Faculty of Engineering
Computer Systems Engineering
CSE452: SOFTWARE ENGINEERING
Instructor: Dr. Ahmed Ayoub
Teaching Assistant: Mai Hassanin ElBaradei

E-commerce Application

Prepared by

Bishoy Magdy
Mahmoud Bahaaeldeen

221521 – 210733

4/12/2023

Table of Contents

1. Project Plan

1.1 Project Objectives

Develop and implement a user-friendly e-commerce application accessible via mobile application or all devices with access to a web browser. Enhance the efficiency of searching for a specific product using various search and filtering methods. Offer multiple payment methods and gateways which enhances the overall customer experience. Ensure a robust and secure system that protects sensitive user data and information.

1.2 Project Scope

The scope of the project includes the development of an e-commerce platform for buyers and sellers, along with a backend system for managing customer orders, seller inventory management, delivery management, and payment management. The e-commerce platform will operate in Egypt and specifically within large cities such as Cairo & Alexandira.

1.3 Stakeholders' Profiles

1- Business owner

2- Sellers

3- Customers

4- Marketing team

5- Technology team

6- Investors

1.4 Stakeholders' Requirements

1-Business Owner:

Needs: Profitability, long-term growing, effective tools to manage his business.

Expectations: sales report that will help them keep track of their company progress, easy website management.

Preferences: Low running costs and a dependable e-commerce program

2-Customers:

Needs: Product variety, secure transactions, user-friendly interface.

Expectations: the program works in different devices like Mobil, computer, accurate product descriptions, real feedback, and reviews.

Preferences: customized recommendations according to their activities on the program, different ways of payment.

3-Sellers:

Needs: sales opportunities, and a reliable platform.

Expectations: Simple product listing, fast payment processing, and customer assistance.

Preferences: different shipping options, and low transaction fees.

4-Marketing Team:

Needs: Reach statistics, engagement.

Expectations: Data analytics, marketing tools to reach the target audience

Preferences :automated email marketing,

5-Technoalogy Team:

Needs: Reliability, security, and system's potential to grow.

Expectations: continuous updates, data protection.

Preferences: API integration, cloud-based solutions, and developer-friendly tools

6-Investors:

Needs: Potential for growth, and risk management.

Expectations: Financial transparency, growth strategies, and performance m

Preferences: Regular reports, strong market study and a strong management team.

1.5 Software development model

The **Incremental** software development model has been chosen for the development of our application. This approach involves breaking down the project into small increments, allowing for the delivery of **functional** components in phases. With each increment, new features are integrated, and regular feedback from stakeholders is obtained, facilitating adjustments and improvements.

2. Project Use Cases

2.1 Use Case Scenarios

Use case 1: Secure User Registration

1. Initial State: A new user opens the application and first chooses between buyer or seller registration.
2. Normal Flow: The user fills in the required information such as e-mail address, phone number, full name, address, and date of birth. If they are a seller, they are asked for extra information such as their tax number and company ID and set up a password. The system saves the user's information, and the user is redirected to the main page.

3. Potential Issues and Handling: if the user enters incomplete or inaccurate information, they are prompted with an error message based on the issue.
4. Other Activities: Other users can access the application while new users are completing their registration.
5. System State on Completion: The user is successfully registered and is now able to access the platform's features such as adding to cart, wish listing, and checking out or adding to inventory, checking sales and confirm orders for buyers or sellers respectively. **Sellers can use all features buyers have.`**

Use case 2: Checking out.

1. Initial State: The user has items in their shopping cart and is ready to complete a purchase and checkout.
2. Normal Flow: The user selects the shopping cart, reviews the items, and proceeds to checkout. They provide shipping and payment information such as their address, credit card information, gift card information, etc. The system verifies the payment details, processes the transaction, and sends a confirmation receipt to the user.
3. Potential Issues and Handling: If the user's payment information is invalid or the transaction cannot be processed, they are alerted with an error message, and they have the option to correct their details or try another payment method.
4. Other Activities: While the user is checking out, other users can continue browsing and making purchases.

