

BSSE FINAL PROJECT

Requirements Specification

Sportswearxpress



Project Advisor

Anum Mustaqeem

Presented by:

Group ID: F24SE002

Student Reg#

L1S21BSSE0072

L1F21BSSE0564

L1F21BSSE0476

Student Name

M Awais Akram

Muhammad Muneeb Butt

Muhammad Qasim Abbass

Faculty of Information Technology

University of Central Punjab

Software Requirements Specification

Version 1

SportwearXpress Advisor:
Anum Mustaqeem

F24SE002

Member Name	Primary Responsibility
Awais Akram	Functional Requirements
Muneeb Butt	Non-functional Requirements & Overall Description
Qasim Abbas	Introduction and Background

Table of Contents

Table of Contents	i
Revision History	ii
Abstract	iii
1. Introduction and Background	2
1.1 Product (Problem Statement)	2
1.2 Background	2
1.3 Scope	3
1.4 Objective(s)/Aim(s)/Target(s)	3
1.5 Challenges	4
1.6 Learning Outcomes	4
1.7 Nature of End Product	5
1.8 Completeness Criteria	5
1.9 Business Goals	6
1.10 Related Work/ Literature Survey/ Literature Review	6
1.11 Document Conventions	7
2. Overall Description	8
2.1 Product Features	8
2.2 User Classes and Characteristics	8
2.3 Operating Environment	9
2.4 Design and Implementation Constraints	11
2.5 Assumptions and Dependencies	12
3. Functional Requirements	13
3.1 User Authentication and Profile Management	15
3.2 Add to Cart	16
3.3 Product Customization	17
3.4 Save Product Design	18
3.5 Adjust Pricing	19
3.6 Track Design Changes	20
3.7 Place Order	21
3.8 Track Order	22
3.9 Interact with Seller	23
3.10 Product Reviews and Ratings	24
3.11 Analytics and Reporting	25
3.12 Cancel Order	26
3.13 Edit Profile	27
3.14 Remove From Cart	28
3.15 Create Profile	29
3.16 Send Refund	30
3.17 Requirements Analysis and Modeling	31
4. Nonfunctional Requirements	38
4.1 Performance Requirements	38
4.2 Safety Requirements	38
4.3 Security Requirements	38
4.4 Additional Software Quality Attributes	39
5. Other Requirements	39
6. Revised Project Plan	40
7. References	40
Appendix A: Glossary	41
Appendix B: IV & V Report	41

Revision History

Name	Date	Reason For Changes	Version

Abstract

This project, SportswearXpress, focuses on developing an AI-driven e-commerce platform designed to transform the sportswear customization experience by addressing critical industry challenges. Precise design tools and high customer satisfaction are paramount in this space, yet existing platforms often lack advanced design features, dynamic pricing, and real-time communication, resulting in miscommunication and unmet customer expectations.

To address these issues, SportswearXpress integrates AI-powered design tools that guide users through the customization process step-by-step. An intelligent chatbot recommends appropriate tools and actions tailored to each task, ensuring efficient and accurate completion of personalized designs. After finalizing the design, the platform dynamically adjusts pricing based on design complexity and the number of tools used, offering transparency and fair pricing.

Key knowledge areas include artificial intelligence for personalized design suggestions, real-time communication systems to streamline user-seller interactions, and dynamic pricing algorithms that align cost with customization complexity. Leveraging technologies like TensorFlow.js for AI, OpenCV for image processing, and Socket.io for instant communication, this platform will enhance user satisfaction, minimize design errors, and optimize the customization workflow. The result is a user-friendly, engaging platform that advances both customer satisfaction and operational efficiency in the sportswear industry.

1. Introduction and Background

1.1 Product (Problem Statement)

SportswearXpress is an AI-powered e-commerce platform designed to solve the challenges associated with customizing sportswear. Traditional customization systems lack intuitive design guidance, real-time communication, and transparent pricing, leading to unmet customer expectations and inefficiencies. SportswearXpress leverages AI to provide users with step-by-step design recommendations, dynamic pricing based on customization complexity, and seamless real-time communication with sellers. This solution aims to enhance user satisfaction and streamline the customization process, setting a new standard for personalized sportswear shopping.

1.2 Background

The sportswear customization domain is rapidly evolving, driven by a growing demand for personalized products. However, current e-commerce platforms in this space often lack sophisticated design tools, dynamic pricing systems, and real-time communication features, resulting in customer dissatisfaction and operational inefficiencies. SportswearXpress aims to address these challenges by providing an AI-enhanced platform that allows for accurate and intuitive customization, improving customer experience and seller efficiency.

1.3 Scope

The scope of SportswearXpress includes essential features aimed at enhancing the sportswear customization experience:

- **User Customization Interface:** A web platform for creating personalized sportswear designs with various options.
- **AI-Powered Design Tools:** Intelligent tools that guide users through the customization process.
- **AI Chatbot:** Real-time assistance and recommendations throughout the design journey.
- **Dynamic Pricing Model:** Automatic adjustments of prices based on customization complexity.
- **Real-Time Communication:** Direct interaction between customers and sellers for immediate support.
- **User Account Management:** Secure accounts for saving designs and tracking order history.
- **Administrative Dashboard:** Tools for sellers to manage products and analyze customer interactions.
- **Testing and Deployment:** Comprehensive testing before launch and ongoing maintenance for continuous improvement.

1.4 Objective(s)/Aim(s)/Target(s)

The objectives of SportswearXpress are to:

- Develop an AI-driven platform for sportswear customization.
- Integrate AI tools to guide users in personalizing sportswear designs.
- Implement a dynamic pricing model based on design complexity.
- Enable real-time, seamless communication between users and sellers.
- Improve customer satisfaction and operational efficiency.

1.5 Challenges

The development of SportswearXpress entails several challenges that must be addressed to ensure effective implementation and user satisfaction. Key challenges include:

- **AI Algorithm Complexity:** Designing and implementing AI-driven design tools for user customization requires overcoming challenges in model training, optimization, and ensuring responsiveness during user interactions.
- **Real-Time Communication:** Establishing seamless real-time communication between users and sellers is critical, requiring robust backend integration and reliable performance under varying network conditions.
- **Dynamic Pricing Algorithm:** Developing a dynamic pricing model that accurately reflects customization complexity while maintaining transparency and user trust poses significant challenges in algorithm design and implementation.
- **User Experience Design:** Creating an intuitive and user-friendly interface that caters to diverse customer preferences demands thorough user testing and iterative design adjustments.
- **Scalability:** Ensuring the platform can efficiently handle increased user traffic and complex design requests without compromising performance is essential for future growth.

1.6 Learning Outcomes

Through the development of SportswearXpress, the project team aims to achieve the following learning outcomes:

- **Software Engineering Practices:** Gain practical experience in software engineering methodologies, including requirements gathering, system design, implementation, testing, and deployment.
- **AI and Machine Learning Proficiency:** Acquire proficiency in leveraging AI and machine learning technologies to enhance user experience, particularly in the areas of design recommendations and real-time user interactions.
- **User-Centered Design Expertise:** Develop skills in user-centered design principles to create intuitive and engaging interfaces that cater to the diverse needs of sportswear customers.
- **Dynamic Pricing Strategies:** Learn to design and implement dynamic pricing models that adapt based on customization complexity and user preferences.
- **Collaborative Skills Enhancement:** Enhance collaboration and communication skills through teamwork, stakeholder engagement, and effective project management.

1.7 Nature of End Product

The final product is an advanced e-commerce platform allowing users to design customized sportswear using AI-based tools, communicate directly with sellers in real-time, and receive dynamically adjusted pricing. This product aims to set a new standard in sportswear customization, making the process efficient, transparent, and engaging for users.

1.8 Completeness Criteria

The project will be considered complete when:

- The platform offers AI-powered design recommendations.
- Real-time communication and chat features function seamlessly.
- The dynamic pricing model adjusts accurately based on design complexity.
- User interface and backend are fully integrated and operational.
- Accessibility and user experience testing confirm platform usability.
- The website meets usability and accessibility standards, as validated through user testing and feedback from sportswear market experienced individuals.
- Comprehensive testing has been conducted to ensure the reliability, performance, and security of the application across different devices and usage scenarios.
- Documentation, including user manuals, developer guides, and technical specifications, is complete and accessible to stakeholders for reference and future maintenance.

