

List of Figures

Figure 2.2.1 SDLC	4
Figure 3.6.1 Requirement understanding	4
Figure 3.7.1 Gantt Chart.....	4
Figure 4.6 Key Actors	4
Figure 4.8 Case Scenarios	4
Figure 4.8.9 Use case overview.....	4
Figure 5.1.4 Database and schema design.....	4
Figure 5.1.5 Class diagrams	4
Figure 6.1.1 Login screen admin	4
Figure 6.1.2 Login screen user	4
Figure 6.1.3 Admin frontend page.....	4
Figure 6.1.4 Admin product module.	4
Figure.6.1.5 Admin site setting page.....	4
Figure.6.1.6 Customer login Screen (Account Details).....	4
Figure.6.1.7 Customer address book	4
Figure.6.1.8 Cart details	4
Figure.6.1.9 Order history	4
Figure.6.1.10 Customer order track.....	4
Figure.6.1.11 Admin product details manage.....	4
Figure.6.1.12 manage images	4
Figure.6.1.13 manage brand	4
Figure.7.1.1 Testing product module.....	4
Figure.7.1.2 Testing encoder	4
Figure.7.1.3 Testing adding users.....	4

List of Tables

Table 5.1 Hardware Requirements	4
Table 5.2 Software Requirements	4
Table 6.2.1 Role Table	4
Table 7.2.2 Test Cases	4

List of Abbreviations

UI	User Interface
SDLC	Software development life cycles
SP	Service Providers
STS	Spring tool suite
REST	Represent full state transfer
API	Application programming Interface

TABLE OF CONTENT

Certificate.....	I
GTU Completion Certificate.....	II
Offer Letter Company	II
Company Certificate	III
Declaration.....	IV
Acknowledgement	V
Abstract	VI
List of Figures	VII
List of Tables	VIII
List of Abbreviations	IX
Table of Contents.....	X
Chapter-1: About the company	1
1.1 About.....	1
1.2 What we do?	1
1.3 Industries.....	2
1.4 Mission.....	2
1.5 Vision.....	2
1.6 Website	3
Chapter-2: Internship Overview	4
2.1 Objectives	4
2.2 Prepare schematic layout for the flow	4
2.3 Software/tools table...	8

Chapter-3:Introduction to internship and management.....	11
3.1 Internship work and along side project.....	12
3.1.1 PROJECT SUMMARY.....	13
3.1.2 PURPOSE.....	14
3.1.3 OBJECTIVE	16
3.1.4 Scope.....	18
3.1.5 Technology and Review	26
3.1.6 Project Effort and Time	29
3.1.7 Different roles	30
3.1.8 Project Scheduling	31
Chapter-4:System Analysis.....	11
4.1 Study of the current system.....	57
4.2 Problems and weakness of current system.....	57
4.3 System feasibility	57
4.4 Activity in the new system.....	57
4.5 Features of the new system	58
4.6 List main modules of the system	60
4.7 Understand the requirements	60
4.8 Case scenarios	62
4.9 Functional requirements.....	66
Chapter-5:System Analysis and Design.....	11
5.0 Selection of Hardware.....	69
5.1 Selection of Database Schema	71

5.2 Project Setup	69
5.3 Project Architecture overview	71
5.4 Class diagrams	69
Chapter-6: Implementation / Environment	73
6.1 Implementation platform.....	73
6.2 Work	75
Chapter-7:Testing	73
7.1 Test snapshots	73
7.2 Test results and analysis.....	75
Chapter-8: Conclusion	77
Chapter-9: References.....	78
Chapter-10 : Mentor's Feedback.	79
Chapter-11 : Internal Guide Feedback.....	80

Internship Overview

2.1 Objective:

- To gain practical experience in software development by working with a team of experienced software developers and participating in various development projects.
- To develop skills in programming languages such as Java, Spring boot ,Rest Api, Spring MVC , security concepts and frontend.
- To learn about software development life cycle (SDLC) phases, including requirements gathering, design, development, testing, and deployment.
- To understand software design principles and patterns, such as SOLID principles, MVC, and modular coding.
- To enhance communication and teamwork skills by collaborating with colleagues on various development projects and presenting ideas to senior management.
- To gain Understanding the work environment for specific Technology.
- To learn work principles of a professional environment and to enhance my expertise in Backend development field.
- Analyzing my proficiency in the professional environment and seeking new areas for improvement.
- To learn work principles of a professional environment and getting experience by working in a project.

2.2 PREPARE SCHEMATIC LAYOUT WHICH SHOWS THE SEQUENCE OF OPERATION FOR MANUFACTURING OF END PRODUCT.

The production is carried out in the following steps

1. Planning
2. Analysis
3. Design
4. Implementation
5. Testing and Integration
6. Maintenance

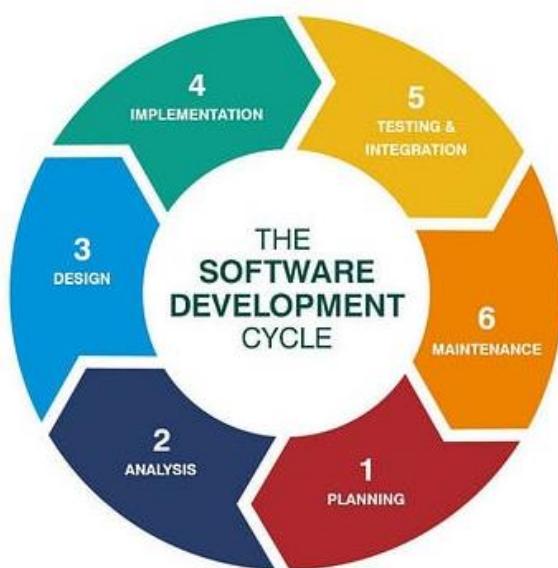


Figure 2.2.1 SDLC

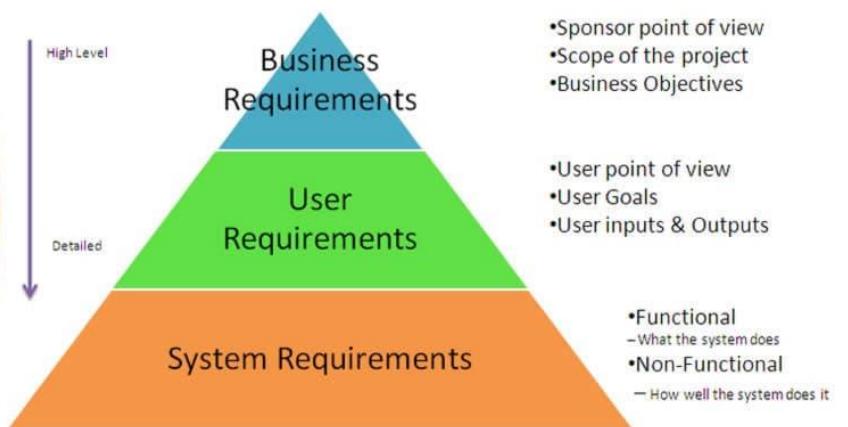


Figure 2.2.2 Requirements understanding

2.3 EXPLAIN IN DETAIL ABOUT EACH STAGE OF PRODUCTION.

1. Requirement Gathering and Analysis

We have collected all the information regarding the project. Once requirement gathering is done, an analysis is done to check the feasibility of the development of a product. Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created. This document should be thoroughly understood by the developers and also be reviewed by the customer.

2. Design

In this phase, the requirement gathered in the document is used as input and the web application architecture that is used for implementing system development is derived. I have designed all the pages like service requirements from client and different roles , Client interaction page, etc.

3. Implementing or Coding

Implementation/Coding started according to the requirement. The web application design is translated into source code. All the components of the software are implemented in this phase. Maven is used for dependency implementation. I have used MVC Structure for implementation through spring boot.

4. Testing

Testing starts once the coding is complete and the modules are released for testing. In this phase, the developed application is tested thoroughly and any defects found are assigned back to get them fixed. Testers refer to the SRS document to make sure that the software is as per the customer's standard.

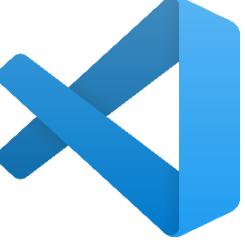
5. Deployment

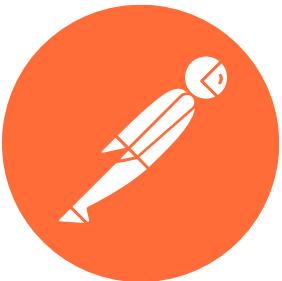
Once the product is tested, it is deployed in the production environment or the first UAT (User Acceptance testing) is done depending on the customer's expectation.

6. Maintenance

After the deployment of a product in the production environment, maintenance of the product i.e., if any issue comes up and needs to be fixed or any enhancement is to be done, it is taken care by the developers.

2.3 Software tool's table:

Index	Tool Name	Logo/Image	Use Case
1.	Visual Studio Code	 The logo for Visual Studio Code is a blue icon featuring a large, stylized white 'X' shape inside a blue rounded rectangle.	Visual Studio Code is a free coding editor that helps you start coding quickly. Use it to code in any programming language, without switching editors. Visual Studio Code has support for many languages, including Python, Java, C++, JavaScript, and more.
2.	Sublime	 The logo for Sublime Text is a dark gray square containing a bright orange 'S' shape.	Sublime Text is one of the most popular text editors in the world. It's jam-packed with powerful features like multi-line editing, build systems for dozens of programming languages, regex find and replace, a Python API for developing plugins, and more.
3.	Spring tool suite	 The logo for Spring Tool Suite (STS) features a green gear with a leaf-like pattern inside, set against a dark blue background. Below the gear, the text "SpringToolSuite4" is written in a white, sans-serif font.	It is used to create stand-alone and production-ready spring applications. It provides an easier and faster way to set up, configure, and run simple and complex web-based applications. Spring Tool Suite (STS) is an IDE popularly known as STS IDE used to create a Spring Boot project.
4.	GitHub	 The GitHub logo is a white octocat icon inside a black circle, positioned on a light gray background with a subtle checkered pattern.	GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

5.	Postman		Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs-faster. Some of Postman's advantages include the collection feature and the possibility to create different testing environments.
6.	MySQL RDBMS		It allows programmers to use SQL to create, modify, and extract data from the relational database. By normalizing data in the rows and columns of the tables, MySQL turns into a scalable yet flexible data storage system with a user-friendly interface that can manage lots of data.

[2.4.0.1 Software & Tools Table]

3.0 INTRODUCTION TO PROJECT / INTERNSHIP AND PROJECT / INTERNSHIP MANAGEMENT

3.1 PROJECT SUMMARY

The purpose of this project is to develop a web application called ShopMe using Spring Boot, focusing on the e-commerce domain. ShopMe aims to provide a seamless and user-friendly online shopping experience for customers. The application will include features such as product browsing, cart management, order placement, and payment processing.

It will also incorporate secure user authentication and authorization mechanisms to ensure data privacy and security. The project will leverage the power of Spring Boot to develop a robust and scalable e-commerce platform.

3.2 PURPOSE

The purpose of the ShopMe web application is to offer a convenient and efficient way for customers to browse and purchase products online. By utilizing Spring Boot, the application will benefit from its features and capabilities for building robust and scalable web applications. The goal is to provide a user-friendly interface, smooth shopping process, and secure transaction handling.

3.3 OBJECTIVE

The objectives of the project are as follows:

- Develop a fully functional e-commerce web application using Spring Boot.
- Implement essential features such as product catalog, shopping cart, order management, and payment processing.
- Ensure a seamless user experience with intuitive navigation and responsive design.
- Integrate secure user authentication and authorization mechanisms.
- Employ Spring Boot's capabilities for efficient database integration and data management.
- Optimize application performance and scalability to handle a large number of concurrent users.
- Adhere to industry best practices and standards in e-commerce application development.

3.4 SCOPE

The scope of the ShopMe project includes:

- Developing the front-end components of the web application using HTML, CSS, and JavaScript.
- Utilizing Spring Boot to create the back-end logic and RESTful APIs for seamless data exchange.
- Implementing a database management system, such as MySQL or PostgreSQL, to store product and user-related information.
- Incorporating secure user authentication and authorization using Spring Security.
- Integrating payment gateways for secure and reliable online transactions.
- Implementing robust error handling and exception management mechanisms.
- Ensuring responsive design and compatibility across different devices and browsers.
- Optimizing the application for performance, scalability, and search engine optimization (SEO).
- The ShopMe project aims to leverage the capabilities of Spring Boot to deliver a feature-rich and secure e-commerce platform that enhances the online shopping experience for customers.

3.5 TECHNOLOGY AND LITERATURE REVIEW

Technology Used For Development :

For the development of the ShopMe web application, the following technologies and tools were utilized:

Spring Framework:

The ShopMe application is built using the Spring Framework, which is a popular Java framework for building enterprise-level applications. It provides a comprehensive programming and configuration model that simplifies the development process and promotes code reusability and modularity.

Java:

Java is the primary programming language used for developing the ShopMe application. It is a robust and widely adopted language that provides the necessary tools and libraries for building scalable and secure applications.

Database Management System:

A database management system, such as MySQL or PostgreSQL, is used to store and manage product information, user profiles, and other relevant data for the e-commerce platform.

Spring Security:

Spring Security is a powerful security framework that provides authentication and authorization features for web applications. It ensures secure access control and protects sensitive data by implementing various security mechanisms.

HTML, CSS, JavaScript:

The front-end components of the ShopMe application are developed using HTML, CSS, and JavaScript. HTML is used for structuring the content, CSS is used for styling and layout, and JavaScript is used for implementing interactive functionality and client-side validation.

jQuery:

jQuery is a fast and lightweight JavaScript library that simplifies the process of manipulating HTML documents, handling events, and making asynchronous requests. It is used in the ShopMe application to enhance the user interface and provide a smooth and interactive user experience.

Bootstrap:

Bootstrap is a popular CSS framework that provides a collection of pre-built responsive components and styles. It enables developers to create modern and visually appealing user interfaces with ease. Bootstrap is utilized in the ShopMe application for achieving a consistent and mobile-friendly design.

Spring Boot:

Spring Boot is a framework built on top of the Spring Framework that simplifies the development of Java applications by providing out-of-the-box configurations and conventions. It eliminates the need for manual setup and allows developers to quickly create production-ready applications.

Annotations:

Annotations are used extensively in the ShopMe application to leverage the power of the Spring Framework. Annotations such as @Controller, @RequestMapping, and @Autowired are used to define controllers, map URLs, and inject dependencies, respectively. These annotations help in reducing boilerplate code and improving code readability.

Maven:

Maven is a build automation tool used for managing dependencies, building, and packaging Java applications. It simplifies the project configuration and ensures consistent build processes, making it easier to manage and maintain the ShopMe application.

JUnit Testing:

JUnit is a popular testing framework for Java applications. It provides a set of annotations and assertions that enable developers to write unit tests to verify the correctness of individual components and functionalities in the ShopMe application.

Literature Review:

During the project, relevant literature and resources were consulted to gain insights and best practices in e-commerce application development, Spring Framework, Java, web technologies, jQuery, Bootstrap, and the effective use of annotations. These resources included official documentation, tutorials, books, and online forums, which provided guidance on architectural patterns, design principles, and techniques for utilizing these technologies.

By utilizing the Spring Framework, Java, HTML, CSS, JavaScript, jQuery, Bootstrap, and annotations, the ShopMe project aims to deliver a feature-rich and user-friendly e-commerce web application. The chosen technologies and tools ensure efficient development, maintainability, and scalability, while incorporating modern web design and interactive functionality for an enhanced user experience.

3.6.2 Project Effort And Time

When we started the project we lacked knowledge regarding the technology we have used. It is the base of the whole project. So first we learned as much as possible regarding the same and soon as we gained confidence we started with some basic features to implement. We faced many difficulties from starting but as we moved further and had some little knowledge gained we were able to solve the problems. We needed some help from our external guides sometimes but soon we developed the logic and were able to tackle it. We then moved to implement advanced features but we were not able to figure out what particular approach we should take to implement some features. In the process, we researched many new features provided by the framework and we started implementing them. Also in some cases, we found more than one method and we were able to judge which method is more feasible and efficient to use. But the time required for all this was longer than we thought. The time taken to implement was less but the time taken to solve bugs and the research work took much more time than we thought. But it was for our own good as the outcome of the process and project is satisfactory.

3.6.3 Roles And Responsibilities

Different Roles:

- Analysis
- Designing
- Frontend
- Backend
- Testing
- Documentation

3.7 PROJECT SCHEDULING (GANTT CHART)

A Gantt chart is a special type of bar chart where each bar represents an activity. The bars are drawn along a timeline. The length of each bar is proportional to the duration of time planned for the corresponding activity. In the Gantt chart used for software project management, each bar consists of a white and shade part. The shaded part of the bar shows the length of time each task is estimated to take.

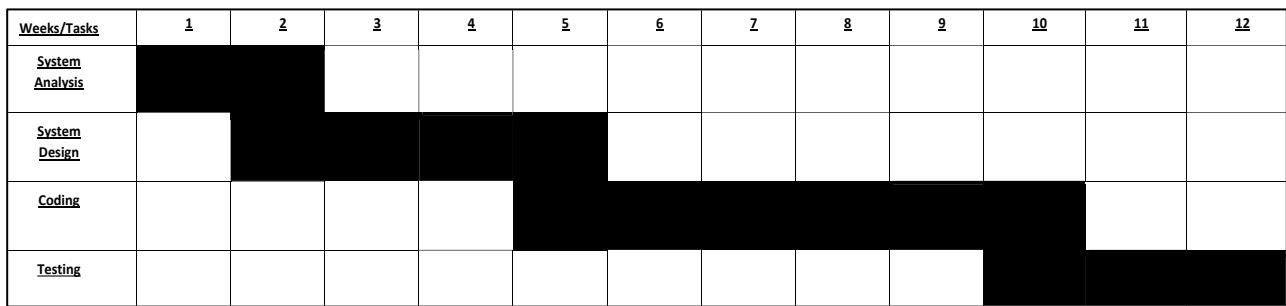


Figure 3.7.1 Gantt Chart

4.0 SYSTEM ANALYSIS

4.1 STUDY OF THE CURRENT SYSTEM

The current system is an e-commerce web application built using Spring Boot. It comprises two applications: the Admin application and the Shopping application. The Admin application is responsible for managing users, categories, brands, products, customers, orders, sales reports, reviews, questions, and settings. On the other hand, the Shopping application enables customers to browse products, place orders, manage their orders, and handle registration.

4.2 PROBLEMS AND WEAKNESSES OF THE CURRENT SYSTEM

One of the weaknesses of the current system is its limited availability. As a web application, it can only be accessed through devices with an internet connection. This can be a constraint for users who do not have access to stable internet connectivity or prefer offline access to the application. Additionally, the current system may lack certain features and functionalities that could enhance the user experience and streamline business processes.

4.3 SYSTEM FEASIBILITY

Operational Feasibility:

The proposed system is designed to be user-friendly and intuitive, ensuring that users can navigate the application easily without requiring extensive training.

Technical Feasibility:

The system is developed using the Spring Boot framework, which provides a robust and scalable platform for building web applications. It leverages the power of Java and various libraries and technologies associated with Spring Boot.

Economical Feasibility:

The cost-effectiveness of the system is achieved through the utilization of open-source technologies, such as Spring Boot and its associated libraries. This allows for reduced licensing costs and increased flexibility for customization.