5. System State on Completion: The user's order is successfully placed, and the items are removed from the shopping cart. The user can track the order status in their order history.

Use case 3: Adding a new item to inventory.

1. Initial State: A seller is logged into their account and wants to add a new product to their inventory.
2. Normal Flow: The seller fills in details about the product, such as name, description, price, quantity, and uploads images. They then submit the information. The system validates the input and adds the item to the seller's inventory.
3. Potential Issues and Handling: If the seller misses required information or there is an issue with the product details, they receive an error message prompting them to complete the missing fields or correct the errors.
4. Other Activities: While a seller is adding a new item to their inventory, other users can continue shopping and making purchases.
5. System State on Completion: The new item is successfully added to the seller's inventory, and it is now available for buyers to view and purchase.

Use case 4: Secure payment.

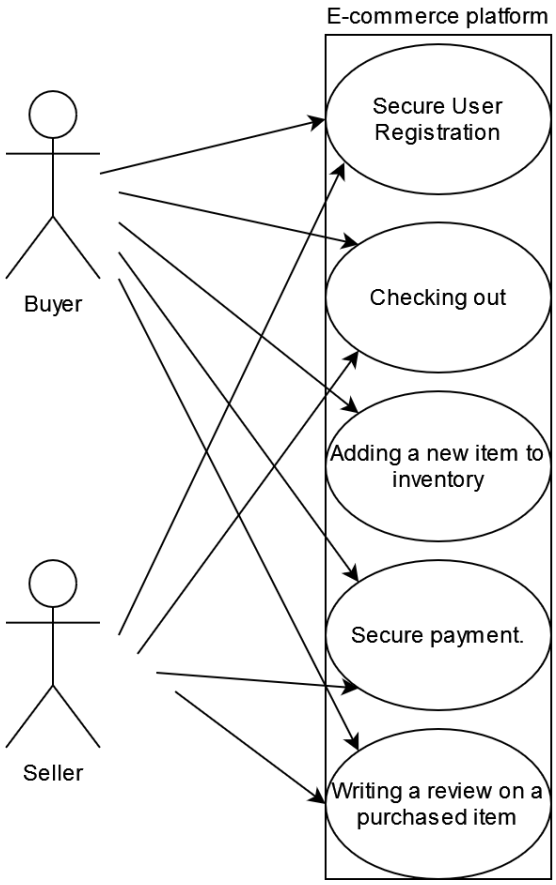
1. Initial State: The user has selected items to purchase and is at the checkout stage.
2. Normal Flow: The user enters their payment information, which includes credit card details or other accepted payment methods. The system securely processes the payment and sends an order confirmation.

3. Potential Issues and Handling: If there is an issue with the payment information, such as an expired card or insufficient funds, the user is notified, and they have the option to update the payment method or contact customer support for assistance.
4. Other Activities: While the user is making a secure payment, other users can continue using the application.
5. System State on Completion: The payment is successfully processed, and the user receives an order confirmation. The items are prepared for shipping or made available for digital download, depending on the nature of the purchase.

Use case 5: Writing a review on a purchased item.

1. Initial State: The user has purchased an item and wants to write a review for it after they have received it.
2. Normal Flow: The user navigates to the item they purchased, selects the "Write a Review" option, and provides their review and a rating. They submitted the review. The system records the review and displays it on the item's page.
3. Potential Issues and Handling: If the review submission encounters an error, such as inappropriate content, the user is notified, and they have the option to revise their review.
4. Other Activities: While a user is writing a review, other users can browse and make purchases, and other reviews can be read on the platform.
5. System State on Completion: The user's review is successfully submitted and displayed on the item's page, providing feedback to other users.

2.2 Use Case Diagram



3. User Requirements

1- Friendly User Interface: When browsing and making purchases on a website or app, users must interact with a friendly user interface.

2- Product Search and Filtering: search or filter the products to reach the desired product quickly.