S.No.	Criteria	Weightage %
1	Frontend and Backend development	35
2	Integrate canvas with machine learning algorithm and compose pricing criteria.	15
3	Integration of canvas to user interface	20
4	Machine learning algorithm integration for chatbot.	20
5	Integration of user interface	10

1.9 Business Goals

SportswearXpress aims to achieve the following business objectives:

- **Enhancing Customer Satisfaction:** Elevate customer satisfaction by providing intuitive and precise customization options that cater to individual preferences.
- **Minimizing Design Errors:** Decrease design errors and miscommunication through the implementation of AI-driven guidance and real-time communication tools.
- **Optimizing Operational Efficiency:** Streamline operations by automating design assistance and dynamic pricing adjustments, allowing for a more efficient workflow.
- **Setting Industry Standards:** Establish a competitive benchmark in the sportswear sector by delivering innovative, AI-powered e-commerce solutions that redefine customer engagement.

1.10 Related Work/ Literature Survey/ Literature Review

Before the development of SportswearXpress, a thorough literature review and survey of existing solutions in the domain of AI-driven e-commerce and sportswear customization were conducted. This included an examination of current platforms, research papers, and industry reports on topics such as:

E-Commerce Personalization: Exploration of platforms that provide customization options for consumers and how user preferences are captured.

AI-Powered Design Tools: Investigation of AI technologies enhancing user experience in design applications.

Real-Time Communication Systems: Analysis of tools enabling real-time interaction between users and sellers to reduce misunderstandings.

Dynamic Pricing Strategies: Review of pricing models that adjust based on customization complexity.

By analyzing these existing solutions and research findings, SportswearXpress aims to leverage strengths while addressing limitations identified in prior works.

Related Projects:

Zazzle: Provides a wide range of customizable products, including apparel, with a user-friendly design interface.

Custom Ink: Specializes in custom apparel with an engaging design tool that facilitates collaborative creation and sharing of designs.

1.11 Document Conventions

This SRS follows the standard format for software requirements specifications.

The document utilizes

the following conventions:

Font Styles: Section headings are in bold font of 14 times New Roman, while subsection headings are in bold of size 12. Regular text is used for descriptions and content.

Highlighting: Important terms or concepts may be highlighted using bold or italic font for emphasis.

Numbering: Sections and subsections are numbered sequentially for organization and reference purposes.

Nomenclature: Where applicable, technical terms or acronyms are defined upon first use to ensure clarity and understanding throughout the document.

2. Overall Description

2.1 Product Features

AI-Powered Design Tools: Intuitive tools guide users step-by-step through the customization process, ensuring precision and ease.

Dynamic Pricing Model: Automatic pricing adjustments based on the complexity of each design, ensuring transparency and fairness.

Real-Time Communication: Direct interaction between users and sellers to provide immediate support and clarify customization needs.

User Customization Interface: A user-friendly web platform with a wide array of customization options for personalized sportswear designs.

AI Chatbot: Provides real-time assistance and personalized recommendations throughout the design process.

User Account Management: Secure user accounts for saving designs and tracking order history.

Administrative Dashboard: Tools for sellers to manage products, track customer interactions, and analyze performance.

2.2 User Classes and Characteristics

1. Casual Users

- **Frequency of Use:** Occasional
- **Functions Used:** Basic customization tools and browsing
- **Technical Expertise:** Low to moderate
- **Characteristics:** Seek easy-to-use interfaces and quick customization options. Prefer straightforward and engaging experiences.

2. Frequent Customizers

- **Frequency of Use:** Regular
- **Functions Used:** Advanced design tools, frequent design adjustments
- **Technical Expertise:** Moderate to high
- **Characteristics:** Comfortable with complex customization features. Value precision and detailed design options. Often return for multiple orders

3. Professional Designers

- **Frequency of Use:** High
- **Functions Used:** Full suite of design tools, bulk customization
- **Technical Expertise:** High
- **Characteristics:** Demand robust design functionalities and seamless workflow integration. Focus on efficiency and high-quality outputs.

4. Sellers/Admins

- **Frequency of Use:** Daily
- **Functions Used:** Product management, customer interaction, analytics
- **Technical Expertise:** High
- **Characteristics:** Require comprehensive tools for inventory management, real-time communication, and sales analytics. Prioritize operational efficiency and customer satisfaction.

5. AI Chatbot Users

- **Frequency of Use:** Varies
- **Functions Used:** Customer support and design guidance
- **Technical Expertise:** Low to moderate
- **Characteristics:** Seek quick and helpful responses to inquiries. Rely on AI guidance for design and troubleshooting.

2.3 Operating Environment

The SportswearXpress platform is designed to operate within a specific environment that includes hardware platforms, operating systems, and other software components. The operating environment for the software is as follows:

Hardware Platform:

- SportswearXpress is primarily intended for use on both desktop and mobile devices, including smartphones and tablets.
- The application should be compatible with a range of hardware configurations, including devices with different processing power, memory capacity, and screen sizes.
- External hardware components, such as IP-based cameras or wearable sensors, may be integrated with the application to enhance its functionality.

Operating System and Versions:

- The application should be compatible with popular operating systems, including:
 - **Windows:** Versions 10 and above.
 - **macOS:** Versions 10.13 (High Sierra) and above.
 - **IOS:** Versions 11 and above (for compatibility with iPhones and iPads).
 - **Android:** Versions 10.0 and above (for compatibility with a wide range of Android devices).
- Compatibility with the latest operating system versions is preferred to ensure access to the latest features and security updates.

Other Software Components or Websites: SportswearXpress may require integration with external software components or applications to provide certain functionalities, such as:

System Requirements: The specific system requirements, including minimum hardware specifications and recommended configurations, will be detailed in the application documentation and user guides provided with the software.

2.4 Design and Implementation Constraints

1. Corporate or Regulatory Policies:

- Compliance with industry regulations and standards related to e-commerce and data privacy (e.g., GDPR, CCPA) is mandatory.
- Adherence to corporate policies regarding user data handling and storage to ensure privacy and security.

2. Hardware Limitations:

- Must operate efficiently on devices with limited processing power and memory capacity, ensuring performance across various hardware specifications.
- Timing requirements for real-time communication and dynamic pricing adjustments may pose constraints on system performance.

3. Interfaces to Other Applications:

- Integration with existing e-commerce platforms or third-party services (e.g., payment gateways, shipping services) may limit customization options.
- Dependencies on APIs for functionalities like map services, voice recognition, and image processing.

4. Specific Technologies, Tools, and Databases:

- Utilizing technologies like TensorFlow.js, OpenCV, and [Socket.io](#) limits the choice of programming languages and development frameworks.
- Databases must support rapid data retrieval and real-time updates, influencing the choice of database management systems.

5. Parallel Operations:

- The platform must support simultaneous user interactions and real-time updates, requiring robust concurrency management.
- Parallel operations may necessitate load balancing and efficient resource allocation to maintain performance.

6. Communications Protocols:

- Realtime communication features depend on reliable and secure protocols to ensure seamless interaction between users and sellers.

7. Design Conventions or Programming Standards:

- Adhering to standard design conventions and programming best practices to ensure maintainability and scalability.
- Using a modular design approach to facilitate future enhancements and integrations.

2.5 Assumptions and Dependencies

1. Assumed Factors:

- **Third-Party Components:** It is assumed that third-party components such as [TensorFlow.js](#) for AI functionalities, OpenCV for image processing, and [Socket.io](#) for real-time communication will continue to be supported and updated.
- **Development Environment:** Assumes that development will occur in a stable environment with access to necessary tools, libraries, and frameworks.
- **Operating Environment:** Assumes compatibility with popular operating systems including Windows, macOS, iOS, and Android, and that these systems will remain stable and continue to support the application requirements.

2. Dependencies:

External Software Components:

The project relies on external software components like messenger platforms for communication between buyers and sellers. This dependency means that changes or disruptions in these platforms could impact payment discussions and updates on orders.

Technology Stack:

MongoDB for data storage, Konva.js for enabling product design tools, and AI libraries like TensorFlow.js are critical to the platform. Updates or deprecations in these technologies may require significant modifications to maintain functionality.

APIs and Services:

Integration with APIs for AI-driven customization, image processing, and real-time communication is essential. Changes or interruptions in these APIs could impact features like product design guidance, chatbot interaction, and design visualization.