4.4 ACTIVITY IN THE NEW SYSTEM

The proposed system aims to address the weaknesses of the current system and enhance its functionality. The new system will focus on the following features and improvements:

Mobile and Desktop Availability:

The new system will be designed to be responsive, ensuring that users can access the application seamlessly from both mobile devices and desktop computers. This will provide flexibility and convenience to users, allowing them to shop and manage their orders from any device.

Improved User Experience:

The new system will prioritize user experience by implementing a user-friendly interface, intuitive navigation, and smooth checkout processes. It will also include features such as product recommendations, personalized offers, and seamless integration with popular payment gateways to enhance the overall shopping experience.

Advanced Analytics and Reporting:

The new system will include robust analytics and reporting capabilities, providing administrators with insights into sales performance, customer behavior, popular products, and more. This data will enable informed decision-making and help optimize business strategies.

Streamlined Admin Management:

The new system will enhance the Admin application by introducing features for efficient user, product, and order management. It will provide tools for managing customer inquiries, handling reviews, and generating comprehensive sales reports. This will streamline administrative tasks and improve operational efficiency.

By implementing these improvements, the new system aims to provide a seamless and enjoyable shopping experience for customers while empowering administrators with powerful management tools.

4.5 FEATURES OF THE NEW SYSTEM

- Data processing has been the key factor in this application as data has to be sent continuously from the client side to the database side and both client panel and roles panel will make database calls as per their needs where creating categories , products and many more function available in the panel. We have to make sure all the data and accurate data are sent from the client side and properly received on the coach's side.
- If any of the data is missing, it might affect other data also because of join tables. We have to make sure the data coming to the coach side is accurate or not. We need to make sure data received on the database side is managed properly and placed properly onto the UI elements.
- The GUI of the new system is good and understandable to the user.
- Fast, Reliable and Accurate system

4.7 LIST THE MAIN MODULES / COMPONENTS / PROCESSES / TECHNIQUES OF NEW SYSTEM

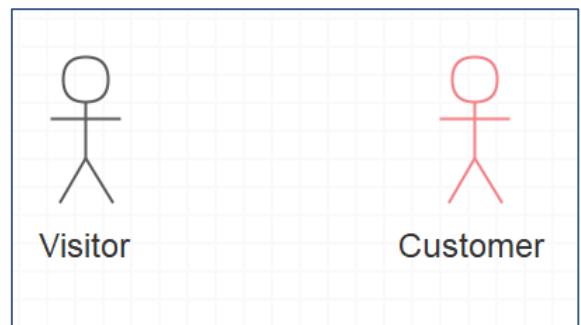
Understanding Requirements :

Key Actors

Back End



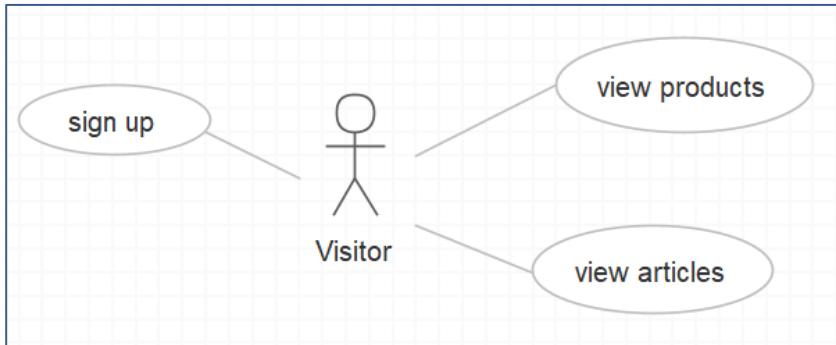
Front End



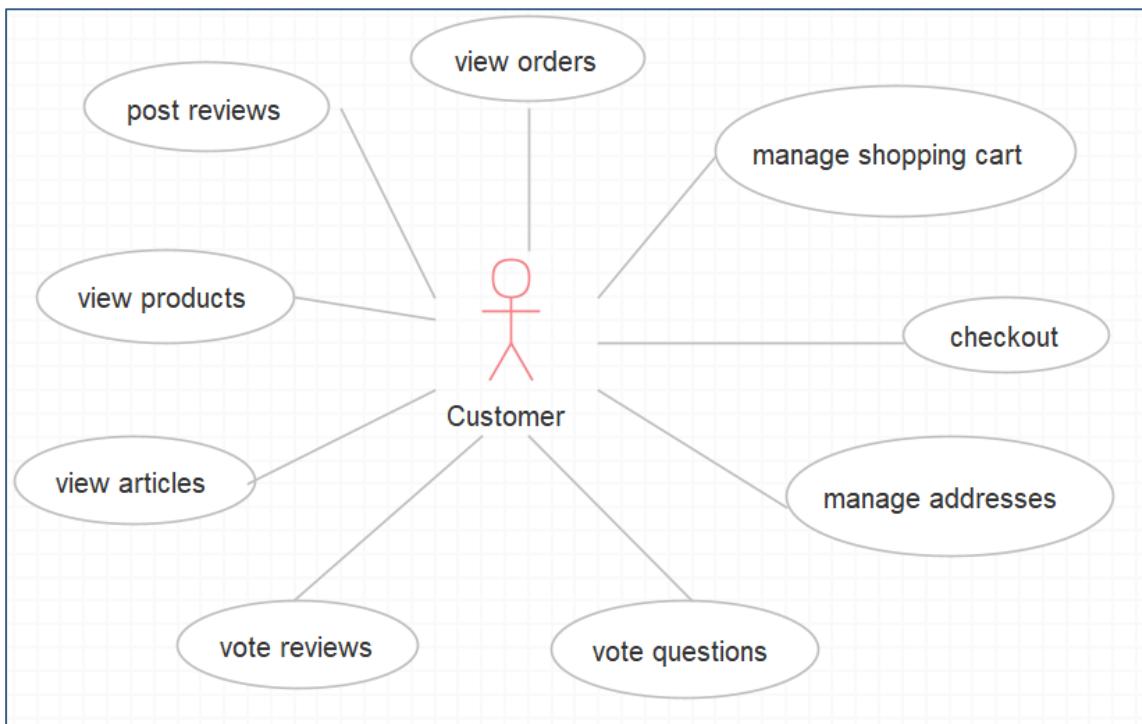
No	Role Name	Description
1	Admin	manage everything
2	Salesperson	manage product price, customers, shipping, orders and sales report
3	Editor	manage categories, brands, products, articles and menus
4	Shipper	view products, view orders and update order status
5	Assistant	manage questions and reviews

4.8 Case Scenarios :

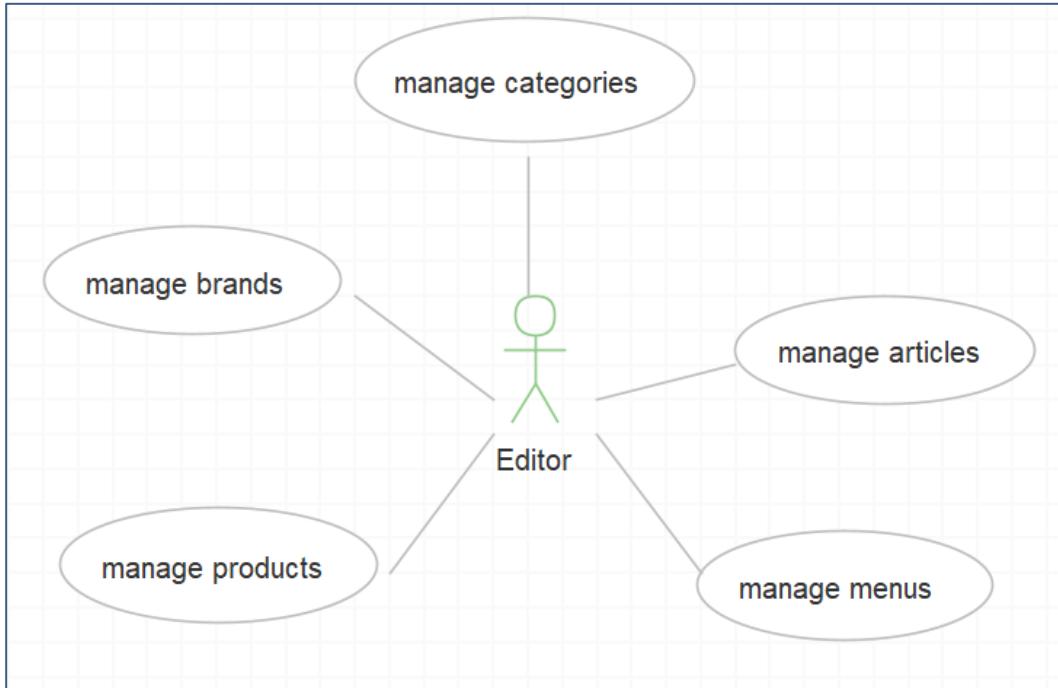
Front End > Use cases for visitors



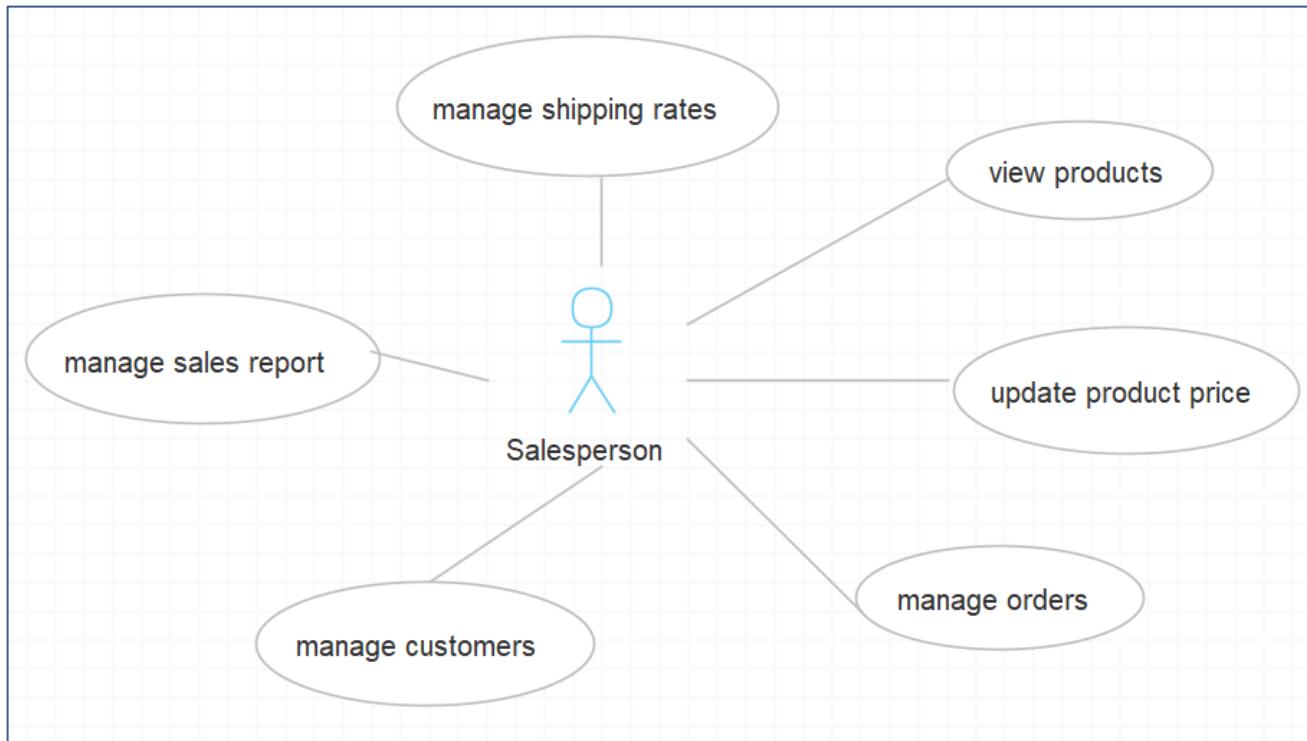
Front End > Use cases for customers



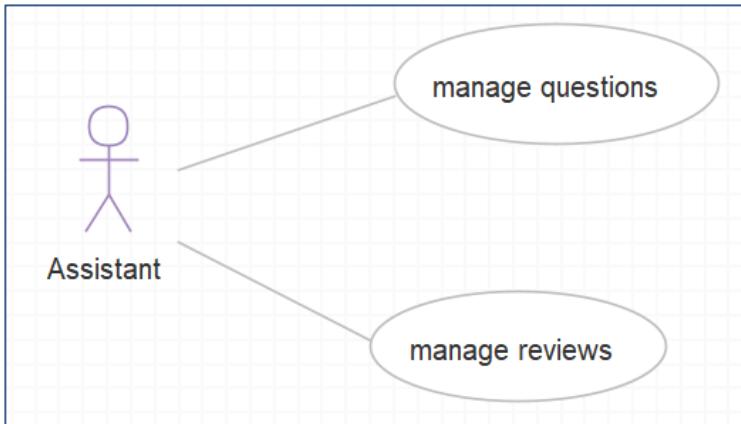
Back End > Use cases for editors



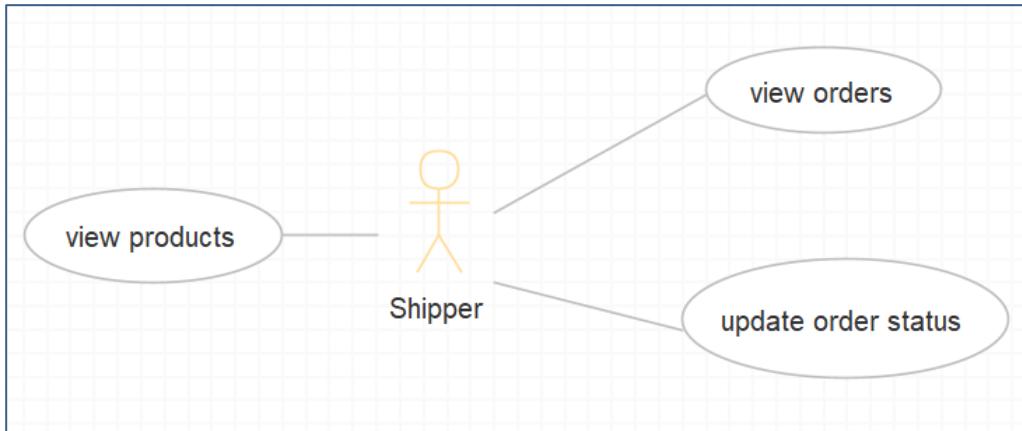
Back End > Use cases for salespersons



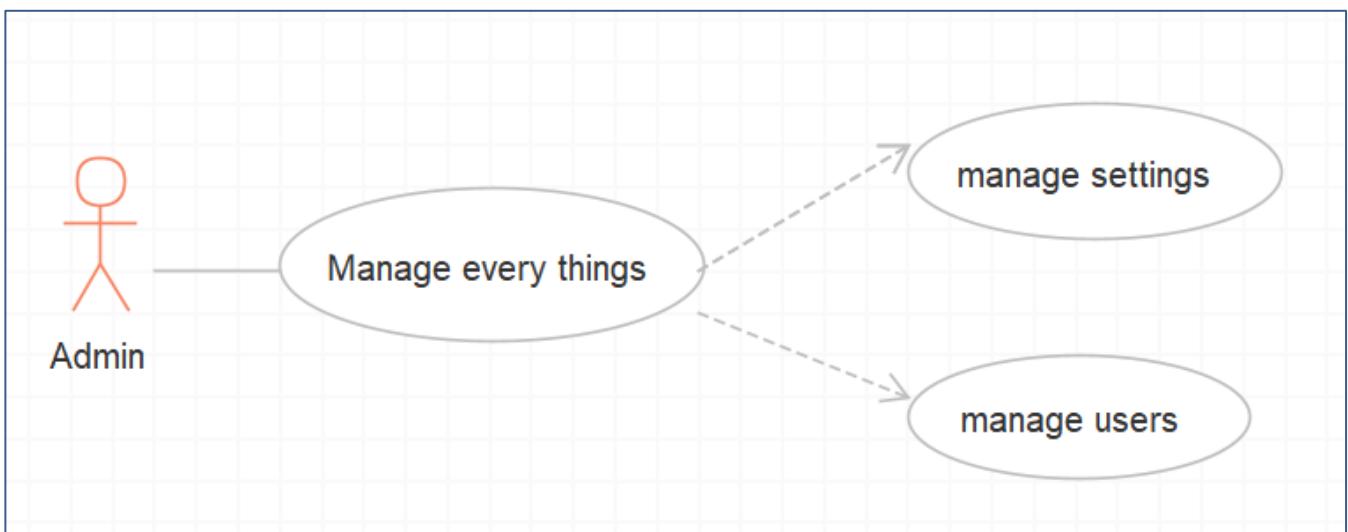
Back End > Use cases for assistants



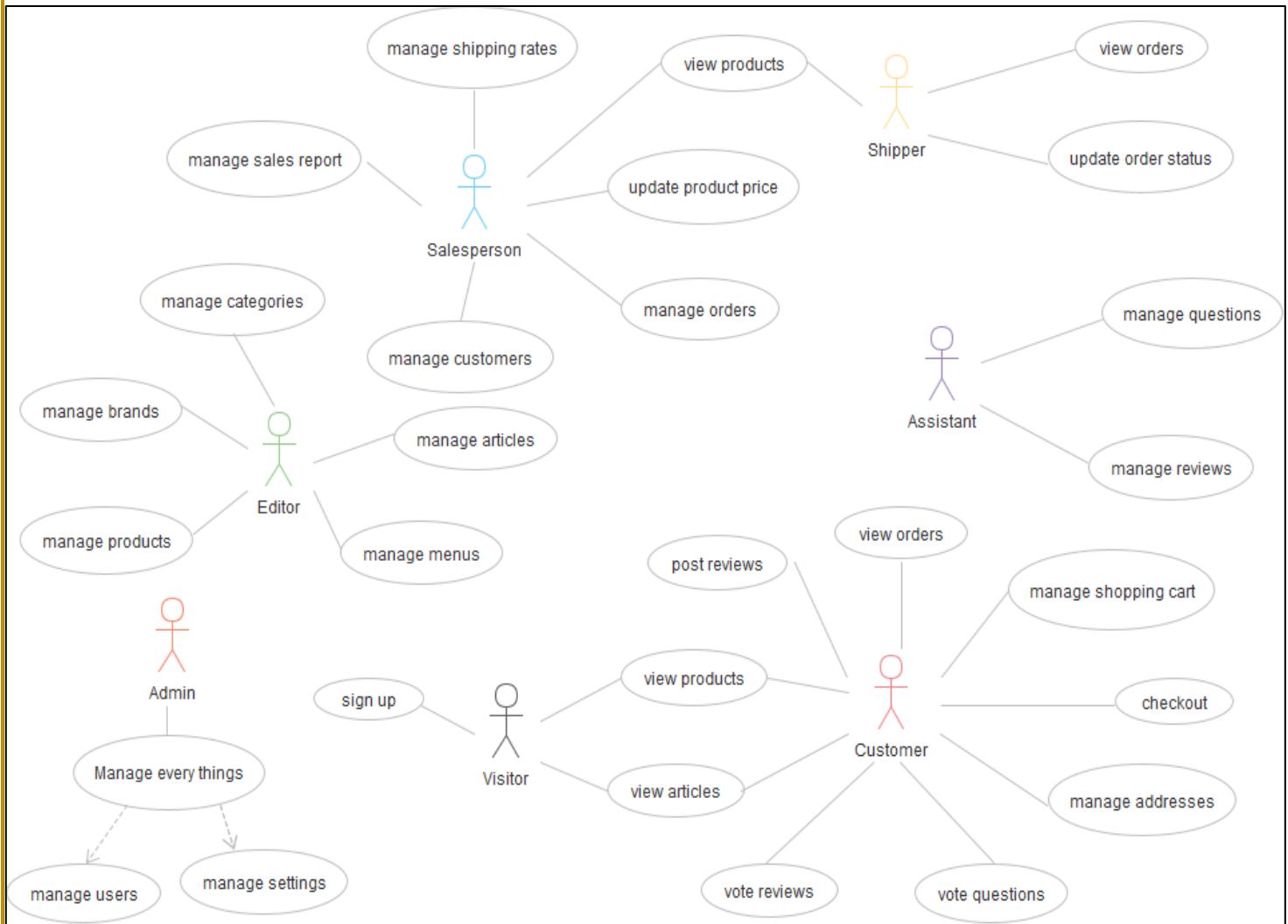
Back End > Use cases for shippers



Back End > Use cases for administrators



Overview Use Cases



4.9 1. Functional Requirements

User Management Service:

- Sign up:
 - Input: User's credentials (e.g., email, password, personal information)
 - Output: Confirmation of successful registration
- Login:
 - Input: User's login credentials (e.g., username, password)
 - Output: Access granted to the system

Admin and Roles Service:

- Create, update, and delete user accounts
- Assign roles and permissions to users

Category Management Service:

- Create, update, and delete categories

Brand Management Service:

- Create, update, and delete brands

Product Management Service:

- Create, update, and delete products

Article Management Service:

- Create, update, and delete articles

Menu Management Service:

- Create, update, and delete menus

Salesperson Service:

- Product Price Management:
 - Set and update product prices
- Customer Management:
 - Create, update, and delete customer accounts

Sales Report Generation Service:

- Generate sales reports based on specified criteria

Editor Service:

- Category Management:
 - Create, update, and delete categories
- Article Management:
 - Create, update, and delete articles

Shipper Service:

- Product Viewing:
 - Access and view product information
- Order Viewing:
 - Access and view assigned orders and their details
- Order Status Update:
 - Update order status based on the shipping process (e.g., shipped, delivered)

Assistant Service:

- Question Management:

- Respond to customer questions and inquiries

- Review Management:

- Manage and moderate product reviews

8. Admin:

- Currency Management:

- Admin can manage available currencies and exchange rates for international transactions

-

• Shipping Rates Management:

- Admin can configure and manage shipping rates for different regions or shipping methods

• Export User Data:

- Admin can export user data (e.g., customer information, order history) for analysis or backups

• Export Product Data:

- Admin can export product data (e.g., name, price, description) for inventory management or external integrations

- User Management:

- Admin can enable/disable user accounts and manage their roles and permissions

Salesperson: • Summary Boxes Page:

- Salesperson can view summary boxes displaying key sales metrics (e.g., total sales, top-selling products) relevant to their role

Editor: Summary Boxes Page:

- Editor can view summary boxes displaying key content metrics (e.g., total articles, pending reviews) relevant to their role

Shipper: Summary Boxes Page:

- Shipper can view summary boxes displaying key order metrics (e.g., total orders, pending shipments) relevant to their role

Assistant: Summary Boxes Page:

- Assistant can view summary boxes displaying key metrics (e.g., total questions, pending reviews) relevant to their role.

- Admin: Manage Site Menus:

- Admin can manage the navigation menus of the website, including adding, editing, and deleting menu items.

- Manage Country and State Lists:

- Admin can manage the list of countries and states/provinces available for selection during the checkout or registration process.

- Mail Server Configuration:

- Admin can configure the mail server settings to enable email communication, including SMTP server details, authentication, and encryption.

- Mail Templates Management:

- Admin can manage email templates for various notifications (e.g., order confirmation, password reset) sent to users, including customization of email content and design.

- Payment Gateway Integration:

- Admin can integrate and configure payment gateways (e.g., PayPal, Stripe) for online transactions, including managing API credentials and settings.

2. Non-Functional Requirements

- Availability: The app will be available 24*7.
- Flexibility: Data Added/Edited will be updated instantly.
- Scalability :
 - Apps can be scaled on demand and running well under loads
 - Back-end and Front-end apps can be scaled separately
- Performance :
 - Fast response time
 - No requests take longer than 4 seconds
- Security :
 - Authentication required for all users (except visitors)
 - Authorization required in the admin CP (editor, assistant, salesperson, shipper...)
 - Plain text Passwords are not stored in database
- Accessibility :
 - Apps can be accessible from any devices connected to the Internet: PC, laptop, tablet, smartphone...

5.0 SELECTION OF HARDWARE / SOFTWARE FOR DEVELOPMENT

Hardware Requirements	
Processor	i7
RAM	16 GB
Hard Disk	512 GB
Graphic Card	4 GB
Display	VA PANEL

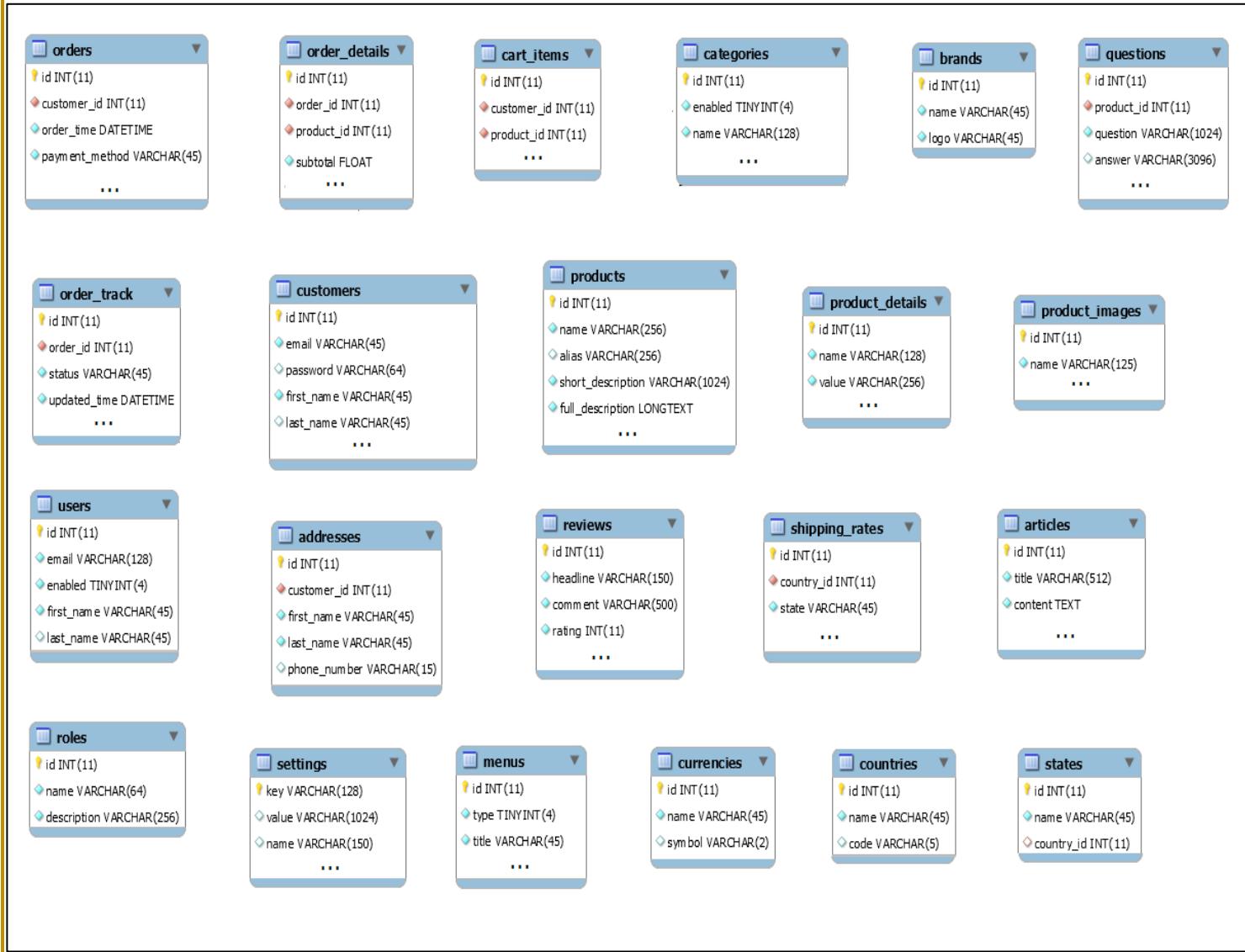
Table 5.1 Hardware Requirements

Software Requirements	
WINDOWS OS	11
SPRING BOOT	V4+,V3+
CHORME BROWSER	113.0.56
POSTMAN	V9.4
LANGUAGE	JAVA
DATABASE	MYSQL
DEPENDENCY MANAGEMENT TOOLS	MAVEN

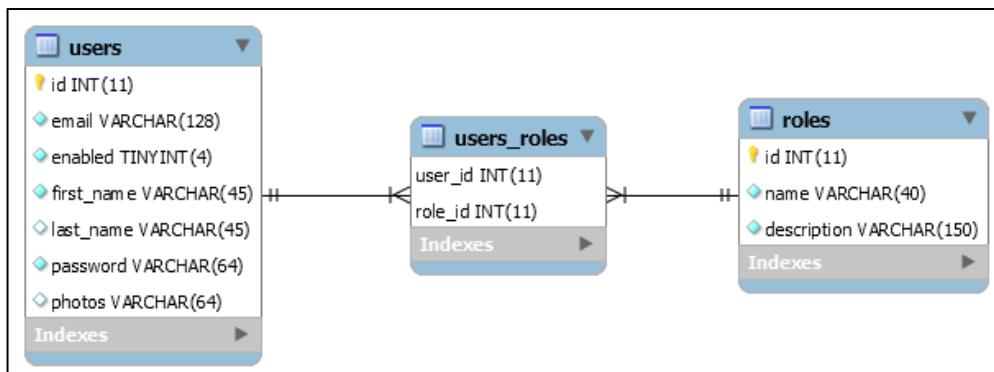
Table 5.2 Software Requirements Learning

5.1 SELECTION OF DATABASE DESIGN AND SCHEMA OVERVIEW :

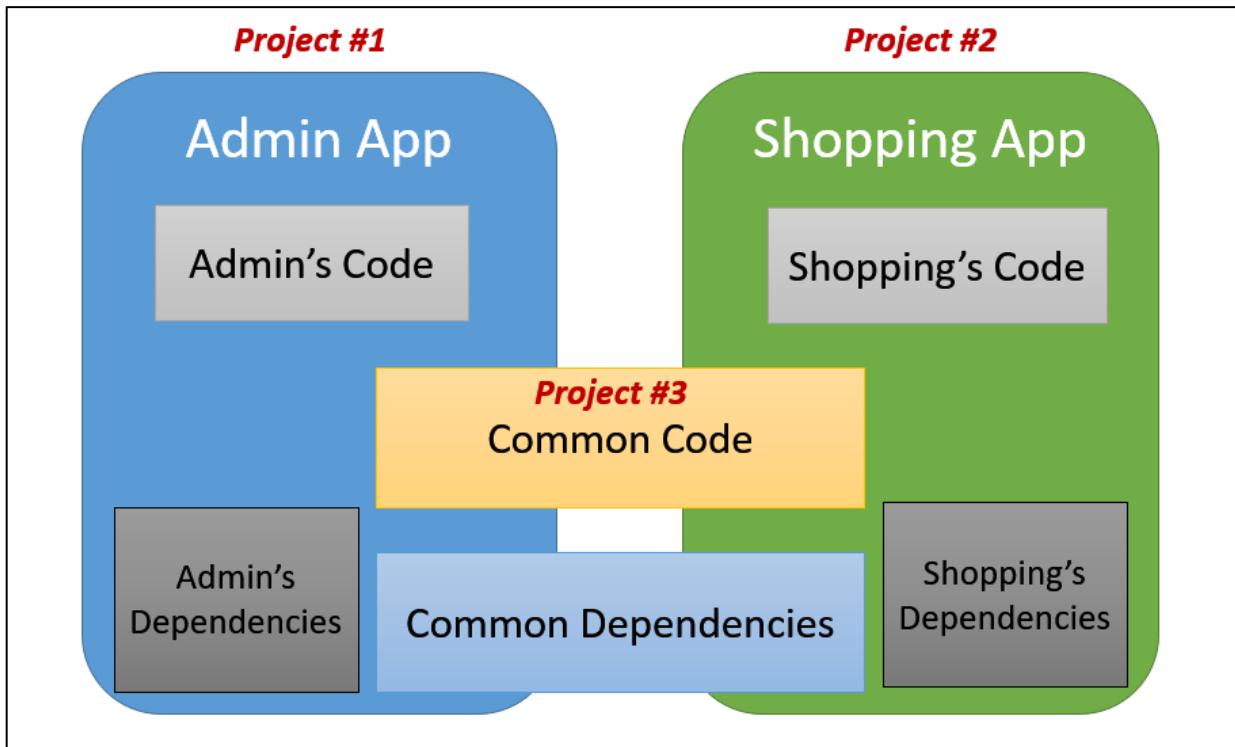
Tables generated / created in project :



