Software Design Description for

Ahmed Hossam, George Ayman, Nader Amir,
Nouran Mohamed, Zeina Hesham

Supervised by: Dr. Salwa Osama, Dr. Abdelnasser RiadZaied,
Eng. Sarah Hatem, Eng. Nada Ayman, Eng. Shereen ElBohy

December 19, 2023

Table 1: Document version history

Version	Date	Reason for Change
1.0	16-Dec-2023	SDD first version's description are defined.

GitHub: `https://github.com/SWE-Project-2023/App`

Contents

1 Introduction **4**
- 1.1 Purpose . . . 4
- 1.2 Scope . . . 4
- 1.3 Intended audience . . . 4
- 1.4 Reference Material . . . 5
- 1.5 Definitions and Acronyms . . . 5

2 System Overview **5**
- 2.1 System Scope . . . 6
- 2.2 System objectives . . . 6
- 2.3 System Timeline . . . 7

3 Design viewpoints **8**
- 3.1 Context viewpoint . . . 8
  - 3.1.1 Offered Services . . . 8
  - 3.1.2 Actors . . . 8
- 3.2 3.2 Composition Viewpoint . . . 8
  - 3.2.1 Architectural Design . . . 8
  - 3.2.2 Subsystem Responsibilities . . . 9
  - 3.2.3 Design Rationale . . . 10
- 3.3 Logical viewpoint . . . 10
  - 3.3.1 Logical Model: . . . 10
  - 3.3.2 Design Concerns: . . . 11
- 3.4 Patterns use viewpoint . . . 11
  - 3.4.1 Design Rationale . . . 16
- 3.5 Algorithm viewpoint . . . 16
- 3.6 Interaction viewpoint . . . 17
- 3.7 Interface viewpoint . . . 17

4 Data Design **18**
- 4.1 Data Description . . . 18
- 4.2 Database design description . . . 19

5 Human Interface Design **19**
- 5.1 User Interface . . . 19
  - 5.1.1 User Registration and Login . . . 19
  - 5.1.2 Product Catalog . . . 21
  - 5.1.3 Shopping Cart . . . 21
  - 5.1.4 Checkout . . . 21
- 5.2 Screen Images . . . 21
- 5.3 Screen Objects and Actions . . . 24

6 Requirements Matrix **25**

7 APPENDICES **26**
7.1 Github 26

Abstract

The Qanaa Pharmacy Web Application utilizes familiar technologies such as HTML, CSS, JavaScript, Node.js, EJS, and SQL to streamline medication procurement and enhance product accessibility within the domain of pharmacy websites. Focused on facilitating user-friendly prescription ordering and efficient product search, this platform prioritizes a secure, functional, and convenient user experience. By leveraging established technology, our goal is to offer a dependable addition to online pharmaceutical services. The project caters to routine healthcare requirements, providing customers with a practical and accessible means to browse and purchase pharmacy products.

1 Introduction

1.1 Purpose

The Software Design Description (SDD) document serves as a detailed blueprint illuminating the architecture and system design for the Qanaa Pharmacy Web Application. It plays a crucial role in guiding our development team in constructing a robust online medicine platform by outlining intricate design components, functionalities, and technological implementations. This document delves into technical nuances, emphasizing system architecture and design patterns while adhering to prevalent standards within the pharmacy website domain. It aims to complement the broader project overview presented in the Software Requirements Specification (SRS), focusing on intricate technical aspects and architectural delineations specific to system design and development.

1.2 Scope

The Software Design Description (SDD) document delineates the comprehensive design viewpoints and system architecture for the Qanaa Pharmacy Web Application within the context of existing pharmacy websites. It encompasses intricate functionalities, intricate design patterns, and architectural considerations while adhering to established norms and expectations within this domain. Notably, this document explicitly excludes the development of a mobile application and integrations beyond the established technology stack and specified features. This limitation is crucial for managing expectations and aligning the design boundaries within conventional pharmacy website paradigms.

1.3 Intended audience

This SDD document is crafted for our team of five dedicated students responsible for the website's development, alongside our Software Engineering professor, who provides invaluable guidance and assessment.

Additionally, the document targets the pharmacy owner, a pivotal stakeholder offering crucial insights into specific requirements and operational needs. Their input shapes the project, ensuring alignment with the pharmacy's goals.

Primarily, this document serves as a reference for the development team, academic supervisors, and industry stakeholders, facilitating effective communication and project alignment for the Qanaa Pharmacy Web Application.

1.4 Reference Material

- Development of a Data-Driven Marketing Strategy for an Online Pharmacy. **Holmer2022**
- Measures of success of computerized clinical decision support systems: An overview of systematic reviews.**JI2021196**
- E-Health Care & Online Pharmacy.**Islam2017**
- E-PHARMACY IMPACTS ON SOCIETY AND PHARMA SECTOR IN ECONOMICAL PANDEMIC SITUATION: A REVIEW.**Singh_Majumdar_Malviya_2020**
- Improving the pharmaceutical supply chain: Assessing the reality of e-quality through e-commerce application in hospital pharmacy.**Breen_Crawford_2005**

1.5 Definitions and Acronyms

Provide definitions of all terms, acronyms, and abbreviations that might exist to properly interpret the SDD. These definitions should be items used in the SDD that are most likely not known to the audience.

Term	Definition
Software Design Document (SDD)	Used as the primary medium for communicating software design information.
Design Entity	An element of a design that is structurally and functionally distinct from other elements.
Design rationale	Information capturing the reasoning of the designer that led to the system as designed, including design options, trade-offs considered, decisions made, and the justifications of those decisions. .

2 System Overview

The Qanaa Pharmacy Website is a comprehensive online pharmaceutical platform that consists of multiple key components. The primary components include a user-friendly web interface for customers to browse, search, and order medications. It also offers secure user authentication, enabling users to create and manage profiles. The system includes a robust database powered by SQL for managing medication inventory and user accounts.

