ICT712/MITS5502

Developing Enterprise applications

Part B: ThriftTech Software Requirements Specification

Project title: ThriftTech – A Business-to-Consumer (B2C) Online Store for Technology Products

Student ID	Full Name	Team Leader Y/N
56616	Ahmed Hassan	Y
56489	Waseem Iqbal	N
55260	Shuja Khalil	N
56150	Ahmed Bilal	N

Table Of Contents:

1. Introduction

1.1 Purpose

The document explains ThriftTech B2C online store system through detailed specifications within its Software Requirements Specification (SRS) format. This document specifies all functional and non-functional needs together with system architectural plans along with design specifications for creating an e-commerce platform which delivers technology products. This document functions as a vital reference tool which maintains necessary agreement between developers testers project managers and business owners about project objectives and needs.

1.2 Scope

The ThriftTech system serves as an internet-based e-commerce website that facilitates the sale of fresh and restored technology products from smartphones to laptops and headphones together with various accessories. Users can access a safe and easy-to-use interface through the system to browse products and search for new items to purchase and maintain their account management functions. Administrators will be able to use the system for complete management of products and orders.

J2EE specifications with MVC architecture enable the system development which connects to MySQL database for data storage purposes. The scope includes:

- User registration and authentication
- Search and browse functionality
- Shopping cart functionality
- Order processing
- Payment processing (simulated)
- User profile (Viewable)

2. System Overview

2.1 System Description

As a B2C e-commerce platform ThriftTech combines superior technology products by connecting new alongside refurbished goods to end users. The e-commerce system offers customers a smooth shopping journey while providing administrators with strong tools to manage their stores properly.

Users will encounter multiple roles on the platform which includes:

1. The consumer role includes buying customers who search products through the site and then establish accounts to add items to their carts for purchase.
2. The Administrator controls both product management and order administration.

ThriftTech concentrates its operations on the resale of technology products with accessories as follows:

- Smartphones
- Laptops
- Power Banks
- Headphones
- Other Accessories

Customers need Description, pricing details, stock status and pictures of all products to reach well-informed purchasing choices.

2.2 System Context

ThriftTech will operate within the following context:

- **End Users:** Primarily tech-savvy consumers looking for competitive prices on technology products
- **Business Model:** B2C (Business to Consumer)
- **Market Segment:** Budget-conscious technology consumers and environmentally conscious buyers interested in refurbished products
- **Technical Environment:** Web-based application accessible via standard web browsers
- **External Systems:** Payment processing interface (simulation only for this project)

2.3 System Goals and Objectives

The primary goals and objectives of the ThriftTech system are:

1. Create a secure, responsive, and user-friendly e-commerce platform for thrifting technology products
2. Implement comprehensive product management features for administrators
3. Provide robust search and filtering capabilities to enhance product discovery
4. Develop an efficient shopping cart and checkout process
5. Ensure secure user authentication
6. Create an attractive and intuitive user interface
7. Apply and showcase enterprise Java development best practices

2.4 Problems it aims to solve

1. People need an established safe platform to obtain inexpensive second-hand technology goods through online transactions.
2. Due to poorly designed search functionality and filtering options in current online marketplace platforms users encounter problems when they need to thrift tech items.

3. Functional Requirements

3.1 User Management

3.1.1 User Registration

- **FR1:** User registration must be available to visitors through the system
- **FR2:** Registration into the system needs these mandatory pieces of information:
    - The application requires users to submit their email addresses which the system uses for account registration usernames
    - Password (with confirmation)
    - Full name
    - Contact number (optional)
    - Shipping address
    - Phone Number
- **FR3:** An interactive registration process within the system requires field validation for all inputs:
    - The email address requirement demands a proper format.
    - Users must create passwords minimum eight characters

- **Output Result:** Upon successful registration the system confirms account creation using the provided email as the username and stores validated user details. If any required fields are missing or invalid (e.g: improper email format or short password) the system prompts the user to correct the inputs.

3.1.2 User Authentication

- **FR1:** User authentication through the system depends on stored data to validate credentials
- **FR2:** The system shall authenticate users by verifying credentials against stored data
- **FR3:** After users provide invalid login credentials the system must present suitable error messages
- **FR4:** User login status maintenance is managed through specific sessions for each user.
- **FR5:** Through the system users can access a dedicated logout option that severs their active session before sending them to the login page.

- **Output Result:** Upon login the system validates user credentials and grants access by initiating a user session. If login fails an error message is shown, users can log out anytime ending the session and redirecting to the login page.

3.1.3 Admin User Management

- **FR1:** The system will establish individual authentication mechanisms for administrating users through specific login credentials (e.g: root/admin)
- **FR2:** When admin users successfully log in the system must automatically transfer them to an administrative dashboard.
- **Output Result:** Admin users log in with specific credentials and are redirected to the admin dashboard upon successful authentication.

3.2 Product Management

3.2.1 Product Display

- **FR1:** Product listings will display with essential information including name price and image

- **FR2:** Consumers will see complete product information through detailed pages on the system per requirement
- **FR3:** The system shall show product availability status based on inventory

- **Output Result:** The system displays products with names prices and images while detailed pages show full product info including availability status based on inventory.

3.3 Search and Browse Functionality

3.3.1 Product Search

- **FR1:** The system shall provide a search function for products
- **FR2:** The system shall allow searching by product name
- **FR3:** The system shall allow searching by matching description
- **FR4:** The system shall show relevant product information in search results

- **Output Result:** Users can search for products by name or matching description and the system returns relevant results displaying products.

3.4 Shopping Cart

3.4.1 Cart Management

- **FR1:** The system shall allow customers to add products to a shopping cart
- **FR2:** The system shall allow customers to view their shopping cart at any time
- **FR3:** The system shall allow customers to update product quantities in the cart

- **FR4:** The system shall allow customers to remove items from the cart
- **FR5:** The system shall maintain the shopping cart state across the user session
- **FR6:** The system shall display the subtotal and total price in the cart view

- **Output Result:** Customers can add update or remove products in their cart view it anytime and see the subtotal prices.

3.5 Order Processing

3.5.1 Checkout Process

- **FR1:** The system shall provide a checkout process for customers with items in their cart
- **FR2:** The system shall display an order summary during checkout
- **FR3:** The system shall offer a simulated payment interface
- **FR4:** The system shall confirm successful order placement
- **Output Result:** Customers proceed through a checkout process with an order summary and simulated payment interface. Upon success the system confirms the order placement.

3.4.2 Order Management

- **FR1:** The system shall record all orders in the database
- **FR2** The system shall allow customers to view their orders.
- **Output Result:** All customer orders are recorded in the database and users can view their orders through the system.