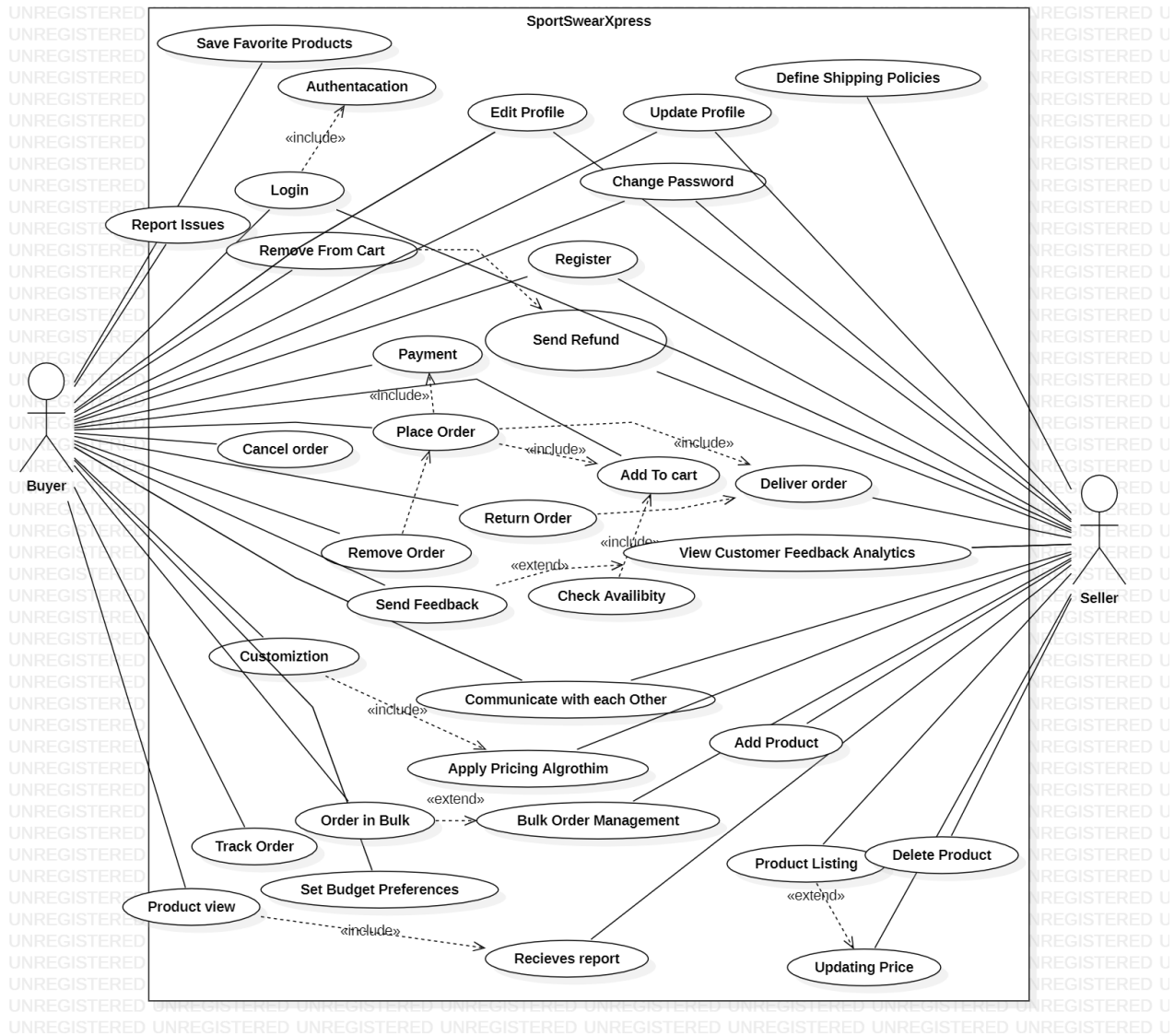
Hardware Dependencies:

Users need compatible devices with sufficient processing power and stable internet connectivity to utilize the interactive design tools and communicate with the seller efficiently. Limitations in user hardware or connectivity might affect the overall experience.

3. Functional Requirements

1. User Registration
2. User Login
3. Product Customization
4. Add to Cart
5. Place Order
6. Payment Processing
7. Order Tracking
8. Product Review
9. Interact with Sellers
10. Dynamic Pricing Model
11. Database Integration
12. Report Generation
13. Check Customization Status
14. Save Product Design

Use-Case Diagram:



3.1 User Authentication and Profile Management

Identifier	UC-1	
Purpose	To allow users to securely log in, log out, and manage their profile.	
Priority	High	
Pre-conditions	User must have an account with a registered email and password.	
Post- conditions	User is logged in and has access to profile management features, or receives an error message if login fails.	
Typical Course of Action		
S#	Actor Action	System Response
1	User navigates to the login page.	System displays the login form.
2	User enters email and password and submits.	System validates credentials.
3	Credentials are correct.	System logs user in and redirects to profile page.
4	User navigates to the profile management page.	System displays profile information and edit options.
4	User updates profile information and submits.	System saves changes and confirms update success.
Alternate Course of Action		
S#	Actor Action	System Response
1	User enters incorrect email or password.	System displays an error message.
2	User attempts to access profile management without logging in.	System redirects to login page.
3	User forgets password and clicks "Forgot Password."	System sends a password reset email.
...		

Table 1: UC-1

3.2 Add to Cart

Purpose		To allow users to add items to their shopping cart for purchase.
Priority		High
Pre-conditions		<ol style="list-style-type: none"> 1. User must be logged into their account. 2. The item to be added must be in stock.
Post- conditions		<ol style="list-style-type: none"> 1. The item is added to the cart. 2. Cart reflects the updated total cost and quantity.
Typical Course of Action		
S#	Actor Action	System Response
1	User selects an item to add to the cart.	System fetches item details.
2	User clicks the "Add to Cart" button.	System checks item availability.
3		System adds the item to the cart and updates the cart details (item count and total price).
4	System confirms the addition of the item.	Display confirmation message: "Item added to cart successfully."
Alternate Course of Action		
S#	Actor Action	System Response
1	User selects an out-of-stock item.	System displays a message: "Item is out of stock."
2	User tries to add an invalid item.	System displays a message: "Invalid item."
3	User adds the last available unit of an item.	System marks the item as "Out of Stock" after addition.

Table 2: UC-2

3.3 Product Customization

Identifier	UC-3	
Purpose	To allow users to customize products (e.g., colors, text, designs) before adding them to their cart.	
Priority	High	
Pre-conditions	<div>1. User must be logged into their account.</div> <div>2. He product must support customization.</div> <div>3. The customization tool should be available and functional.</div>	
Post- conditions	<div>1. Customization is saved and linked to the product in the cart.</div> <div>2. The system displays the customized product for review.</div>	
Typical Course of Action		
S#	Actor Action	System Response
1	User selects a product that supports customization.	System fetches and displays the customization interface.
2	User chooses customization options such as color, text, or design.	System updates the preview to reflect the selected customizations.
3	User confirms the customization.	System saves the customization and calculates the updated price (if applicable).
4	User clicks "Add to Cart" after customization.	System adds the customized product to the cart and displays a confirmation message.
Alternate Course of Action		
S#	Actor Action	System Response
1	User selects a product that does not support customization.	System displays a message: "Customization not available for this product."
2	User inputs invalid customization details (e.g., invalid characters or file formats).	System prompts an error message: "Invalid customization input. Please correct it."
3	User attempts to save customization during a system error.	System displays a message: "Unable to save customization. Please try again later."

Table 3: UC-3

3.4 Save Product Design

Identifier	UC-4	
Purpose	To allow users to save a customized product design for future use or purchase.	
Priority	Medium	
Pre-conditions	1. User must be logged into their account. 2. Customization is completed and ready to save.	
Post- conditions	1. The product design is saved in the user's profile under "Saved Designs".	
Typical Course of Action		
S#	Actor Action	System Response
1	User clicks "Save Design" after completing product customization.	System validates the customization details and displays a message: "Saving design..."
2	User provides a name for the design (optional).	System saves the design with the provided name or assigns a default name if none is provided.
3	Typical: System confirms: "Design saved successfully" and makes it available in the "Saved Designs" section.	Typical Continuation: Saved design appears in the user's account for future access.
Alternate Course of Action		
S#	Actor Action	System Response
1	Alternative: User attempts to save without completing customization.	Alternative Response: System displays an error: "Customization incomplete. Please complete your design before saving."
2	Alternative: User attempts to save during a system error (e.g., connection loss).	Alternative Response: System displays: "Unable to save the design. Please check your connection and try again later."
3	Alternative: User attempts to save a duplicate design name.	Alternative Response: System prompts: "Design name already exists. Please choose a unique name."

