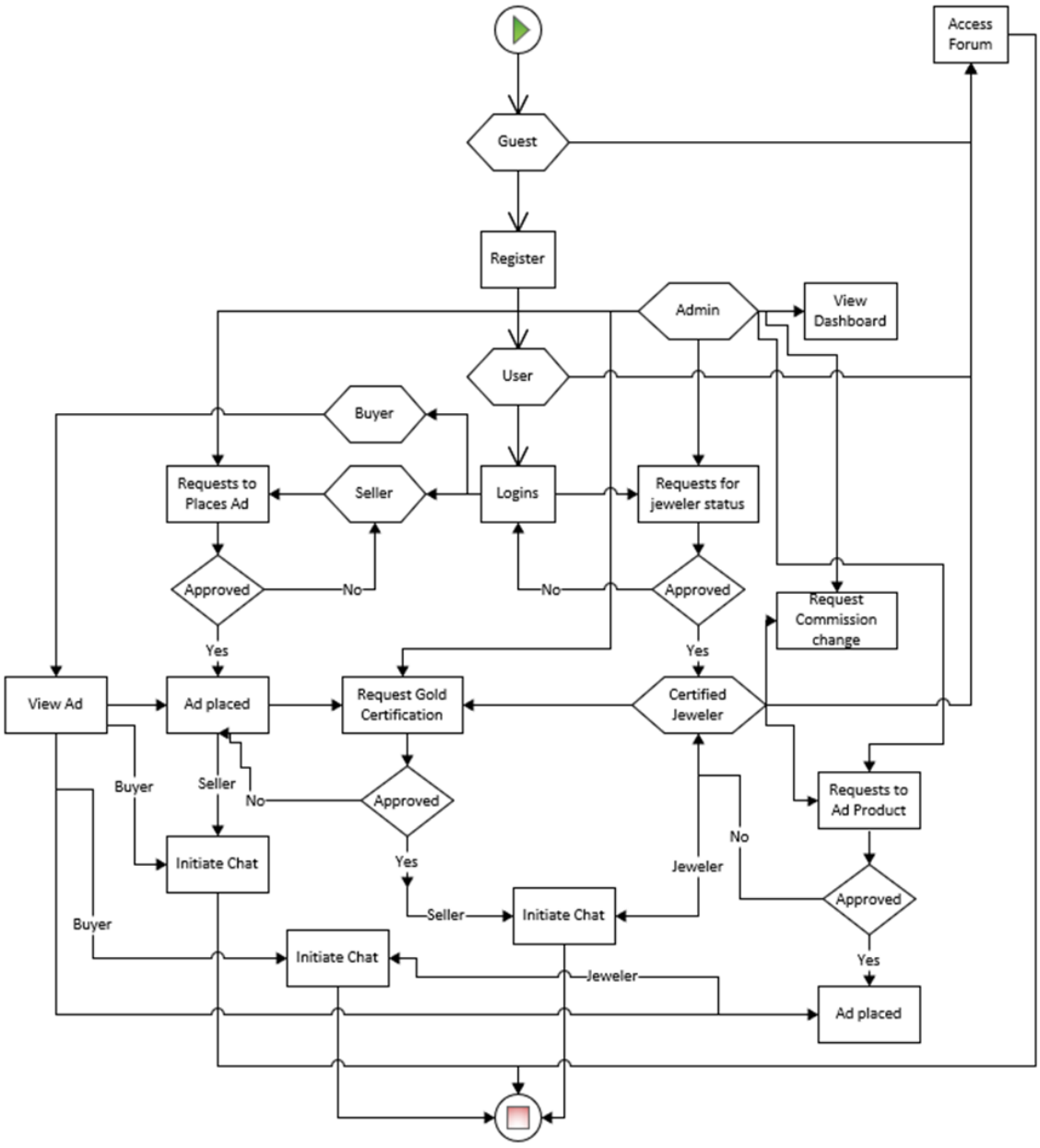

Figure 4.29: Flow Diagram

4.5 Activity Diagram

Swim-lane activity diagrams are a great way of understanding system working. The first diagram in section 4.5.1 depicts how a guest can elevate his status to user (buyer/seller) and jeweler and admin's role in it. The second diagram in section 4.5.2 depicts the process of how users buy and sell and what role the admin and the certified jewelers play in it.