3.6 Payment Processing

3.6.1 Payment Simulation

- **FR1:** The system shall simulate payment processing
- **FR2:** The system shall simulate payment confirmation
- **Output Result:** The system simulates processing and displays a dummy payment confirmation popup during checkout.

4. Non-Functional Requirements

4.1 Performance Requirements

- **NFR1:** The system shall load pages within 2 seconds under normal load conditions
- **NFR2:** Database queries shall execute within 1 second
- **NFR:** The system shall optimize image loading for product displays

- **Metrics & Constraints:** The system must ensure all pages load within 2 seconds and database queries execute within 1 second under normal load while images for product displays must be optimized to load without delaying page rendering.

4.2 Security Requirements

- **NFR1:** The system shall implement secure authentication using credential validation
- **NFR2:** All passwords shall be hidden while registering.
- **Metrics & Constraints:** The system must authenticate users via credential validation within seconds and ensure password fields are masked during registration to prevent on-screen disclosure.

4.3 Usability Requirements

- **NFR1:** The user interface shall be intuitive and easy to navigate
- **NFR2:** The system shall implement responsive design for optimal viewing on different devices (desktop, tablet, mobile)
- **NFR3:** Error messages shall be clear and neat
- **NFR4:** The system shall maintain consistent design patterns across all pages
- **NFR5:** The checkout process shall require minimal steps for ease

- **Metrics & Constraints:** The system must provide a responsive UI with consistent design patterns clear error messages and a streamlined checkout process requiring minimal steps ensuring optimal usability.

4.4 Reliability Requirements

- **NFR1:** The system shall maintain 99% uptime during normal operation
- **NFR2:** The system shall handle errors gracefully without crashing

- **Metrics & Constraints:** The system must maintain at least 99% uptime under normal conditions and handle unexpected errors gracefully without leading to crashes or data loss.

4.5 Maintainability Requirements

- **NFR1:** The code shall follow Java coding standards and best practices
- **NFR2:** The system shall implement MVC architecture
- **NFR3:** The code shall include essential comments
- **NFR4:** The system shall be designed with modules for easier maintenance

- **Metrics & Constraints:** The system must follow Java coding standards with essential comments and ensure maintainable code structure for ease of updates and debugging.

5. System Architecture

5.1 Architectural Overview

ThriftTech will be implemented using the Model-View-Controller (MVC) architectural pattern, which separates the application into three interconnected components:

1. **Model:** Represents the data and business logic of the application
2. **View:** Presents the data to the user through user interfaces
3. **Controller:** Handles user requests and manages the flow between model and view

The system will be built using J2EE technologies with the following tiers:

1. **Presentation Tier:** JSP pages, HTML, CSS for user interface
2. **Business Logic Tier:** Servlets, Enterprise JavaBeans (EJB)
3. **Data Access Tier:** Hibernate framework for database operations
4. **Database Tier:** MySQL database server

5.2 Components Description

5.2.1 Presentation Components

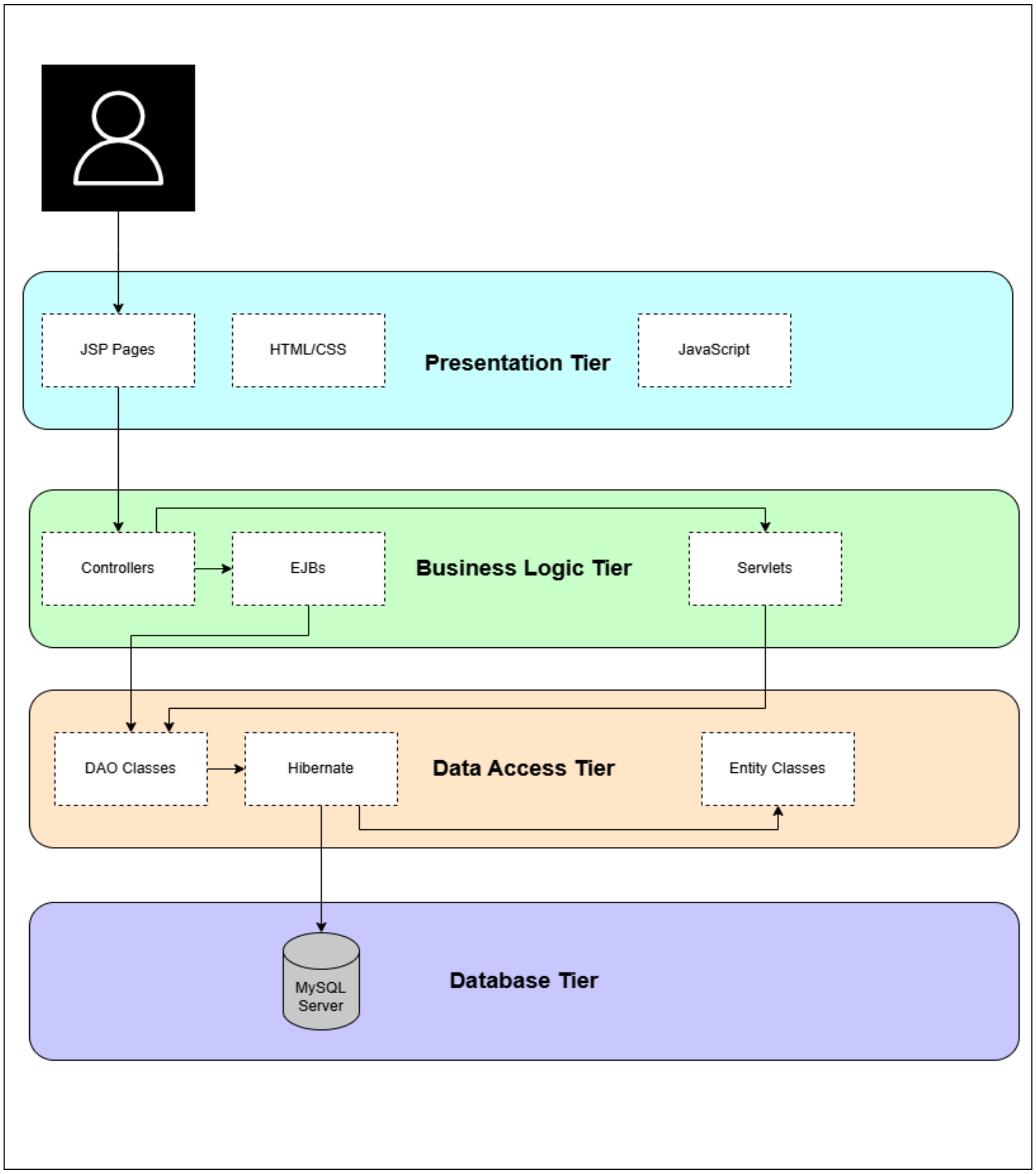
- **JSP Pages:** Dynamic web pages that display content to users
- **HTML/CSS:** Structure and styling of web pages
- **JavaScript:** Client-side validation and enhanced user experience