Key features and functionalities encompass prescription ordering, health information access, secure payment processing, and a dynamic search function to find products efficiently. Additionally, the system provides pharmacy staff tools for order management, inventory control, and seamless customer engagement. Altogether, the Qanaa Pharmacy Web Application aims to deliver a user-centric, secure, and efficient online pharmacy experience, bridging the gap between healthcare seekers and pharmaceutical services in the digital age.

2.1 System Scope

The Qanaa Pharmacy Website system is designed to offer a seamless online pharmaceutical experience while addressing the following key objectives:

- Prescription Ordering: The system will allow customers to conveniently order medications online.
- Health Information Access: Customers can access vital health information and insights through the platform.
- Secure Payment Processing: The system will integrate secure payment processing for a seamless user experience.
- Dynamic Search Function: Users will have access to a robust search feature to efficiently find pharmaceutical products.

Internal Pharmacy Operations: The system will provide tools for pharmacy staff to manage orders, control inventory, and engage with customers.

However, it's important to clarify that the system will not include:

- Mobile Application Development: The development of a mobile application version is outside the defined scope.

2.2 System objectives

- Develop a user-friendly application accessible through personal laptops, providing step-by-step guidance for average-skilled pharmacy shoppers to complete purchases, aiming for a 20% improvement in completion rates within the first six months, and completing the development within eight months.
- Implement efficient search and filter functions within the application to enable seamless browsing, targeting a 15% increase in user satisfaction within the first four months, and completing the implementation within five months.
- Display the user's cart and wishlisted items within the application, with a goal of a 25% increase in items added to the cart and wishlist combined within the first three months.
- Implement a system to display warnings for administrators regarding low-stock inventory, aiming for a 30% reduction in instances within the first two months, and completing the implementation within three months.
- Provide administrators with statistical insights into sales performance, targeting a 15% increase in overall sales within the first six months, and ensuring availability of sales statistics within seven months.

2.3 System Timeline

Table 2: Project name time plan

Id	Task	Start Date	Number of Days	Team Member
1	Home Page	01/11/2023	16	Nouran,Zeina,George,Nader
2	Header, Footer	01/11/2023	14	Zeina,Nader,Nouran
3	GUI	01/11/2023	20	Nader,Nouran, Zeina,George
4	SignUp- SignIn	04/11/2023	10	George, Nader
5	Authentication	04/11/2023	5	George
6	Product Details	15/11/2023	5	Nouran
7	Product List	17/11/2023	5	Nouran,Ahmed
8	Wish-list	20/11/2023	5	Zeina,Nouran
9	Cart	20/11/2023	5	Nouran,George
10	Product CRUD	27/11/2023	10	George
11	User CRUD	27/11/2023	10	Ahmed,George
12	Checkout	1/12/2023	5	George
13	SDD	4/12/2023	8	Zeina,Nader,Nouran,George,Ahmed
14	Order CRUD	6/12/2023	5	Ahmed
15	Dynamic Menu	7/12/2023	2	George
16	Account Page	9/12/2023	2	Nader,George
17	Deployment	18/12/2023	1	Ahmed

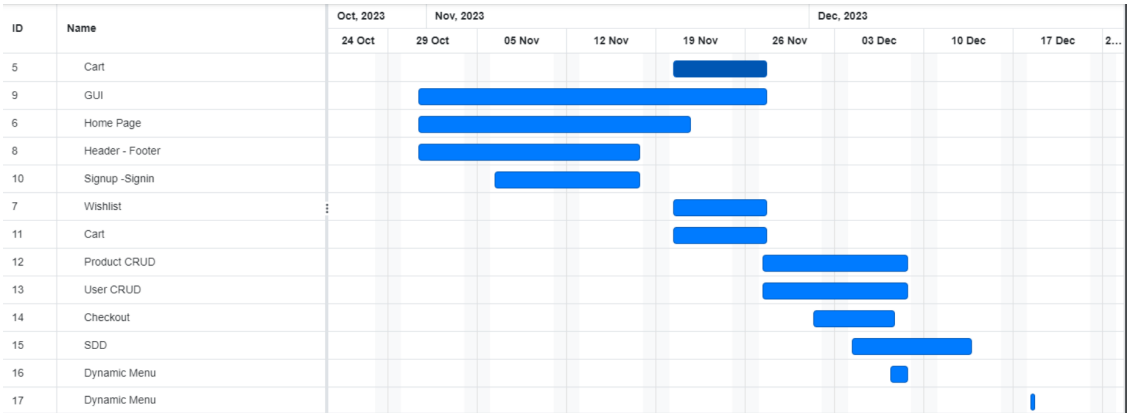

Figure 1: Project name GANTT Chart

3 Design viewpoints

3.1 Context viewpoint

The Context view serves as the initial phase in the system design process, offering a comprehensive depiction of the Qanaa Pharmacy web application's services and users. This perspective is crafted by referencing actors, encompassing users and various stakeholders, engaging with the application in its environment. The Context viewpoint adopts a "black box" approach, providing a holistic overview of the design subject.

3.1.1 Offered Services

The Qanaa Pharmacy Store is an online platform designed to facilitate the purchase of pharmaceutical products, creating a seamless and efficient experience for customers.

1. **Product Catalog:** Explore a diverse catalog of medicines with detailed information, including dosage, usage instructions, and potential side effects.
2. **Search and Navigation:** Utilize an efficient search functionality to quickly find specific medicines or browse through categories for a straightforward shopping experience.
3. **Online Ordering:** Place orders for medicines directly through the online platform, specifying quantities and any additional instructions.
4. **Order Confirmation:** Receive instant confirmation upon successful placement of orders.
5. **Order Status Notification:** Stay informed with automated notifications about the status of placed orders, from processing to delivery.
6. **User Account Management:** Create and manage user accounts for a personalized shopping experience, including order history and preferences.

3.1.2 Actors

1. **Customers:** Individuals seeking pharmaceutical products, both prescription and over-the-counter.
2. **System Administrators:** Oversee the overall functionality of the Qanaa Pharmacy Store platform, ensuring system reliability and security.

Design concerns: Systems services and users.