Example : User Module → Database

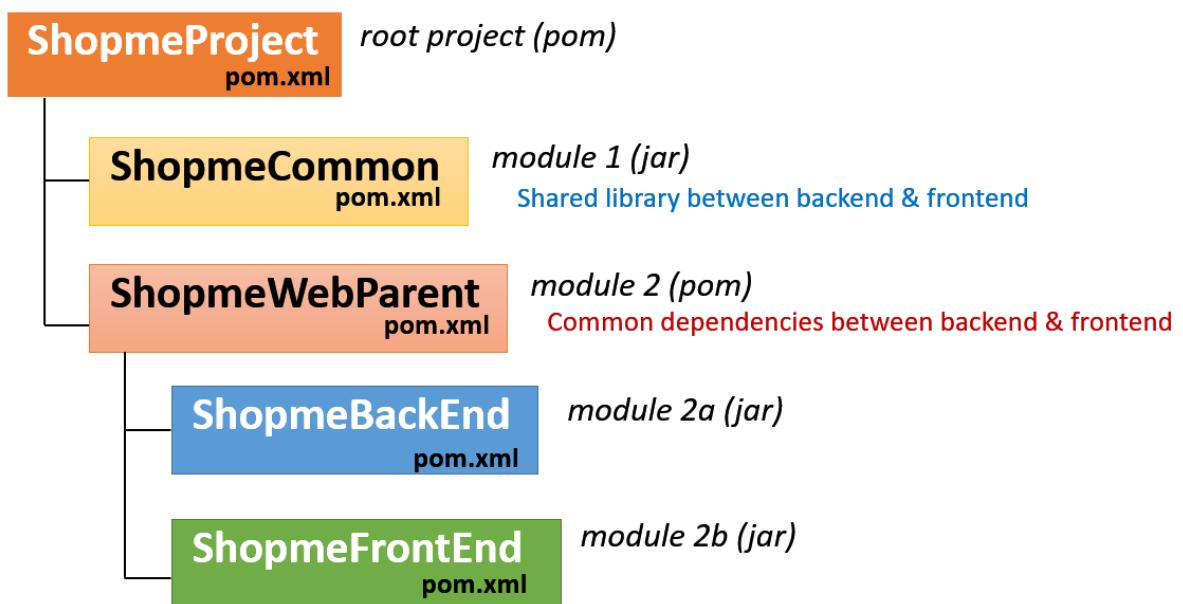


5.2 PROJECT SETUP OVERVIEW :

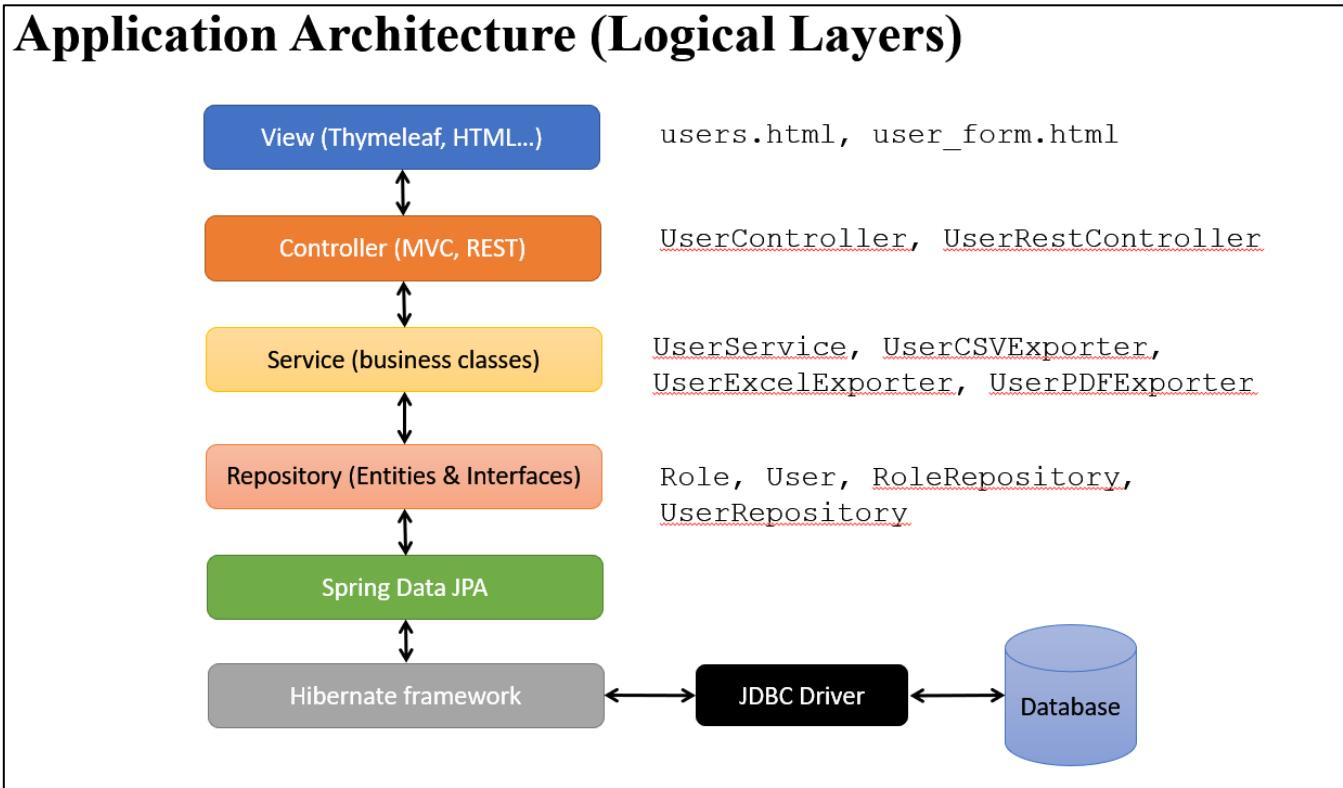


5.3 PROJECT ARCHITECTURE OVERVIEW :

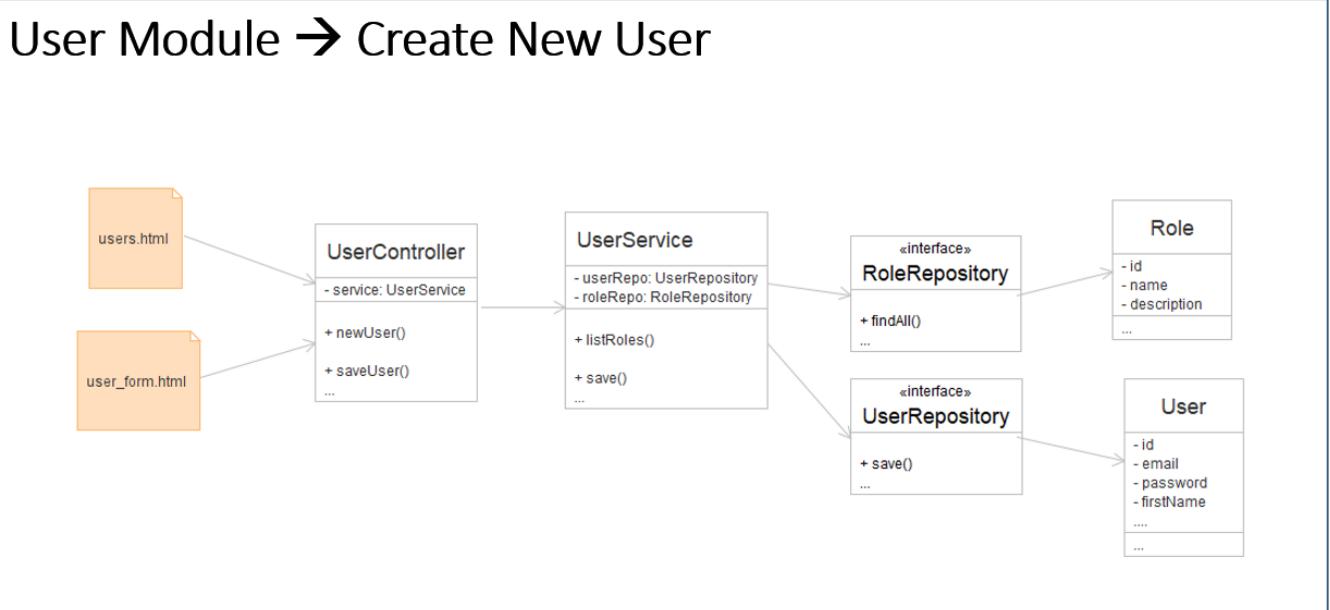
Creating Multi-Module Maven Project

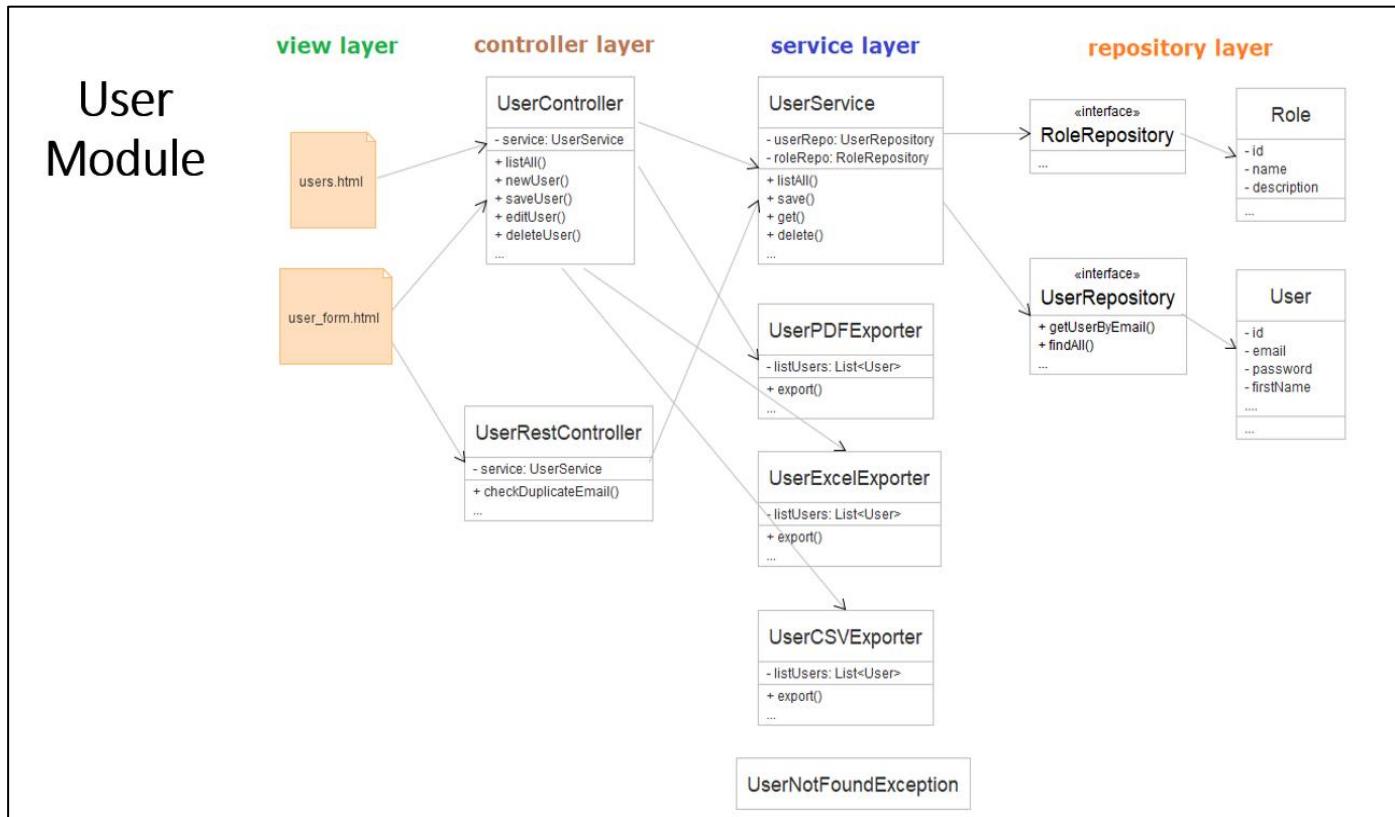


EXAPMLE USERS MODULE :

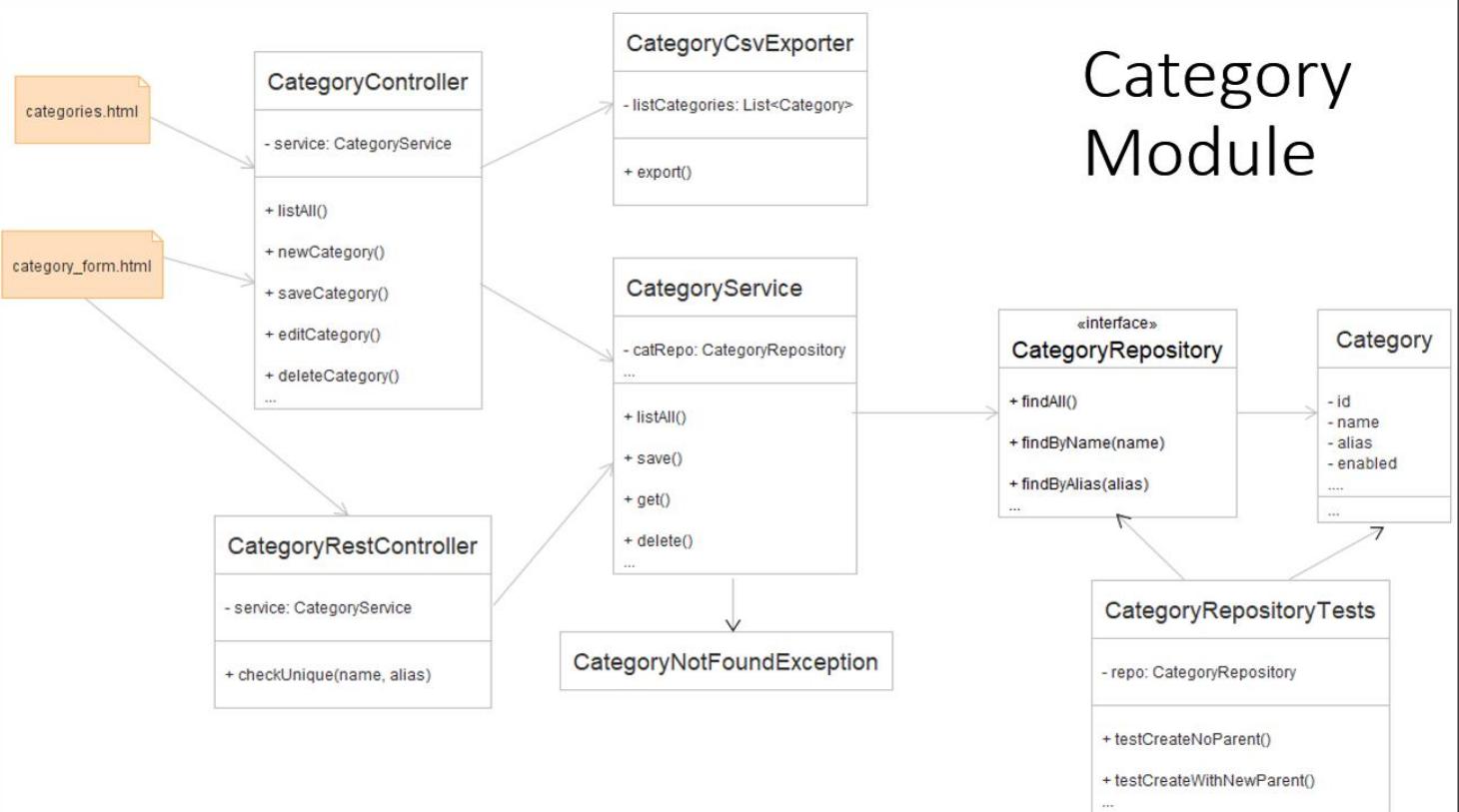


5.4 MVC PATTERN CLASS DIAGRAMS FOR THE PROJECT FOR DIFFERENT MODULES :

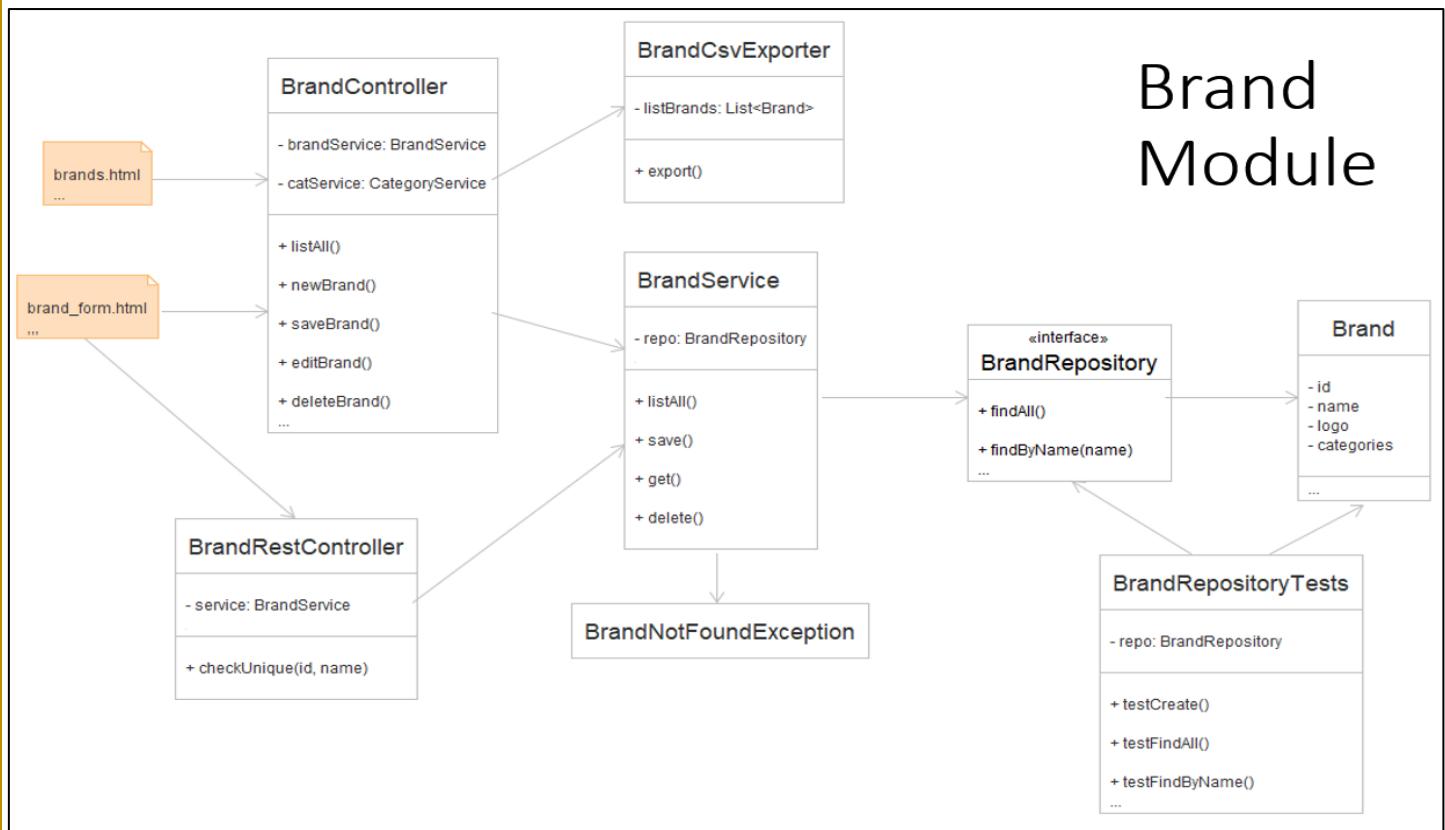




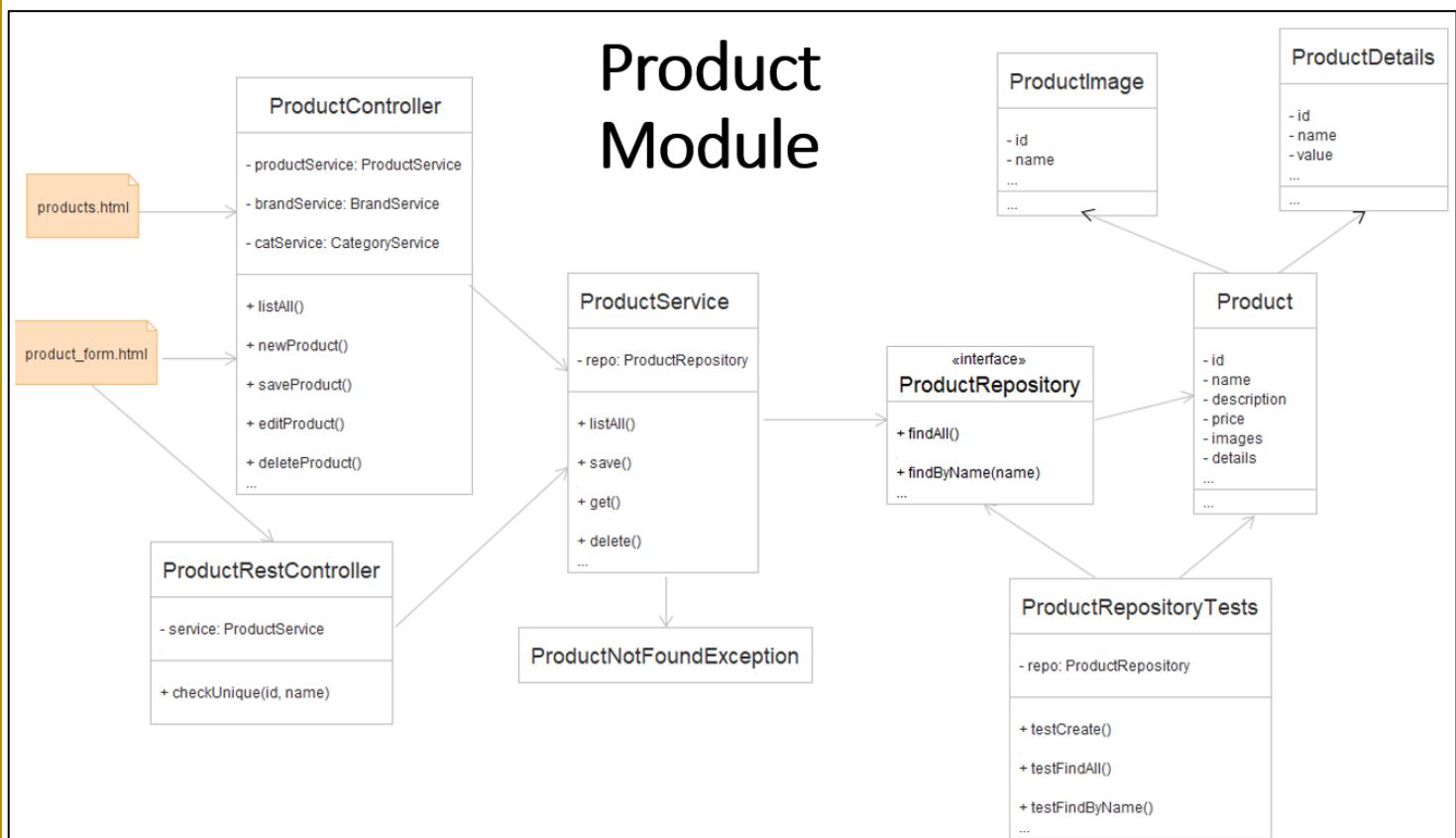
Category Module :



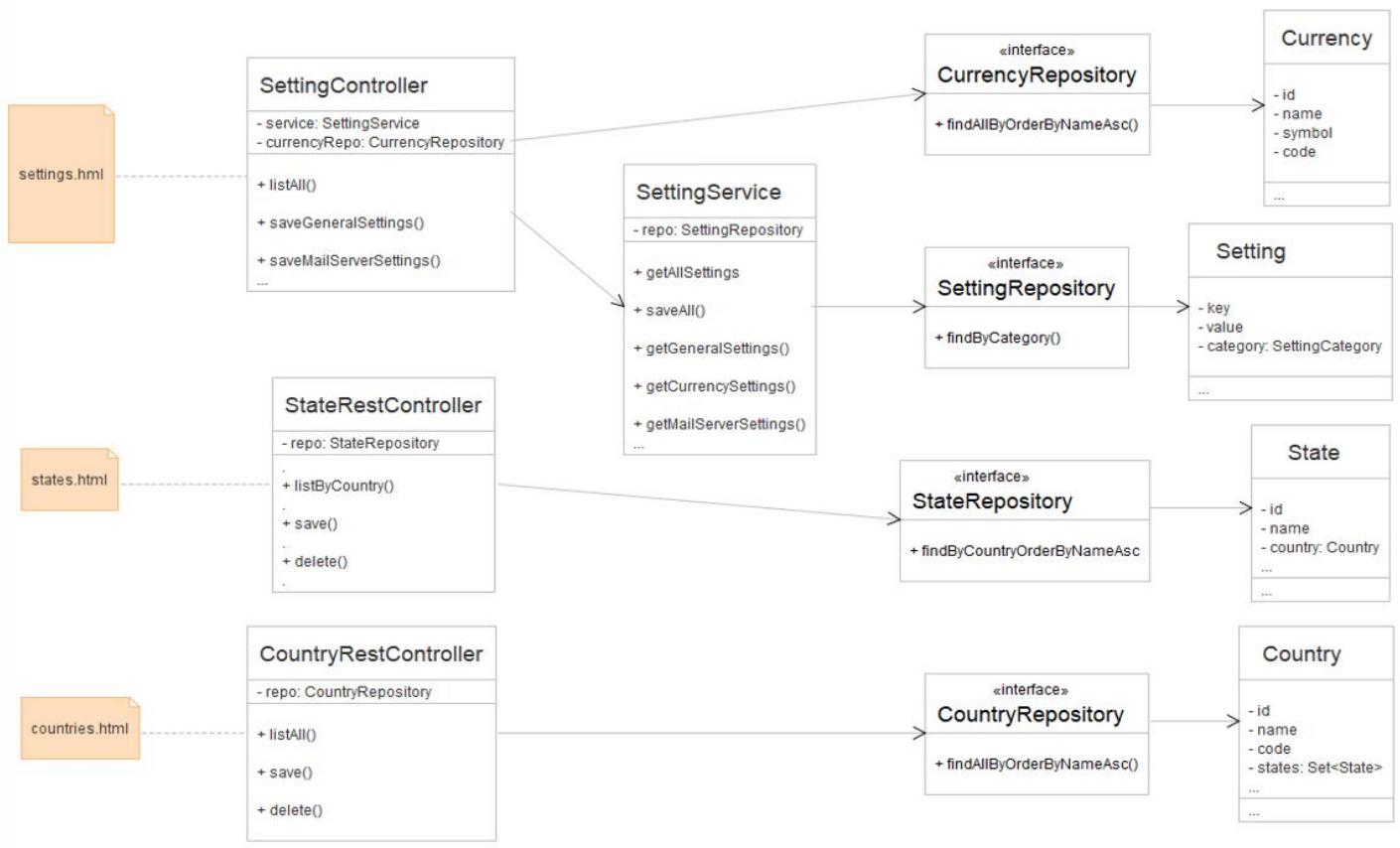
Brand Module :