Table 4: UC-4

3.5 Adjust Pricing

Identifier	UC-5	
Purpose	To allow administrators to adjust product pricing dynamically based on factors such as demand, discounts, or promotions.	
Priority	High	
Pre-conditions	1. Admin must be logged into the system with the necessary permissions. 2. Product must already exist in the system.	
Post- conditions	1. The updated price is reflected in the system and visible to customers.	
Typical Course of Action		
S#	Actor Action	System Response
1	Admin selects a product to adjust pricing.	System displays the current price and pricing history.
2	Admin inputs a new price or applies a discount percentage.	System validates the new price (e.g., checks against minimum price thresholds).
3	Admin confirms the adjustment.	System updates the product price and logs the change in the pricing history. Customers see the updated price.
Alternate Course of Action		
S#	Actor Action	System Response
1	Admin selects a product to adjust pricing.	System displays the current price and pricing history.
2	Admin inputs a new price or applies a discount percentage.	System validates the new price (e.g., checks against minimum price thresholds).
3	Admin confirms the adjustment.	System updates the product price and logs the change in the pricing history. Customers see the updated price.

Table 5: UC-5

3.6 Track Design Changes

Identifier	UC-6	
Purpose	To allow users to track and review the changes made to a product design during customization.	
Priority	Medium	
Pre-conditions	1. User must be logged into their account. 2. Design changes must be saved in the system.	
Post- conditions	1. User can view and revert to a previous version of the design if required.	
Typical Course of Action		
S#	Actor Action	System Response
1	User accesses the "Track Changes" option for a customized product.	System displays a list of all changes made to the design in chronological order.
2	User selects a specific change to review.	System provides a preview of the design at that specific stage.
3	User reverts to a previous design version (if required).	System confirms the reversion and updates the current design to reflect the selected version.
Alternate Course of Action		
S#	Actor Action	System Response
1	User tries to access changes for a product without saved versions.	System displays: "No design changes available for this product."
2	User attempts to revert during a system error.	System displays: "Unable to process the request. Please try again later."

Table 6: UC-6

3.7 Place Order

Identifier		UC-7
Purpose		To allow users to place an order for customized or standard products.
Priority		High
Pre-conditions		1. User must be logged in. 2. Cart must contain at least one product.
Post- conditions		1. The order is placed, and the user receives an order confirmation.
Typical Course of Action		
S#	Actor Action	System Response
1	User proceeds to checkout.	System displays the order summary and total cost.
2	User enters shipping and payment details.	System validates the details and confirms the payment.
3	User confirms the order.	System processes the order and displays a confirmation message with an order ID.
Alternate Course of Action		
S#	Actor Action	System Response
1	User tries to checkout with an empty cart.	System displays: "Your cart is empty. Please add items before proceeding."
2	User enters invalid payment details.	System prompts: "Payment failed. Please check your details and try again."
3	User cancels the order during checkout.	System discards the order and returns the user to the cart.

Table 7: UC-7

3.8 Track Order

Identifier	UC-8	
Purpose	To allow users to track the status of their placed orders.	
Priority	Medium	
Pre-conditions	1. User must be logged in. 2. User must have at least one active order.	
Post- conditions	1. User can view the current status and estimated delivery time.	
Typical Course of Action		
S#	Actor Action	System Response
1	User accesses the "Track Order" section.	System displays a list of all active orders.
2	User selects a specific order to track.	System displays the current status, location, and estimated delivery time.
Alternate Course of Action		
S#	Actor Action	System Response
1	User tries to track an order that does not exist.	System displays: "No orders found to track."
2	User tries to track during a system outage.	System displays: "Unable to fetch order details. Please try again later."

Table 8: UC-8

3.9 Interact with Seller

Identifier	UC-9	
Purpose	To allow users to communicate with sellers for queries about products or orders.	
Priority	Medium	
Pre-conditions	1. User must be logged in. 2. Seller contact must be enabled for the product.	
Post- conditions	1. Messages are successfully sent and received between the user and the seller.	
Typical Course of Action		
S#	Actor Action	System Response
1	User selects the "Contact Seller" option for a product or order.	System opens a chat or messaging interface.
2	User sends a message to the seller.	System delivers the message to the seller and confirms delivery.
3	User receives a reply from the seller.	System notifies the user of the new message and displays it in the chat interface.
Alternate Course of Action		
S#	Actor Action	System Response
1	User tries to contact a seller for a product without contact support.	System displays: "This seller cannot be contacted directly."
2	User sends a message during a system error.	System displays: "Message failed to send. Please try again later."

Table 9: UC-9

3.10 Product Reviews and Ratings

Identifier	UC-10	
Purpose	To allow users to leave reviews and ratings for purchased products.	
Priority	Medium	
Pre-conditions	1. User must have purchased the product.	
Post- conditions	1. Review and rating are saved and displayed on the product page.	
Typical Course of Action		
S#	Actor Action	System Response
1	User selects a purchased product to review.	System opens the review and rating interface.
2	User writes a review and provides a rating (e.g., 1–5 stars).	System validates the input and displays a confirmation message.
3	User submits the review.	System saves the review and updates the product’s review section.
Alternate Course of Action		
S#	Actor Action	System Response
1	User tries to review a product they have not purchased.	System displays: "Only purchased products can be reviewed."
2	User submits an incomplete review (e.g., no text or rating).	System displays: "Please provide both a rating and a review before submitting."
3	User attempts to submit during a system outage.	System displays: "Unable to submit your review. Please try again later."

Table 10: UC-10

3.11 Analytics and Reporting

Identifier		UC-11
Purpose		To provide administrators with analytics and reports on user behavior, sales, and system performance.
Priority		Medium
Pre-conditions		Admin must be logged in.
Post- conditions		Admin views detailed reports and analytics.
Typical Course of Action		
S#	Actor Action	System Response
1	Admin accesses the analytics section.	System loads available reports and data visualization tools.
2	Admin selects a report to view (e.g., sales, user activity).	System generates and displays the selected report.
3	Admin filters or sorts the report data.	System updates the report view according to the selected filters.
4	Admin exports the report data to a file (e.g., CSV).	System generates and downloads the requested report file.
Alternate Course of Action		
S#	Actor Action	System Response
1	Admin tries to access unavailable report data.	System displays an error message and suggests alternative options.
2	Admin selects an invalid date range for the report.	System prompts the admin to choose a valid date range.
3	Admin’s request for a large dataset fails.	System notifies the admin about performance issues and suggests a smaller dataset.

Table 11: UC-11

3.12 Cancel Order

Identifier	UC-12	
Purpose	To allow users to cancel an order before it has been processed for shipping.	
Priority	High	
Pre-conditions	1. User must be logged into their account. 2. The order must be in a cancellable state (e.g., not yet shipped).	
Post- conditions	1. The order is canceled, and the user receives a confirmation. 2. Refunds (if applicable) are initiated.	
Typical Course of Action		
S#	Actor Action	System Response
S#	Actor Action	System Response
1	User navigates to the "Orders" section.	System displays a list of all active orders.
2	User selects an order they wish to cancel.	System checks the order status to verify if it can be canceled.
3	User confirms the cancellation of the order.	System cancels the order, updates its status to "Canceled," and sends a confirmation notification.
Alternate Course of Action		
S#	Actor Action	System Response
1	User tries to cancel an order that has already been shipped.	System displays: "Order cannot be canceled as it has already been shipped. Please initiate a return after delivery."
2	User attempts to cancel during a system error.	System displays: "Unable to process the cancellation. Please try again later."
3	User decides not to confirm the cancellation after selecting an order.	System retains the order and returns to the order details page without making changes.

Table 12: UC-12

3.13 Edit Profile

Identifier		UC-13
Purpose		To allow users to update their profile information such as name, email, contact details, or password.
Priority		Medium
Pre-conditions		1. User must be logged into their account.
Post- conditions		1. Updated profile information is saved and reflected in the system.
Typical Course of Action		
S#	Actor Action	System Response
S#	Actor Action	System Response
1	User navigates to the "Edit Profile" section.	System displays the user's current profile information in editable fields.
2	User updates one or more fields (e.g., name, email, or password).	System validates the inputs (e.g., valid email format, password strength).
3	User clicks the "Save" button to confirm the changes.	System saves the updated information and displays a success message: "Profile updated successfully."
Alternate Course of Action		
S#	Actor Action	System Response
1	User inputs invalid data (e.g., email without "@" or weak password).	System displays an error message: "Invalid input. Please correct the highlighted fields."
2	User attempts to save changes without completing required fields.	System prompts: "Please fill in all required fields before saving."
3	User tries to update the profile during a system error.	System displays: "Unable to save changes. Please try again later."