3.2 3.2 Composition Viewpoint

3.2.1 Architectural Design

Overall Architecture: The Qanaa Pharmacy Store adopts a simplified architecture based on the Model-View-Controller (MVC) pattern, primarily incorporating Views and Controllers. The architecture is intentionally lightweight to maintain simplicity and efficiency.

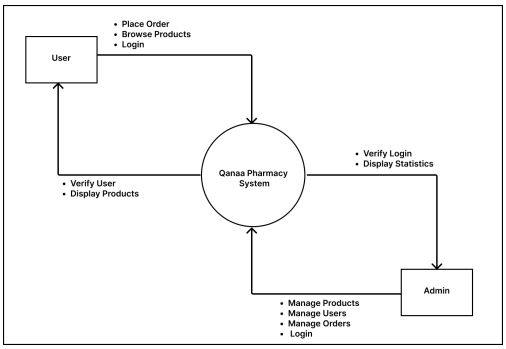

Figure 2: Context Diagram for the Inventory Management System

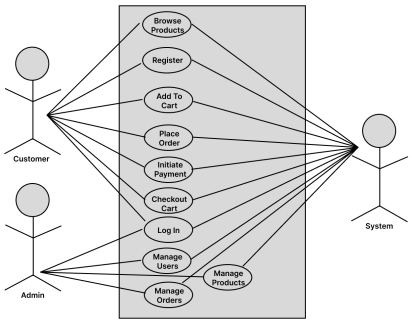

Figure 3: Use Case Diagram Example

Decomposition into Subsystems and Components:

Views: - Responsible for presenting information to users. - Handles the display of the user interface for browsing products, placing orders, and receiving order status.

Controllers: - Manages user inputs and business logic. - Orchestrates the interaction between Views and the database for seamless data flow.

Architecture Patterns: The chosen architecture aligns with the MVC pattern, focusing on a two-tier structure with Views responsible for presentation and Controllers managing user inputs and interactions with the database.

3.2.2 Subsystem Responsibilities

Views: - Render the user interface for product catalog, order placement, and order status. - Accept user inputs and forward them to Controllers for processing. - Display information retrieved from the database.

Controllers: - Receive and process user inputs from Views. - Implement business logic, including order processing and validation. - Interact directly with the database to insert and retrieve data.

Subsystem Collaboration: Views and Controllers collaborate closely to ensure a smooth user experience: - Views send user inputs to Controllers for processing. - Controllers interact with the database to retrieve product information and update order status. - Views display relevant information based on the responses from Controllers.

Design concerns: Composition and modular assembly of systems in terms of subsystems and (pluggable) components, buy vs. build, reuse of components.

3.2.3 Design Rationale

The system is structured into several key modules including user authentication, product management, and order processing. Each module encapsulates specific functionalities, facilitating scalability and maintainability. For instance, the "User Authentication" module ensures secure access to the system, while the "Order Processing" module handles transactions and inventory updates. The design rationale emphasizes modularity to enable future expansions and modifications without disrupting the entire system.

3.3 Logical viewpoint

The Logical viewpoint delineates the system's data model and entity relationships, elucidating the logical organization of key components within the Qanaa Pharmacy Web Application.

The system's logical structure revolves around essential entities such as "User," "Product," and "Order." These entities define relationships governing how users interact with the platform, including placing orders for specific products and managing their accounts.

3.3.1 Logical Model:

User Entity: The "User" entity represents individuals registered within the system and includes attributes such as username, email, and password.

Admin Entity: The "Admin" entity encapsulates details about administrators authorized to manage the system, containing attributes like admin ID, username, and administrative privileges.

Product Entity: The "Product" entity encapsulates details about pharmaceutical products available on the platform, comprising attributes like product name, description, price, and stock quantity.

Order Entity: The "Order" entity captures information regarding user purchases, containing attributes such as order ID, product IDs, user details, and transaction records.

3.3.2 Design Concerns:

The Logical viewpoint is crucial for addressing the design and implementation of appropriate abstractions within the system. It focuses on leveraging existing types for domain abstractions while considering the creation of new types for specific functionalities. The primary concern lies in making informed decisions about abstractions and their integration with existing types to ensure proper system functionality and potential for reuse.

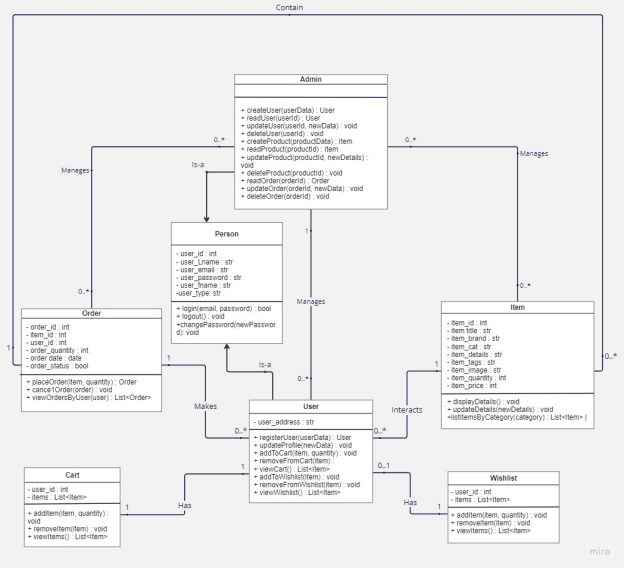

Figure 4: UML Class Diagram

3.4 Patterns use viewpoint

This viewpoint addresses design ideas focusing on the used design patterns. UML class diagram and the UML package diagram can be used here to illustrate the used design patterns.

This viewpoint discusses the design patterns utilized within the Qanaa Pharmacy Web Application and their significance in enhancing the system's structure and functionality.

The system incorporates various design patterns aiming to optimize code organization, improve maintainability, and facilitate future modifications. For instance, the application utilizes the Singleton pattern to ensure that specific critical classes have only one instance, such as the database connection manager. This pattern aids in efficient resource utilization and centralized management of shared resources throughout the application.

