**Software Requirements and Design Document**

**for**

Solo Store

**Prepared by Muhammad Abbas, Shaffin Imam, Furqan Jahangir**

**NUCES FAST Islamabad**

**Date: 27-11-2024**

**Table of Contents**

**Table of Contents ii**

**1.** **Introduction 1**

1.1 Purpose 1

1.2 Product Scope 1

1.3 Title 1

1.4 Objectives 1

1.5 Problem Statement 1

**2.** **Overall Description 1**

2.1 Product Perspective 1

2.2 Product Functions 2

2.3 List of Use Cases 2

2.4 Extended Use Cases 2

2.5 Use Case Diagram 2

**3.** **Other Nonfunctional Requirements 2**

3.1 Performance Requirements 2

3.2 Safety Requirements 2

3.3 Security Requirements 2

3.4 Software Quality Attributes 2

3.5 Business Rules 3

3.6 Operating Environment 3

3.7 User Interfaces 3

**4.** **Domain Model 3**

**5.** **System Sequence Diagram 3**

**6.** **Sequence Diagram 3**

**7.** **Class Diagram 4**

**8.** **Package Diagram 4**

**9.** **Deployment Diagram 4**

# **Introduction**

## **Purpose**

*The purpose of SoloStore is to develop a* ***user-friendly e-commerce application*** *that allows users to browse, search, and purchase products online seamlessly. It eliminates the need for traditional shopping by providing an organized platform where:*

* *Customers can search for products, add them to their cart, and place orders.*
* *Store owners can manage their store inventory, update product details, and handle orders efficiently.*
* *Administrators can oversee the system and manage user roles, ensuring a secure and organized experience.*

*The project aims to cater to the needs of both* ***customers and store owners****, providing convenience, accessibility, and efficiency through modern software technology.*

## **Product Scope**

*SoloStore is an e-commerce application enabling users to browse, search, purchase products, and leave reviews. Store owners can manage their products, and administrators oversee the system. The system aims to enhance the online shopping experience and streamline store management.*

## **Title**

*SoloStore*

## **Objectives**

*Provide an intuitive interface for customers to browse and purchase products.*

*Enable store owners to manage inventory and analyze sales data.*

*Ensure secure transactions and user data protection.*

*Improve customer satisfaction with review functionality.*

## **Problem Statement**

*Traditional shopping methods are time-consuming and lack convenience. SoloStore addresses these issues by offering a digital marketplace that connects customers and store owners, reducing manual efforts and improving accessibility.*

# **Overall Description**

## **Product Perspective**

*SoloStore is a* ***standalone application*** *developed as a self-contained product but with potential for future integration into larger e-commerce ecosystems.*

* ***Context****: The product uses* ***JavaFX*** *for the user interface and* ***PostgreSQL*** *as the database to ensure high performance and scalability.*
* ***Relation to Existing Systems****: It replaces manual inventory tracking and traditional shopping methods with a streamlined digital system.*
* ***System Components****:*
  + ***Client Application****: The JavaFX front-end, where customers and store owners interact with the system.*
  + ***Database****: PostgreSQL for managing users, products, orders, and reviews.*
  + ***Business Logic****: Features like product management, reviews, and search functionality ensure smooth operations..*

## **Product Functions**

*The major functions of SoloStore include:*

1. ***Product Management****:*
   * *Store owners can add, update, and delete products.*
   * *Customers can browse products based on categories or search keywords.*
2. ***Cart and Checkout****:*
   * *Customers can add products to their cart and proceed with checkout seamlessly.*
   * *Real-time price calculations ensure an accurate checkout experience.*
3. ***Search and Filtering****:*
   * *Customers can search for products by name or description.*
   * *Filtering options (e.g., price range or category) can be implemented to refine results.*
4. ***Order Management****:*
   * *Customers can place and track orders.*
   * *Store owners can manage order statuses (e.g., pending, shipped).*
5. ***Review System****:*
   * *Customers can leave reviews for purchased products to help others make informed decisions.*
6. ***Role-Based Access****:*
   * *Users are assigned roles (e.g., customer, store owner, admin), each with specific permissions.*

## **List of Use Cas**e**s**

The use cases for SoloStore summarize the key interactions between users and the system. Below is a list of the primary use cases:

1. **Browse Products**:
   * Customers can view all available products on the homepage, organized in a grid or list.
2. **Search Products**:
   * Customers can search for products by name or description using the search bar.
3. **Add Product to Cart**:
   * Customers can select desired products and add them to their shopping cart.
4. **Manage Products**:
   * Store owners can add, update, or delete products in their store.
5. **Place Order**:
   * Customers can place an order after adding items to their cart.
6. **View Order History**:
   * Customers can view their past orders, including order details and delivery status.
7. **Leave Product Review**:
   * Customers can leave reviews for products they have purchased.
8. **Manage Store**:
   * Store owners can edit store information, such as store name and description.
9. **Role-Based Access Control**:
   * Admins can manage user roles, ensuring secure access to system features.
10. **Login/Signup**:
    * Users can register for an account or log in to access their personalized features.

## **Extended Use Cases**

Here are detailed descriptions of some key use cases:

#### 2.4.1 Browse Products

**Actors**: Customers  
**Precondition**: The user is logged in or is accessing the homepage.  
**Basic Flow**:

1. The system displays all available products with images, names, descriptions, and prices.
2. Customers can scroll through the list and click on any product for more details. **Postcondition**: The user can view product details or proceed to add a product to their cart.

#### 2.4.2 Search Products

**Actors**: Customers  
**Precondition**: The user is on the homepage with access to the search bar.  
**Basic Flow**:

1. The customer enters a keyword into the search bar.
2. The system fetches and displays products matching the keyword.
3. The customer can click on a result to view more details or add it to their cart. **Postcondition**: Products matching the search term are displayed.

#### 2.4.3 Add Product to Cart

**Actors**: Customers  
**Precondition**: The customer is browsing products.  
**Basic Flow**:

1. The customer clicks "Add to Cart" on a product.
2. The system updates the shopping cart with the selected product and quantity.
3. The customer can proceed to checkout or continue shopping. **Postcondition**: The product is added to the cart, ready for checkout.

#### 2.4.4 Manage Products

**Actors**: Store Owners  
**Precondition**: The store owner is logged in.  
**Basic Flow**:

1. The store owner navigates to the product management page.
2. The system allows the store owner to:
   * Add a new product by providing its name, description, price, and image.
   * Update details of an existing product.
   * Delete a product. **Postcondition**: The product list is updated in the database.

## **Use Case Diagram**

# **Other Nonfunctional Requirements**

## **Performance Requirements**

**Response Time**: The system must retrieve product details and search results within **2 seconds** for up to 1000 simultaneous users.

**Concurrent Users**: The platform should handle **up to 5000 concurrent users** without a significant drop in performance.

**Order Processing**: Orders must be processed within **5 seconds** after submission.

**Database Queries**: All database queries, including product browsing, should complete within **1 second**.

These performance requirements ensure the system remains efficient and responsive under normal and peak usage.

## **Safety Requirements**

*The system must include* ***data backup functionality*** *to prevent data loss due to system failures.*

*Validation should ensure no unauthorized access to sensitive user data, such as passwords or payment details.*

*If a crash occurs, the system should log the issue and safely close all active connections without data corruption.*

*Adherence to industry-standard safety protocols, such as* ***OWASP guidelines****, is mandatory.*

## **Security Requirements**

***User Authentication****: The system must use* ***hashed passwords*** *(e.g., bcrypt) and enforce* ***strong password policies****.*

***Access Control****: Role-based access control ensures that store owners, admins, and customers can only access features assigned to their roles.*

***Encryption****: All sensitive data, such as user information and payment details, must be transmitted using* ***TLS/SSL encryption****.*

***Audit Logs****: Maintain logs of all admin and store-owner activities for auditing purposes.*

***Data Protection****: Adhere to* ***GDPR (General Data Protection Regulation)*** *and* ***CCPA (California Consumer Privacy Act)*** *for user data privacy.*

## **Software Quality Attributes**

***Usability****: The system should feature an intuitive interface, ensuring ease of use for all users with minimal training.*

***Reliability****: SoloStore must operate with* ***99.9% uptime****, ensuring consistent availability.*

***Maintainability****: The codebase must follow standard naming conventions and include comments for ease of future updates.*