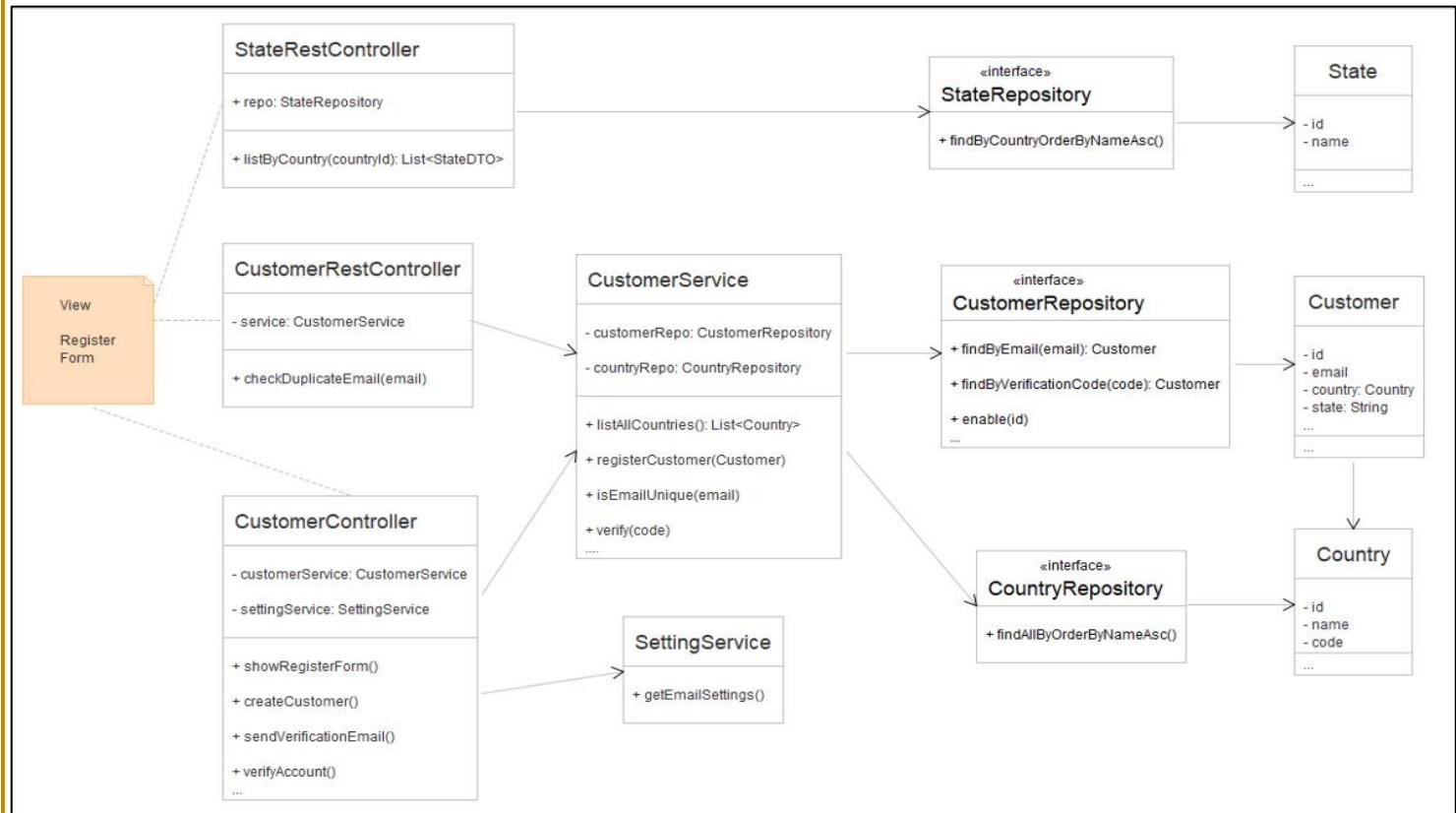
Product Module :



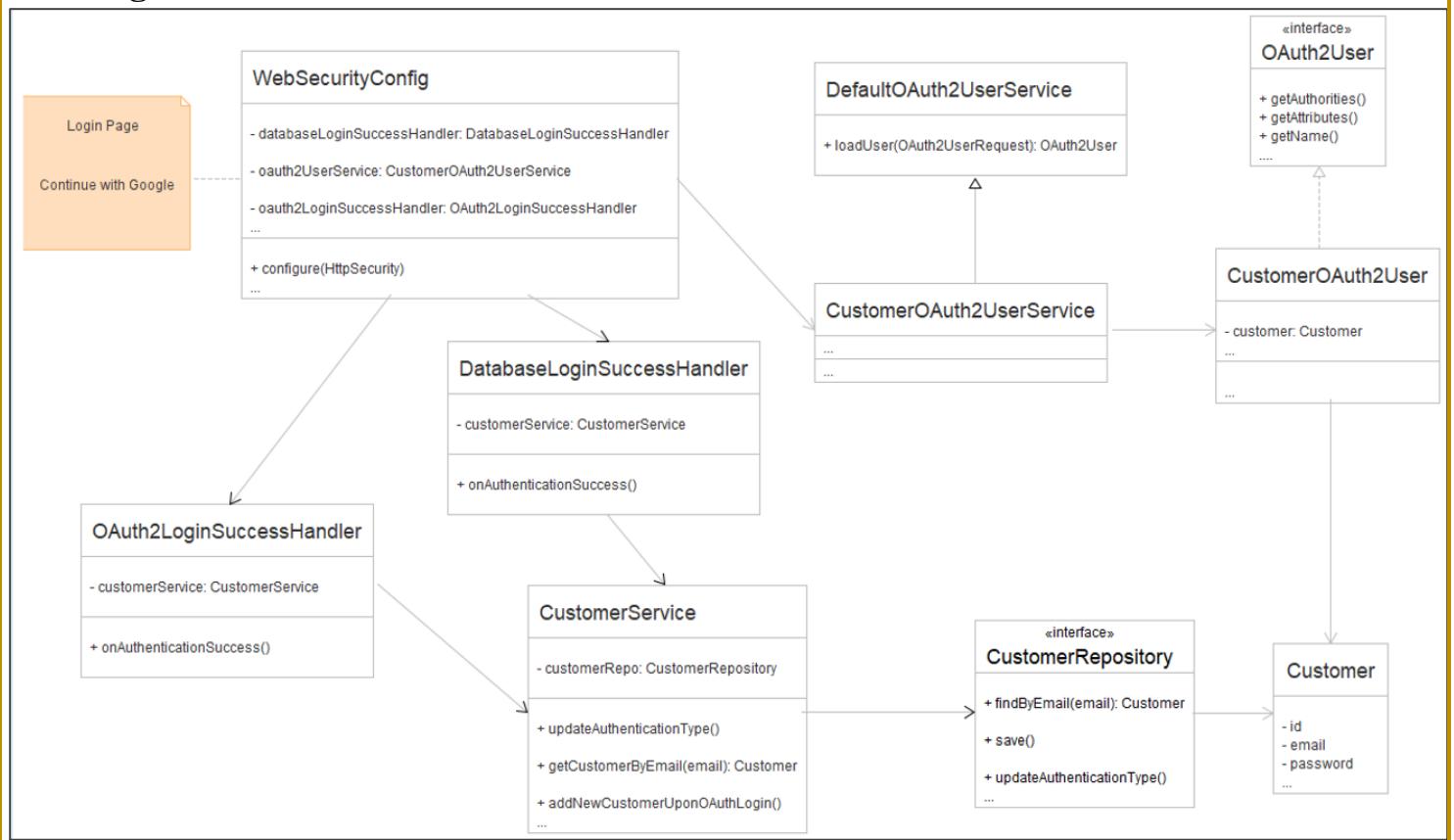
Settings Module :



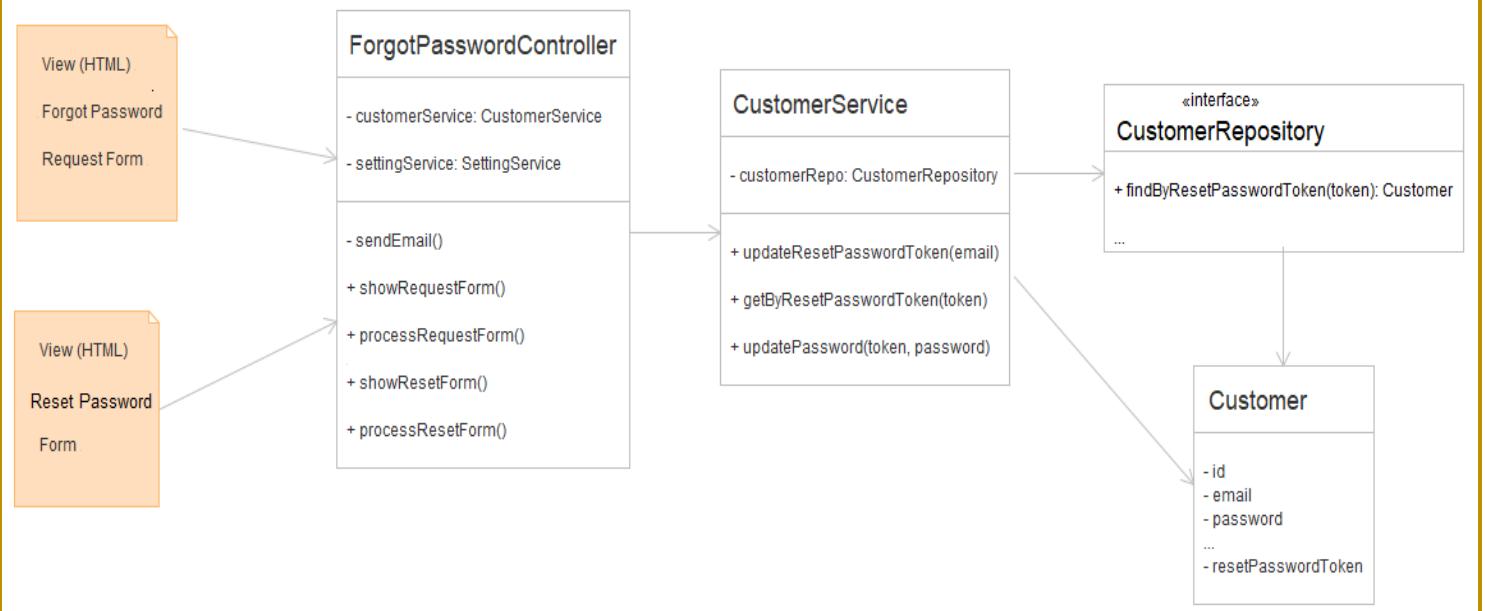
Customer Registration :



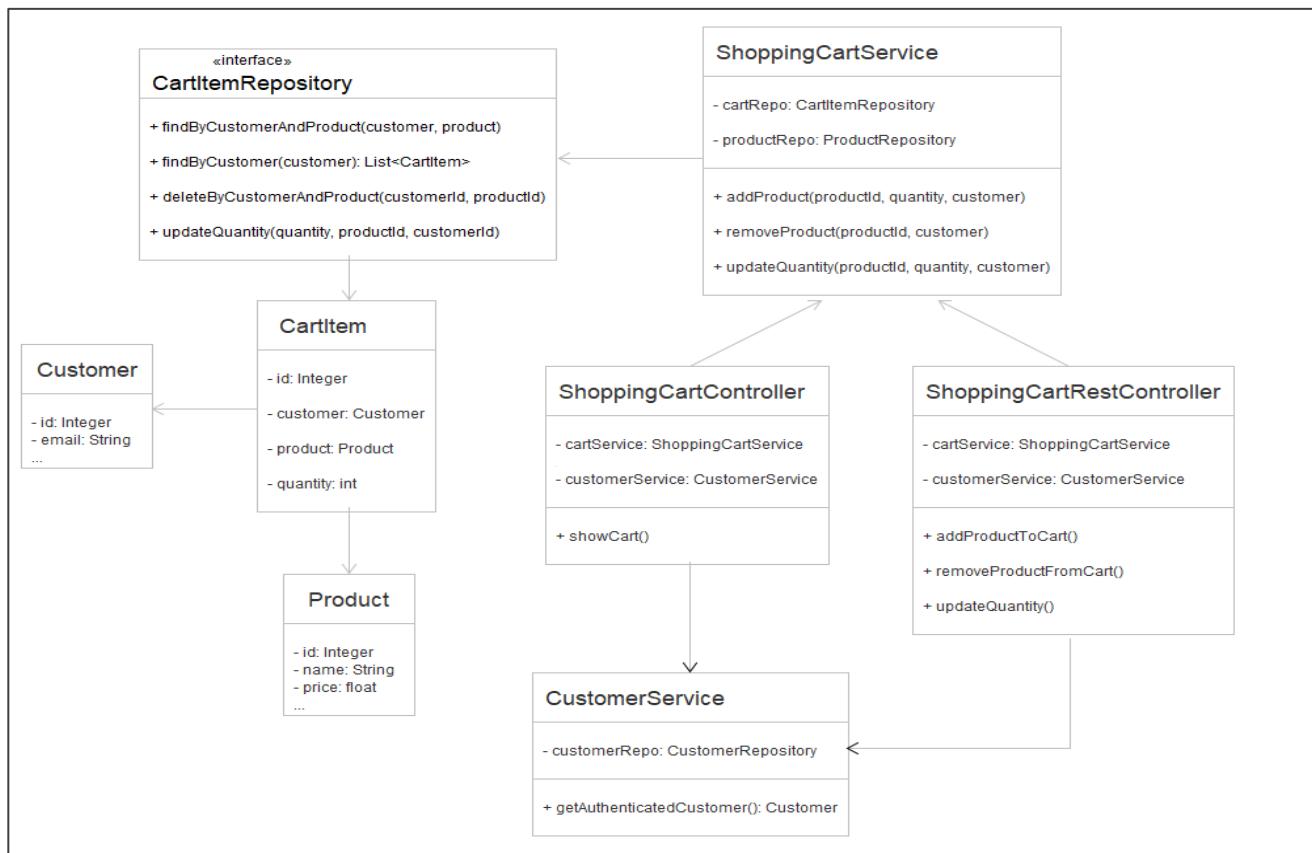
Login Module:



Forget Password Function :



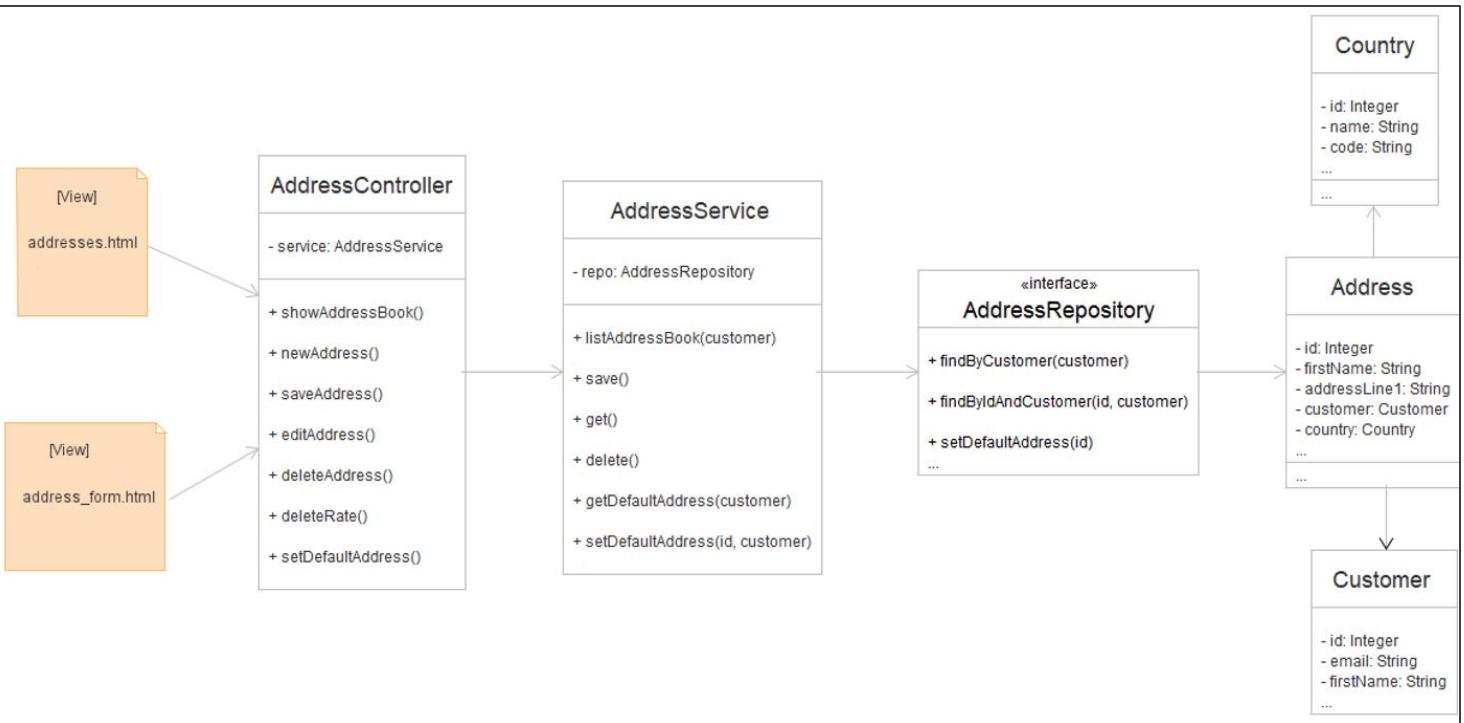
Shopping Cart Class Diagram :