3- Details of the Product :product reviews, pictures, and descriptions to help user decide what to buy.

4- Shopping Cart: Features to save items for future purchase and manage the shopping cart easily.

5- Easy Checkout Procedure: a simple, safe checkout procedure with a range of payment choices.

6- Discounts and Promotions :continuous notify the user about sales, discounts, and exclusive deals.

7- Order Tracking: Real-time tracking of orders, and delivery time.

8- User opinions and recommendations: users can offer suggestions and feedback.

9- Reviews and Ratings: User can write their reviews and ratings to build trust and help decision-making.

10- Customer profile: should contain User info such as name, address, phone number and order history, and saved preferences for personalized experience.

11- Security and Privacy: user info and financial info should be private and secure.

12- Customer Service: Availability of helpful customer support for questions, help, and problem solving.

13- Seller can Set product prices and apply discounts. Run promotional campaigns with coupon codes.

14- Sellers can trace their sales and access the sales history of the products.

4. System Requirements

1- To store product and customer data. The development team must use a reliable database system such as MySQL.

2- Hardware: development team must use a sufficient server and hardware that can handle the user's traffic and support scalability for growing in the future

3- The system shall provide user registration and allow users to enter their email address, phone, address and keep their data private and safe.

4- The system shall provide a filter and sorting features according to price , category, and other details for better experience.

5- The system shall provide a user-friendly shopping cart to add and remove items and also continuously calculate the total payment number.

6- The system shall provide an easy and secure check out and support various ways for payment such as credit and cash payment.

7- System shall provide order management. Tracking of customers order and support order verification by sending an email to user to confirm his order.

8- System shall provide a shipping feature and provide the delivery date.

9- The system shall provide a feature that help users to write their feedback and reviews for the product they bought .this feature helps in decision making.

10- System shall provide Return and Refund Management that help users to return and refund their money if they want.

11- The system shall provide a feature for sellers to trace their sales and make their discounts on their products.

4.1 Functional Requirements

1- User Registration

Description and Priority: Users can register for an account on the system with this functionality. Given that it is necessary for user engagement, it has top priority.

Stimulus/Response Sequences: User enters registration information (password, email address, and name).The system verifies the data input. The system generates a user account and sends a confirmation email if the input is valid. Invalid input causes the system to display the relevant error messages.

Functional Requirements: To ensure system security, it should securely collect and store user information, including username, email address, and password. Passwords should be hashed for added security. The system should validate email addresses for correct formatting, ensure each email is linked to a single account, and enforce minimum password requirements. Additionally, email confirmation should be sent to the user's provided address. Effective error handling is important, displaying clear error messages and assisting users in resolving issues. Lastly, registration actions should be logged for auditing purposes.

2-Product Catalog

Description and Priority: A product catalog is provided by this functionality. Product display and ease of user browsing are top priorities.

Stimulus/Response Sequences: User visits the product catalog as a stimulus. Reaction: The system shows a detailed product list. Users employs filters or search as a stimulus. Reaction: Based on user parameters, the system modifies the product list that is visible.

Functional Requirements: A product database containing information such as name, price, description, and availability should be kept up to date by the system. It is best to categorize products to make navigation easier. Products can be sorted, filtered, and searched for using a variety of criteria.

3-Shopping Cart and Checkout

Description and Priority: Users are able to add products to a cart and finish transactions with this functionality. It is very important to the shopping process.

Stimulus/Response Sequences: The user adds things to the cart as a stimulus. Reaction: The system adds the chosen products to the cart. Stimulus: The customer checks out. Reaction: The order summary is shown on the checkout page by the system.

Functional Requirements: Items in a user's shopping cart should be able to be updated, removed, or added. The entire order cost, including taxes and delivery charges, should be computed by the system. It should be possible for users to finish the transaction through a safe checkout process.

4- Order Management

Description and Priority: Users and administrators can manage orders with this capability. For post-purchase operations, it has a medium importance.