***Scalability****: The system architecture should support* ***horizontal scaling*** *to handle an increased load as the user base grows.*

***Interoperability****: SoloStore should integrate seamlessly with third-party payment gateways and external APIs.*

## **Business Rules**

*Only* ***store owners*** *can add, update, or delete products.*

*Only* ***customers*** *can place orders and leave reviews.*

*Orders can only be placed for products that are in stock.*

***Admins*** *are responsible for managing user roles and monitoring activity logs.*

*Products cannot be deleted if they are associated with pending orders.*

## **Operating Environment**

***Hardware Requirements****: The system must run on servers with at least* ***16 GB RAM, 4 CPU cores, and 500 GB SSD storage****.*

***Operating System****: Compatible with* ***Windows Server 2016+****,* ***Ubuntu 18.04+****, and* ***macOS****.*

***Software Dependencies****:*

* *Java 17+*
* *PostgreSQL 14+*
* *Apache Tomcat or similar servlet container for deployment.*
* *JavaFX for the user interface.*

## **User Interfaces**

***Homepage****: Displays featured and categorized products with their images, prices, and descriptions.*

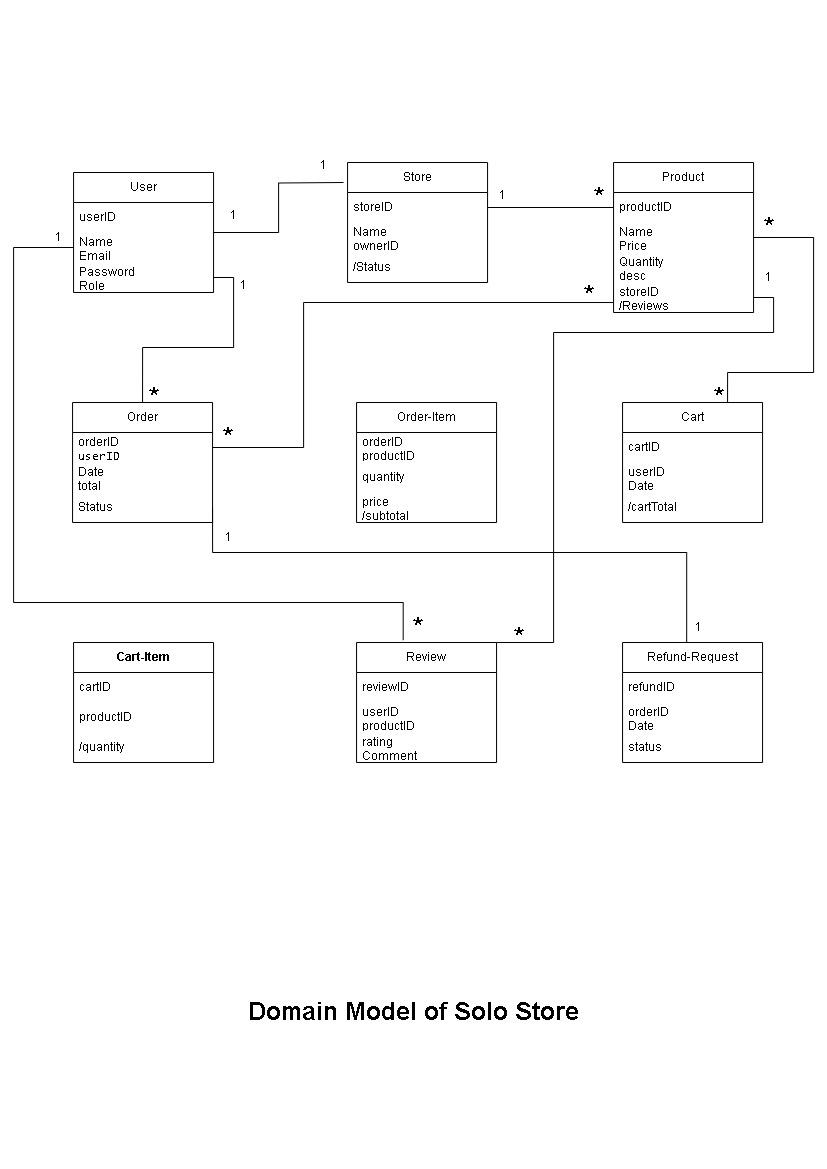
***Product Details Page****: Shows detailed information, including reviews, for selected products.*