Shipping Rate Module :

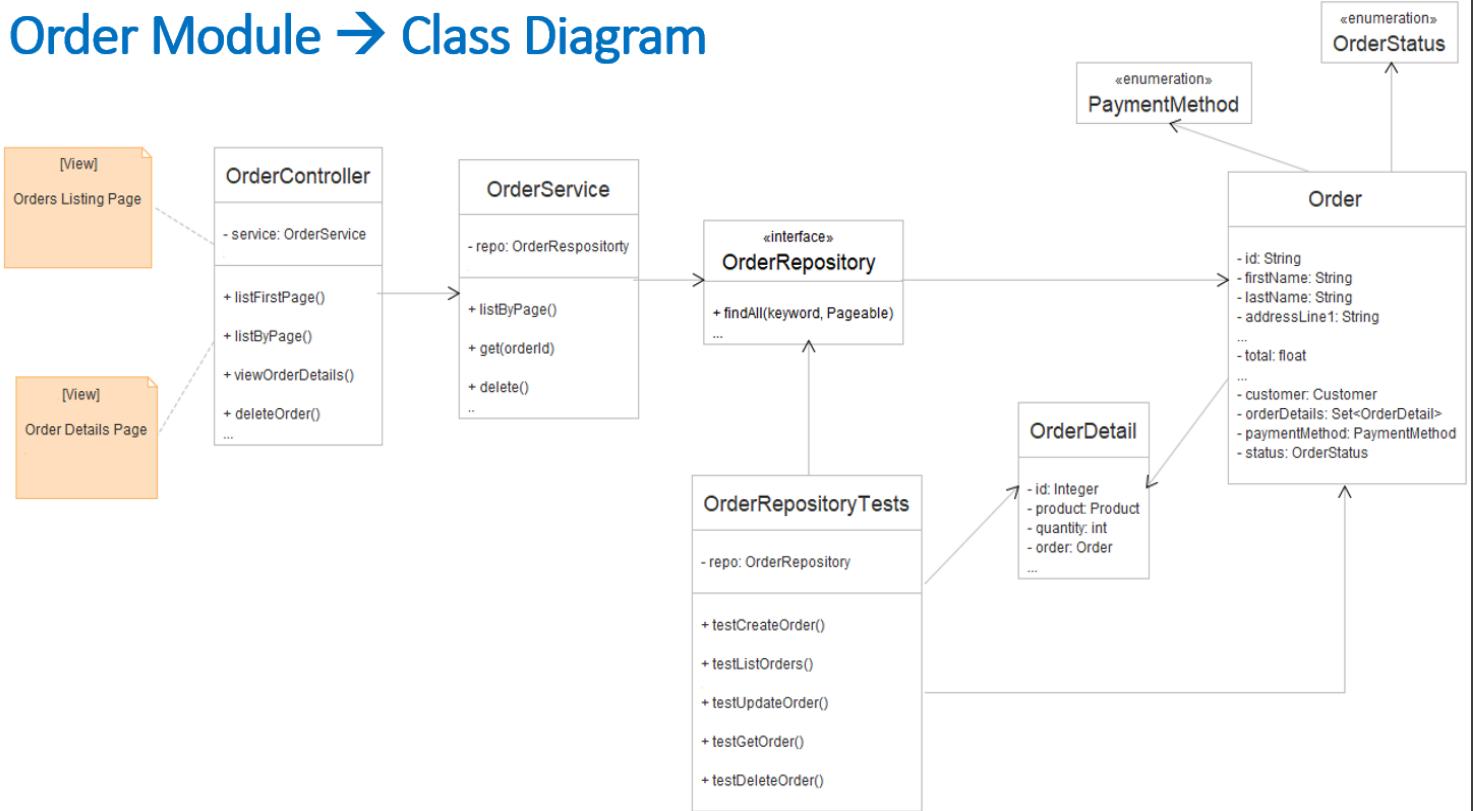


Address book Module :

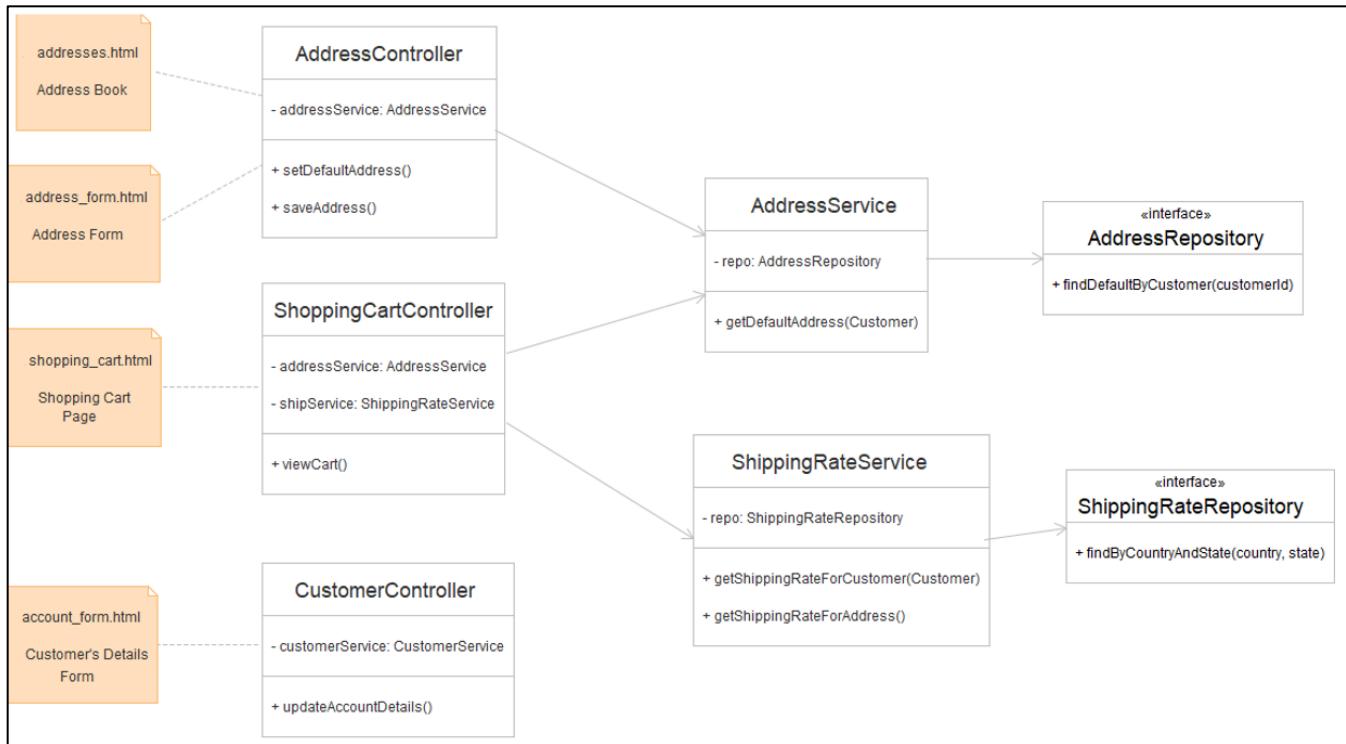


Order Module :

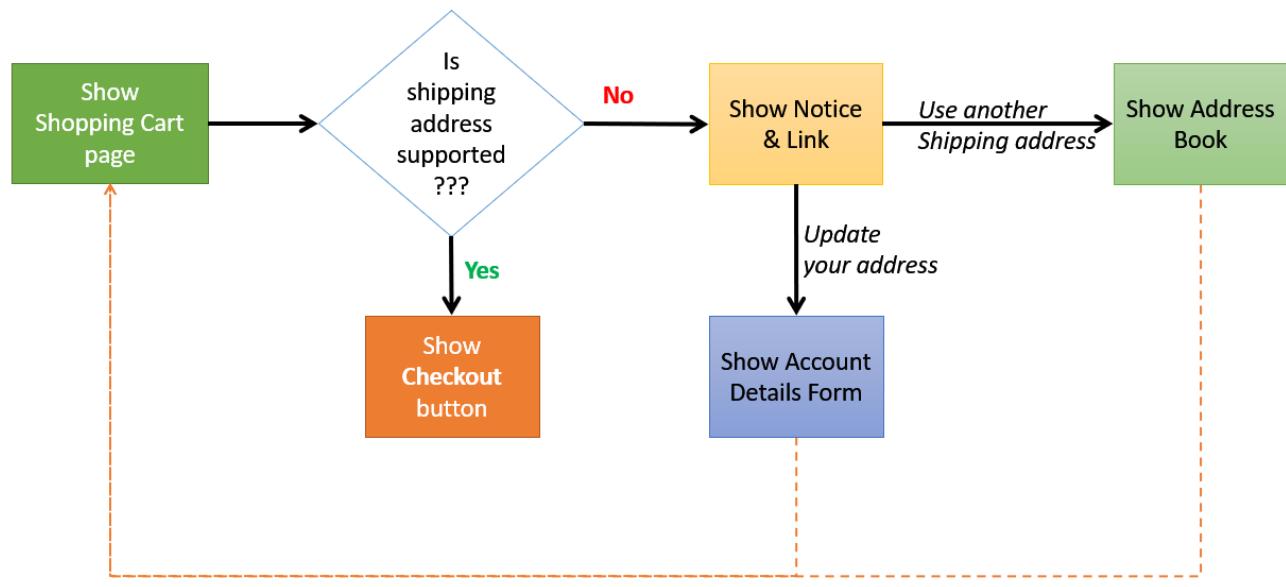
Order Module → Class Diagram



Checkout Module :



Checkout Module → Shopping Cart Workflow



6.0 IMPLEMENTATION

6.1 IMPLEMENTATION PLATFORM / ENVIRONMENT

Login Screen (Role-based Authentication) :

The login screen for the admin panel of the e-commerce Spring Boot application allows role-based authentication, ensuring secure access based on user roles. Users will be required to enter their credentials to log in.

The authentication process validates the entered credentials against the stored user data in the database. It checks the username and password combination for accuracy and verifies the user's assigned role.

Upon successful authentication, the user is granted access to the admin panel with appropriate privileges based on their role. If the entered credentials are incorrect or the user does not have the necessary role, an error message will be displayed.

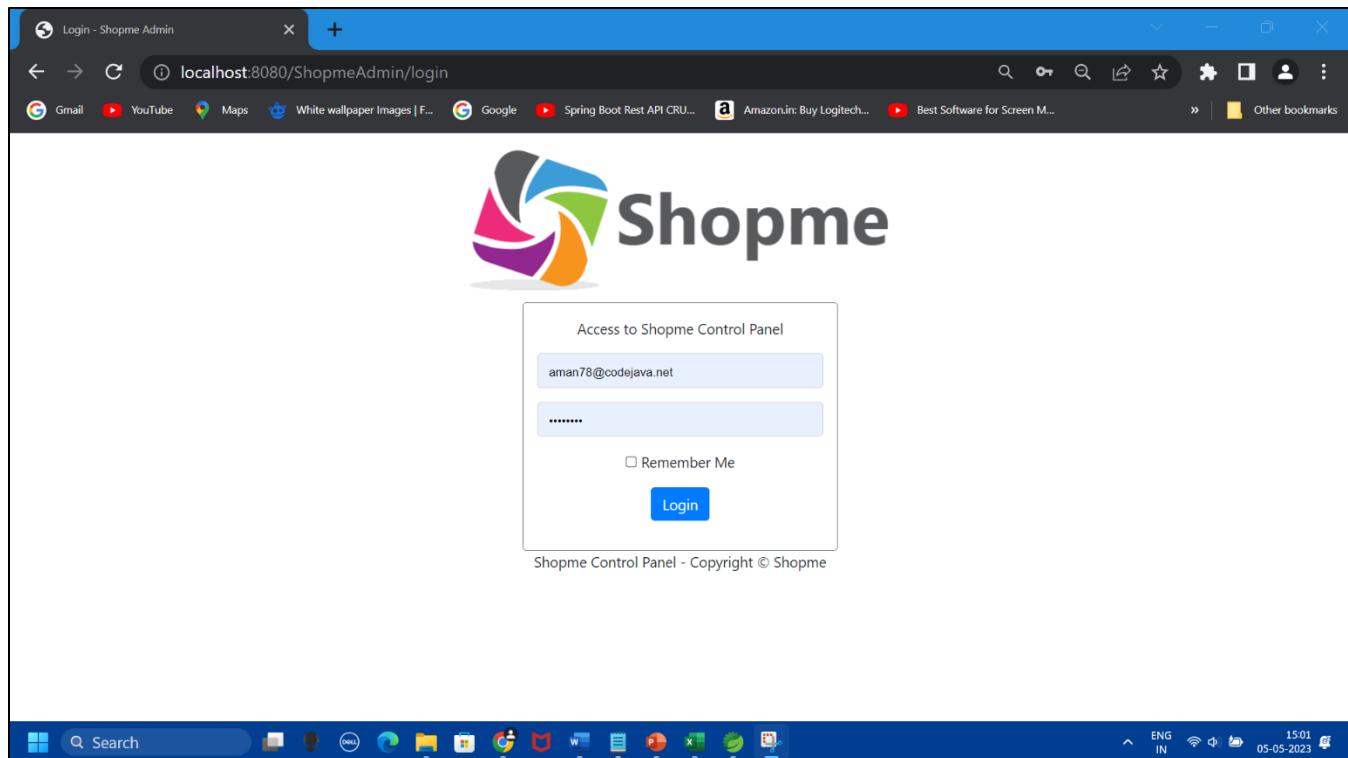


Figure.6.1.1 Login Screen Admin

User Sign Up (Frontend Application) :

The frontend application of the e-commerce Spring Boot app provides a user sign-up feature, allowing new users to create an account.

Users will be required to provide specific information such as their name, email address, password, and other relevant details. The entered information will be validated for correctness and uniqueness.

Once the user submits the sign-up form, the application will process the data and create a new user account in the system. The account details will be stored securely in the database for future authentication and personalized user experience.

After successful sign-up, users will be redirected to the login page to access the features and functionalities of the e-commerce application.

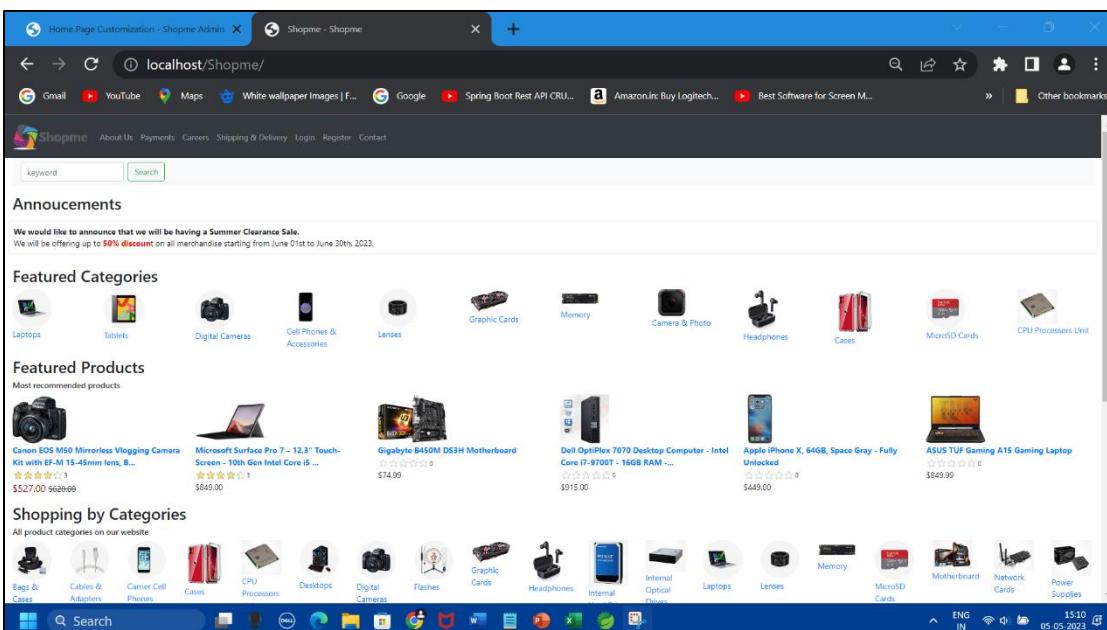
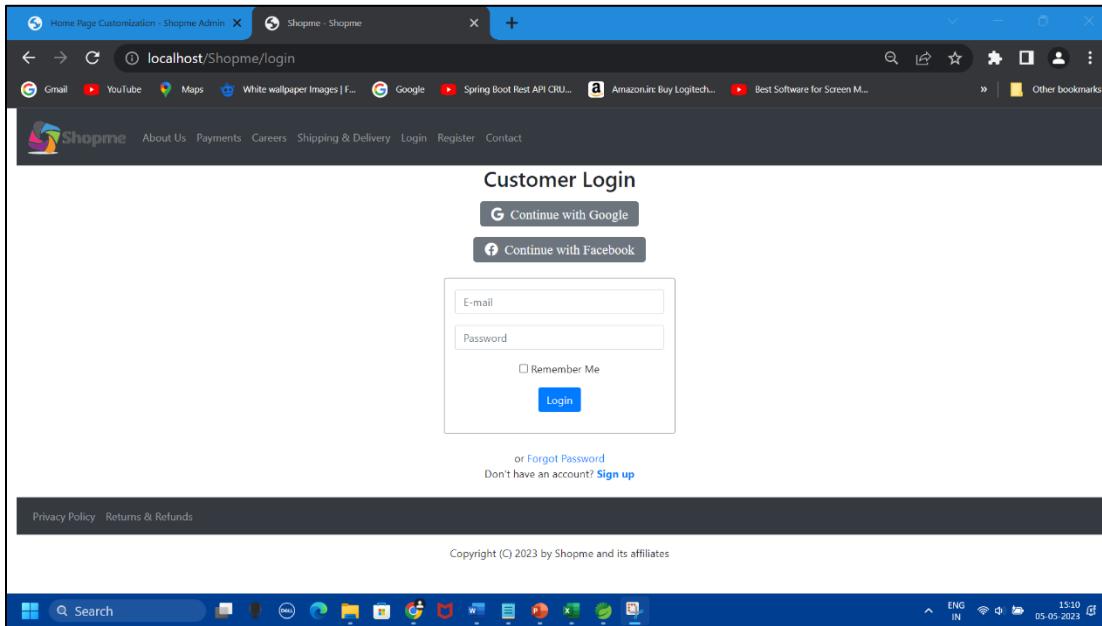


Figure.6.1.2 Login Screen User

Figure.6.1.2 Frontend Page

Home Screen :

The home screen of the Spring Boot e-commerce ShopMe application is designed to provide a user-friendly interface with various sections and features for seamless navigation and access to different functionalities.

The key components of the home screen are as follows:

Client Activities: Located in the top right corner of the screen, this section presents a summary of the client's recent activities, including their orders, wishlist, and shopping cart status.

Client Active Status: Positioned in the bottom right corner, this section indicates the current online status of the client, showing whether they are actively browsing the application.

Featured Categories: Positioned at the bottom section of the home screen, this section highlights the featured product categories, allowing users to explore and navigate to specific product categories of interest.

Shop by Brands: Also located at the bottom section, this feature enables users to browse products based on their preferred brands. It provides a convenient way to discover products from specific brands available in the e-commerce store.

Articles: This section presents articles related to various topics, such as product reviews, buying guides, and industry news. It offers informative content to enhance the user's shopping experience and provide valuable insights.