5.2.2 Data Access Components

- **Hibernate Framework:** ORM tool for database operations
- **DAO Classes:** Data Access Objects for database interactions
- **Entity Classes:** Java classes representing database entities

5.2.3 Security Components

- **Authentication Filters:** Check for admin credentials
- **Session Management:** Track user sessions
- **Validation Components:** Ensure data integrity



5.3 Database Design

The ThriftTech system will use a MySQL database to efficiently manage and store data. The database will be designed using a normalized schema to reduce redundancy and ensure data integrity. It will include appropriate relationships to support fast query performance.

5.3.1 Entities

1. Users
2. Products
3. Orders
4. ShoppingCart
5. Payment

5.3.2 Entities & attributes with chosen (PK & FK)

1. **Users Table**
    - UserID (PK)
    - Email
    - FullName
    - ContactNumber
    - UserType
    - Address
    - Password
2. **Products Table**
    - ProductID (PK)
    - ProductName
    - Description
    - Price
    - StockQuantity

3. **Categories Table**
   - CategoryID (PK)
   - CategoryName
   - Description

4. **Orders Table**
   - OrderID (PK)
   - UserID (FK)
   - OrderDate
   - TotalAmount
   - Status
   - PaymentMethod

5. **OrderItems Table**
   - OrderItemID (PK)
   - OrderID (FK)
   - ProductID (FK)
   - Quantity
   - PriceAtPurchase

6. **ShoppingCart Table**
   - CartID (PK)
   - UserID (FK)
   - CreatedDate
   - TotalPrice

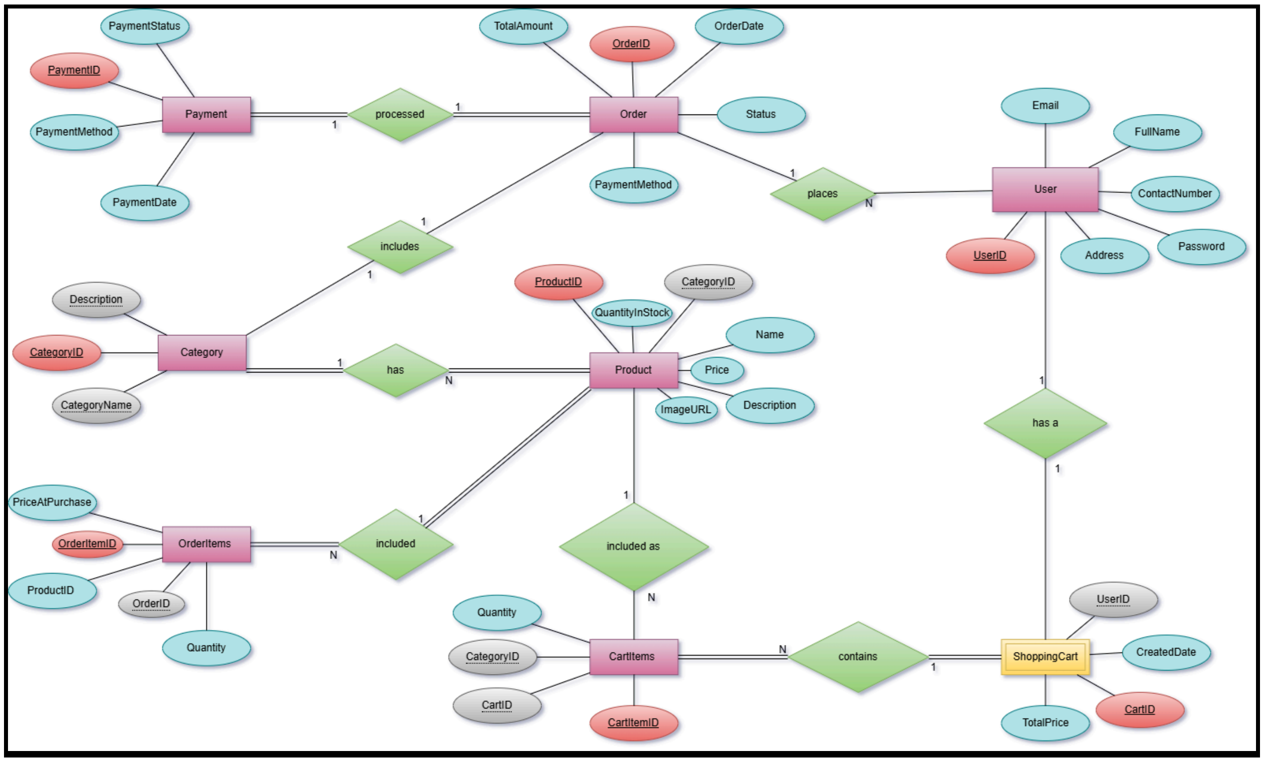
7. **CartItems Table**
    - CategoryID (PK)
    - CartID (FK)
    - ProductID (FK)
    - Quantity

8. **Payment**
    - PaymentID (PK)
    - PaymentDate
    - PaymentStatus
    - PaymentMethod

5.3.3 Relationships

- Users → Orders: One user can place many orders.
- Users → ShoppingCart: One user can have one shopping cart.
- Products → Categories: Each product belongs to one category.
- Orders → OrderItems: One order can have multiple order items.
- Products → OrderItems: One product can appear in many order items.
- ShoppingCart → CartItems: One cart can contain multiple cart items.
- Products → CartItems: One product can appear in many cart items.
- Orders → Payment: Payment method in orders refers to a method in the Payment table.
- Categories → Products: One category can have multiple products.