Additionally, the Observer pattern is employed to maintain consistency between the user interface and backend functionalities. This pattern enables automatic updates on the user interface whenever there are changes in the system, ensuring real-time synchronization of data and a responsive user experience.

The strategy design pattern is also integrated into the system architecture. It allows the application to dynamically switch between different algorithms or behaviors based on specific user

Table 3: User Class Details

Abstract or Concrete:	Concrete
Superclasses	-
Subclasses	-
Purpose	Represents individuals registered within the system and includes attributes such as username, email, and password.
Collaborations	Used in collaboration with other entities for various operations, e.g., placing orders.
Attributes	• user_id: int(3) • user_fname: varchar(20) • user_Lname: varchar(20) • email: varchar(50) • user_password: varchar(255) • user_address: text • user_isAdmin: tinyint(1) DEFAULT 0
User Operations	• viewUserProfile • updateProfile • viewCart • addToCart • removeFromCart • updateCartItemQuantity • viewWishlist • addToWishlist • removeFromWishlist • checkout • viewOrderHistory

Table 4: Admin Class Details

Abstract or Concrete:	Concrete
Superclasses	-
Subclasses	-
Purpose	Encapsulates details about administrators authorized to manage the system, containing attributes like admin ID, username, and administrative privileges.
Collaborations	-
Attributes	• admin_id: int(3) • admin_email: varchar(50) • admin_fname: varchar(20) • admin_lname: varchar(20) • admin_password: varchar(50)
Operations	• viewAllUsers • viewUser • editUser • deleteUser • viewAllOrders • viewOrder • editOrder • deleteOrder • viewAllProducts • viewProduct • editProduct • deleteProduct

Table 5: Item Class Details

Abstract or Concrete:	Concrete
Superclasses	-
Subclasses	-
Purpose	Encapsulates details about pharmaceutical products available on the platform, comprising attributes like product name, description, price, and stock quantity.
Collaborations	Used in collaboration with other entities for various operations, e.g., placing orders.
Attributes	• item_id: int(5) • item_title: varchar(250) • item_brand: varchar(250) • item_cat: varchar(250) • item_details: text • item_quantity: int(3) • item_price: int(10) • item_offers: int(1) DEFAULT 0
Operations	-

Table 6: Orders Class Details

Abstract or Concrete:	Concrete
Superclasses	-
Subclasses	-
Purpose	Captures information regarding user purchases, containing attributes such as order ID, product IDs, user details, and transaction records.
Collaborations	Used in collaboration with other entities for various operations, e.g., tracking orders.
Attributes	• order_id: int(11) • item_id: int(11) • user_id: int(11) • order_quantity: int(3) • order_date: date DEFAULT current_timestamp() • order_status: tinyint(1) DEFAULT 0
Operations	-

Table 7: Cart Class Details

Abstract or Concrete:	Concrete
Superclasses	-
Subclasses	-
Purpose	Represents items added to the user's shopping cart with details like user ID, item ID, and quantity.
Collaborations	Used to manage the user's shopping cart and interact with the 'user' and 'item' entities.
Attributes	• cart_id: int(11) • user_id: int(11) • item_id: int(11) • quantity: int(3)
Operations	-

Table 8: Wishlist Class Details

Abstract or Concrete:	Concrete
Superclasses	-
Subclasses	-
Purpose	Represents items added to the user's wishlist with details like user ID and item ID.
Collaborations	Used to manage the user's wishlist and interact with the 'user' and 'item' entities.
Attributes	• wishlist_id: int(11) • user_id: int(11) • item_id: int(11)
Operations	-

preferences or system states. For instance, it enables flexible payment processing methods, adapting to various payment gateways without significant code changes.

These design patterns collectively contribute to the system's robustness, scalability, and adaptability by providing effective solutions to common design problems encountered in the development of the Qanaa Pharmacy Web Application.

3.4.1 Design Rationale

Utilizing the Model-View-Controller (MVC) design pattern ensures separation of concerns, enhancing maintainability and scalability. The MVC pattern divides the application into three interconnected components – Model (data handling), View (user interface), and Controller (business logic). This approach fosters code reusability and facilitates parallel development by different team members, contributing to the project's efficiency.

3.5 Algorithm viewpoint

This viewpoint discusses algorithms employed for critical functionalities. For instance, the system implements search algorithms to enable efficient product discovery. These algorithms enable users to search for specific products based on keywords or categories, enhancing the overall user experience.

Decision tables and flowcharts, "pseudo-code," or (actual) system code may also be used.