Admin Panel :

Admin Panel Screen

The Admin Panel screen of the e-commerce application provides a comprehensive interface for managing various aspects of the online store. It allows administrators to perform administrative tasks efficiently. Here are the key sections and functionalities available in the Admin Panel:

Summary Boxes:

This section displays important summary information such as total users, customers, orders, sales, and other relevant metrics. It provides a quick overview of the store's performance and key statistics.

Users Management:

This section allows administrators to manage user accounts, including creating new accounts, updating existing user information, and deleting user accounts if necessary. It provides functionalities for user authentication and role-based access control.

Customers Management:

This section enables administrators to manage customer accounts. They can view customer details, update customer information, and handle customer-related tasks such as managing addresses, tracking orders, and handling customer inquiries.

Brands Management:

This section allows administrators to manage brands available in the e-commerce store. They can add new brands, edit existing brand information, and delete brands if required. It helps in organizing and categorizing products by brand.

Categories Management:

This section enables administrators to manage product categories. They can create new categories, update existing category details, and delete categories as needed. It helps in structuring and organizing the product catalog.

Products Management:

This section provides functionalities for managing products in the e-commerce store. Administrators can add new products, edit product details, manage inventory, set prices, and handle product variations or attributes. It allows for efficient product catalog management.

Menus Management:

This section allows administrators to manage the website menus and navigation. They can create and modify menus, define menu items, and arrange the menu structure. It helps in organizing and enhancing the website's navigation for a better user experience.

Homepage Settings:

This section provides options to customize the homepage of the e-commerce store. Administrators can configure featured products, promotional banners, and other elements to create an attractive and engaging homepage.

Articles Management:

This section enables administrators to manage articles or blog posts related to the products, industry, or other relevant topics. They can create, edit, and delete articles, as well as manage article categories and tags.

Shipping Rates:

This section allows administrators to manage shipping rates and options for different regions or shipping methods. They can set shipping rules, define rates based on weight, location, or other factors, and configure shipping options for customers.

Sales Report:

This section provides comprehensive sales reports and analytics. Administrators can generate sales reports based on various criteria such as date range, product category, or customer segment. It helps in analyzing sales performance and making informed business decisions.

The Admin Panel screen of the e-commerce application empowers administrators to efficiently manage various aspects of the online store, ensuring smooth operations and providing a seamless shopping experience for customers

The screenshot shows the Shopme Control Panel interface. At the top, there are two tabs: "Home - Shopme Admin" and "Shopme - Shopme". The URL in the address bar is "localhost:8080/ShopmeAdmin/". The page title is "Shopme Control Panel". A welcome message "Welcome Aman Gameti [Admin]" is displayed. The main content area is divided into several sections:

- General Information**: Site Name: Shopme, Currency Name: United States Dollar, Currency Symbol: \$, Decimal Point Type: POINT, Decimal Digits: 2, Thousands Point Type: COMMA.
- Users: 23**: Add User, Enabled: 23, Disabled: 0, View All.
- Categories: 31**: Add Category, Root Categories: 2, Enabled: 31, Disabled: 0, View All.
- Brands: 54**: Add Brand, View All.
- Products: 105**: Add Product, Enabled: 105, Disabled: 0, In Stock: 104, Out of Stock: 1, View All.
- Questions: 22**: View All, Approved: 22, UnApproved: 0, Answered: 11, UnAnswered: 11.
- Reviews: 52**: View All, Reviewed Products: 32.
- Customers: 40**: View All, Enabled: 37, Disabled: 3.
- Shipping Rates: 28**: Add Shipping Rate, COD Supported: 17, View All.
- Orders: 160**
- Articles: 9**
- Menu Items: 6**
- Storefront Sections: 7**
- Settings**

Figure.6.1.3 Admin Frontend Page

The screenshot shows the "Manage Products" page. The URL in the address bar is "localhost:8080/ShopmeAdmin/products/page/1?sortField=id&sortDir=asc&categoryId=0". The page title is "Products - Shopme Admin". The main content area displays a table of products:

ID	Main Image	Product Name	Brand	Category	Enabled	Action
1		Canon EOS M50 Mirrorless Vlogging Camera Kit with EF-M 15-45mm lens, B...	Canon	Digital Cameras	<input checked="" type="checkbox"/>	
2		Sony ZV-1 Camera for Content Creators, vlogging and YouTube with flip ...	Sony	Digital Cameras	<input checked="" type="checkbox"/>	
3		Panasonic Lumix FZ80 4K Digital Camera	Panasonic	Digital Cameras	<input checked="" type="checkbox"/>	
4		Pelican 1200 Case With Foam (Black)	Pelican	Bags & Cases	<input checked="" type="checkbox"/>	

Figure.6.1.4 Admin Frontend Page Products

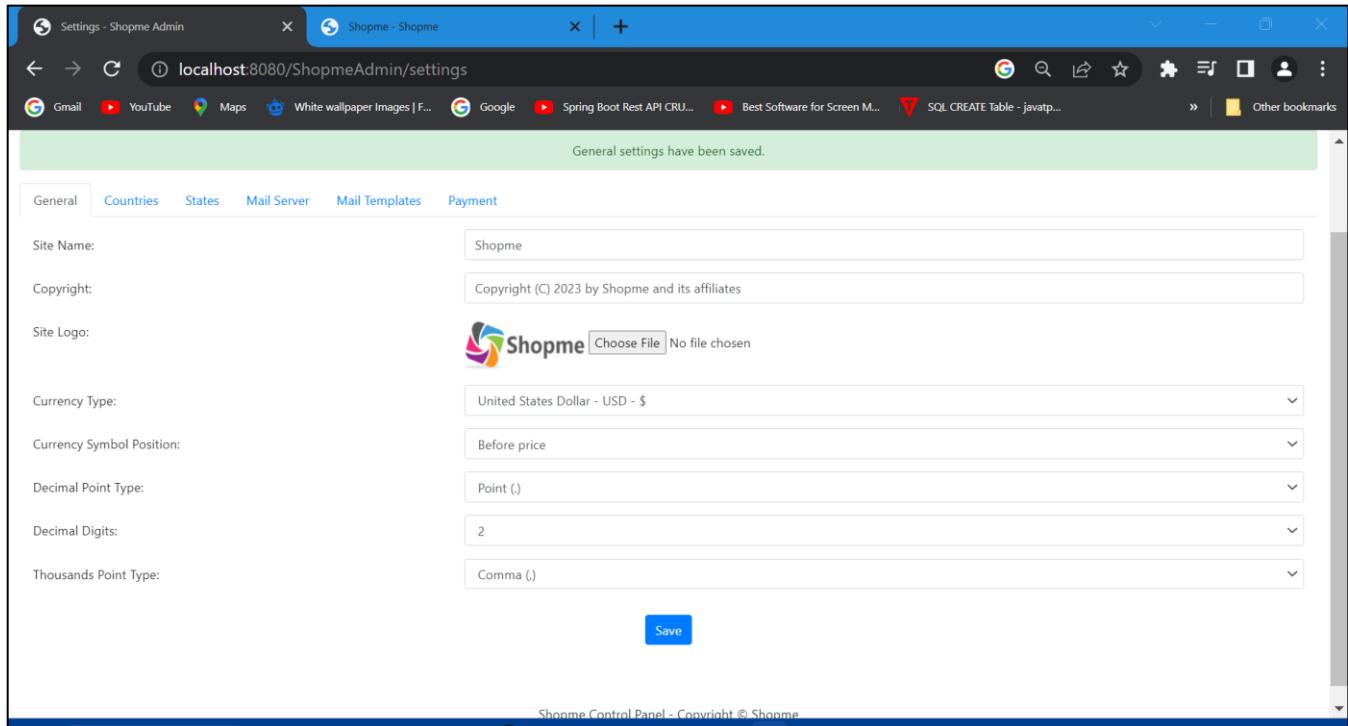


Figure.6.1.5 Admin Frontend Page Settings module

Here above functionalites and options are manged by roles and then only they can select options

Modules Roles \	Users	Categories	Brands	Products	Questions	Reviews	Customers	Shipping	Orders	Reports	Articles	Menus	Settings
Admin													
Salesperson													
Editor													
Shipper													
Assistant													

Allowed Denied

No	Role Name	Description
1	Admin	manage everything
2	Salesperson	manage product price, customers, shipping, orders and sales report
3	Editor	manage categories, brands, products, articles and menus
4	Shipper	view products, view orders and update order status
5	Assistant	manage questions and reviews

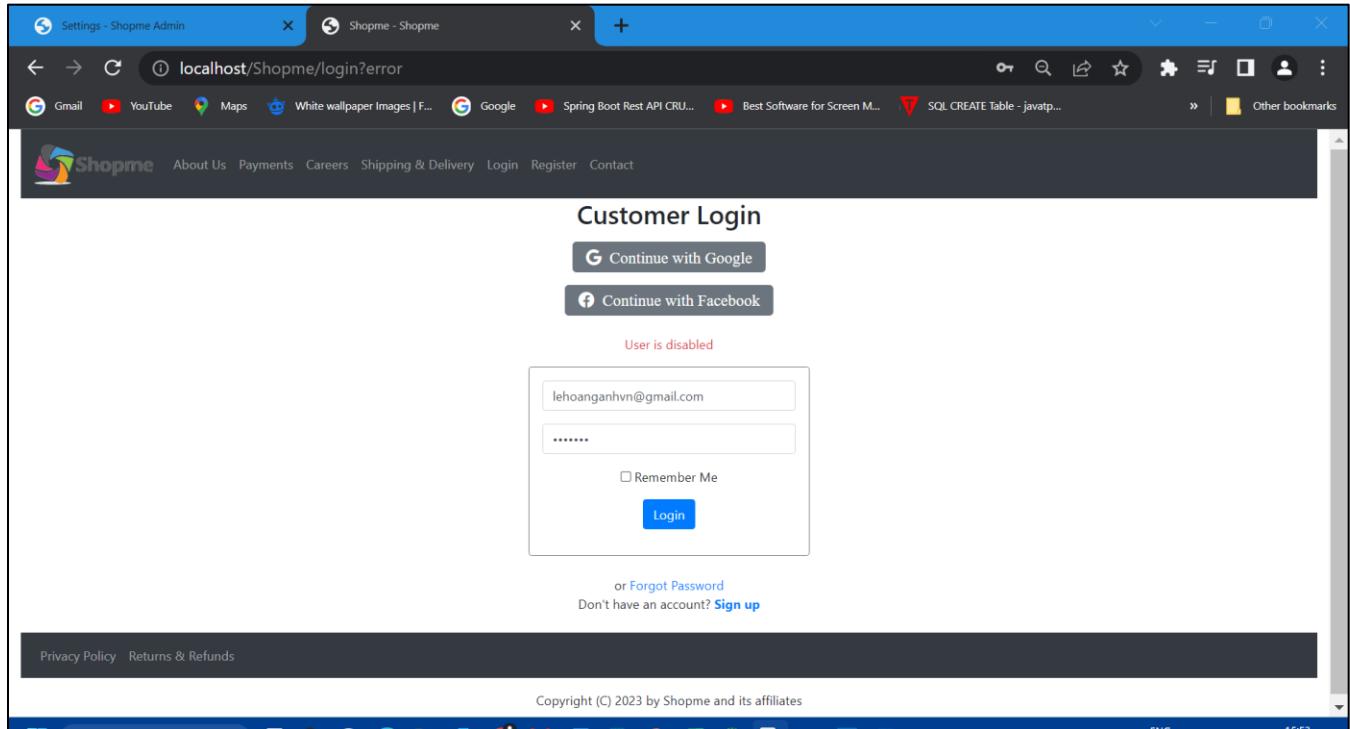


Figure.6.1.6 Frontend Page customer Login

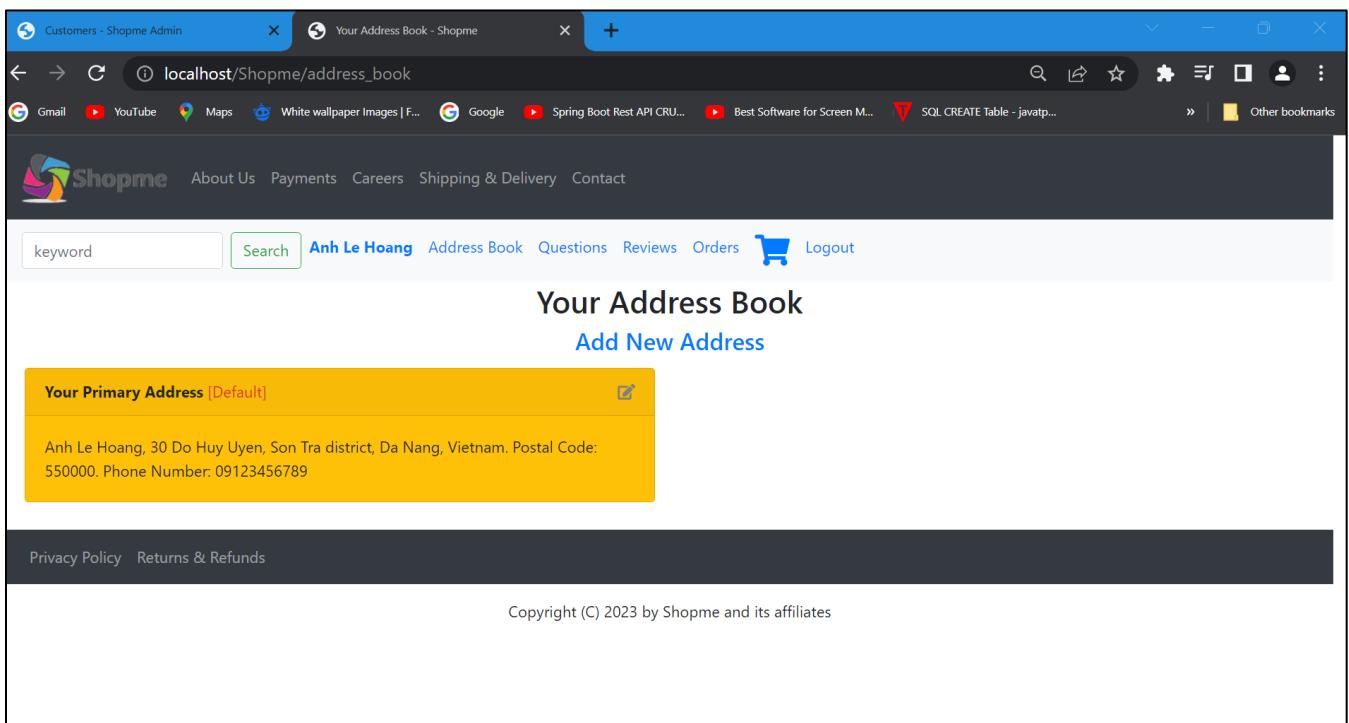


Figure.6.1.7 Frontend Page customer Login and address book

The screenshot shows a web browser window for the Shopme platform. The title bar indicates the user is logged in as 'Anh Le Hoang'. The main content is titled 'Your Shopping Cart' and lists two items:

- Canon EOS M50 Mirrorless Vlogging Camera Kit with EF-M 15-45mm lens, B...**
Quantity: 1
Price: X \$527.00 \$620.00
Subtotal: = \$2,108.00
- Microsoft Surface Pro 7 – 12.3" Touch-Screen – 10th Gen Intel Core i5 ...**
Quantity: 2
Price: X \$849.00
Subtotal: = \$849.00

The total estimated amount is displayed as **Estimated Total: \$2,957.00**. A red 'Check Out' button is visible.

Figure.6.1.8 Frontend Page customer cart details

The screenshot shows a web browser window for the Shopme platform. The title bar indicates the user is logged in as 'Anh Le Hoang'. The main content is titled 'My Orders' and displays a table of recent purchases:

ID	Order Time	Products	Total	Status
18	2021-03-10 18:11:21	<ul style="list-style-type: none"> Dell OptiPlex 7070 Desktop Computer - Intel Core i7-9700T - 16GB RAM -... Bosch BT150 Compact Extendable Tripod with Adjustable Legs BT 150 	\$966.62	DELIVERED
17	2021-03-08 22:28:53	<ul style="list-style-type: none"> Samsung Electronics EVO Select 256GB microSDXC UHS-I U3 100MB/s Full H... 	\$61.20	DELIVERED
16	2021-03-04 15:56:56	<ul style="list-style-type: none"> Lenovo Legion 5 Gaming Laptop 	\$2,017.48	DELIVERED
15	2021-03-04 15:53:50	<ul style="list-style-type: none"> Dell Inspiron 27 AIO, 27 inch, FHD Touchscreen, Intel Core i7-10510U, ... 	\$4,953.75	DELIVERED