5.3.1 Entity Relationship Diagram



5.4 Class Diagram

The class diagram represents the object-oriented structure of the ThriftTech system and visualizes the key classes their attributes methods and relationships. It follows the UML (Unified Modelling Language) standards and highlights the high-level design of the system. Below is an overview of the main classes:

5.4.1 Main Classes

1. **Class: User**

- **Attributes:**
  - userID: int
  - email: string
  - fullName: string
  - contactNumber: string
  - userType: string
  - address: string
  - password: string

- **Methods:**
  - register()
  - login()

2. **Class: Product**

- **Attributes:**
  - productID: int
  - categoryID: int
  - productName: string
  - description: string
  - price: float
  - stockQuantity: int

- **Methods:**
  - updateStock()
  - viewDetails()

---

3. **Class: Category**

- **Attributes:**
  - categoryID: int
  - categoryName: string

  - description: string

- **Methods:**
  - addCategory()
  - editCategory()

---

4. **Class: Order**

- **Attributes:**
    - orderID: int
    - userID: int
    - orderDate: date
    - totalAmount: float
    - status: string
    - paymentMethod: string

- **Methods:**
    - placeOrder()
    - cancelOrder()
    - updateStatus()

---

5. **Class: OrderItem**

- **Attributes:**
    - orderItemID: int
    - orderID: int
    - productID: int
    - quantity: int
    - priceAtPurchase: float

- **Methods:**
    - calculateSubtotal()

---

6. **Class: ShoppingCart**

- **Attributes:**
    - cartID: int
    - userID: int
    - createdDate: date
    - price: float

- **Methods:**
    - addItem()
    - removeItem()
    - calculateTotal()

---

7. **Class: CartItem**

- **Attributes:**
    - cartID: int
    - productID: int
    - quantity: int

- **Methods:**
    - updateQuantity()

---

8. **Class: Payment**

- **Attributes:**
  - paymentID: int
  - paymentDate: date
  - paymentStatus: string
  - paymentMethod: string

- **Methods:**
  - processPayment()
  - confirmPayment()

5.4.2 Relations

1. **User → Order**:

   Type: Association, **Multiplicity:** User (1) — (0..*) Order

   A user can place multiple orders, but each order belongs to one user.

2. **Order → OrderItem**:

   Type: Composition, **Multiplicity:** Order (1) — (1..*) OrderItem

   An order contains multiple order items, and cannot exist without them.

3. **OrderItem → Product**:

   Type: Association, **Multiplicity:** Product (1) — (0..*) OrderItem

   Each order item refers to one product, but a product can appear in many orders.

4. **User → ShoppingCart**:

   Type: Association, **Multiplicity:** User (1) — (0..1) ShoppingCart

   A user can have one shopping cart, but it's not mandatory.

5. **ShoppingCart → CartItem**:

   Type: Aggregation, **Multiplicity:** ShoppingCart (1) — (0..*) CartItem

   A shopping cart contains multiple cart items, but items can exist independently.

6. **CartItem → Product**:

   Type: Association, **Multiplicity:** Product (1) — (0..*) CartItem

   Each cart item refers to one product, and a product can appear in multiple cart items.

7. **Product → Category**:

   Type: Association, **Multiplicity:** Category (1) — (0..*) Product

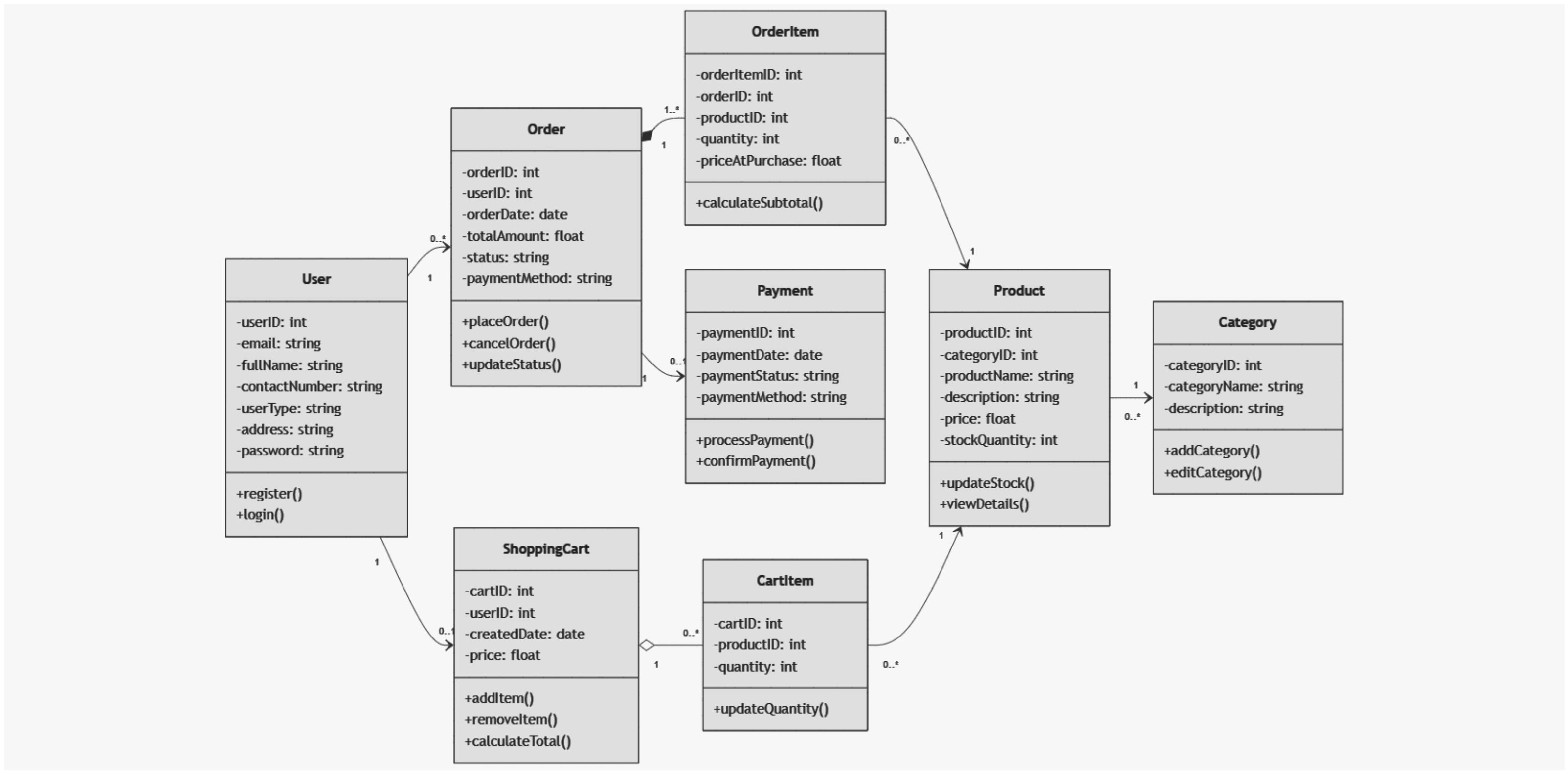
   A product belongs to one category, and a category can have many products.