3.6 Interaction viewpoint

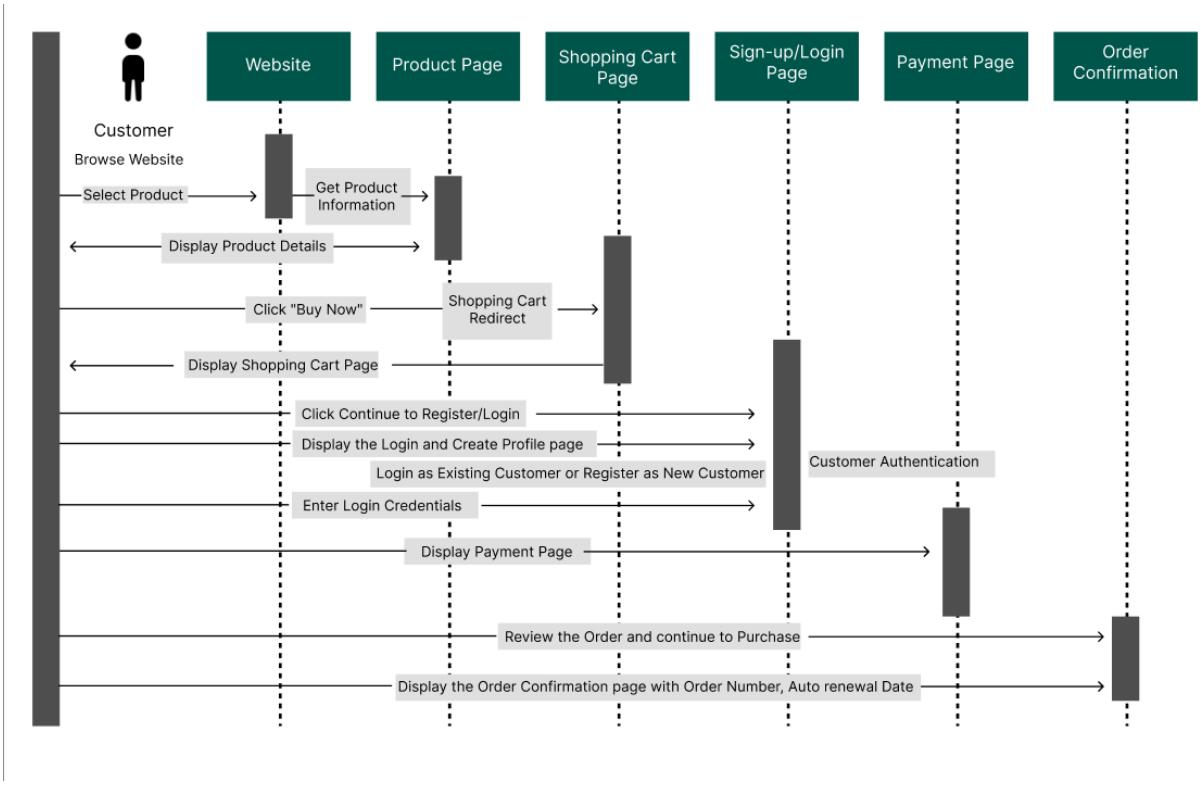

Figure 5: UML Sequence Diagram

3.7 Interface viewpoint

The Interface viewpoint details the specifications for various entities within the Qanaa Pharmacy Web Application. It acts as a guide for programmers and testers, ensuring accurate utilization of provided services.

User Authentication Service Interface:

- **Purpose:** Provides functionalities for user registration, login, and profile management.
- **Methods:** `registerUser()`, `loginUser()`, `updateProfile()`, `resetPassword()`.
- **Input Parameters:** User credentials, profile information.
- **Output Parameters:** Authentication status, user profile data.

Product Management Service Interface:

- **Purpose:** Manages product listings, including addition, modification, and removal.

- **Methods:** `addProduct()`, `updateProduct()`, `removeProduct()`.
- **Input Parameters:** Product details, identification.
- **Output Parameters:** Confirmation status, updated product data.

Order Processing Service Interface:

- **Purpose:** Handles order placement, processing, and tracking.
- **Methods:** `placeOrder()`, `processPayment()`, `trackOrder()`.
- **Input Parameters:** Selected products, payment information, shipping details.
- **Output Parameters:** Order confirmation, payment status, tracking information.

4 Data Design

4.1 Data Description

The information domain of the Qanaa Pharmacy Web Application, which includes the multifaceted data essential to pharmacy e-commerce activities, is transformed into a well-planned data structure. At its core, the system employs a relational database management system (RDBMS) to map real-world entities such as users, products, and transactions into structured tables.

User data is securely stored in a user table, with sensitive information like passwords hashed for protection. Administrative details are segregated in an admin table, delineating user access rights and maintaining stringent security measures. Products are cataloged in an item table that details product names, brands, categories, and prices, which ties directly into inventory management by tracking available quantities. Transactions are meticulously recorded in the orders table, capturing essential details such as order quantity, dates, and statuses, thus streamlining the order management process. User engagement is further fostered through cart and wishlist tables, which record current and future purchase intents, respectively. The system also features an item-images table, which decouples product imagery from item details, facilitating an adaptive and visually engaging user interface.

For real-time data entry, the system utilizes web forms. Customers input their information on the registration, login, and checkout pages. These forms are also used for administrative tasks like inventory management, where admins can add or update product information.The form inputs are processed by the server and then stored in the respective tables in the database—user information in the user table, product information in the item table, orders placed by customers in the orders table, current shopping cart details in the cart table, saved items for future purchase in the wishlist table, and the association of images with their respective products in the item-images table

In the database schema for the Qanaa Pharmacy Web Application, each table has a uniquely assigned primary key (ID) that follows an auto-incrementing integer format, such as admin-id for

the admin table or user-id for the user table, to uniquely identify each record. Tables that hold related data use these primary keys as foreign keys to establish relationships; for instance, the orders table uses user-id to link orders to users. These keys are indexed to ensure efficient data retrieval and maintain referential integrity, ensuring that all relationships within the database are consistent and valid.

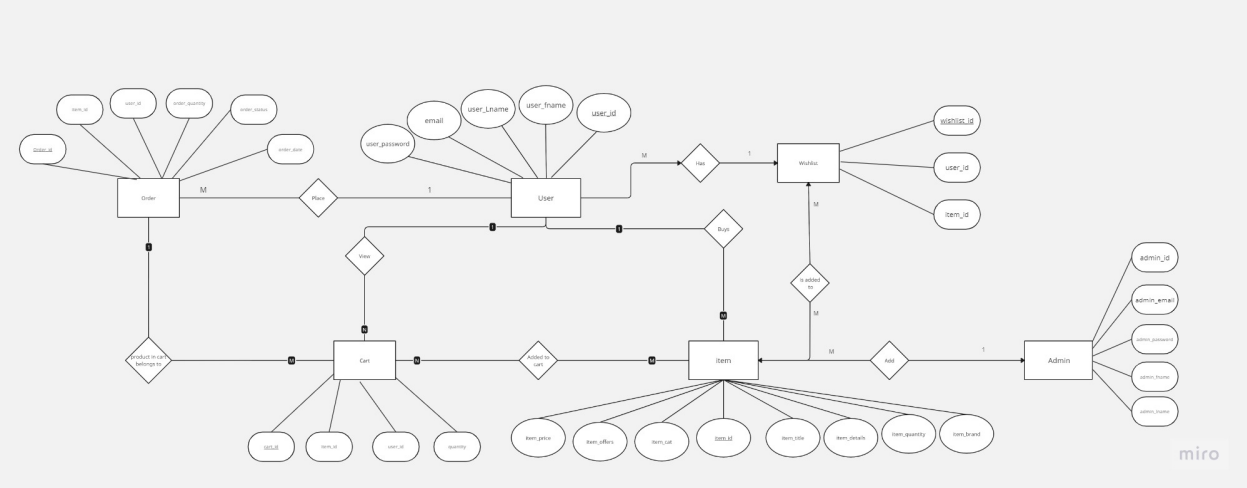

Figure 6: Entity Relationship Diagram (ERD)

Provide data models diagrams such as Entity Relationship Diagram (ERD) in this section.