Figure.6.1.8 Frontend Page customer order history

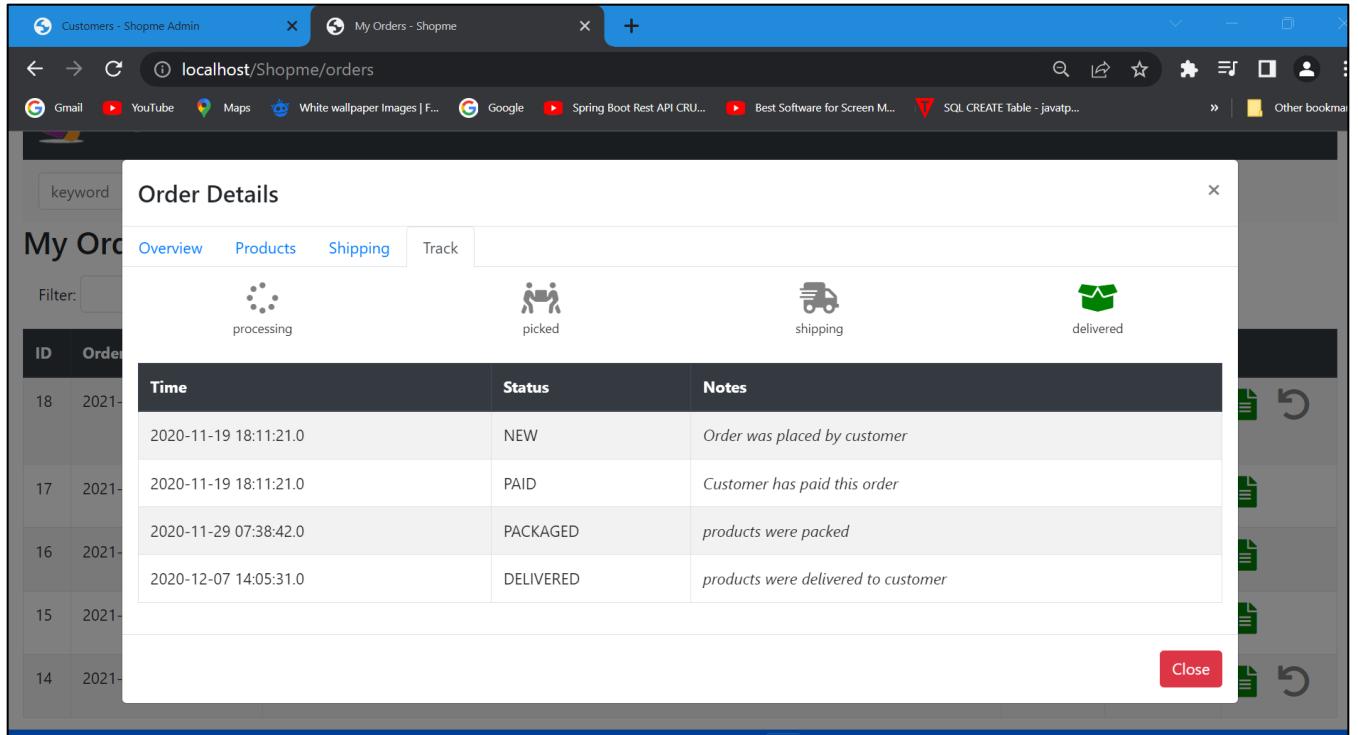


Figure.6.1.10 Frontend Page customer order track

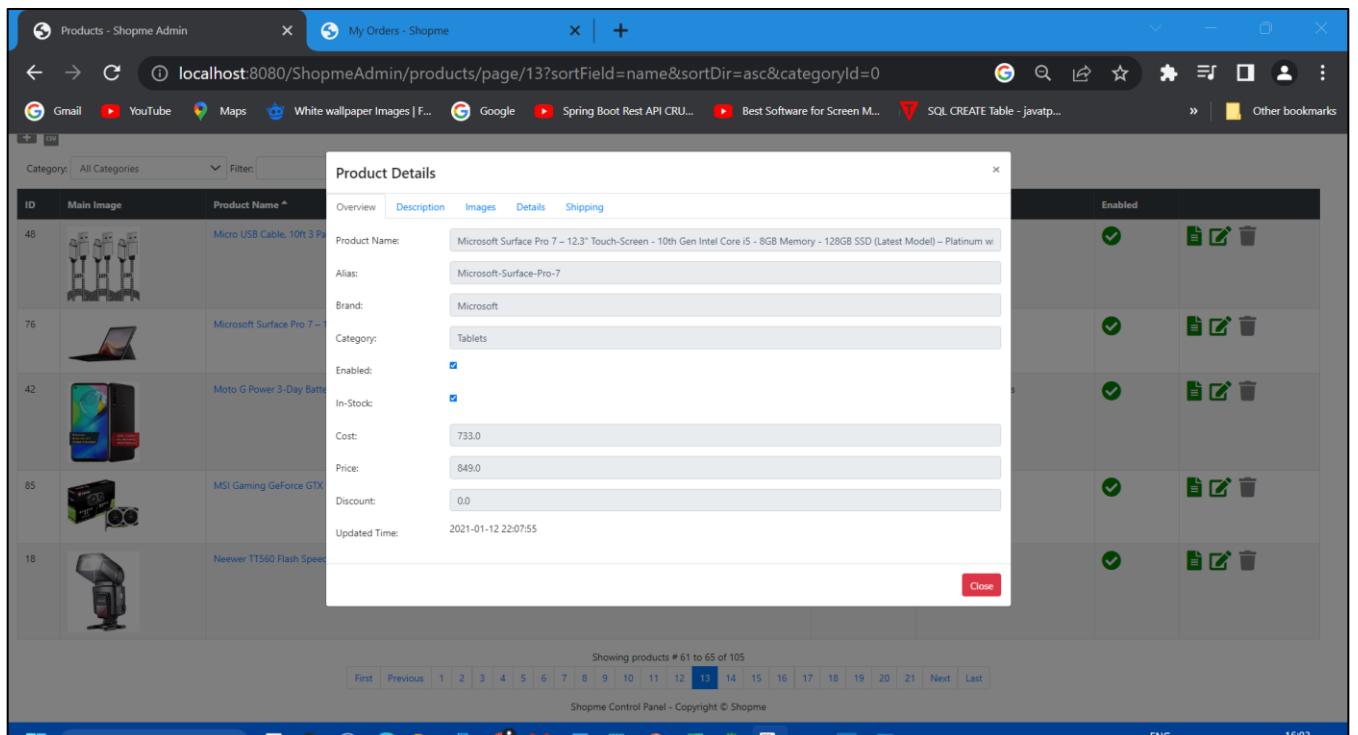


Figure.6.1.11 Admin Frontend Product details mange

The screenshot shows the 'Product Details' modal window overlaid on a grid of products. The modal displays five additional images for a Microsoft Surface Pro 7 product, labeled Extra Image #1 through Extra Image #5. The main image is labeled 'Main Image: Surface Pro 7 main.png'. The modal has tabs for Overview, Description, Images, Details, and Shipping.

ID	Main Image	Product Name	Category	Enabled
48		Micro USB Cable, 10ft 3-Pin		<input checked="" type="checkbox"/>
76		Microsoft Surface Pro 7 - 1		<input checked="" type="checkbox"/>
42		Moto G Power 3-Day Battery		<input checked="" type="checkbox"/>
85		MSI Gaming GeForce GTX 1660 192-Bit HDMI/DP 6GB GDRR5 HDCP Support Dir...	MSI	<input checked="" type="checkbox"/>
18		Neewer TT560 Flash Speedlite	Neewer	<input checked="" type="checkbox"/>

Showing products # 61 to 65 of 105

Figure.6.1.12 Admin Frontend Product details mange images

The screenshot shows the 'Brands' modal window overlaid on a grid of brands. The modal displays categories for a selected brand, specifically Acer. The categories listed are Laptop, Tablets, Desktops, Cables & Adapters, Bags & Cases, CPU Processors Unit, Graphic Cards, and Internal Optical Drives.

Brand ID	Logo	Brand Name	Categories
37		Acer	Laptops Tablets Desktops
28		Aioneus	Cables & Adapters
13		AmazonBasics	Bags & Cases
43		AMD	CPU Processors Unit Graphic Cards
9		Apple	Laptops Unlocked Cell Phones Carrier Cell Phones Tablets Desktops Cell Phones & Accessories
40		ASUS	Laptops Motherboard Tablets Internal Optical Drives Sound Cards Desktops Power Supplies
17		Bosch	Tripods & Monopods
12		CADeN	Bags & Cases
1		Canon	Camera & Photo Digital Camera Flashes Lenses
47		Cosair	Power Supplies Memory

Showing brands # 1 to 10 of 54

Figure.6.1.13 Admin Frontend brands manage

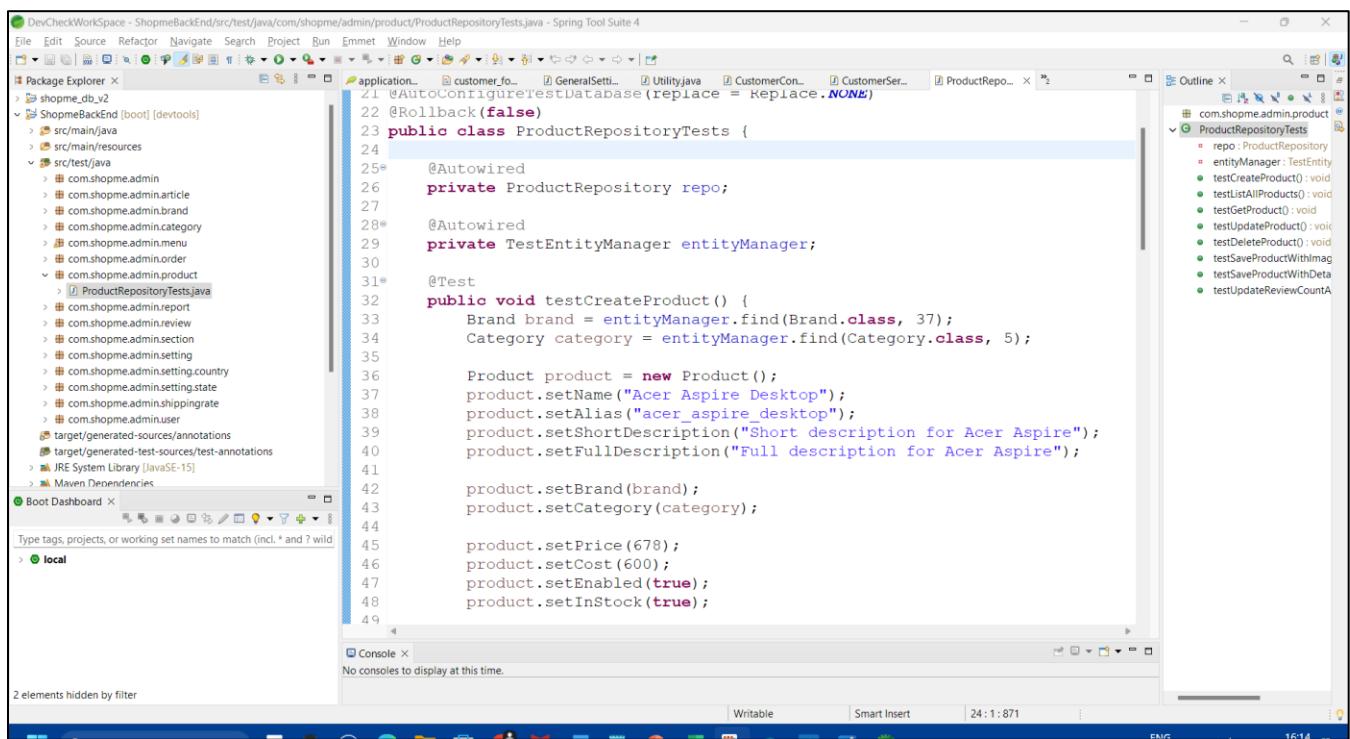
7.0 TESTING

7.1 TESTING PLAN / STRATEGY

Testing Strategy

While my web app was customized in the system, testing revealed that this customization or customization did not result in any malicious processing or breach. Subsequent care was taken while the application was made on a local machine. We can eliminate these mistakes by thinking about current or possible situations but if they do happen, our request is strong enough to handle it. Testing the app for all possible situations that it can / cannot handle and our application remains firm is what we mean.

Test snapshots :



```

21 @AutoConfigureTestDatabase(replace = Replace.NONE)
22 @Rollback(false)
23 public class ProductRepositoryTests {
24
25     @Autowired
26     private ProductRepository repo;
27
28     @Autowired
29     private TestEntityManager entityManager;
30
31     @Test
32     public void testCreateProduct() {
33         Brand brand = entityManager.find(Brand.class, 37);
34         Category category = entityManager.find(Category.class, 5);
35
36         Product product = new Product();
37         product.setName("Acer Aspire Desktop");
38         product.setAlias("acer_aspire_desktop");
39         product.setShortDescription("Short description for Acer Aspire");
40         product.setFullDescription("Full description for Acer Aspire");
41
42         product.setBrand(brand);
43         product.setCategory(category);
44
45         product.setPrice(678);
46         product.setCost(600);
47         product.setEnabled(true);
48         product.setInStock(true);
49
50     }
51
52     @Test
53     public void testUpdateProduct() {
54         Product product = repo.findById(1L);
55         product.setPrice(1000);
56         repo.save(product);
57
58         assertEquals(1000, product.getPrice());
59     }
60
61     @Test
62     public void testDeleteProduct() {
63         Product product = repo.findById(1L);
64         repo.delete(product);
65
66         assertNull(repo.findById(1L));
67     }
68
69     @Test
70     public void testSaveProductWithImage() {
71         Product product = new Product();
72         product.setName("Acer Aspire Desktop");
73         product.setAlias("acer_aspire_desktop");
74         product.setShortDescription("Short description for Acer Aspire");
75         product.setFullDescription("Full description for Acer Aspire");
76
77         product.setBrand(brand);
78         product.setCategory(category);
79
80         product.setPrice(678);
81         product.setCost(600);
82         product.setEnabled(true);
83         product.setInStock(true);
84
85         product.setImage("imagePath");
86
87         repo.save(product);
88
89         assertEquals("imagePath", product.getImage());
90     }
91
92     @Test
93     public void testSaveProductWithoutImage() {
94         Product product = new Product();
95         product.setName("Acer Aspire Desktop");
96         product.setAlias("acer_aspire_desktop");
97         product.setShortDescription("Short description for Acer Aspire");
98         product.setFullDescription("Full description for Acer Aspire");
99
100        product.setBrand(brand);
101        product.setCategory(category);
102
103        product.setPrice(678);
104        product.setCost(600);
105        product.setEnabled(true);
106        product.setInStock(true);
107
108        repo.save(product);
109
110        assertEquals(null, product.getImage());
111    }
112
113    @Test
114    public void testGetProduct() {
115        Product product = repo.findById(1L);
116
117        assertEquals("Acer Aspire Desktop", product.getName());
118        assertEquals("acer_aspire_desktop", product.getAlias());
119        assertEquals("Short description for Acer Aspire", product.getShortDescription());
120        assertEquals("Full description for Acer Aspire", product.getFullDescription());
121        assertEquals(678, product.getPrice());
122        assertEquals(600, product.getCost());
123        assertEquals(true, product.isEnabled());
124        assertEquals(true, product.isInStock());
125    }
126
127    @Test
128    public void testListAllProducts() {
129        List<Product> products = repo.findAll();
130
131        assertEquals(10, products.size());
132    }
133
134    @Test
135    public void testUpdateReviewCount() {
136        Product product = repo.findById(1L);
137
138        product.setReviewCount(5);
139
140        repo.save(product);
141
142        assertEquals(5, product.getReviewCount());
143    }
144
145    @Test
146    public void testDeleteReviewCount() {
147        Product product = repo.findById(1L);
148
149        product.setReviewCount(5);
150
151        repo.delete(product);
152
153        assertEquals(null, product.getReviewCount());
154    }
155}

```

Figure.7.1.1 Testing add Products in product repository

Testing has been carried out on main controllers for different modules throughout the application development

The screenshot shows the Spring Tool Suite interface with the following details:

- File Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Emmet, Window, Help.
- Toolbars:** Standard, Selection, Search, Navigator, Outline, Properties, Status Bar.
- Left Sidebar:** Package Explorer (com.shopme.admin.user), JUnit (PasswordEncoderTest).
- Central Area:**

```

1 package com.shopme.admin.user;
2
3 import static org.assertj.core.api.Assertions.assertThat;
4
5
6 public class PasswordEncoderTest {
7     @Test
8     public void testEncodePassword() {
9         BCryptPasswordEncoder passwordEncoder = new BCryptPasswordEncoder();
10        String rawPassword = "Aman2020";
11        String encodedPassword = passwordEncoder.encode(rawPassword);
12
13        System.out.println(encodedPassword);
14
15        boolean matches = passwordEncoder.matches(rawPassword, encodedPassword);
16
17        assertThat(matches).isTrue();
18    }
19
20}
21
22

```
- Right Sidebar:** Outline (com.shopme.admin.user, PasswordEncoderTest, testEncodePassword(), void).
- Bottom Panels:** Failure Trace, Boot Dashboard, Console (terminated > PasswordEncoderTest [JUnit] D:\spring-tool-suite-4-4.17.1.RELEASE-e4.26.0-win32.win32.x86_64.self-extracting\contents\sts-4.17.1.RELEASE\plugins\org.eclipse.jst.j2ee\2.10.0\src\main\java\com\shopme\admin\user\PasswordEncoderTest.java).

Figure.7.1.2 Testing password encoder on plain text

The screenshot shows the Spring Tool Suite interface with the following details:

- File Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Emmet, Window, Help.
- Toolbars:** Standard, Selection, Search, Navigator, Outline, Properties, Status Bar.
- Left Sidebar:** Package Explorer (com.shopme.admin.product, com.shopme.admin.report, com.shopme.admin.review, com.shopme.admin.section, com.shopme.admin.setting, com.shopme.admin.shippingrate, com.shopme.admin.user, target/generated-sources/annotations, target/generated-test-sources/test-annotations, JRE System Library [JavaSE-15], Maven Dependencies).
- Central Area:**

```

38
39        assertThat(savedUser.getId()).isGreaterThan(0);
40    }
41
42    @Test
43    public void testCreateNewUserWithTwoRoles() {
44        User userRavi = new User("ravi@gmail.com", "ravi2020", "Ravi", "Kumar");
45        Role roleEditor = new Role(3);
46        Role roleAssistant = new Role(5);
47
48        userRavi.addRole(roleEditor);
49        userRavi.addRole(roleAssistant);
50
51        User savedUser = repo.save(userRavi);
52
53        assertThat(savedUser.getId()).isGreaterThan(0);
54    }
55
56    @Test
57    public void testListAllUsers() {

```
- Right Sidebar:** Outline (com.shopme.admin.user, UserRepositoryTests, repo: UserRepository, entityManager: TestEntityManager, testCreateNewUserWithTwoRoles(), testGetUserByEmail(), testGetUserById(), testUpdateUserDetails(), testUpdateUserRoles(), testDeleteUser(), testDisableUser(), testCountByJd(), testEnableUser(), testListFirstPage(), testSearchUsers(), void).
- Bottom Panels:** Failure Trace, Boot Dashboard, Console (terminated > PasswordEncoderTest [JUnit] D:\spring-tool-suite-4-4.17.1.RELEASE-e4.26.0-win32.win32.x86_64.self-extracting\contents\sts-4.17.1.RELEASE\plugins\org.eclipse.jst.j2ee\2.10.0\src\main\java\com\shopme\admin\user\PasswordEncoderTest.java).

Figure.7.1.3 Testing adding users in database

7.2 TEST RESULTS AND ANALYSIS

7.2.1 Test Cases (Test Id, Test Condition, Expected Output, Actual Output, Remark)

TEST CASES

Sr. No.	Purpose	Input	State	Expected Output	Actual Output	Test Result
1	User Login	Registered Email, Correct Password, Submit	Logout	Success	Success	Pass
2	User Login	Unregistered Email, Password	Logout	Ask user to Register First	Ask user to Register First	Pass
3	User Register	Username, Email, Password	New User	Success	Success	Pass
4	Users Profile	Users Profile Picture	Client data	Success	Success	Pass
5	Product add	Adding product details	Product added	notification shown	Success	Pass
6	Category add	Adding category details	Category added	Notification shown	Success	Pass
7	Add User	Adding user detail	User Added	Notification shown	Success	Pass
8	Customer register	Adding duplicate emails	duplicate error	Success	Successfully message shown in form	Pass
9	Customer Login	Adding valid details	Access allow	Success	Success	Pass

8.0 CONCLUSION AND DISCUSSION

8.1 OVERALL ANALYSIS OF INTERNSHIP / PROJECT VIABILITIES

From the project we made these are the following things which can be concluded:

Overall, the spring boot e-commerce web application provides a robust platform for managing users, customers, products, categories, menus, and other essential aspects of an e-commerce business. It empowers administrators to efficiently manage and operate their online store while providing a seamless shopping experience for customers.

During the development of the REST API, I gained valuable insights and learning experiences. I learned how to design and implement robust and scalable APIs using the principles of REST architecture.

Overall, working on the REST API helped me enhance my skills in API development, enabling me to build efficient and reliable web services.

Web Development:

- 1) HTML
- 2) CSS
- 3) JavaScript
- 4) Frontend libraries

Soft Skills:

- 1) Communication
- 2) Time Management
- 3) Adaptability
- 4) Problem Solving

Backend:

- 1) Spring boot
- 2) Rest API
- 3) Database management
- 4) Debugging

- Developing a complete shopping website using Java, Spring Boot, Thymeleaf, Bootstrap, jQuery

and MySQL database.

- Developing the Shopping application allowing customers to browse products and place orders
- Coding the Admin application for managing users, categories, brands, products, customers, orders, sales report...
- Learning how to use Spring RESTful Webservices on the client side

REFERENCES:

<https://www.javatpoint.com/spring-boot-annotations>

<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>

<https://www.baeldung.com/rest-with-spring-series>

<https://stackoverflow.com/questions/tagged/spring-boot>

<https://start.spring.io/>