Stimulus/Response Sequences: Stimulus: The order history is accessed by the user or administrator. Reaction: The system shows a detailed summary of previous orders. Stimulus: The order status is updated by the administrator. Reaction: The order status is updated by the system appropriately.

Functional Requirements: Order details and history should be accessible to administrators and users alike. Users should receive emails from the system with order confirmations. Order management, order status updates, and refund processing ought to be under the purview of administrators.

5- Payment Processing

Description and Priority: Payment transactions are handled securely using this feature. When it comes to finishing purchases, it is really important.

S**timulus/Response Sequences**: Stimulus: The customer pays at the point of sale. The payment choices are provided by the system. The user chooses a payment option and enters payment information as the stimulus. Reaction: The payment is processed safely by the system, and a confirmation is shown.

Functional Requirements: PayPal, credit cards, and other payment methods should all be safely processed by the system.

Payment processing needs to adhere to industry security standards.

Payment confirmations should be sent to users after successful transactions.

6-Payment and Financial Reporting:

Description and Priority: This function manages sales payments and gives vendors financial reporting. Encouraging transparent financial management is crucial.

Stimulus/Response Sequences: Seller looks at their financial dashboard as a stimulus.

Reaction: Sales, earnings, and payment history are shown on the system. The seller asks for payment as a stimulus. In response, the payout process is started by the system.

Functional Requirements: A financial dashboard that displays sales, earnings, and transaction history should be available to sellers. Sellers should be able to request payouts using a secure mechanism on the system. Periodic financial summaries and reports that are required for sellers.

4.2 Non-functional Requirements

4.2.1 Performance Requirements

- The platform should be designed such that any page such as the main page, item page, or checkout page should load in no more than 4 seconds under general internet connection conditions.
- The platform should be able to handle a minimum of 15,000 simultaneous users during peak traffic periods.

4.2.2 Safety Requirements

- The system should be able to detect products that do not comply with safety regulations and protect users from purchasing such products by not allowing them to be listed.

4.2.3 Security Requirements

- The system should have robust intrusion detection and prevention mechanisms that prevent any kind of data breach which insures users' data.
- The platform should support two-factor authentication and biometric authentication to further solidify its security.

4.2.4 Software Quality Attributes

- The platform should have a user-friendly interface with a clear and intuitive design to help users navigate easily and smoothly.
- The platform should be maintainable with commented and documented code which ensures development and code scalability.

5. System Design

5.1 Design Goals

- Mobile Optimization:
    - Most of the users of the application will be mobile users. Therefore, it is important to optimize a user-friendly mobile interface considering the large demographic.
- Ease of Use:
    - The application must be designed with simplicity in mind, focusing on intuitive navigation and minimal learning curve. Ensuring that any user would be able to navigate easily through the application.
- Speed and Responsiveness:
    - The application must be optimized for speed and responsiveness. This is to keep users from leaving the application due to slowness or loading issues. This would be done by sacrificing capital due to running high quality servers in all regions the application runs in.