4.2 Database design description

Describe any databases (provide database schema diagram) and/or description of other data storage items.

5 Human Interface Design

5.1 User Interface

Our online pharmacy's user interface is made to be simple, effective, and intuitive to provide users the best possible experience. From the user's point of view, the functionality is outlined in the following features:

5.1.1 User Registration and Login

- Users can create an account by providing necessary information.
- Returning users can log in securely using their credentials.

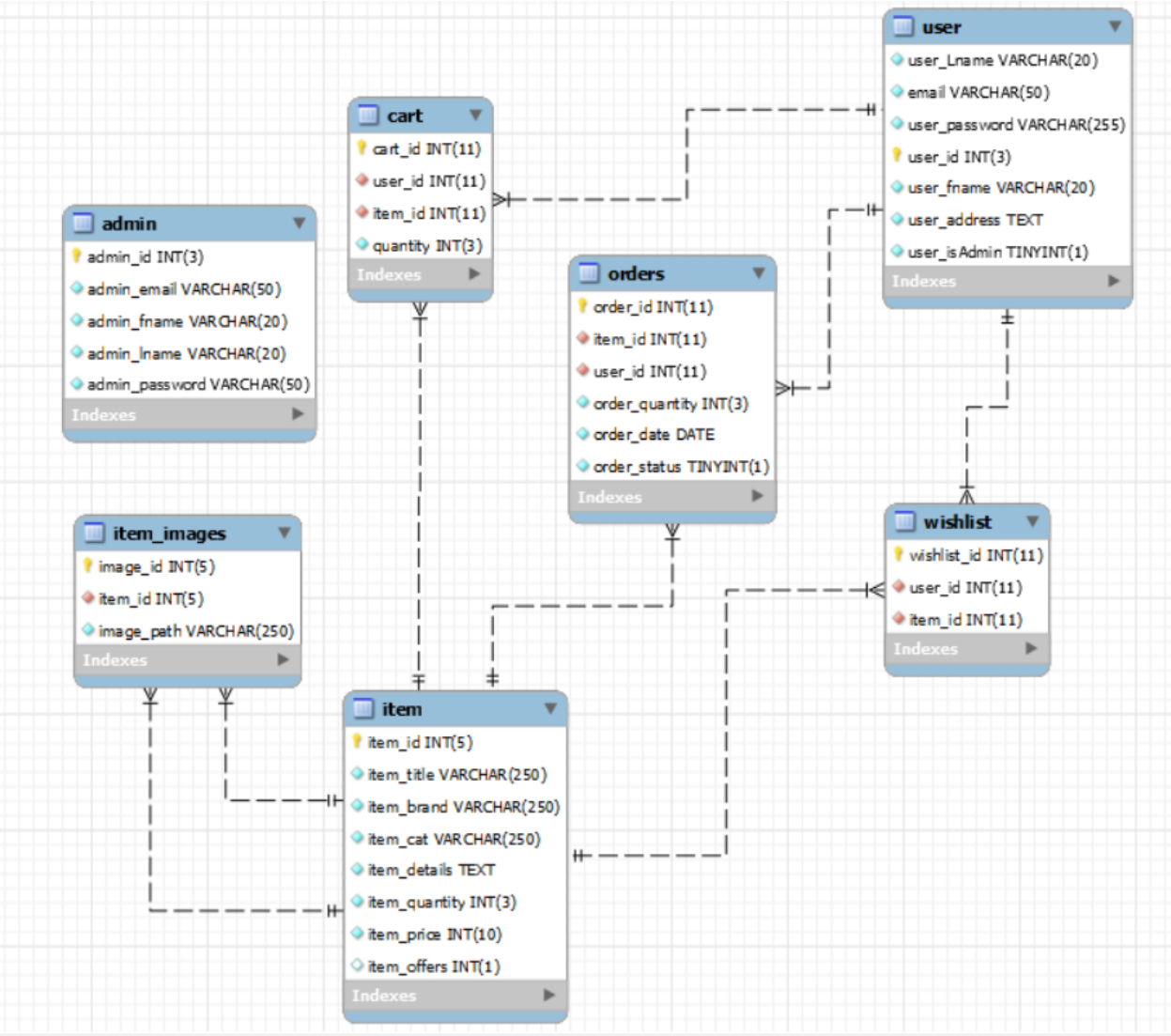

Figure 7: Database Schema Diagram

5.1.2 Product Catalog

- A search bar allows users to quickly find specific medications.
- Users can browse medications by category, brand or name

5.1.3 Shopping Cart

- Users can add medications to their shopping cart for easy management.
- The shopping cart displays a summary of selected items, quantities, and total cost.

5.1.4 Checkout

- Seamless checkout process with multiple payment options.
- Users can review and confirm their orders before finalizing the purchase.