***Admin Dashboard****: Provides analytics and tools for managing products, orders, and users.*

***Store Management Page****: Allows store owners to manage their inventory.*

***Order History Page****: Displays order details for customers.*

# **Domain Model**

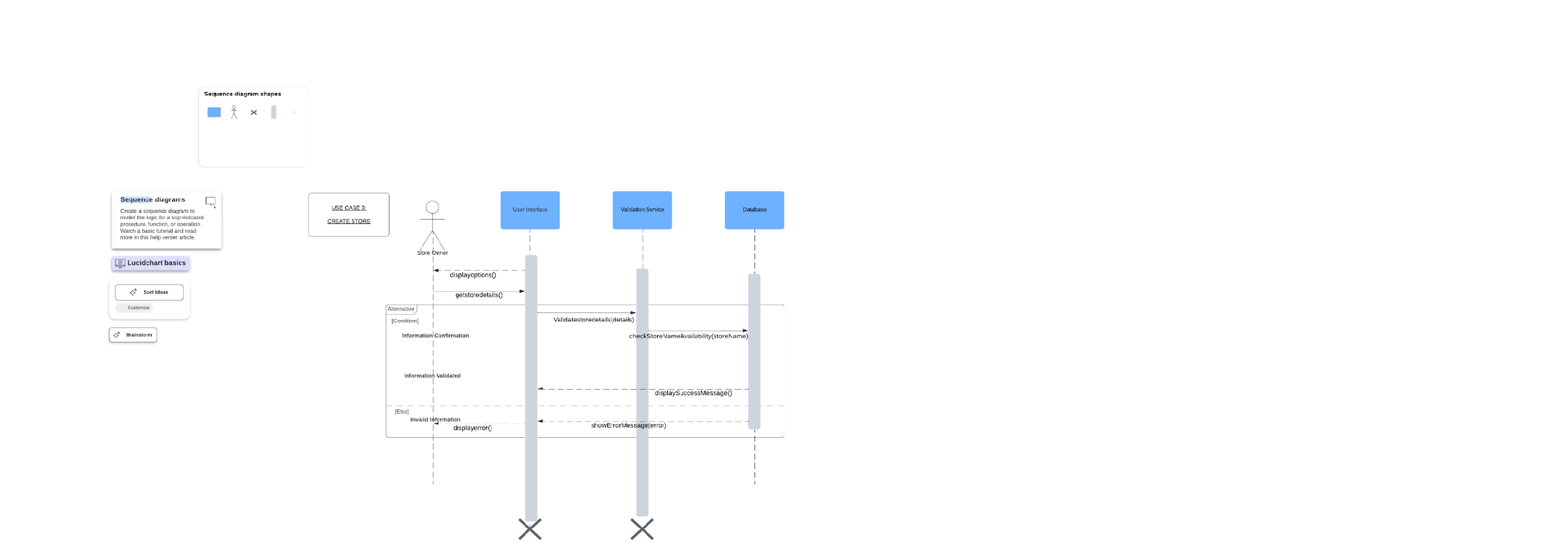


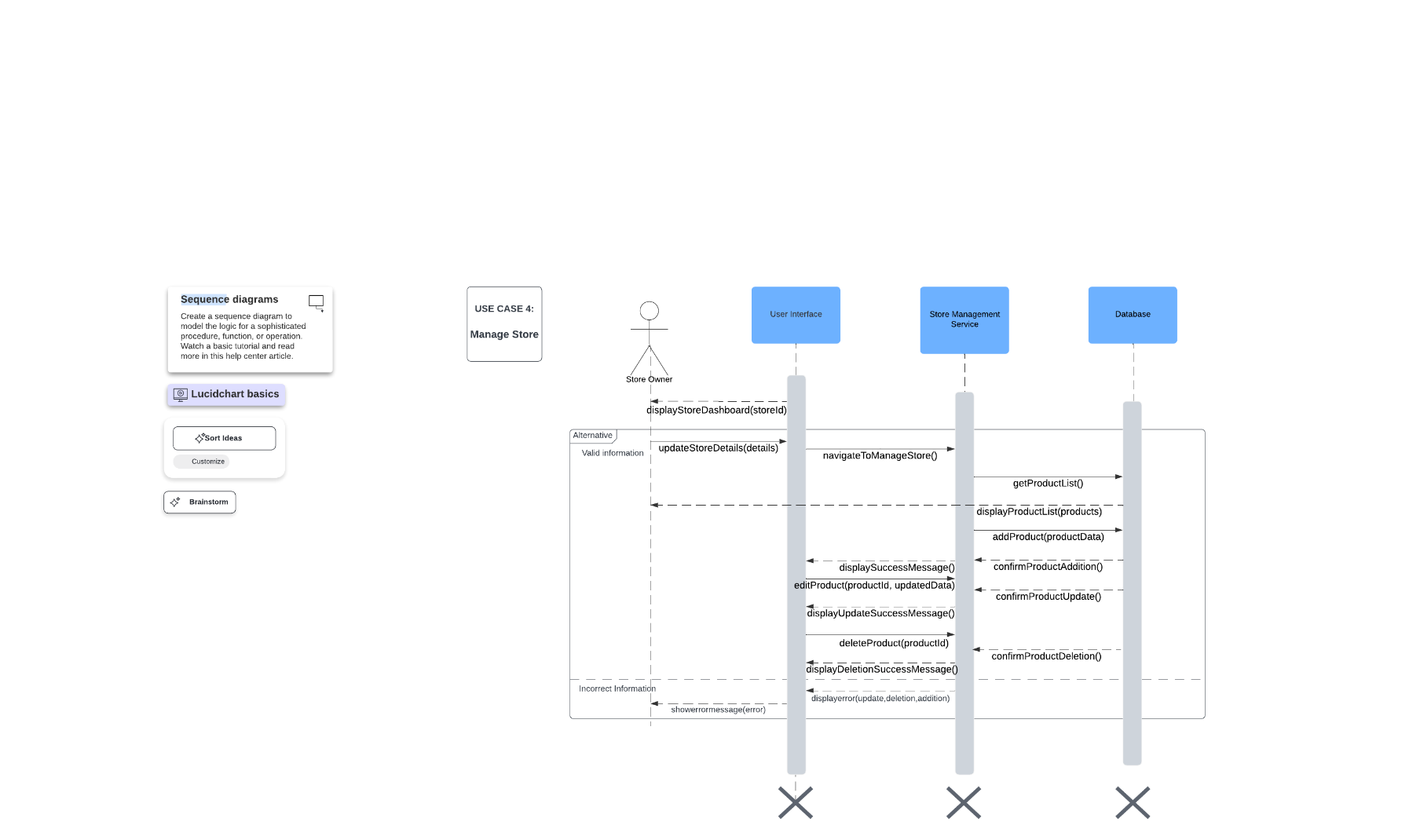
# **System Sequence Diagram**