5.2 Proposed Software Architecture

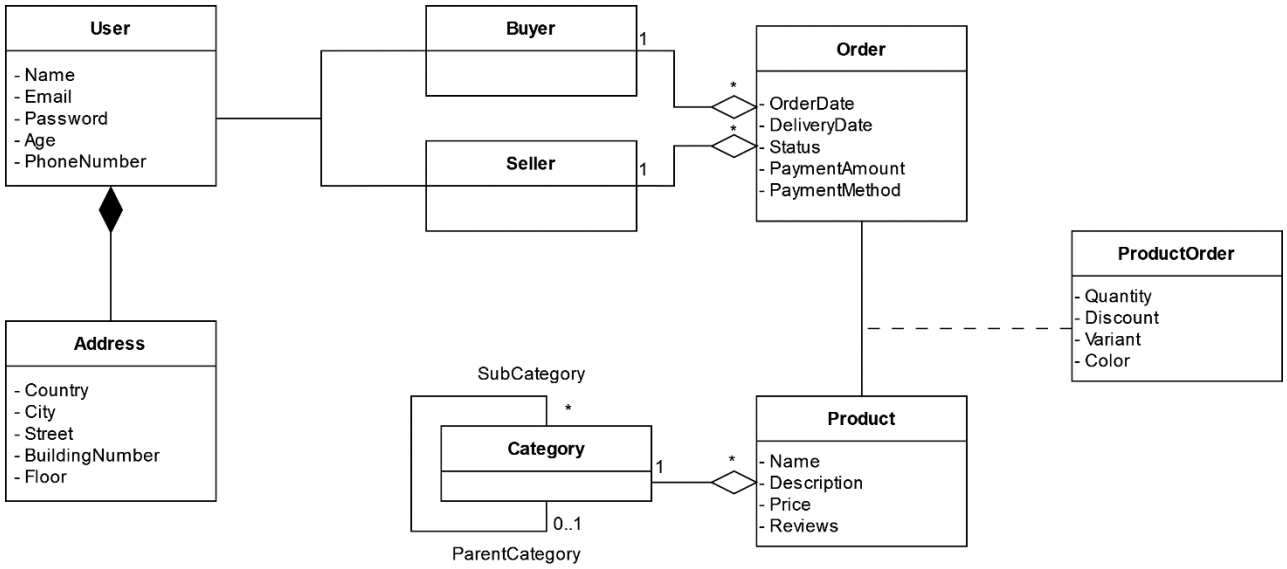
5.2.1 Overview

The **layered architecture** is the most suitable architecture as it divides the e-commerce application into distinct layers—presentation, business logic, and data storage. Each layer has specific responsibilities, contributing to the overall functionality.

- How it Works:
    - **Presentation Layer**: Focuses on delivering a user-friendly interface with an emphasis on simplicity and straightforward access.
    - **Business Logic Layer**: Manages core functionalities like product searches and order processing, payment processing, order tracking optimizing for ease of use and responsiveness.
    - **Data Storage Layer**: Handles data persistence, ensuring speed and responsiveness using high-quality servers in different regions.
- Why Layered Architecture:
    - **Modularity**: Facilitates easy maintenance and updates to specific components without disrupting the entire system, aligning with the ease-of-use goal.
    - **Scalability**: allows layers to scale independently, adapting to variations in user load without impacting the program.

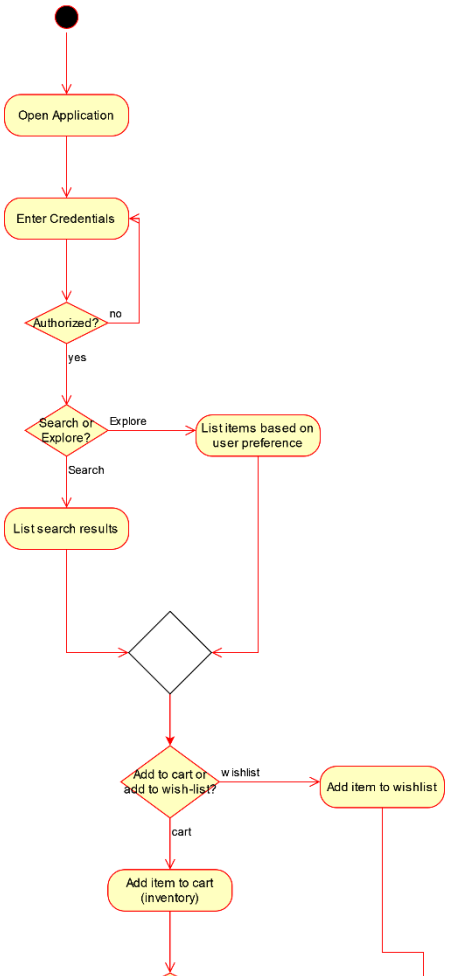
- **Separation of Concerns**: Promotes a clear division of responsibilities, preventing changes in one layer from directly impacting others, aligning well with the design goals.