4.5.1 Actor Process

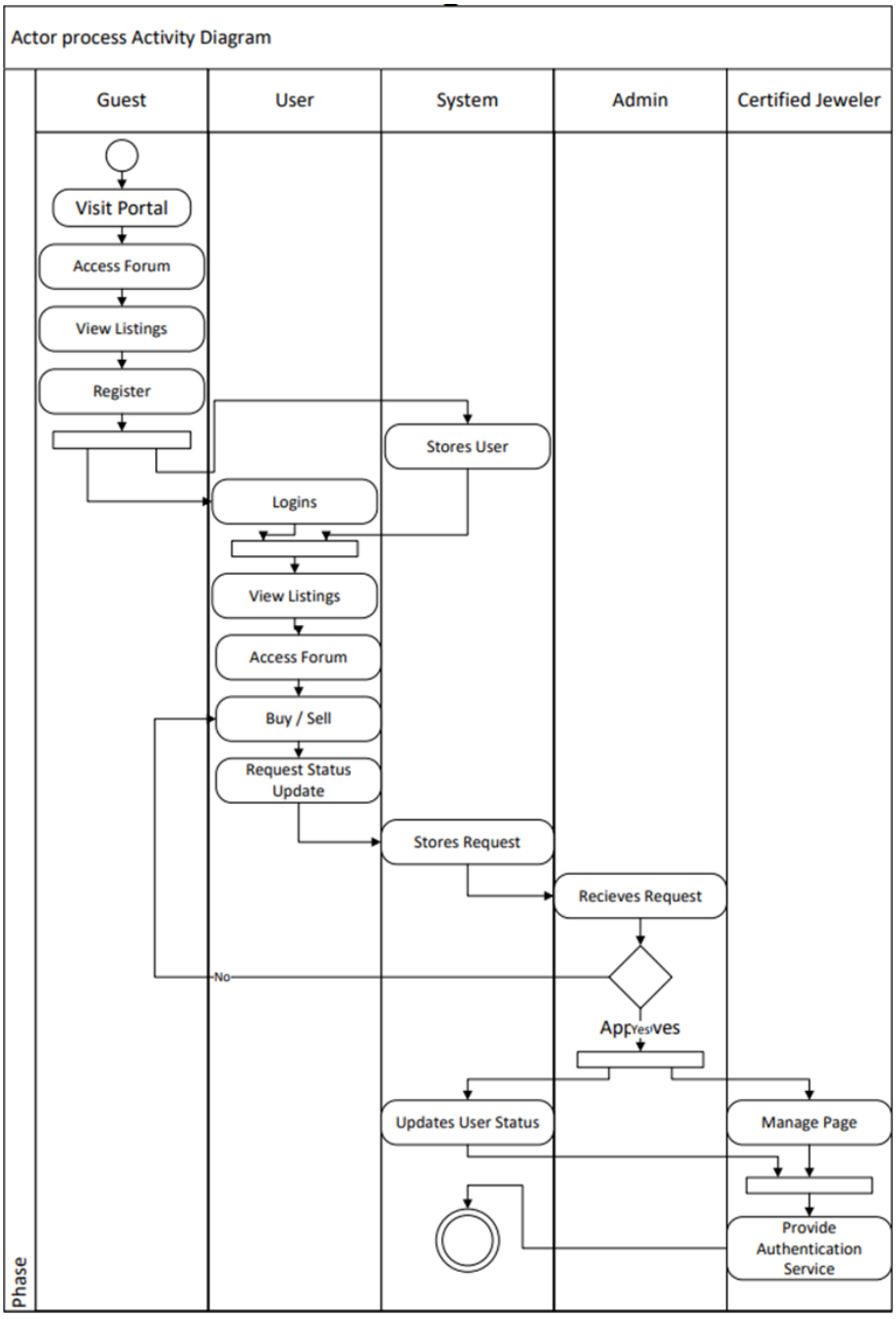

Figure 4.30: Actor Process Activity Diagram

4.5.2 Buy / Sell Process

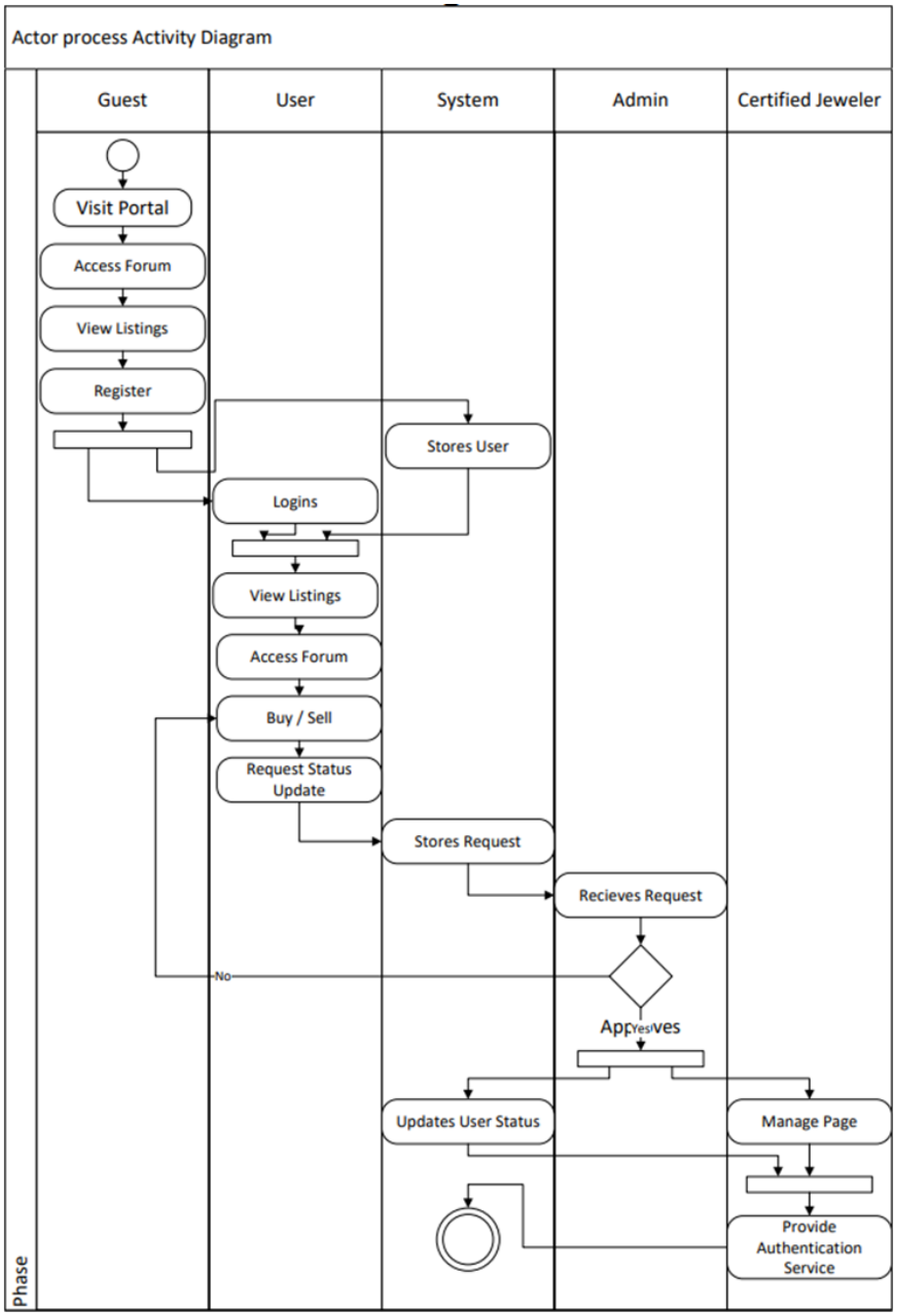

Figure 4.31: Buy / Sell Process Activity Diagram

4.6 WireFrames

Wireframes provide an idea for the layout and how the application User Interface would look after development. This helps to know and define the areas and sections of each page and helps us identify any functionality that is to be added in the view.

4.6.1 Registration Page

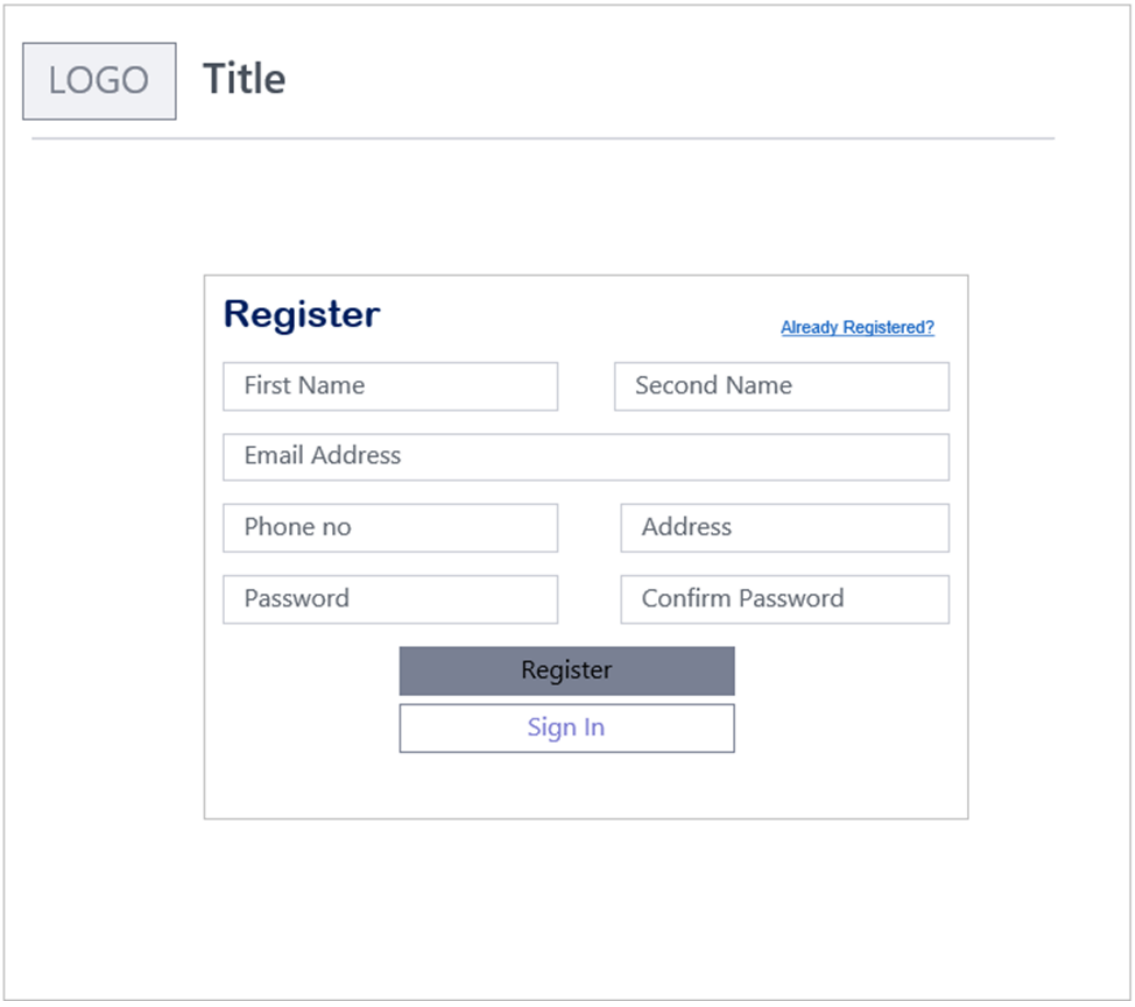

Figure 4.32: Registration Page Wireframe

The wireframe above is for the registration page when a guest first comes on the platform. It depicts the logo and title in the top left corner in the navbar along with a registration form in the middle. The registration form consists of required fields, a register button and options if the user is already registered.