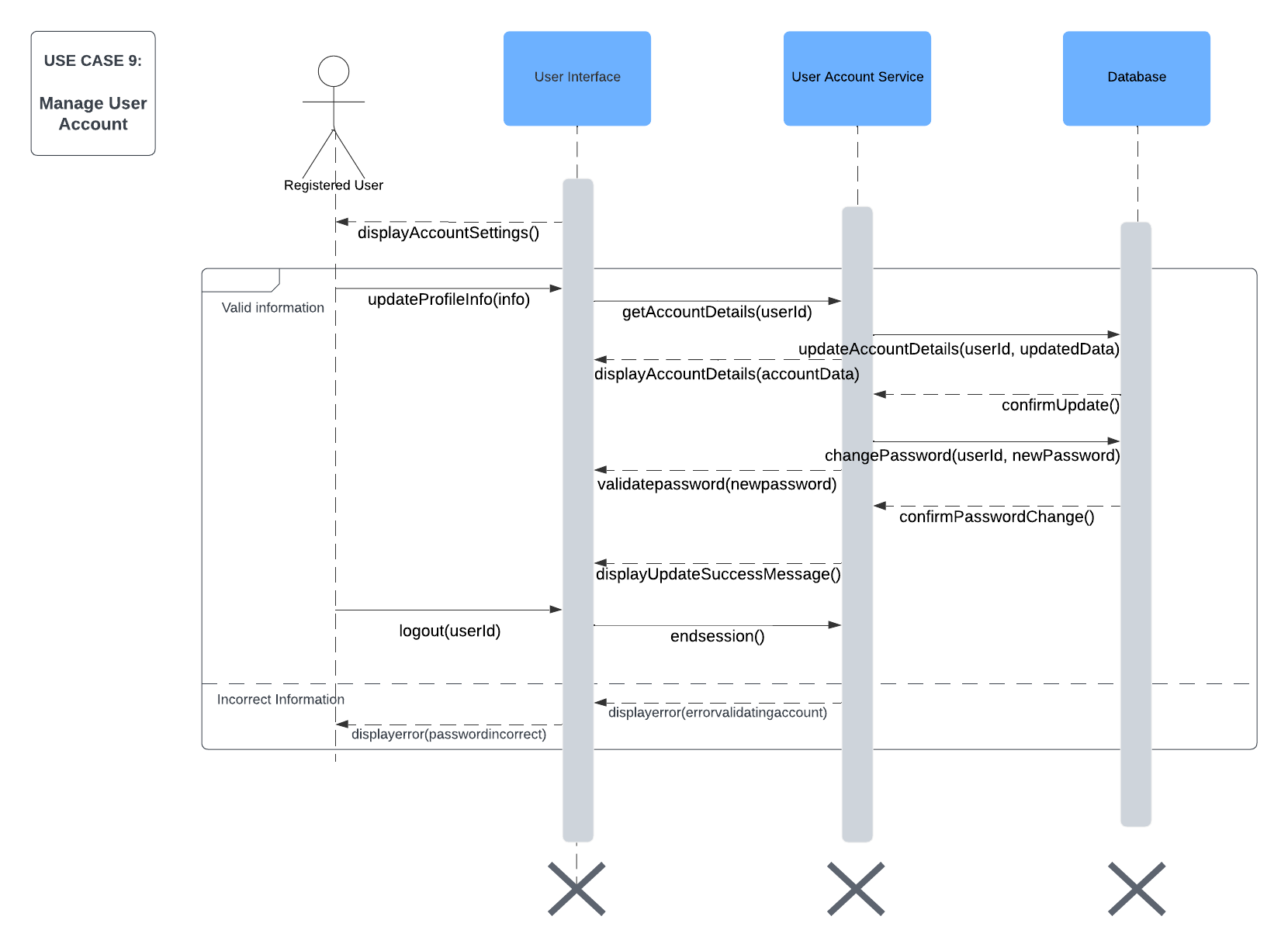
A screenshot of a computer

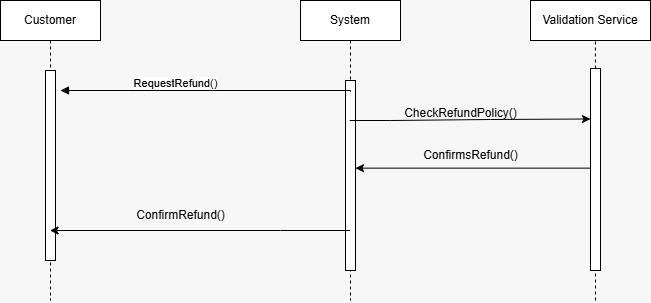
Description automatically generated

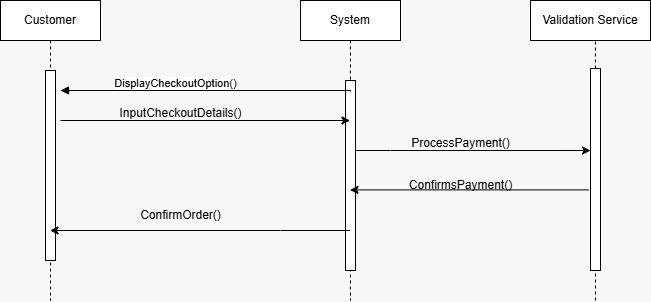