Table 13: UC-13

3.14 Remove From Cart

Identifier	UC-14	
Purpose	To allow users to remove items from their shopping cart before proceeding to checkout.	
Priority	High	
Pre-conditions	1. User must be logged into their account. 2. The shopping cart must contain at least one item.	
Post- conditions	1. The selected item is removed from the cart. 2. The cart's total cost and item count are updated accordingly.	
Typical Course of Action		
S#	Actor Action	System Response
S#	Actor Action	System Response
1	User navigates to the shopping cart page.	System displays a list of all items in the cart.
2	User selects an item to remove.	System fetches the details of the selected item.
3	User clicks the "Remove" button next to the item.	System removes the item from the cart.
Alternate Course of Action		
S#	Actor Action	System Response
1	User attempts to remove an item from an empty cart.	System displays an error: "Your cart is empty."
2	User removes the last item in the cart.	System displays a message: "Cart is now empty."
3	User attempts to remove an item during a system error.	System displays: "Unable to process request. Try again later."

Table 14: UC-14

3.15 Create Profile

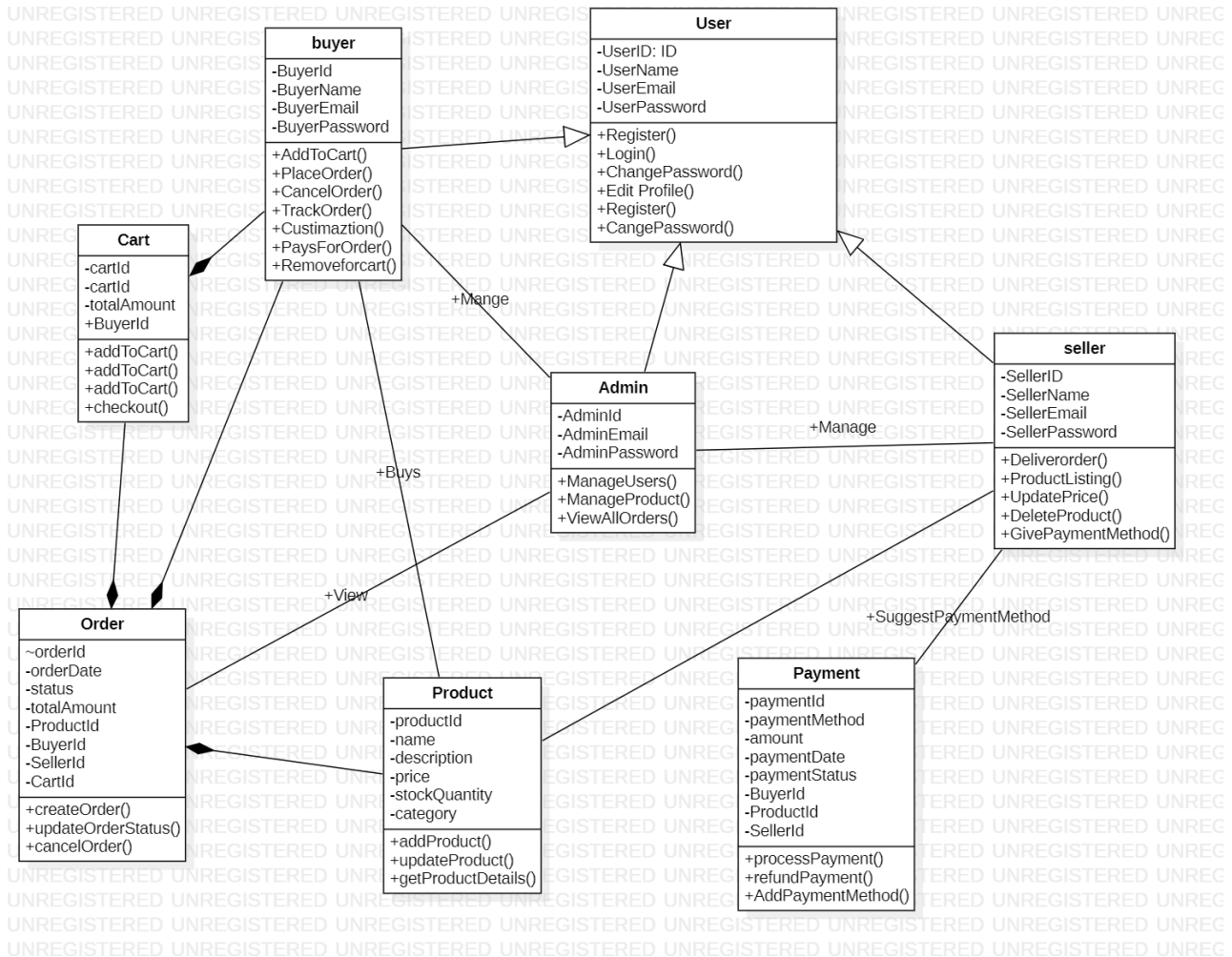
Identifier	UC-15	
Purpose	To allow new users to create a profile by registering on the platform.	
Priority	High	
Pre-conditions	1. User must not already have an account on the platform.	
Post- conditions	1. A new user profile is created, and the user is logged in.	
Typical Course of Action		
S#	Actor Action	System Response
S#	Actor Action	System Response
1	User selects the "Sign Up" option on the platform.	System displays the registration form with required fields (e.g., name, email, password).
2	User fills in the required information and submits the form.	System validates the information (e.g., checks email format and password strength).
3	User successfully completes any additional verification steps (e.g., email or phone OTP verification).	System creates the profile and displays a success message: "Profile created successfully."
4	User is automatically logged in or prompted to log in manually.	System redirects the user to the home/dashboard page.
Alternate Course of Action		
S#	Actor Action	System Response
1	User submits the form with invalid data (e.g., weak password, invalid email).	System displays an error message: "Invalid input. Please correct the highlighted fields."
2	User tries to register with an email already associated with an existing account.	System displays: "This email is already in use. Please log in or use a different email."
3	User exits the process before completing registration.	System discards any unsaved data and returns to the main page.