8. **Order → Payment**:

   Type: Association, **Multiplicity:** Order (1) — (0..1) Payment

   An order may or may not have a payment associated with it.

5.4.3 Class Diagram



6. User Interfaces

6.1 UI Design Elements

6.1.1 Project Scope & Goals

The primary goal of ThriftTech consists of developing an online platform to let customers browse and purchase pre-owned technology items through an easy-to-use marketplace. Users of the platform could find products using easy search filters. The marketplace configuration of ThriftTech provides services to customers who love technology and to cost watchful shoppers.

User Interface development creates an efficient system through which customers can enjoy effortless shopping at ThriftTech. Customers need an easy-to-use product search system on the website together with organized product categories and plain checkout process. ThriftTech pours effort into developing their user interface as well as designing a responsive system and visually appealing portals to attract new customers and keep existing ones.

6.1.2 Create User Personas

Developing user personas for ThriftTech requires dividing the customer groups by distinct needs and behavior patterns. Various user personas for ThriftTech include:

- Tech Enthusiast Tom:
- Age: 28
- Occupation: Software Developer

The way Tom behaves shows he has technical skills which lead him to search for excellent high-end refurbished electronic gadgets including smartphones, laptops and additional accessories. Product descriptions maintain high importance to this customer.

- Budget Shopper Sarah:
- Age: 22
- Occupation: College Student

The behavior of Sarah includes constant searching for cheap electronics together with their accessories. Her main concern is finding merchandise within her budget. Sarah enjoys easy checkout procedures

The personas demonstrate typical user needs of ThriftTech users which will direct the UI design process to fulfill their respective requirements. Tom will find prominent product filters at the same time Sarah will be able to get affordable products.

6.1.3 Common UI Components

ThriftTech will feature a modern, clean, and responsive user interface with the following design elements:

- Navigation Bar: Fixed at the top with logo, search bar, cart icon, and user account links
- Footer: Contact information, quick links, and copyright information
- Buttons: Clearly styled with hover effects
- Forms: Clean layout with inline validation
- Cards: Used for product displays with consistent sizing

6.2 Navigation/User Interface Workflows

6.2.1 Customer Navigation

Registration and Login Flow:

1. User navigates to registration page
2. User completes registration form
3. System validates input
4. System creates user account
5. User is redirected to login page

6. User enters credentials
7. System authenticates user

8. User is redirected to home page

Product Browsing Flow:

1. User navigates to product category
2. User views product listings
3. User can filter or sort products
4. User clicks on product for details
5. User views detailed product information

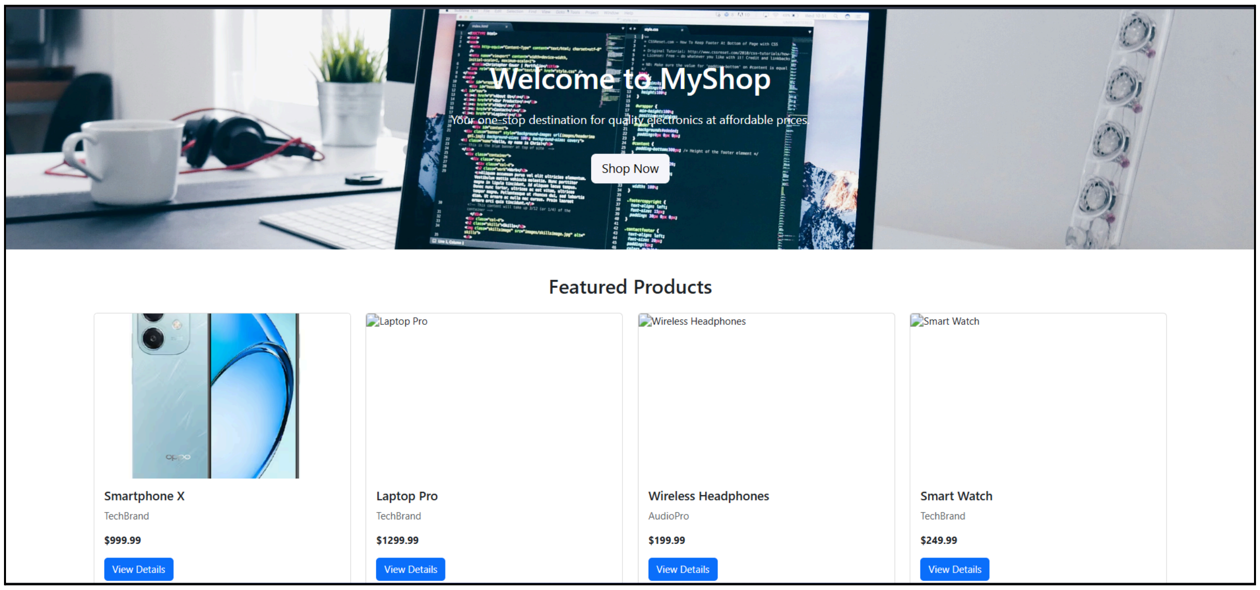
Purchase Flow:

1. User adds products to cart
2. User reviews cart contents
3. User proceeds to checkout
4. User confirms shipping details
5. User enters payment information
6. System processes payment (simulated)
7. System confirms order
8. User receives dummy order confirmation pop-up

6.3 Visual Design

The visual design idea is as follow:

Main homepage featuring products along with register page and an Empty Shopping Cart page would look something like this:



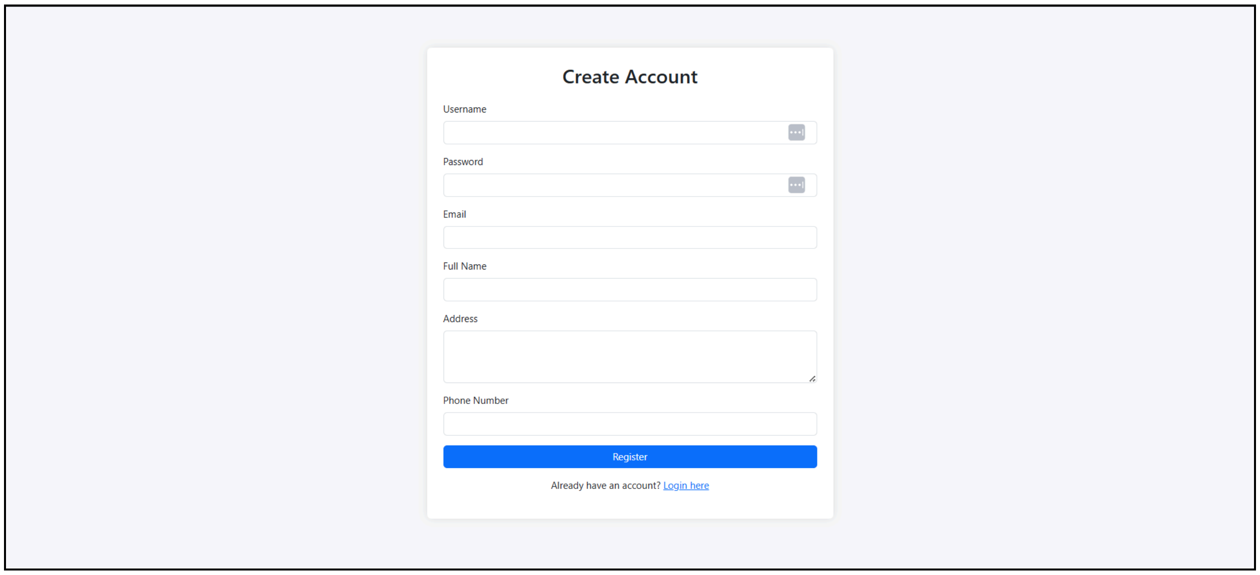
Welcome to MyShop

Your one-stop destination for quality electronics at affordable prices

Shop Now

Featured Products

Smartphone X	Laptop Pro	Wireless Headphones	Smart Watch
TechBrand	TechBrand	AudioPro	TechBrand
$999.99	$1299.99	$199.99	$249.99
View Details	View Details	View Details	View Details



Create Account

Username

Password

Email

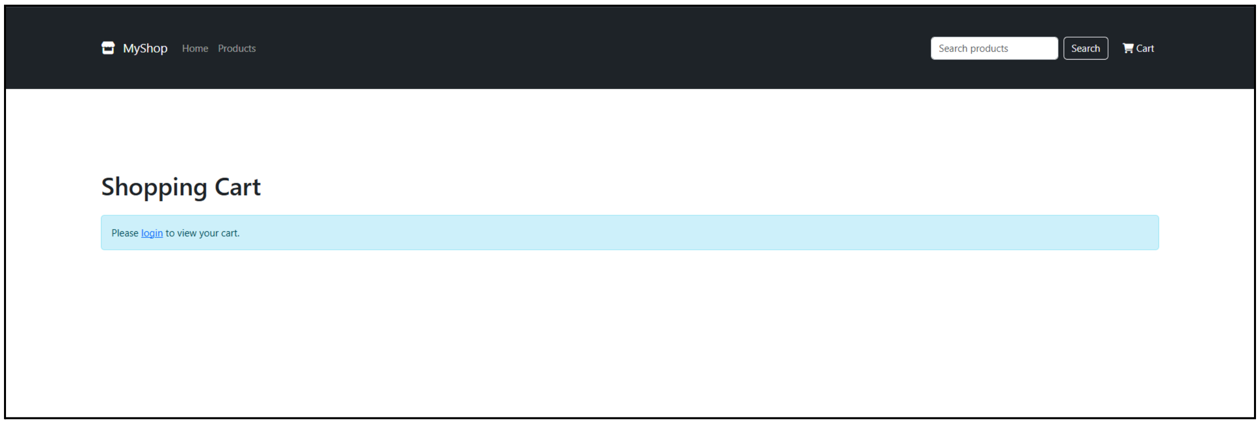
Full Name

Address

Phone Number

Register

Already have an account? Login here



Shopping Cart

Please login to view your cart.

7. Data Requirements

7.1 Data Entities

7.1.1 User Entity

- Stores customer and administrator information
- Contains authentication details and personal information
- Includes attributes: UserID, Email, Password, FullName, ContactNumber, UserType

7.1.2 Product Entity

- Stores product information including details, pricing, and stock
- Contains product descriptions and categorization
- Includes attributes: ProductID, CategoryID, ProductName, Description, Price, StockQuantity

7.1.3 Order Entity

- Stores order information
- Includes attributes: OrderID, UserID, OrderDate, TotalAmount, Status, PaymentMethod

7.1.4 Cart Entity

- Stores shopping cart information
- Includes attributes: CartID, UserID, CreatedDate, Price

7.2 Data Storage

7.2.1 Database Server

- MySQL database server will be used for data storage
- Database schemas will be properly normalized to reduce redundancy

7.3 Data Access

7.3.1 Data Access Layer

- Hibernate framework will be used for object-relational mapping
- DAO pattern will be implemented for database operations
- JPA annotations will be used to map Java objects to database tables

8. External Interfaces

8.1 User Interfaces

8.1.1 Web Browser Interface

- The system will be accessible via standard web browsers
- Supported browsers: Chrome, Edge (latest versions)
- Responsive design will ensure compatibility with various screen sizes

8.2 Software Interfaces

8.2.1 Database Interface

- JDBC/Hibernate interface to MySQL database
- Connection pooling will be implemented for efficient database access

8.2.2 Payment Gateway Interface (Simulated)

- Interface for processing payments
- Implementation will simulate payment processing for demonstration purposes

8.3 Communication Interfaces

8.3.1 HTTP

- The system will communicate with clients via HTTP protocol
- All operations will be simulated using appropriate protocols

9. System Constraints

9.1 Hardware Limitations

- The system will be developed and tested on standard development hardware
- No specific hardware optimizations will be implemented

9.2 Software Limitations

- The system will be developed using:
  - NetBeans 25 or higher as the IDE
  - Java (J2EE) as the programming language
  - MySQL as the database server
- No third-party libraries outside those specified in the requirements will be used