4.6.2 Login Page

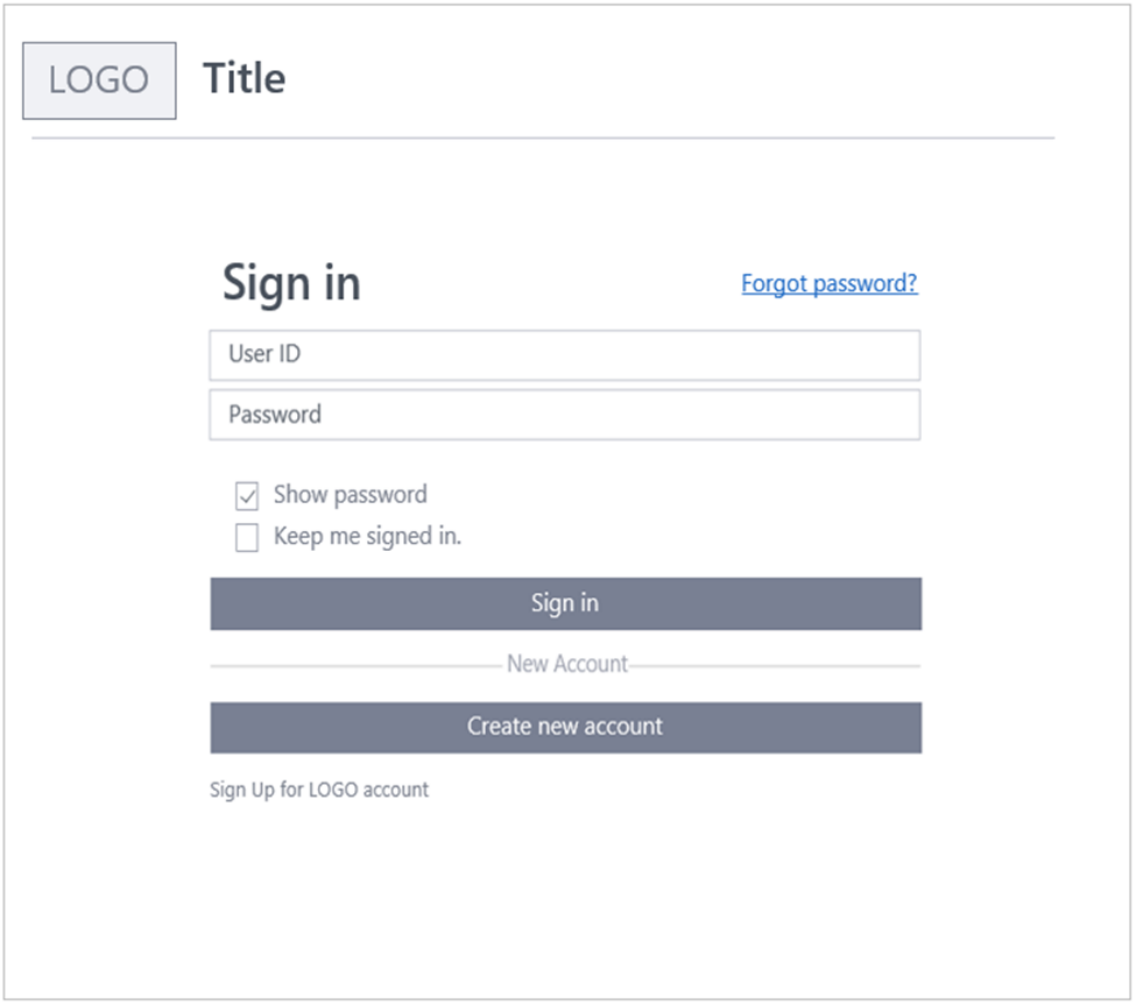

Figure 4.33: Login Page Wireframe

The wireframe above is for the Login page when a registered user wishes to log back in to the platform. It depicts the logo and title in the top left corner in the navbar along with a login form in the middle. The login form consists of required fields, a sign-in button, options to remember the user, for forgot password and options if the user is new and wishes to register.

4.6.3 Home Page

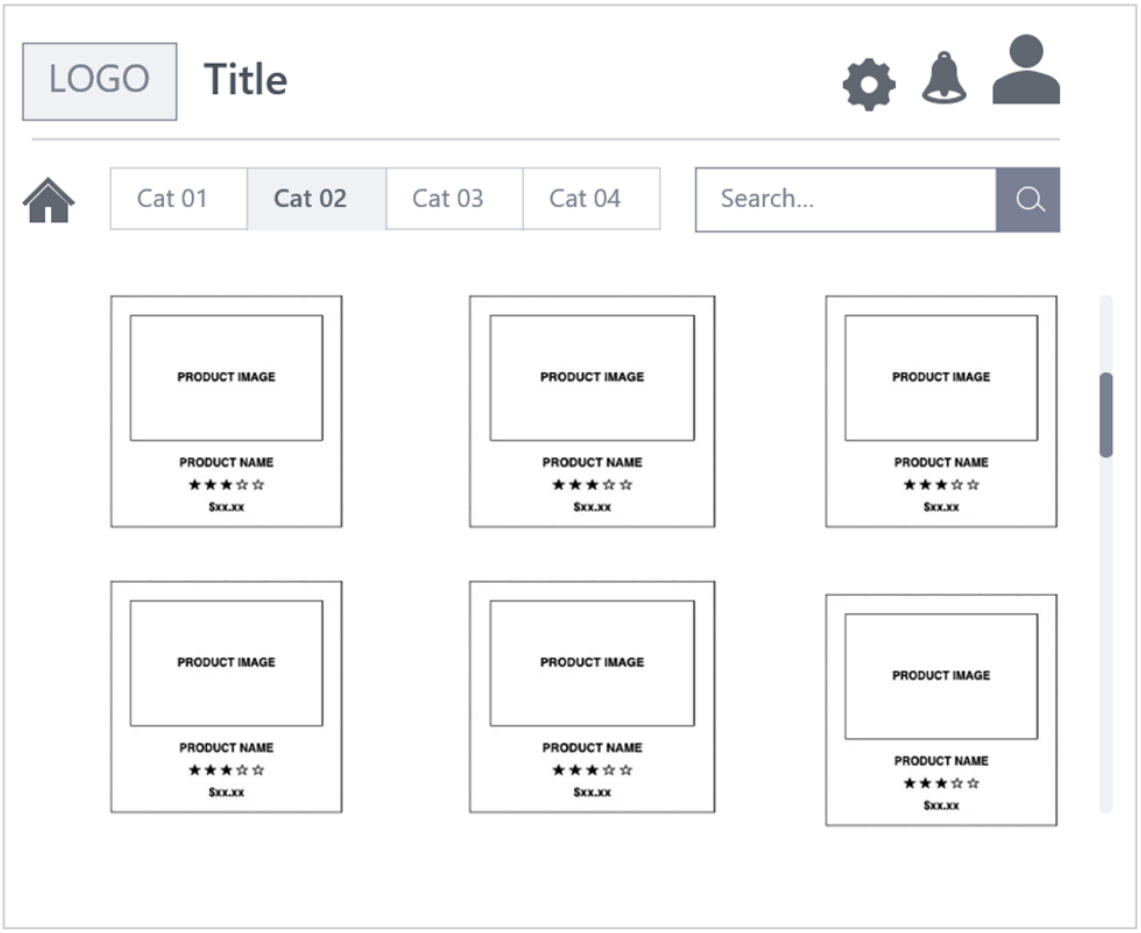

Figure 4.34: Home Page Wireframe

The wireframe above is for the Landing page when a registered user logs in the system. It depicts the logo and title in the top left corner in the navbar along with settings, notifications and account icons on the right. The page consists of options to choose different categories and a search bar to filter listings based on keywords along with some of the latest listings that the users can interact with.

4.6.4 User Profile Page

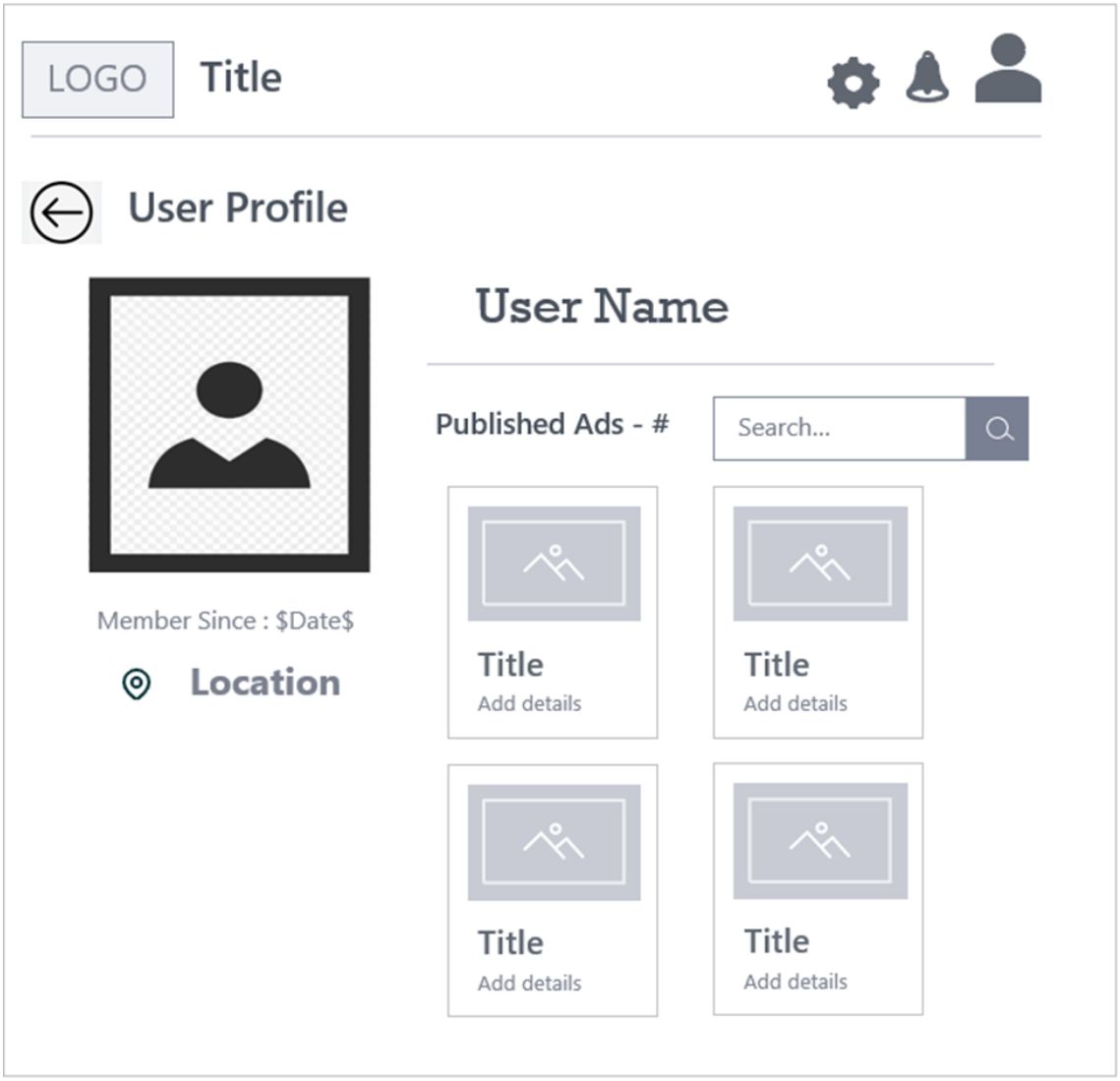

Figure 4.35: User Profile Page Wireframe

The wireframe above is for the User Profile page when a registered user selects my profile in account options. It depicts the logo and title in the top left corner in the navbar along with settings, notifications and account icons on the right. The page consists of user profile image, user name, location, active ads and a search field to search for a specific ad.