5.2.2 Class Diagram

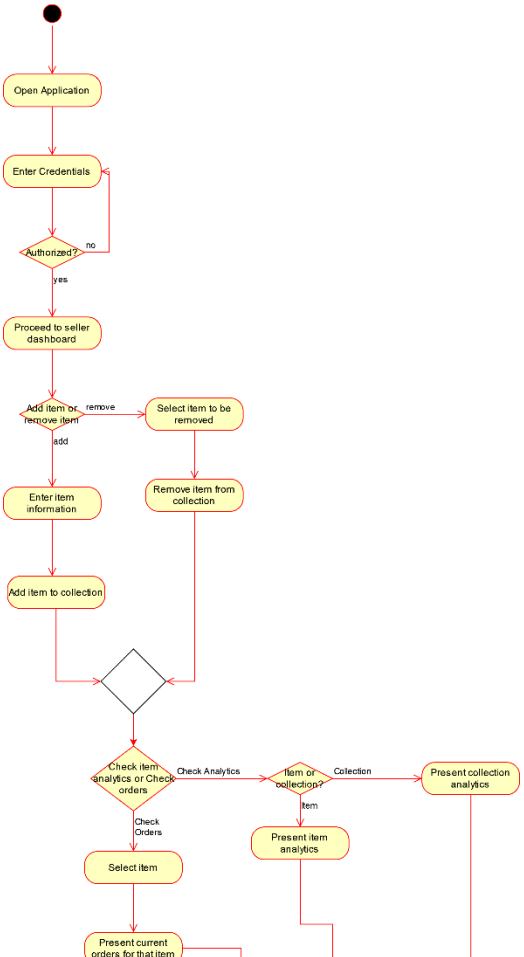


5.2.3 Activity Diagrams

Buyer's Activity Diagram

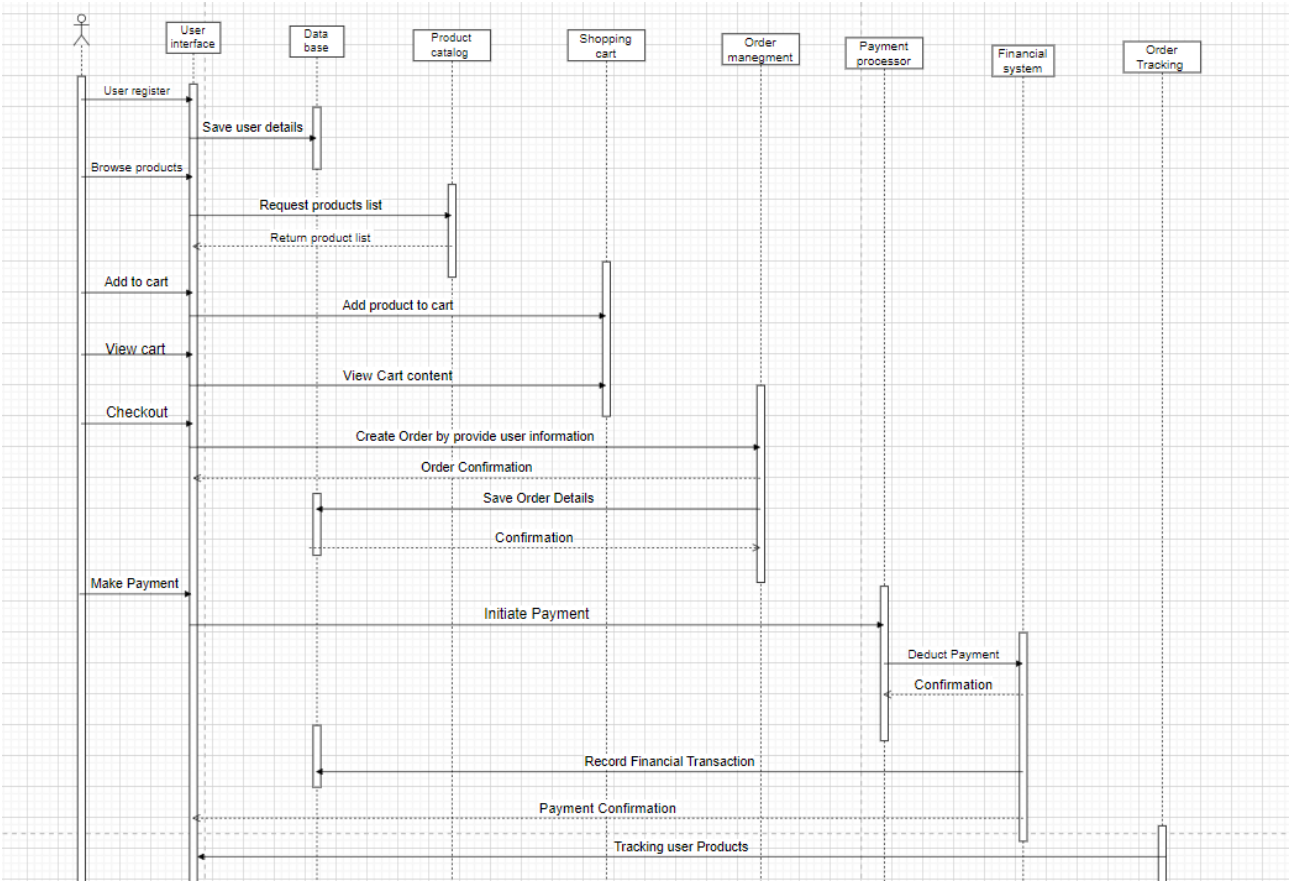


Seller's Activity Diagram

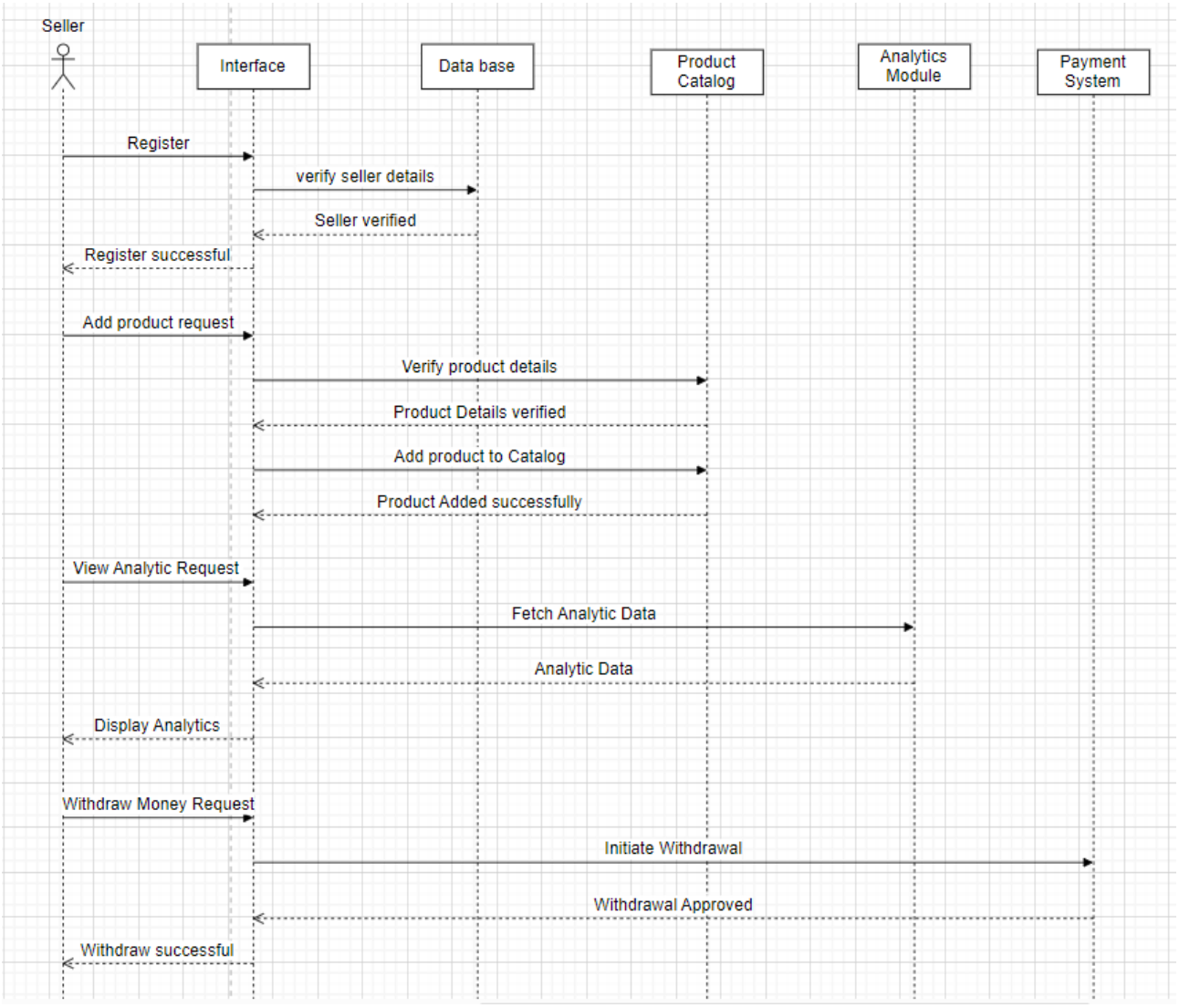


5.2.4 Sequence Diagrams

Buyer's Sequence Diagram



Seller's Sequence Diagram



Test Scenarios

User Registration:

Positive Test Cases:

- Valid Registration:
    - Verify that users can successfully create accounts using valid credentials.
- Confirmation Email Sent:
    - Confirm that a confirmation email is dispatched after successful registration.

Negative Test Cases:

- Registration with Duplicate Email:
    - Attempt to register with an already registered email and verify appropriate error message.
- Registration with Weak Password:
    - Try to register with a password that doesn't meet security criteria and verify error message.

Browse Products:

Positive Test Cases:

- Product Display Accuracy:
    - Ensure accurate display of product images and details.