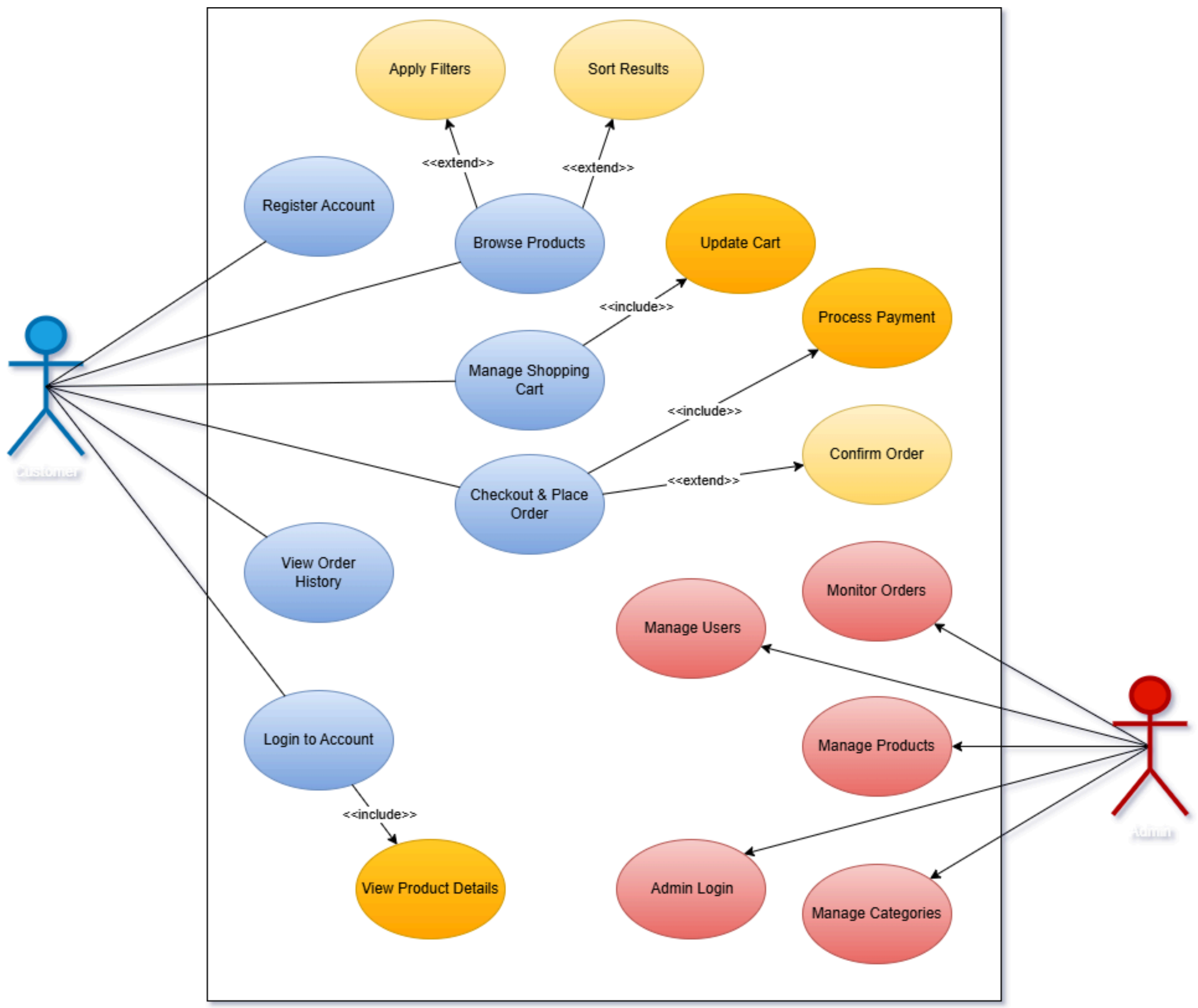
9.3 Design Constraints

- The system must follow MVC architecture
- The system must use Hibernate framework for database operations
- The system must implement Enterprise Java Beans
- The system must include session management

10. Use Cases and Scenarios

10.1 Use Case Diagram

The following use case diagram illustrates the key interactions between actors and the ThriftTech system:



10.2 Detailed Use Cases

10.2.1 Customer Use Cases

UC1: Register Account

- **Actor**: Customer
- **Description**: Customer creates a new account in the system
- **Preconditions**: Customer is not logged in
- **Main Flow**:
  - Customer navigates to registration page
  - Customer fills in required information (email, password, name, address)
  - System validates the information in real-time
  - System creates the account and securely stores credentials

UC2: Login to Account

- **Actor**: Customer
- **Description**: Customer logs in to their account
- **Preconditions**: Customer has registered account
- **Main Flow**:
  - Customer navigates to login page
  - Customer enters email and password
  - System validates credentials against stored data
  - System creates session and redirects to home page

UC3: Browse Products

- **Actor**: Customer

- **Description**: Customer browses products by category or search
- **Preconditions**: None

- **Main Flow**:
  - Customer navigates to category page or uses search function
  - System displays relevant products
  - Customer can filter and sort products
  - Customer selects product to view details
  - System displays comprehensive product information

UC4: Manage Shopping Cart

- **Actor**: Customer
- **Description**: Customer adds, updates, or removes items from cart
- **Preconditions**: Products are available in inventory
- **Main Flow**:
  - Customer adds product to cart from product page
  - System confirms addition and updates cart count
  - Customer navigates to cart page to view contents
  - Customer adjusts quantities or removes items
  - System updates cart totals in real-time

UC5: Checkout and Place Order

- **Actor**: Customer
- **Description**: Customer completes purchase process
- **Preconditions**: Customer has items in cart
- **Main Flow**:
  - Customer initiates checkout from cart page
  - Customer confirms or updates shipping address
  - Customer selects payment method
  - Customer reviews order summary

  - Customer confirms order placement
  - System processes payment (simulated)

UC6: View Order History

- **Actor**: Customer
- **Description**: Customer views past orders and status
- **Preconditions**: Customer is logged in and has placed orders
- **Main Flow**:
  - Customer navigates to order history page
  - System displays list of orders with basic information

10.2.2 Admin Use Cases

UC1: Admin Login

- **Actor**: Admin
- **Description**: Admin authenticates to access administrative functions
- **Preconditions**: Admin has credentials in the system
- **Main Flow**:
  - Admin navigates to admin login page
  - Admin enters credentials
  - System validates admin-level access
  - System redirects to admin dashboard
- **Postcondition**: Admin is authenticated with administrative privileges

UC2: Manage Products

- **Actor**: Admin
- **Description**: Admin creates, updates, or removes products
- **Preconditions**: Admin is logged in
- **Main Flow**:

- Admin navigates to product management section
- Admin views list of existing products
- Admin selects to add new product or edit existing one
- Admin enters/updates product information
- System validates and saves changes

- **Postcondition**: Product catalog is updated

UC3: Monitor Orders

- **Actor**: Admin
- **Description**: Admin views and manages order statuses
- **Preconditions**: Admin is logged in
- **Main Flow**:
    - Admin navigates to order management section
    - System displays list of orders with status information
    - Admin selects specific order to view details
    - Admin updates order status if needed
    - System saves changes
- **Postcondition**: Order statuses are up-to-date

UC4: Manage Users

- **Actor**: Admin
- **Description**: Admin views and manages user accounts
- **Preconditions**: Admin is logged in
- **Main Flow**:
    - Admin navigates to user management section
    - System displays list of registered users
    - Admin can view user details