4.6.5 Jeweler Page

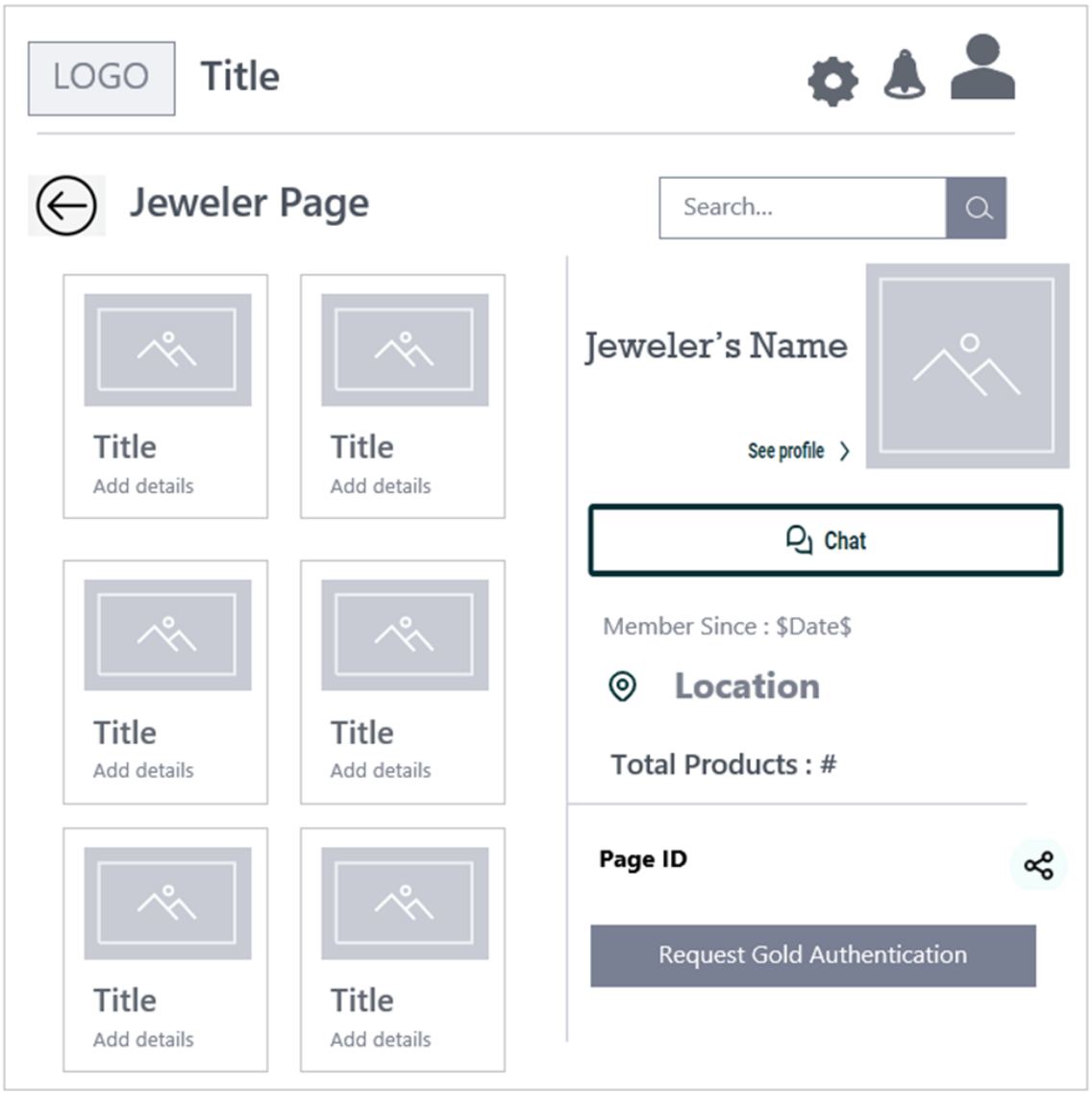

Figure 4.36: Jeweler Page Wireframe

The wireframe above is for the Jeweler Page when a registered user visits a jeweler page. It depicts the logo and title in the top left corner in the navbar along with settings, notifications and account icons on the right. The page consists of jeweler information along with added products including a search field to filter specific products.

4.6.6 Jeweler Profile Page

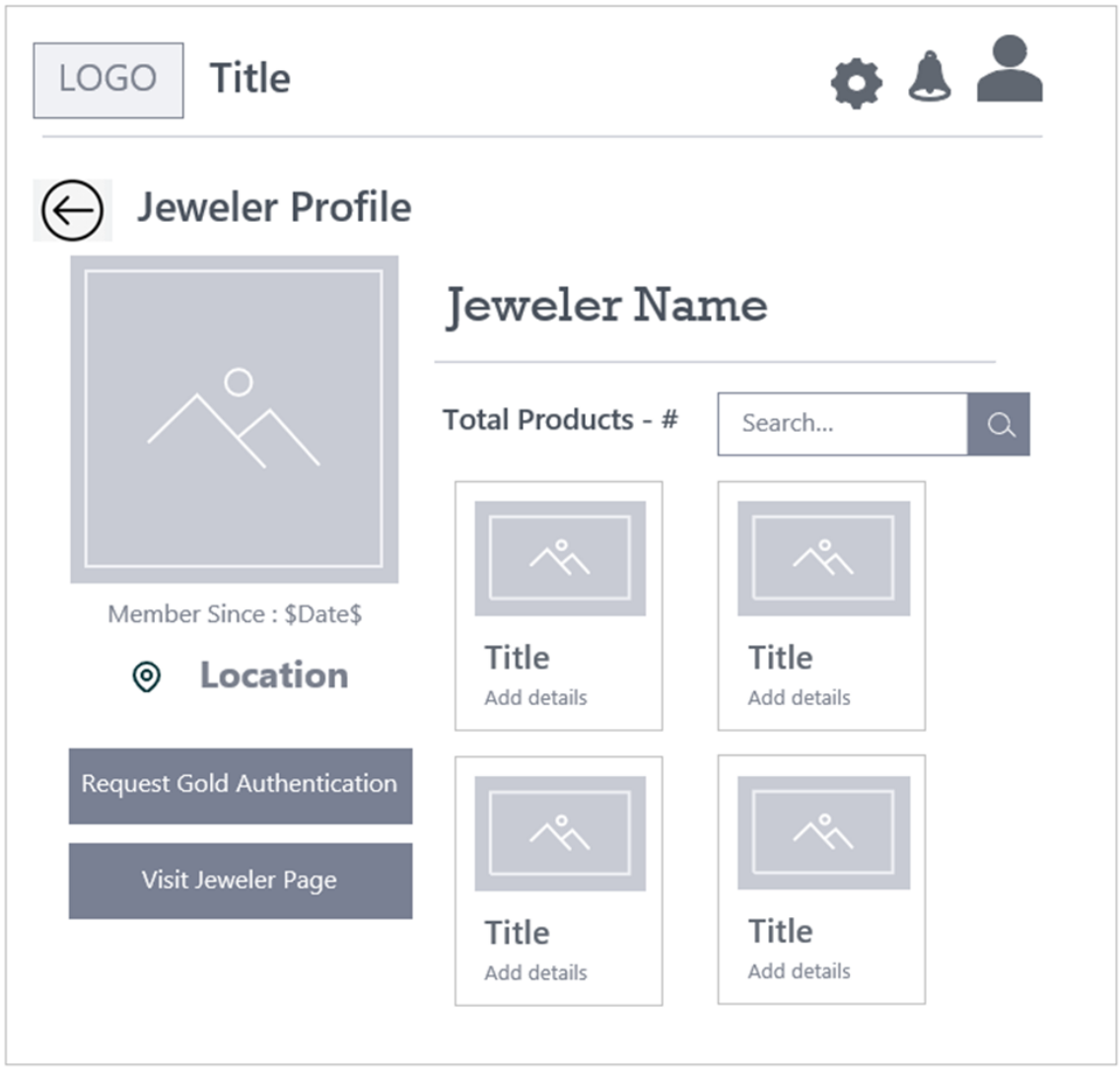

Figure 4.37: Jeweler Profile Page Wireframe

The wireframe above is for the Jeweler Profile page when a registered user selects to view jeweler profile. It depicts the logo and title in the top left corner in the navbar along with settings, notifications and account icons on the right. The page consists of jeweler profile image, jeweler name, location, active ads and a search field to search for a specific ad.