5.2 Screen Images

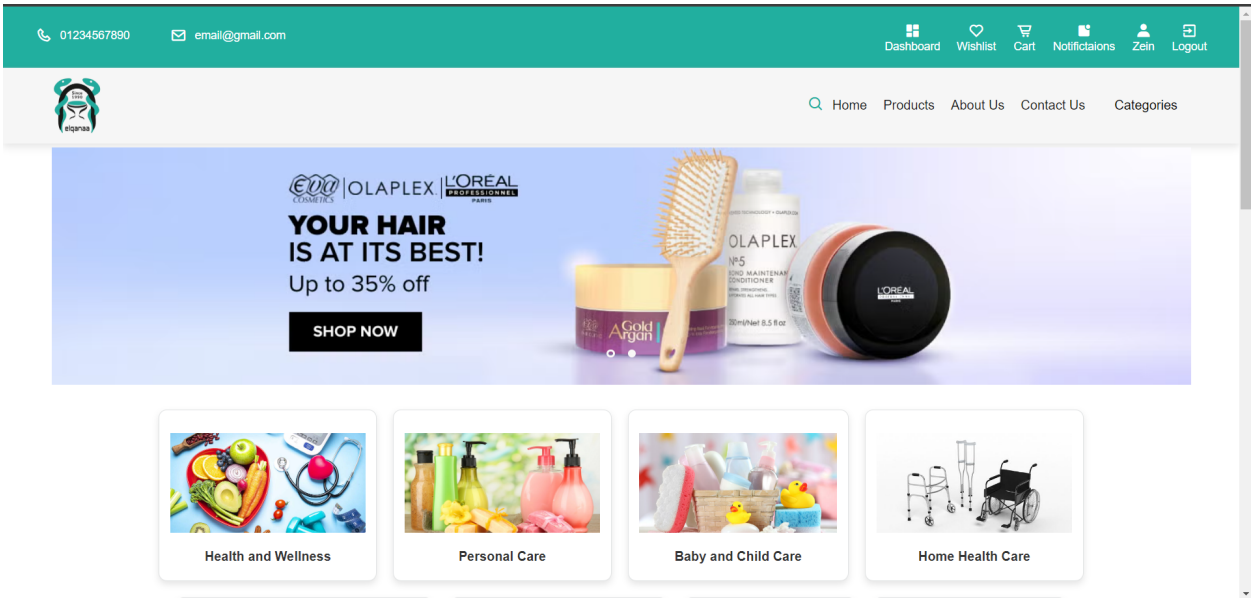

Figure 8: Home Page

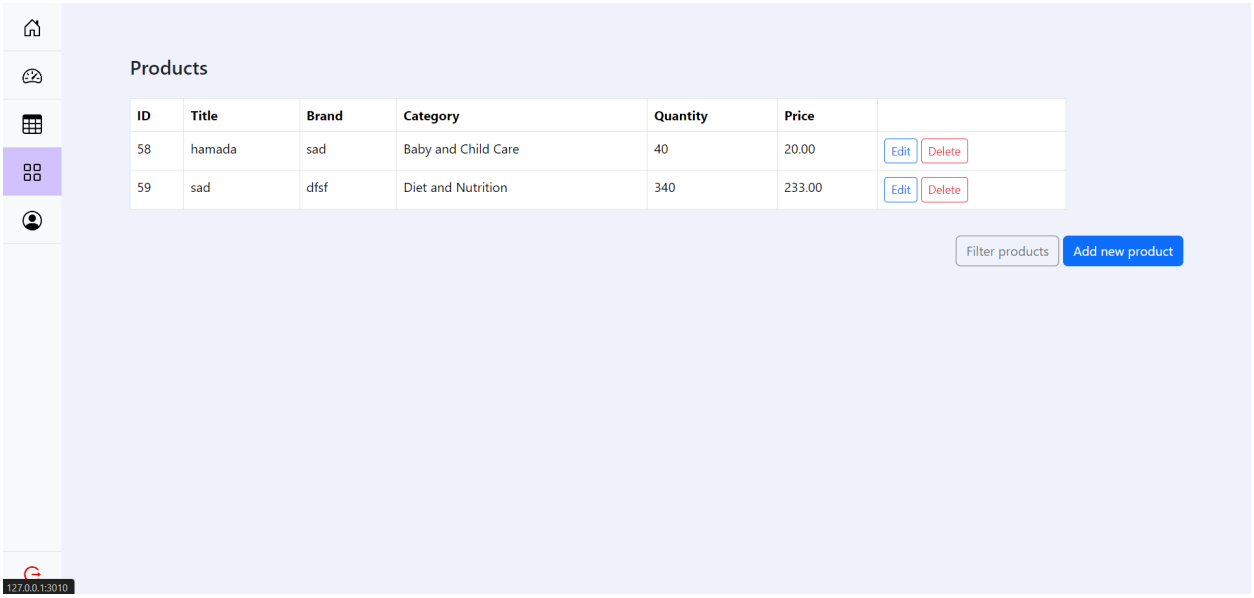

Figure 9: Admin Products

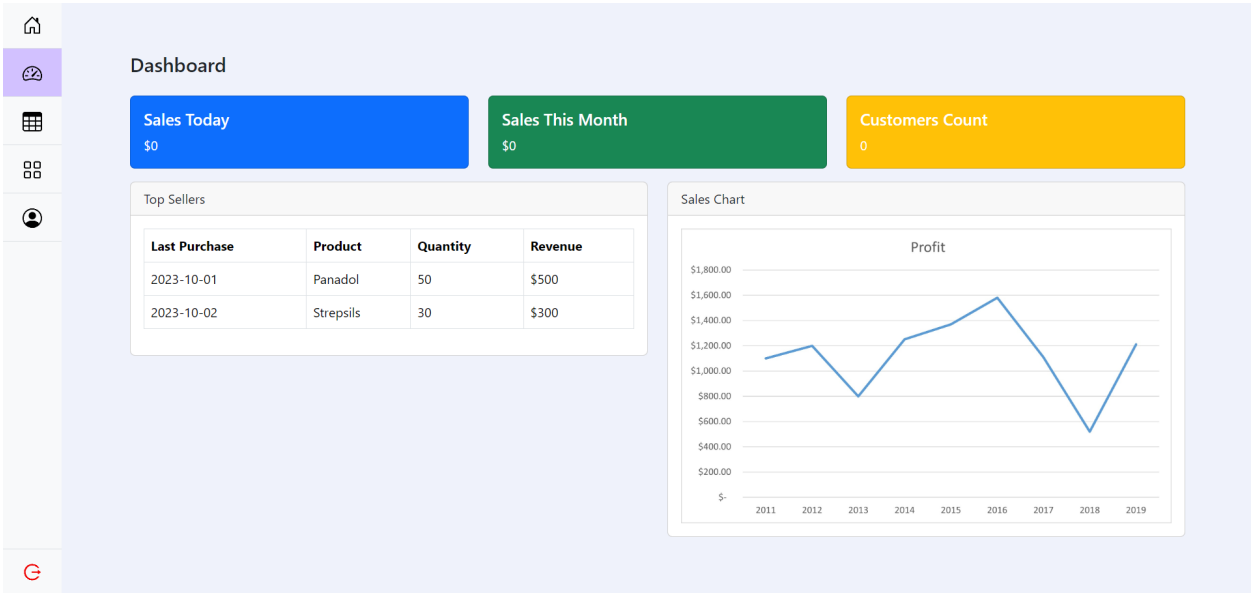

Figure 10: Admin Statistics

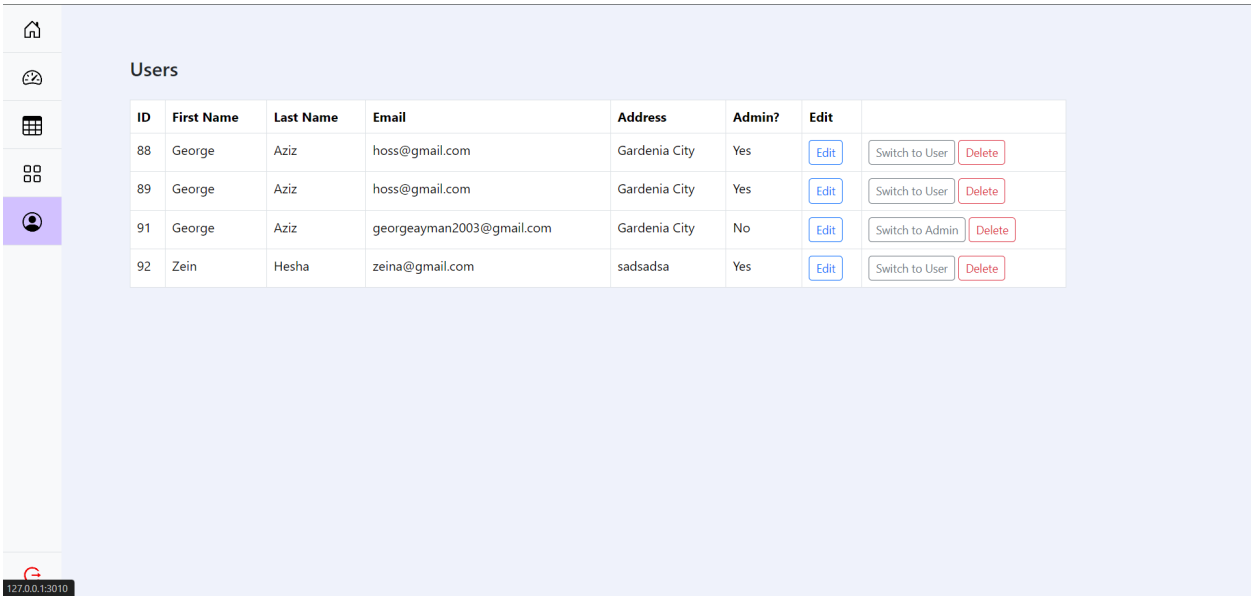

Figure 11: Admin Users

5.3 Screen Objects and Actions

1. Login Screens:

    - Screen Objects:
      Email and password fields are required for user and admin login
      Validation on the login button
    - Actions:
      Users enter their data in the fields
      Users press login to enter the web application

2. Home Page:

    - Screen Objects:
      User can check out the featured product and add it to their cart or wish-list
      User can choose from the categories the category they want
      User can check their notifications
      User can logout
      User can check their profile, cart, or wish-list

    - Actions:
      Upon hovering over the product the user can choose either to view it in full form or to add it directly to the cart or wish-list.
      Users can click on the category they want to be instantly filtered from all the products to the category they want.
      Users can press the notification sbutton and check out their latest notification
      User can press the logout button to be taken to the sign-in page again
      User can click on the profile icon button, the cart icon or the heart icon to be sent to their pages

3. Products Page:

    - Screen Objects:
      All products are featured with the ability to search and filter through them
    - Actions:
      Users enter their queries for search either by product title or brand
      Users also can enter a minimum and maximum price for their displayed products.

4. Admin Dashboard:

    - Screen Objects:
      Edit with admin-user roles
      Edit user
      Delete User

show statistics

- Actions:
  Admin can change the role of a user to become an admin.
  Admin can press the edit icon, and delete icons to subsequently use their functionalities.
  Admin can view the statistics for their inventory and past orders.

6 Requirements Matrix

Table 9: Requirements Ratrix

Req. ID	Requirement Description	Class	Test Cases ID	Status
FR01	User Registration - t user account creation functionality	Auth	TC01, TC02	Developed
FR02	Product Browsing - Develop search and filter functionality	product	TC03, TC04	Developed
FR03	Shopping Cart - Add/remove items from the cart	cart	TC05, TC06	Developed
FR04	Order Placement - Implement checkout and payment process	checkout	TC07, TC08	Developed
FR05	User Authentication - Ensure secure login and authentication	Auth	TC09, TC10	Developed
FR06	Order History - Enable users to view past orders	User	TC11, TC12	Developed
FR07	users can add/remove items to wishlist	wishlist	TC13, TC14	Developed
FR08	Admin CRUD (Users) - and Promote users to admin	Admin	TC15, TC16	Developed
FR09	Admin CRUD (Products)	Admin	TC17, TC18	Developed
FR10	View Statistics - Provide admins with a view of sales and user activity	Admin	TC19, TC20	Developed
FR11	Product Search and Filter - Allow users to search and filter products	Admin	TC21, TC22	Developed

7 APPENDICES

7.1 Github