- Filter and Search Functionality:
    - Verify that filters and search return the correct results.
- Product Sorting Feature:
    - Confirm that users can sort products effectively by various criteria.

Negative Test Cases:

- Search for Non-Existent Product:
    - Search for a non-existent product and expect an appropriate message.
- Conflicting Filters Results:
    - Apply filters that should yield no results and check for an informative message.

Shopping Cart:

Positive Test Cases:

- Adding Products to Cart:
    - Verify that products can be added to the shopping cart.
- Cart Quantity Update:
    - Confirm that product quantities in the cart can be changed.
- Cart Total Price Update:
    - Check that the total price updates when cart items change.

Negative Test Cases:

- Adding Out-of-Stock Items to Cart:
    - Attempt to add more items to the cart than available in stock and expect an error message.
- Checkout with Empty Cart:
    - Try to proceed to checkout with an empty cart and verify that a prompt is displayed.

Payment Process:

Positive Test Cases:

- Multiple Payment Options:
    - Verify that multiple payment options are available and can be selected.

- Successful Transaction with Valid Payment:
    - Confirm successful transaction with valid payment details.
- Payment Confirmation Message:
    - Ensure a confirmation message with details appears post-payment.

Negative Test Cases:

- Invalid Payment Details:
    - Enter invalid payment details and expect to see an error message.

Order Tracking:

Positive Test Cases:

- Valid Order Tracking:
    - Ensure entering a valid order number yields correct tracking information.
- Accurate Shipment Status:
    - Confirm that the shipment status is accurate and updated.

Negative Test Cases:

- Invalid Order Number:
    - Enter an invalid order number and verify that an error message is displayed.
- Tracking Canceled Order:
    - Attempt to track a canceled order and expect a message indicating no tracking info is available.