4.6.7 Forum Page

Figure 4.38: Forum Page Wireframe

The wireframe above is for the Forum page when a guest or registered user visits the forum. It depicts the logo and title in the top left corner in the navbar along with log in and sign up buttons on the right. The page consists of User posts along with search field to filter out specific posts.

4.6.8 Place Ad Page

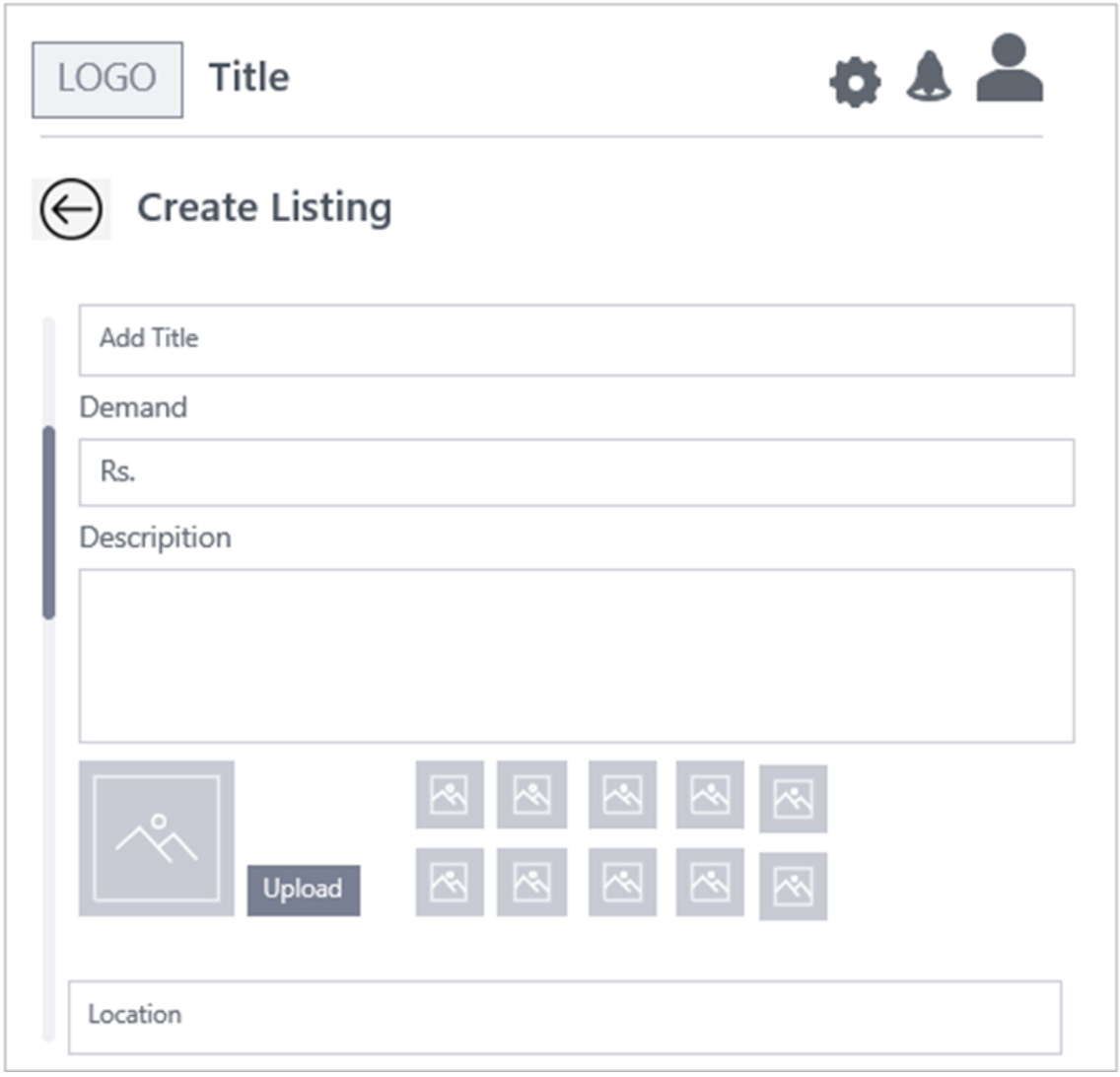

Figure 4.39: Place Ad Page Wireframe

The wireframe above is for the place ad page when a registered user selects to sell his gold. It depicts the logo and title in the top left corner in the navbar along with settings, notifications and account icons on the right. The page consists of a form requiring the user to input information regarding his listing such as title, price, description and images etc.

4.6.9 Ad Details Page

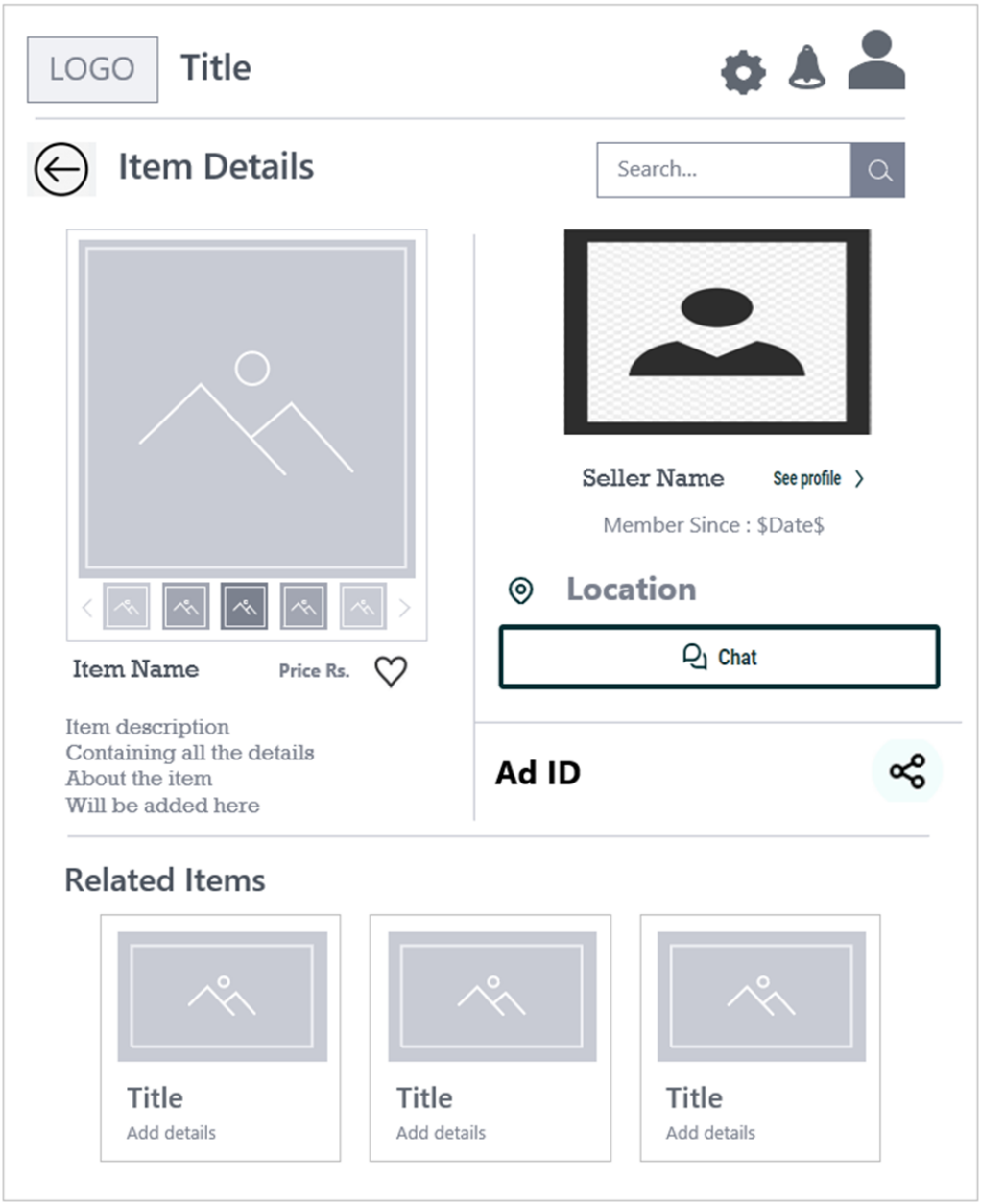

Figure 4.40: Ad Details Page Wireframe

The wireframe above is for the Ad details page when a registered user selects a specific ad to view its details. It depicts the logo and title in the top left corner in the navbar along with settings, notifications and account icons on the right. The page consists of seller profile image, seller name, location, option to chat and time since member. The item details comprises of the images, item name, description, add to favorite button, add ID and similar ads at the end of the page.

Chapter 5

System Implementation

5.1 System Architecture

System architecture represents the system application as it's a high-level logical representation of the system. It includes the internal and external components of the system.

Our proposed system comprises of one main component, "Web Application" which further consists of several internal components.