Table 15: UC-15

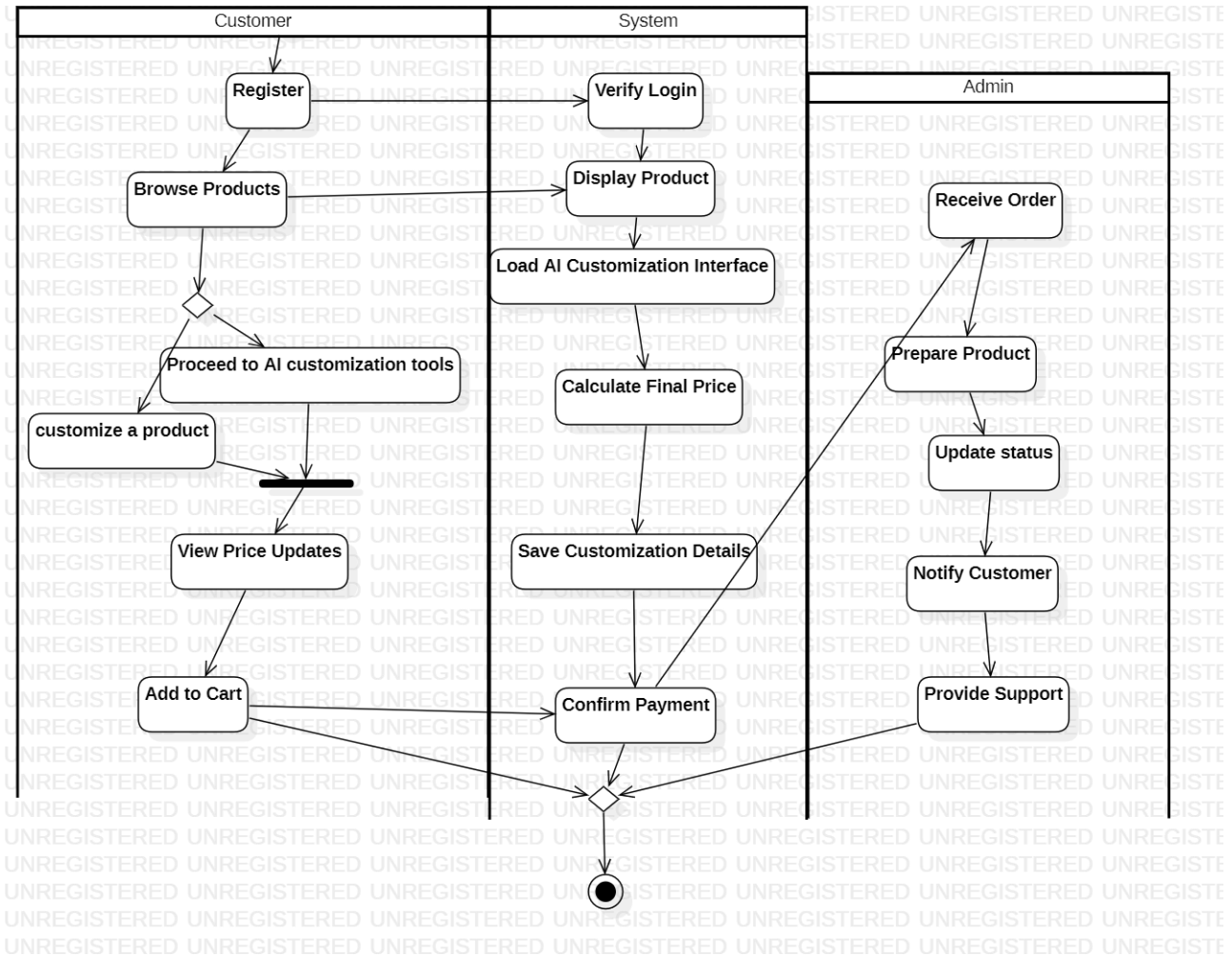
3.16 Send Refund

Identifier	UC-16	
Purpose	To allow administrators or the system to issue refunds to users for canceled orders, returned products, or other valid reasons.	
Priority	High	
Pre-conditions	1. The order or product must be eligible for a refund. 2. The user's payment details must be valid and available in the system.	
Post- conditions	1. The refund is successfully issued to the user's payment method. 2. A confirmation notification is sent to the user.	
Typical Course of Action		
S#	Actor Action	System Response
S#	Actor Action	System Response
1	Admin/User navigates to the refund section.	System displays a list of eligible orders for refunds.
2	Admin/User selects an order or item to refund.	System fetches the order details and calculates the refund amount.
3	Admin/User clicks the "Send Refund" button.	System validates the eligibility and payment details.
4	Admin/User confirms the refund action.	System processes the refund and updates the order status to "Refunded."
Alternate Course of Action		
S#	Actor Action	System Response
1	Admin/User tries to refund an ineligible order.	System displays an error: "Refund not applicable for this order."
2	Admin/User enters invalid refund details.	System displays: "Invalid refund details. Please verify and try again."
3	Admin/User attempts to send a refund during a system error.	System displays: "Refund could not be processed. Please try again later."

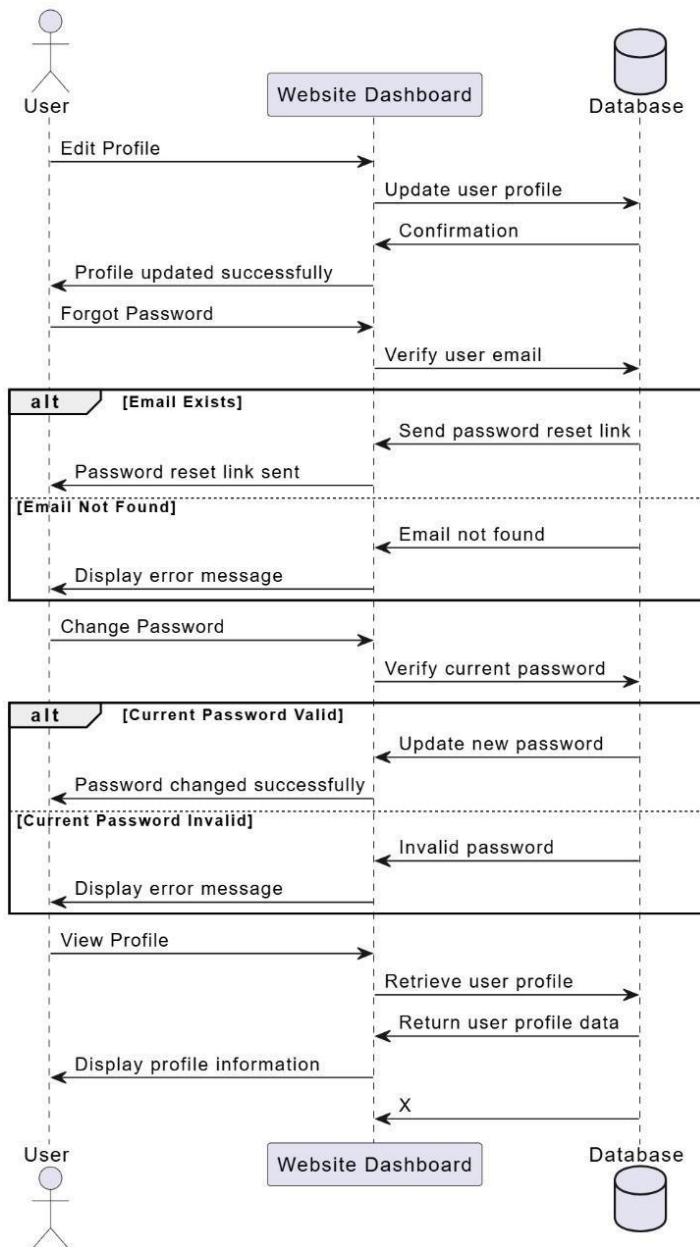
Table 16: UC-16

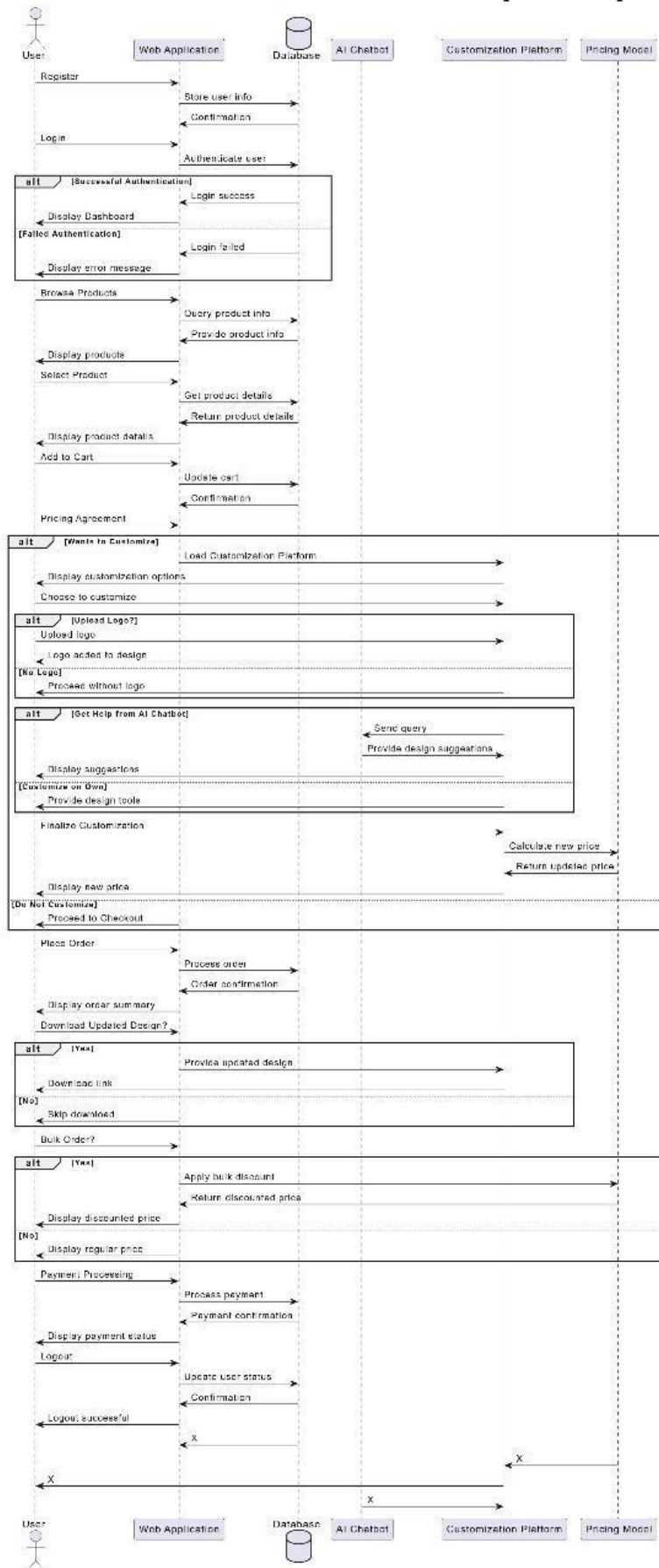
Class Diagram:

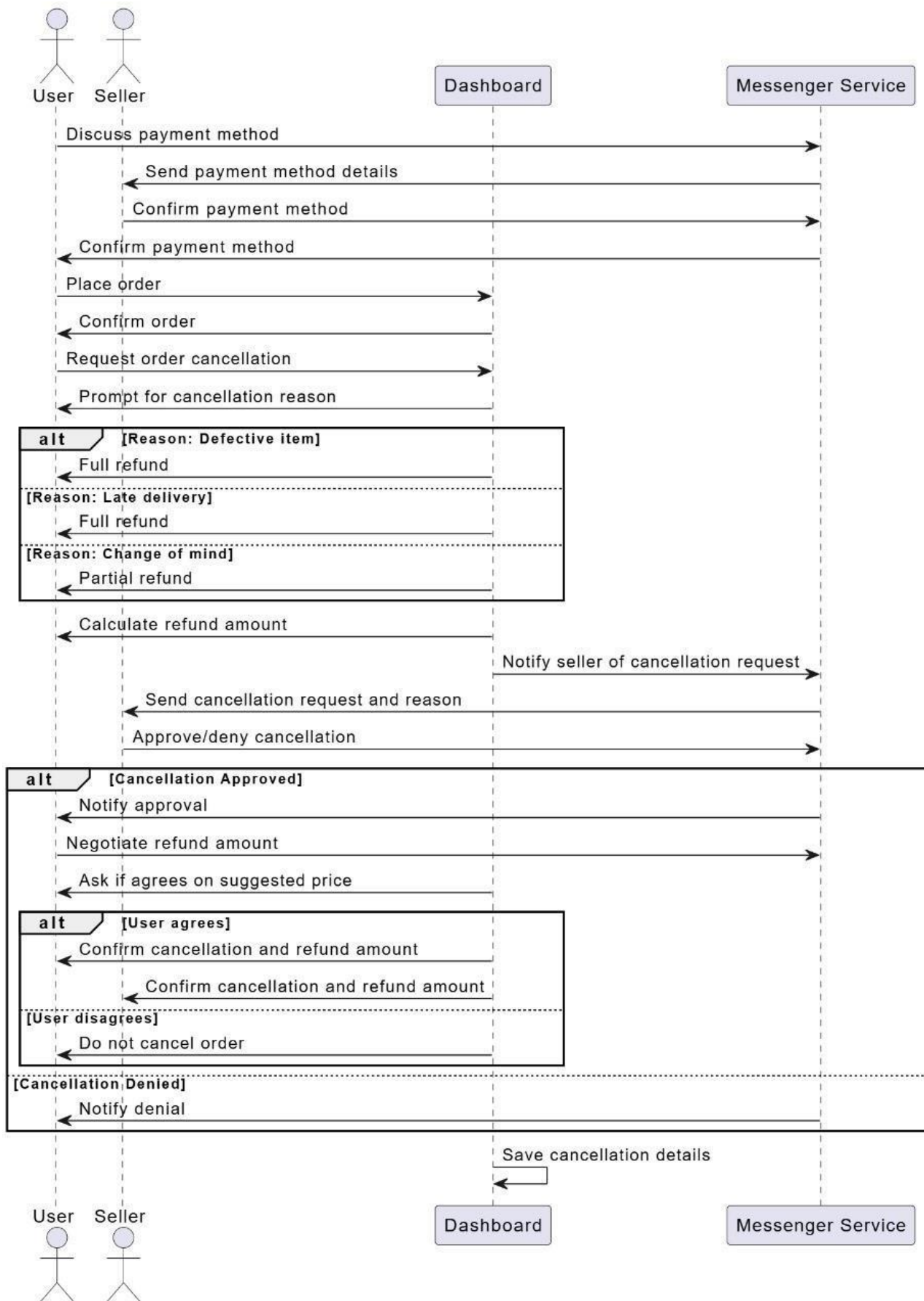
Activity Diagram:



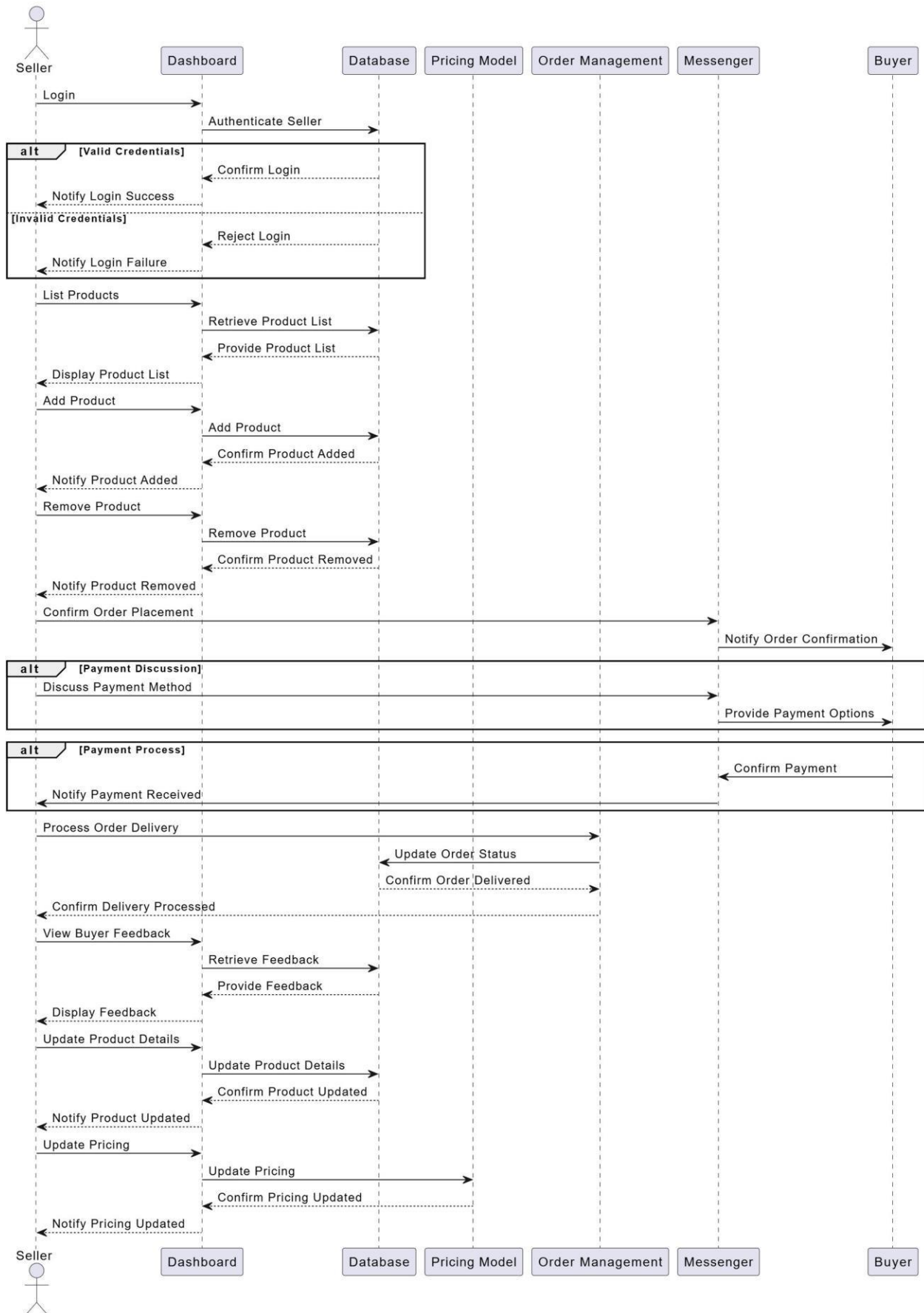
Sequence Diagram:

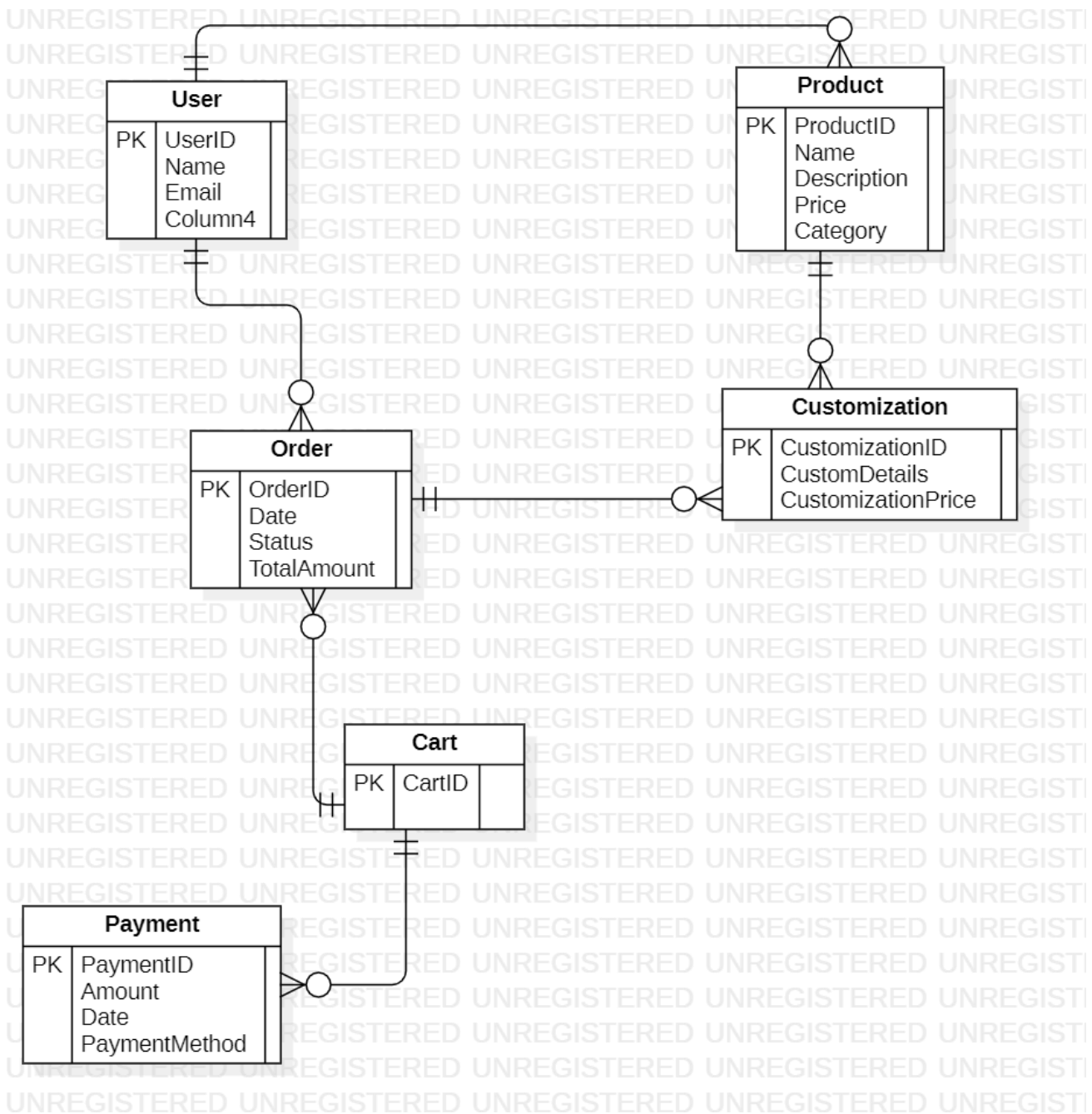






SportsWearXpress



Entity-relationship diagram:

4. Nonfunctional Requirements

4.1 Performance Requirements

Customization Interface Performance:

- **Requirement:** Provide a smooth and responsive experience.
- **Rationale:** Ensures users can design efficiently without delays.

AI-Powered Design Tools:

- **Requirement:** Provide suggestions within 5 seconds.
- **Rationale:** Enhances user satisfaction with real-time guidance.

AI-Powered Design Tools:

- **Requirement:** Deliver quick and helpful suggestions.
- **Rationale:** Enhances creativity and keeps users engaged.

Order Confirmation and Tracking

- **Requirement:** Process orders and provide timely tracking updates.
- **Rationale:** Builds trust and ensures a positive customer experience.

4.2 Safety Requirements Data

Protection and Privacy:

Implement industry-standard encryption (e.g., AES-256) for data storage and TLS for data transfers. Perform regular security audits to minimize risks of unauthorized access.

User Authentication:

Allow users to log in with secure and reliable methods to protect their accounts from misuse. Provide mechanisms to verify user identity effectively to avoid unauthorized logins.

4.3 Security Requirements

Data Protection and Privacy: Keep all user data secure during storage and while being transferred between systems. Protect sensitive information from being accessed by unauthorized individuals.

User Authentication: Use secure processes to confirm user identity and safeguard accounts.

Ensure only authorized users can access their information and perform actions.

Compliance and Certification: Follow national standards for data security and conduct regular checks to identify and fix potential security issues. Provide users with a trustworthy environment where their data and activities remain safe.

4.4 Additional Software Quality Attributes

Usability: The interface should be intuitive and user-friendly, allowing users to complete their tasks efficiently without extensive training or support.

Reliability: The platform should be easy to use, with clear instructions and a simple interface for all users.

Maintainability: Make the system easy to update and fix so it can be improved and expanded over time.

Interoperability: Ensure the system works well with other tools and services, enabling smooth operations.

Scalability: The architecture should support scaling up to accommodate increasing numbers of users and transactions without compromising performance.

5. Other Requirements

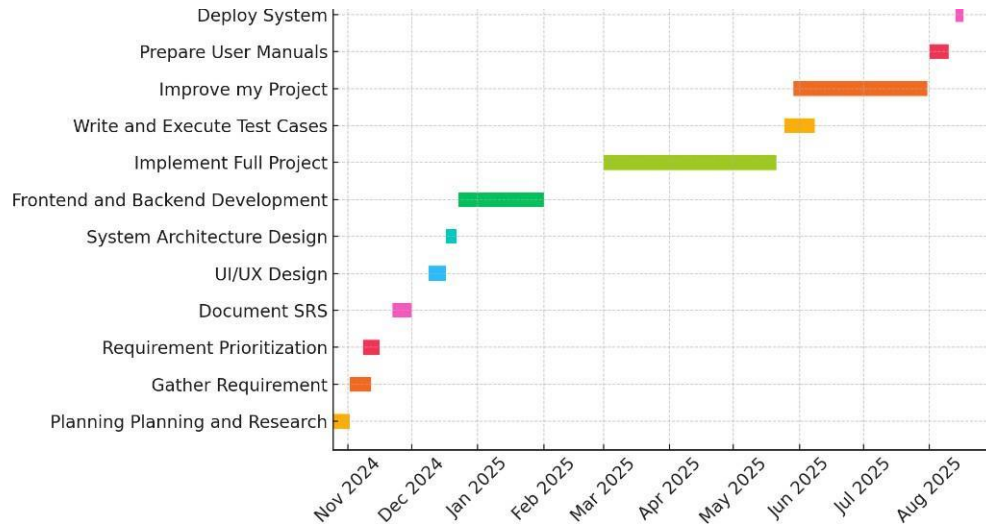
Database Requirements: The system must use a relational database (MongoDB) to store user data, product details, and order information. Data should be regularly backed up to prevent loss.

External Interface Requirements: Include features for direct communication between buyers and sellers, like a messaging system.

Internationalization Requirements: Support the English language and PKR currency to meet the needs of Pakistani users. Adapt date, time, and number formats to the preferences of local users.

Reuse Objectives: The codebase should be modular and reusable to facilitate future projects. Components and libraries should be designed for easy integration and adaptation

6. Revised Project Plan



7. References

[1] Zazzle, “Zazzle - Customize Anything,” [Online]. Available: <https://www.zazzle.com/>. [Accessed: 26-Nov-2024].

[2] Custom Ink, “Custom Ink - Custom T-shirts and More,” [Online]. Available: <https://www.customink.com/>. [Accessed: 26-Nov-2024].

Appendix A: Glossary

Appendix B: IV & V Report

(Independent verification & validation) IV & V
Resource

Name

Signature

S#	Defect Description	Origin Stage	Status	Fix Time	
				Hours	Minutes
1					
2					
3					
...					

Table 2: List of non-trivial defects