Postcondition: User account statuses are updated

UC5: Manage Categories

- **Actor**: Admin
- **Description**: Admin creates, updates, or removes product categories
- **Preconditions**: Admin is logged in
- **Main Flow**:
  - Admin navigates to category management section
  - Admin views list of existing categories
  - Admin selects to add new category or edit existing one
  - Admin enters/updates category information
  - System saves changes

11. Shopping Cart Implementation

11.1 Shopping Cart Features

- Online carts automatically update without needing page reloading.
- Persistent cart storage across sessions for logged-in users
- Anonymous cart functionality with session-based storage
- Quantity adjustment with inventory validation
- Quick product removal with confirmation
- The shopping cart shows details that include calculation of the subtotal amount
- Empty cart detection

11.2 Shopping Cart Workflows

11.2.1 Adding Products

1. A user can add products to their shopping cart by selecting "Add to Cart" from display pages.
2. System validates product availability
3. System increases the quantity count when a product exists already in the shopping cart.
4. The system creates a new cart entry for items at a starting quantity of one.
5. Cart total is recalculated

11.2.2 Updating Quantities

1. User navigates to cart page
2. The User adjusts product counts through either automatic increment-decrement controls.
3. The system checks the new quantity value against available inventory stocks.
4. Cart totals update immediately
5. When available stock reaches its limit the system both applies restrictions on product numbers and alerts the user.

11.2.3 Removing Items

1. The user selects the remove icon which appears next to the carted item.
2. System confirms removal intent
3. Item is removed from cart
4. Cart totals update immediately
5. An empty cart message shows when the cart reaches its empty state.

11.2.4 Proceeding to Checkout

1. User clicks "Checkout" button
2. The system leads users first to the checkout process step.
3. The payment process is simulated and a successful order message is shown as pop-up

11.3 Shopping Cart Design

11.3.1 Technical Implementation

- The shopping cart maintains its data in session storage for each logged on users
- Database persistence for authenticated users

11.3.2 User Interface Design

- The interface shows clear levels through product information as well as pricing details
- The program includes noticeable quantity adjustment controls
- Subtotal calculation at runtime
- The website provides estimates for delivery fees and taxes (dummy tax fee)
- Users who have empty carts will see continue shopping options
- Responsive design for all device sizes

12. Site Map

12.1 Site Map Overview

The ThriftTech site map is organized to provide intuitive navigation and access to all system features:

12.1.1 Main Sections

- **Home**: Landing page with featured products
- **Shop**: Product browsing organized by categories
    - Smartphones
    - Laptops
    - Power Banks
    - Headphones
    - Accessories
- **Account**: User-specific account management
    - Login/Register
    - Profile
    - Orders
- **Cart**: Shopping cart and checkout process
- **Admin**: Administrative functions (restricted access)

12.1.2 Static Pages

- About Us
- Privacy Policy
- Shipping Information
- Return Policy

12.2 Navigation Paths

12.2.1 Main Navigation Paths

- **Home → Category → Product → Cart → Checkout → Confirmation**
  - Primary purchase flow for customers
- **Home → Account → Login → Profile → Orders**
  - Account management and order tracking

12.2.2 Administrator Navigation Paths

- **Login → Admin Dashboard → Product Management → Add/Edit Product**
  - Product catalog management
- **Login → Admin Dashboard → Order Management → Order Details**
  - Order processing workflow

12.2.3 Mobile Navigation Considerations

- Collapsed menu for small screens
- Streamlined checkout process for mobile devices
- Touch-friendly interface elements

13. Testing and Quality Assurance

13.1 Testing Approach

The ThriftTech system will undergo comprehensive testing to ensure quality and reliability:

13.2 Test Cases

13.2.1 Functional Test Cases

- **TC001**: User Registration
  - Verify all required fields are validated
  - Test email uniqueness validation
  - Verify password strength requirements
  - Confirm success message

- **TC002**: User Login
  - Test valid credentials authentication
  - Test invalid credentials handling
  - Verify session creation and management

- **TC003**: Product Management
  - Test adding new products with all details
  - Test updating existing product information
  - Verify product deletion

- **TC004**: Shopping Cart
  - Test adding products to cart
  - Verify quantity updates and validations

    - Test removal of cart items
    - Verify cart persistence across sessions

- **TC005**: Checkout Process
    - Test complete checkout workflow
    - Test payment processing simulation
    - Confirm order creation and confirmation simulation

13.3 Quality Assurance

- Regular system health checks and monitoring
- Performance monitoring and optimization
- Rollback procedures in case of issues
- Daily database backups
- File system backup procedures
- Disaster recovery plan

14. Appendices

14.1 Technology Stack

- **Frontend**:
    - HTML5, CSS3, JavaScript
    - Bootstrap 4.6 for responsive design
- **Backend**:
    - Jakarta EE 11
    - J2EE framework
    - Servlets and JSP
    - Enterprise JavaBeans (EJB)
    - Hibernate

- **Database**:
    - MySQL 8.0.42
    - Connection pooling with JDBC
- **Development Tools**:
    - NetBeans 25 IDE
    - Maven for dependency management

14.2 Development Environment

- **IDE**: Apache NetBeans 25
- **Server**: GlassFish v8.0
- **Database**: MySQL 8.0.42 with MySQL Workbench
- **JDK**: Java Development Kit 21

14.3 Glossary

- **DAO**: Data Access Object - pattern for database interaction
- **EJB**: Enterprise JavaBeans - server-side component architecture
- **Hibernate**: Object-relational mapping framework for Java
- **JPA**: Java Persistence API - Java specification for ORM
- **JSP**: JavaServer Pages - technology for creating dynamic web content
- **MVC**: Model-View-Controller - architectural pattern for UI applications
- **ORM**: Object-Relational Mapping - technique for converting data between incompatible systems
- **SQL**: Structured Query Language - language for database interaction
- **UI/UX**: User Interface/User Experience - design considerations for human-computer interaction
- **Validation**: Process of ensuring data correctness and integrity

References

[1] IEEE. "IEEE Recommended Practice for Software Requirements Specifications," in IEEE Std 830-1998, vol., no., pp.1-40, 20 Oct. 1998, doi: 10.1109/IEEESTD.1998.88286.

[2] Fowler, M. "Patterns of Enterprise Application Architecture," Addison-Wesley Professional, 2002.

[3] NetBeans IDE 25 Documentation. Apache NetBeans. [Online]. Available: https://netbeans.apache.org/kb/docs/index.html

[4] MySQL 8.0 Reference Manual. Oracle. [Online]. Available: https://dev.mysql.com/doc/refman/8.0/en/

[5] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. "Design Patterns: Elements of Reusable Object-Oriented Software," Addison-Wesley Professional, 1994.