5.2 Web Application

In our Web Application, EGold Haven, users ranging from individual sellers to certified jewelers find a comprehensive platform tailored for the transparent and efficient trading of gold jewelry. Designed to accommodate a diverse user base, including buyers, sellers, certified jewelers, and industry professionals, EGold Haven offers a seamless experience for all participants. Sellers benefit from the ability to set prices independently without intermediary fees, empowering them to maximize profits and reach a broader audience. On the other hand, buyers enjoy access to gold jewelry at competitive prices, coupled with the assurance of authenticity provided by certified jewelers. Moreover, certified jewelers find an avenue to expand their market reach and earn commissions for verifying the quality of listed items. EGold Haven not only simplifies the gold jewelry trading process but also fosters trust and transparency within the industry, ultimately revolutionizing how gold jewelry is bought and sold online.

5.3 Tools and Technologies

The following tools and technologies are being used in our proposed system.

5.3.1 Visual Studio Code

A lightweight and powerful code editor used for frontend and backend development.

5.3.2 MongoDB

A NoSQL database management system utilized for data storage and retrieval.

5.3.3 Express.js

A minimalist web application framework for Node.js, handling server-side logic and routing.

5.3.4 React.js

A JavaScript library for building user interfaces, facilitating the creation of interactive and dynamic frontend components.

5.3.5 Node.js

A JavaScript runtime environment for executing server-side code, providing scalability and performance for backend operations.

5.3.6 MVC Pattern

The system architecture of EGold Haven Web Application is structured around the MVC pattern [10], leveraging the MERN (MongoDB, Express.js, React.js, Node.js) stack for development. This architecture facilitates the creation of a scalable, robust, and maintainable web application tailored for gold jewelry trading.

5.3.6.1 Model (M)

The model component of our architecture represents the data layer, where MongoDB serves as the database management system. MongoDB stores various types of data, including user profiles,jeweler profiles, gold item listings, certifications and forum discussions.

5.3.6.2 View (V)

The view component is implemented using React.js, a JavaScript library for building user interfaces. React.js facilitates the creation of interactive and responsive UI components, ensuring an engaging user experience. Views are responsible for rendering data retrieved from the back-end and handling user interactions.

5.3.6.3 Controller (C)

Express.js, a minimalist web application framework for Node.js, acts as the controller layer in our architecture. Express.js handles HTTP requests from clients, routes them to the appropriate endpoints, and interacts with the model layer to retrieve or manipulate data. It also manages the flow of data between the model and view components.

5.4 System Internal Components

5.4.1 Front End Components

5.4.1.1 React Components

React.js components form the frontend architecture, comprising various UI elements such as user authentication forms,jeweler registration,product listings, chat, item listings, certification requests, forum discussions and admin dashboard etc. These components facilitate user interactions and display dynamic content based on data retrieved from the backend.

5.4.1.2 Redux State Management

Redux is utilized for state management in our application, enabling efficient data flow between React components and ensuring consistent application state. This functionality is being used for seamless management of user authentication and status changes of the users.

5.4.2 Back End Components

5.4.2.1 Express.js Middleware

Express.js middleware functions handle request preprocessing, authentication, authorization, error handling, and other cross-cutting concerns. Authentication middleware verifies user credentials during login and ensures authorized access to protected endpoints. Additionally, middleware functions handle validation and processing of data submitted by users.

5.4.2.2 Mongoose ORM

Mongoose is used as an Object-Document Mapping (ODM) library for MongoDB, providing a structured schema definition and simplifying database interactions in the Node.js environment. Mongoose models define the structure of user profiles, gold items, gold products, jeweler profiles, certification records, and forum discussions, enabling CRUD operations on these entities.

5.4.2.3 API Endpoints

Express.js defines RESTful API endpoints for various operations, including user authentication forms,jeweler registration,product listings, chat, item listings, certification requests, forum discussions and admin dashboard etc. These endpoints handle incoming requests from the frontend, execute corresponding business logic, and interact with the database through Mongoose ORM to retrieve or modify data. Functionality such as user registration, login, logout, item listing creation, certification requests etc. are implemented through these endpoints.

5.4.3 Database Layer

5.4.3.1 MongoDB Database

MongoDB stores structured data required for the functioning of the web application. Collections within MongoDB store user profiles,jeweler profiles, gold item details, gold product details, certification records, notification, chat and forum discussions. Functionality related to these collections are managed within MongoDB, ensuring data integrity and scalability.

5.4.4 Security Measures

In our commitment to maintaining robust security standards, we implement advanced encryption techniques and authentication protocols within our system. Through password encryption and the utilization of JSON Web Tokens (JWT), we ensure the protection of user data and secure access to our platform. Let's delve into the specifics of our security measures.

5.4.4.1 Password Encryption

To safeguard user credentials, we utilize bcrypt hashing algorithms for password encryption. This approach ensures that passwords are securely stored within our system, mitigating the risk of unauthorized access, even in the event of a data breach.

5.4.4.2 JWT Tokens

Our system leverages JSON Web Tokens (JWT) for secure user authentication and authorization. By employing JWT tokens, we establish a reliable mechanism for verifying user identity and permissions, enhancing the overall security of our platform.

5.5 Functionalities Integrated into Internal Components

5.5.1 User Profiles and Authentication

Managed through Express.js middleware,users can register, login and user authentication ensures secure access to the platform's features. Authentication status and session management are handled via token-based authentication mechanisms.

5.5.2 Item Listing

Administered through Express.js API endpoints, item listing functionality enables users to create, view, edit, and delete gold jewelry listings. CRUD operations on listings are executed via API requests, with validation and authorization checks enforced by middleware.

5.5.3 Jeweler Profiles and Product Pages

Jewelers can create personalized profiles showcasing their craftsmanship and manage individual store pages to display their gold jewelry offerings. Users can browse jeweler profiles and product pages to explore unique designs and make informed purchasing decisions.

5.5.4 Certification Management

Implemented within Express.js endpoints, certification management functionality empowers certified jewelers to verify the authenticity and quality of listed gold items. Certification requests are processed through API endpoints, with updates reflected in the database.

5.5.5 Forum Interactions

Facilitated by Express.js endpoints, forum interaction functionality allows users to engage in discussions and share insights within the community. Posts and replies are stored in the MongoDB database, with CRUD operations managed through API endpoints.

5.5.6 Chat and Notifications

Users can communicate via real-time chat functionality integrated into the platform, enabling direct communication between buyers, sellers, and certified jewelers. Additionally, users receive notifications for various actions, such as new messages, item updates, and forum activity, enhancing user engagement and interaction.

5.5.7 Admin Dashboard

The Admin Dashboard provides administrators with tools to manage user accounts, oversee certification requests, monitor forum activity, and handle other administrative tasks. Requests related to user management and platform administration are efficiently managed through the Admin Dashboard interface.

5.6 Development Model

The system is evolved through the Incremental Model. In this model, continuous changes are discovered, and they are implemented during development.

5.6.1 Stage 1: Project Initiation and Research

During this initial stage, we conduct thorough research on the domain of gold jewelry trading in Pakistan. This includes understanding market trends, identifying user needs and preferences, and analyzing existing platforms. Project goals and objectives are established, and the scope of the system is defined based on the research findings.

5.6.2 Stage 2: System Design and Prototyping

In this stage, the system architecture and design are conceptualized based on the identified requirements. Wire frames and mock-ups are created to visualize the user interface and user experience. A prototype is developed to demonstrate key functionalities and gather feedback from stakeholders. Testing of the prototype is conducted to validate the design and ensure it aligns with user expectations.

5.6.3 Stage 3: Incremental Development and Testing

During this iterative development phase, the system is built incrementally, with new features and enhancements added in each iteration. Agile development methodologies are employed to facilitate collaboration and adaptability. Continuous testing is carried out to identify and address any bugs or issues early in the development process. User acceptance testing is conducted to validate the system against user requirements and ensure its readiness for deployment.

Chapter 6

Testing

6.1 System Testing

System testing is a critical phase in the software development life-cycle [11], focusing on evaluating the entire system as a whole to ensure that it meets specified requirements and functions as intended. This phase involves testing the integrated system components, functionalities, and interactions to uncover defects, validate system behavior, and verify compliance with user expectations.

6.1.1 Testing Techniques

- Functional Testing
- Compatibility Testing
- Performance Testing
- Security Testing
- Usability Testing
- User Acceptance Testing
- Regression Testing
- Load Testing
- Stress Testing
- Boundary Testing

6.2 Functional testing

Functional testing involves validating the system's functional requirements by executing test cases that verify whether each feature performs as intended. Test cases are designed to assess specific functionalities or user interactions, ensuring that the system behaves correctly and produces the expected outputs for various input scenarios. Through functional testing and the execution of test cases, software testers can identify defects, validate system behavior, and ensure that the software meets the specified functional requirements.