# **Sequence Diagram**



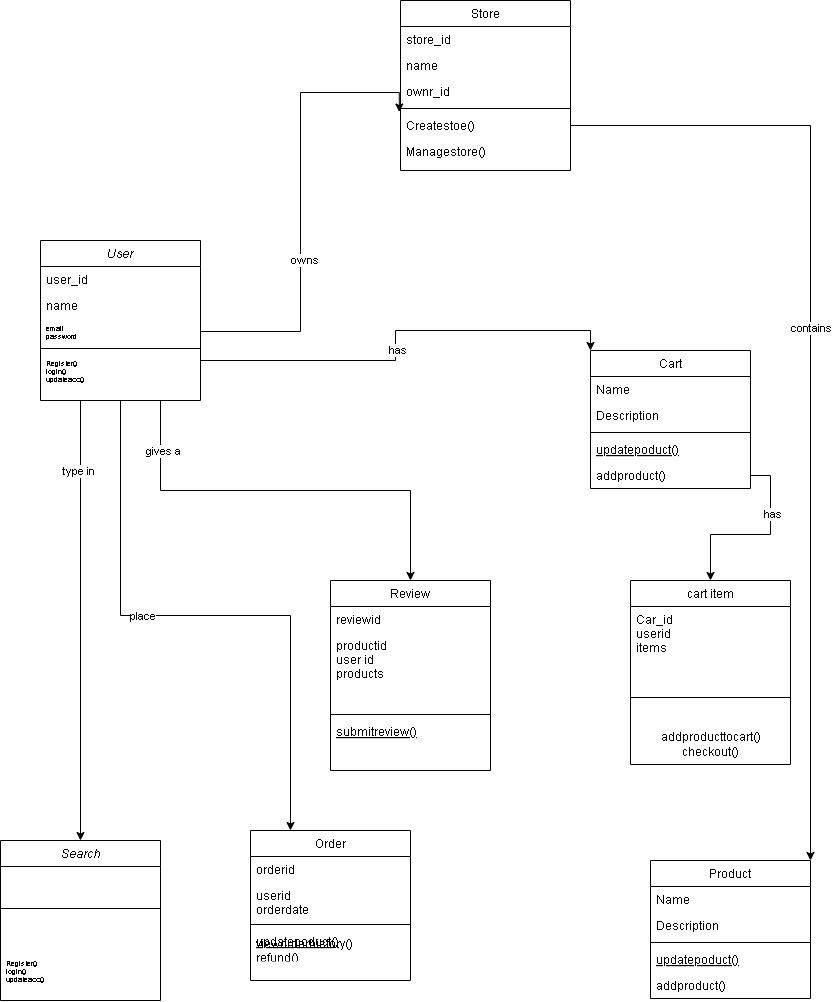






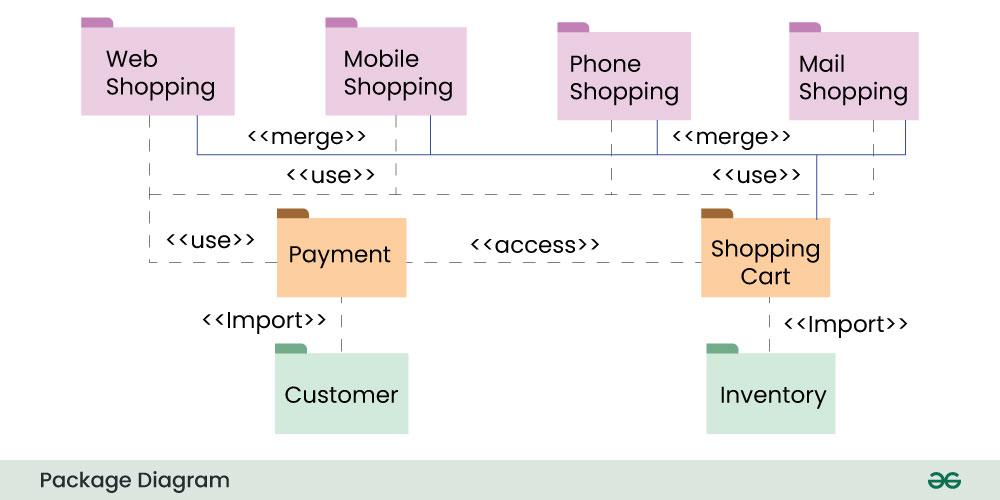


# **Class Diagram**



# **Component Diagram**

# **Package Diagram**



# **Deployment Diagram**