6.3 Test Cases

6.3.1 User Registration

Table 6.1: User Registration Test Case

Test Case ID	EGH-TC1
Test Title	User Registration
Test Case Description	This test case verifies the overall functionality of the user registration process, ensuring that users can successfully register an account and receive confirmation.
Pre-Conditions	User should have internet connection and is on the home page.
Test Steps	1. Navigate to the registration page. 2. Fill out the registration form with valid information (e.g., username, email, phoneno, password and confirm password). 3. Submit the registration form. 4. Verify that a confirmation message is displayed indicating successful registration. 5. Attempt to log in using the newly registered credentials. 6. Verify that login is successful and the user is redirected to the landing page.
Expected Results	The user should be able to register an account successfully, receive a confirmation prompt, and log in without any issues.
Result Status	PASS

6.3.2 User Login

Table 6.2: User Login Test Case

Test Case ID	EGH-TC2
Test Title	User Login
Test Case Description	This test case verifies the overall functionality of the user registration process, ensuring that users can successfully register an account and receive confirmation.
Pre-Conditions	User must be registered on the platform.
Test Steps	1. Navigate to the login page. 2. Enter valid login credentials (email/phoneno and password). 3. Click on the "Login" or "Sign In" button. 4. Verify that the system redirects the user to the landing page after successful login. 5. Attempt to log in with invalid credentials (e.g., incorrect password, non-existent phoneno/email). 6. Verify that appropriate error messages are displayed, and login is not successful.
Expected Results	For valid credentials the user should be redirected to the landing page. For invalid credentials the system should show error message and prevent login.
Result Status	PASS

6.3.3 SSO User Login

Table 6.3: SSO User Login Test Case

Test Case ID	EGH-TC3
Test Title	SSO User Login
Test Case Description	This test case verifies the functionality of Single Sign-On (SSO) login, allowing users to access the application/system using their SSO credentials.
Pre-Conditions	SSO integration is enabled and configured and User has valid SSO credentials.
Test Steps	1. Navigate to the login page. 2. Click on the "Login with Single Sign-On (SSO)" option. 3. Select valid account to process SSO Login.
Expected Results	User should be successfully authenticated via SSO. Upon successful authentication, the user should be logged in to the application/system.
Result Status	PASS

6.3.4 Reset Password

Table 6.4: Reset Password Test Case

Test Case ID	EGH-TC4
Test Title	Reset Password
Test Case Description	This test case verifies the functionality of the password reset process when initiated via an email link.
Pre-Conditions	The user must have a registered account with a valid email address.
Test Steps	1. Navigate to the forgot password page. 2. Enter valid email. 3. Click on the "Send Token" button. 4. User receives a link on his email and clicks on it. 5. User is directed to the reset password page. 6. User enters new password and confirm password. 7. User clicks on "reset password" button. 8. User is prompted about password reset success and is redirected to the login page.
Expected Results	User is prompted about successful password reset and is redirected to the login page.
Result Status	PASS

6.3.5 Edit User Profile

Table 6.5: Edit User Profile Test Case

Test Case ID	EGH-TC5
Test Title	Edit User Profile
Test Case Description	This test case verifies the functionality of editing user profile information.
Pre-Conditions	The user is logged in and is on the edit profile page.
Test Steps	1. Click on the profile icon to access the user profile. 2. Click on the "Edit Profile" option to navigate to the edit profile page. 3. Make changes to the information fields, such as name, email and phoneno. 4. Submit the updated information.
Expected Results	1. The system should validate the information entered by the user. 2. Upon successful validation, the system should update the user's profile information in the database. 3. The system should redirect the user to the profile page. 4. The profile page should display the updated information.
Result Status	PASS

6.3.6 Place Ad

Table 6.6: Place Ad Test Case

Test Case ID	EGH-TC6
Test Title	Place Ad
Test Case Description	This test case verifies the functionality of placing an advertisement for a user's gold on the platform.
Pre-Conditions	The user is logged in.
Test Steps	1. Navigate to the "Place Ad" page. 2. Enter the required information for the ad, including title, price, description, category, and images. 3. Submit the ad for approval. 4. Wait for ad approval.
Expected Results	1. The system should validate the ad's information for completeness and proper conventions. 2. Upon successful validation, the system should add the ad to the database and notify the user of its creation. 3. User should be redirected to the "My Ads" page.
Result Status	PASS

6.3.7 Edit Ad

Table 6.7: Edit Ad Test Case

Test Case ID	EGH-TC7
Test Title	Edit Ad
Test Case Description	This test case verifies the functionality of editing an ad placed by the user.
Pre-Conditions	The user is logged in.
Test Steps	1. Click on the profile icon and navigate to the "My Ads" page. 2. Select the specific ad to edit. 3. Make the desired changes to the ad. 4. Submit the edited ad.
Expected Results	1. The system should perform validations on the new information. 2. Upon successful validation, the system should update the approved ad in the database and notify the user. 3. User should be redirected to the "My Ads" page.
Result Status	PASS

6.3.8 Delete Ad

Table 6.8: Delete Ad Test Case

Test Case ID	EGH-TC8
Test Title	Delete Ad
Test Case Description	This test case verifies the functionality of deleting an ad placed by the user.
Pre-Conditions	The user is logged in.
Test Steps	1. Open the specified ad from the "My Ads" page. 2. Choose the "Delete Ad" option and confirm the action.
Expected Results	1. The system should ask for confirmation before deleting the ad. 2. Upon confirmation, the system should delete the ad from the database. 3. User should be redirected to the "My Ads" page.
Result Status	PASS

6.3.9 Request for Jeweler Status

Table 6.9: Request for Jeweler Status Test Case

Test Case ID	EGH-TC9
Test Title	Request for Jeweler Status
Test Case Description	This test case verifies the functionality of requesting jeweler status update from user to jeweler.
Pre-Conditions	The user is logged in and is on the home/landing page
Test Steps	1. Navigate to the landing page. 2. Select the option to request jeweler status update. 3. Enter the required information, including CNIC, phone number, CNIC images, store images and address etc. 4. Submit the request for approval.
Expected Results	1. The system should validate the entered information. 2. Upon successful validation, the system should send the request for admin approval. 3. User should be redirected to the landing page and prompted to wait for approval notification.
Result Status	PASS

6.3.10 Search Listings

Table 6.10: Search Listings Test Case

Test Case ID	EGH-TC10
Test Title	Search Listings
Test Case Description	This test case verifies the functionality of searching for specific items through all the approved listings on the platform.
Pre-Conditions	The user is logged in and is on the listings page
Test Steps	1. Navigate to the listings page. 2. Search for a specific item or choose a specific category or specification. 3. Verify the updated view to see the listings that match the search criteria.
Expected Results	1. The system should search and retrieve the listings from the database based on the user's search query or selected category. 2. The view should be refreshed, and the retrieved listings should be dynamically loaded on the listings page. 3. User should be able to browse listings based on their preference.
Result Status	PASS

6.3.11 View Listing Details

Table 6.11: View Listing Details Test Case

Test Case ID	EGH-TC11
Test Title	View Listing Details
Test Case Description	This test case verifies the functionality of viewing the details of a specific listing while browsing through listings.
Pre-Conditions	The user is logged in.
Test Steps	1. Navigate to the landing or listings page. 2. Click on a specific listing. 3. Verify that the user lands on a page with detailed listing information, including title, price, description, category, and images.
Expected Results	1. The system should respond to the user's request and retrieve the detailed information about the specific listing. 2. The system should dynamically load the page with the retrieved data, displaying the listing's details. 3. User should be able to view all relevant information about the listing.
Result Status	PASS

6.3.12 Set Favorite

Table 6.12: Set Favorite Test Case

Test Case ID	EGH-TC12
Test Title	Set Favorite
Test Case Description	This test case verifies the functionality of setting favorite listings by the user, allowing them to view these listings later on the "My Favorites" page.
Pre-Conditions	The user is logged in.
Test Steps	1. Navigate to the landing or listings page. 2. Browse through the listings and click to view details. 3. Like a specific listing by clicking on the "Favorite" button.
Expected Results	1. The system should add the liked listing to the user's favorite listings in the database. 2. User should be able to view the liked listing later on the "My Favorites" page.
Result Status	PASS

6.3.13 View Dashboard

Table 6.13: View Dashboard Test Case

Test Case ID	EGH-TC13
Test Title	View Dashboard
Test Case Description	This test case verifies the functionality of the admin viewing statistics of the web portal, including information such as total ads placed, ads sold, and pending requests along with jeweler statistics.
Pre-Conditions	Admin is logged in.
Test Steps	1. Admin selects the "View Statistics" option from their dashboard to refresh stats. 2. Admin is presented with various options for different statistics of the portal. 3. Admin selects a specific option to view detailed information (e.g., live ads, expired ads, registered users, registered jewelers).
Expected Results	1. The system requests the selected items' information from the database. 2. The system populates the view with the fetched information for the admin to view. 3. Admin is presented with the details of the particular item they selected and option to accept or reject the request.
Result Status	PASS

6.3.14 Gold Certification Request

Table 6.14: Gold Certification Request Test Case

Test Case ID	EGH-TC14
Test Title	Gold Certification Request
Test Case Description	This test case verifies the functionality of the user requesting a jeweler on the platform to authenticate their gold for a specific ad.
Pre-Conditions	User is logged in and has a live ad.
Test Steps	1. User opens up the "My Listings" tab from their profile menu. 2. User selects the specific ad for which they want to request gold certification. 3. User clicks on the request certification button. 4. User selects a jeweler and a request is sent to them.
Expected Results	1. The system tracks the request and keeps it in the database until dealt with. 2. A request is sent to the selected jeweler, and the system maintains a chat between the seller and jeweler, recording it in the database. 3. Seller is prompted to wait for the jeweler's response.
Result Status	PASS

6.3.15 Approve/Reject Jeweler Status Request

Table 6.15: Approve/Reject Jeweler Status Request Test Case

Test Case ID	EGH-TC15
Test Title	Approve/Reject Jeweler Status Request
Test Case Description	This test case verifies the functionality of an admin approving or rejecting the status update request of a user to jeweler.
Pre-Conditions	Admin is logged in and is on the inspect jeweler requests page.
Test Steps	1. Admin logs in and lands on the dashboard. 2. Admin navigates to the pending jeweler requests page. 3. Admin views individual requests one by one and validates the information provided in the request. 4. Admin approves or rejects the request.
Expected Results	1. The system responds by retrieving the requests from the database and showing them to the admin. 2. Admin approves or rejects the request. 3. The system discards the request or updates the status in the database based on the admin's decision.
Result Status	PASS

6.3.16 Approve/Reject Ads

Table 6.16: Approve/Reject Ads Test Case

Test Case ID	EGH-TC16
Test Title	Approve/Reject Ads
Test Case Description	This test case verifies the functionality of an admin approving or rejecting ad posting requests submitted by users.
Pre-Conditions	Admin is logged in and is on the inspect listing requests page.
Test Steps	1. Admin logs in and lands on the dashboard. 2. Admin navigates to the pending listing requests page. 3. Admin views individual requests one by one and validates the information provided in the request. 4. Admin approves or rejects the request.
Expected Results	1. The system responds by retrieving the requests from the database and showing them to the admin. 2. Admin approves or rejects the request. 3. The system discards the request or adds the ad to published ads in the database based on the admin's decision.
Result Status	PASS

6.3.17 Initiate Chat

Table 6.17: Initiate Chat Test Case

Test Case ID	EGH-TC17
Test Title	Initiate Chat
Test Case Description	This test case verifies the functionality of initiating a chat between users or between a user and a jeweler.
Pre-Conditions	Both the users (buyers/sellers) are logged in.
Test Steps	1. User (buyer) views listing details. 2. User selects the chat option on the ad. 3. User sends a message to the other user (seller). 4. User (seller) receives the message and replies. 5. User (buyer) receives the reply and can continue the chat. 6. Both users can view this chat along with the chat history.
Expected Results	1. The system responds by keeping a record of the chat in the database. 2. User (seller) receives a notification and replies to the chat. 3. The system also retrieves the previous history of chat from the database and shows it to the users.
Result Status	PASS

6.3.18 Approve/Reject Certification Request

Table 6.18: Approve/Reject Certification Request Test Case

Test Case ID	EGH-TC18
Test Title	Approve/Reject Certification Request
Test Case Description	This test case verifies the functionality of a jeweler approving or rejecting a request to authenticate a particular seller's gold.
Pre-Conditions	Jeweler is logged in and is on the manage store page.
Test Steps	1. Jeweler opens up "Pending Requests" from their page. 2. Jeweler opens each request one by one and views the ad. 3. Jeweler selects "Approve" or "Disapprove".
Expected Results	1. In case of approve , chat is initiated between the jeweler and the seller. 2. In case of disapprove the seller should be notified that the request has been declined.
Result Status	PASS

6.3.19 Add Product

Table 6.19: Add Product Test Case

Test Case ID	EGH-TC19
Test Title	Add Product
Test Case Description	This test case verifies the functionality of a registered jeweler adding products to their allotted store page.
Pre-Conditions	Jeweler is logged in and is on the manage store page.
Test Steps	1. Jeweler selects add a product option from their page. 2. Jeweler fills the required information to add a product and submits.
Expected Results	1. The system validates the information of the product that the jeweler entered. 2. System records the information in the database under the jeweler's store page. 3. Jeweler is on their My Store page and can view the product listed.
Result Status	PASS

6.3.20 Edit Product

Table 6.20: Edit Product Test Case

Test Case ID	EGH-TC20
Test Title	Edit Product
Test Case Description	This test case verifies the functionality of a registered jeweler editing products already listed on their store page.
Pre-Conditions	Jeweler is logged in and is on the manage store page.
Test Steps	1. Jeweler selects a product to view its details. 2. Jeweler clicks on the edit product button. 3. Jeweler makes the desired changes and submits.
Expected Results	1. The system validates the new information that is entered. 2. System updates the new product information in the database. 3. The product information is sent again to the admin for approval. 4. Jeweler is on their My Store page and can view the product listing with updated info and pending status.
Result Status	PASS

6.3.21 Delete Product

Table 6.21: Delete Product Test Case

Test Case ID	EGH-TC21
Test Title	Delete Product
Test Case Description	This test case verifies the functionality of a registered jeweler deleting products already listed on their store page.
Pre-Conditions	Jeweler is logged in and is on the manage store page.
Test Steps	1. Jeweler selects a product to view its details. 2. Jeweler clicks on the delete product button. 3. Jeweler confirms the action.
Expected Results	1. The system asks for confirmation to delete. 2. In case of yes, the product is deleted from the database and jeweler is redirected to the store page. 3. In case of no, the deletion process is cancelled and product detail page stays open.
Result Status	PASS

Chapter 7

Conclusion

Our Gold Haven platform has been meticulously designed and developed to cater to the needs of both buyers and sellers in the gold jewelry market. With a focus on transparency, trust, and efficiency, our platform aims to revolutionize the way gold jewelry is traded and certified in Pakistan.

Throughout the development process, we have prioritized user experience and security, ensuring that both buyers and sellers can transact with confidence. By leveraging cutting-edge technologies such as the MERN stack and robust encryption techniques, we have built a secure and user-friendly platform that meets the highest industry standards.

One of the key features of our platform is its user-friendly interface, which allows buyers to browse through a wide range of gold jewelry listings with ease. Sellers, on the other hand, benefit from a streamlined process for listing their products and managing their store pages. Additionally, our certification management system empowers certified jewelers to verify the authenticity and quality of listed gold items, enhancing trust and credibility within the marketplace.

Looking ahead, we remain committed to continuous improvement and innovation. We recognize the importance of staying abreast of evolving market trends and user preferences, and we are dedicated to incorporating feedback from our users to enhance the platform further. Our goal is to establish Gold Haven as the premier destination for buying and selling gold jewelry in Pakistan, offering unparalleled transparency, trust, and convenience to all stakeholders.

7.1 Future Works

In the future we can work on the following aspects.

- Enhanced user experience to implement user feedback to refine the platform's interface and usability, ensuring a seamless and intuitive browsing experience for all users.
- Exploring the integration of artificial intelligence (AI) technologies such as natural language processing for sentiment analysis [12] in forum discussions and chat interactions.
- Expanding the platform's reach and offerings by on-boarding more certified jewelers and diversifying product categories beyond gold jewelry.
- Developing a dedicated mobile application for EGold Haven to cater to the growing number of users accessing the platform via mobile devices.

References

[1] Xiaocheng Ge, Richard F Paige, Fiona AC Polack, Howard Chivers, and Phillip J Brooke. Agile development of secure web applications. In *Proceedings of the 6th international conference on Web engineering*, pages 305–312, 2006.

[2] Muhammad Ali Jamil. Scope, associated risks, and major platforms of e-commerce in pakistan. 2022.

[3] Burçak Ertimur. *Gold and gold jewelry: Exploration of consumer practices*. Bilkent Universitesi (Turkey), 2003.

[4] KK Ramachandran and Dayana Sajjanan. New age gold jewellery consumers'in the online platform. *Abhinav International Monthly Refereed Journal of Research in Management & Technology*, 4(10), 2015.

[5] Farrah Arif and Sarah Suneel Sarfraz. *Pakwheels.com—A Period of Transition!*, pages 149–184. Springer Singapore, Singapore, 2019.

[6] Maria João Fernandes Abreu. New shopping trends: internet, second-hand trade and olx. Master's thesis, Universidade Catolica Portuguesa (Portugal), 2014.

[7] Robert T Plant. *eCommerce: formulation of strategy*. Prentice Hall Professional, 2000.

[8] Yannis Bakos. The emerging landscape for retail e-commerce. *Journal of economic perspectives*, 15(1):69–80, 2001.

[9] CMK De Silva, AS De Silva, KA I Maduwantha, DAIU Dewpura, DI De Silva, and RRP De Zoysa. Efficiency of an e-commerce web application with mern stack and modern tools. *International Journal of Engineering and Management Research*, 12(5):349–356, 2022.

[10] James Bucanek. Model-view-controller pattern. *Learn Objective-C for Java Developers*, pages 353–402, 2009.

[11] Alan M. Davis, Edward H. Bersoff, and Edward R. Comer. A strategy for comparing alternative software development life cycle models. *IEEE Transactions on software Engineering*, 14(10):1453–1461, 1988.

[12] Wei Yen Chong, Bhawani Selvaretnam, and Lay-Ki Soon. Natural language processing for sentiment analysis: an exploratory analysis on tweets. In *2014 4th international conference on artificial intelligence with applications in engineering and technology*, pages 212–217. IEEE, 2014.