**Final Lab Report**

Course Name: Software Engineering Lab

Course Code: CSC 470

**Submitted To:**

Ferdaus Anam Jibon

Associate Professor, Department of CSE

**Submitted by:**

**Group C**

|  |  |  |
| --- | --- | --- |
| **Serial** | **Name** | **ID** |
| **03** | Ashraful Islam Akash | 22103428 |

|  |
| --- |
|  |
|  |  |

Department of Computer Science and Engineering College of Engineering and Technology

IUBAT–International University of Business Agriculture and Technology

Date of Submission : 24-05-2025

**"Online Shopping Inventory Management System"**

Table of Content

[**Chapter 1** 6](file:///C:\Users\DELL\Downloads\software%20finallabreport%203nd%20draft%20Anni.docx#_Toc198927635)

[**Introduction** 6](file:///C:\Users\DELL\Downloads\software%20finallabreport%203nd%20draft%20Anni.docx#_Toc198927636)

[1.2 **System benefit**: 7](#_Toc198927637)

[**1.3 Software Process model:** 8](#_Toc198927638)

[**Chapter 2** 9](file:///C:\Users\DELL\Downloads\software%20finallabreport%203nd%20draft%20Anni.docx#_Toc198927639)

[**Requirement Engineering** 9](file:///C:\Users\DELL\Downloads\software%20finallabreport%203nd%20draft%20Anni.docx#_Toc198927640)

[**2.1 Requirements Engineering** 10](#_Toc198927641)

[**2.2User requirement:** 10](#_Toc198927642)

[**2.3 System Requirement:** 11](#_Toc198927643)

[2.4 Functional Requirements 13](#_Toc198927644)

[**2.5 Non-Functional Requirements** 14](#_Toc198927645)

[**2.6 Use case diagram:** 16](#_Toc198927646)

[**2.7Use case Text:** 17](#_Toc198927647)

[2.7.1 For User: 17](#_Toc198927648)

[**2.7.2 For Admin:** 18](#_Toc198927649)

[**Chapter 3** 19](file:///C:\Users\DELL\Downloads\software%20finallabreport%203nd%20draft%20Anni.docx#_Toc198927650)

[**Analysis** 19](file:///C:\Users\DELL\Downloads\software%20finallabreport%203nd%20draft%20Anni.docx#_Toc198927651)

[3.1 Analysis Modeling: 20](#_Toc198927652)

[**3.1.2 Activity Diagram:** 25](#_Toc198927653)

[**3.1.3 Swim Lane Diagram:** 26](#_Toc198927654)

[**3.1.4 Sequence diagram:** 27](#_Toc198927655)

[3.1.5 Class Diagram: 28](#_Toc198927656)

[**3.2 Project Management** 29](#_Toc198927657)

[**3.3 Project planning and scheduling** 32](#_Toc198927658)

[**3.4 Project Estimation** 39](#_Toc198927659)

[3.5 System Quality Management for Online Shopping Inventory Management System 41](#_Toc198927660)

[**Chapter 4** 43](file:///C:\Users\DELL\Downloads\software%20finallabreport%203nd%20draft%20Anni.docx#_Toc198927661)

[**Designing** 43](file:///C:\Users\DELL\Downloads\software%20finallabreport%203nd%20draft%20Anni.docx#_Toc198927662)

[4.1 Data Flow Diagram: 44](#_Toc198927663)

[4.1.1Context Level Diagram: 44](#_Toc198927664)

[**4.1.2Level 1 DFD:** 45](#_Toc198927665)

[**4.1.3 Level 2 DFD:** 46](#_Toc198927666)

[4.2 ERD: 47](#_Toc198927667)

[4.3 Database Field Design: 48](#_Toc198927668)

[**Chapter 6** 54](file:///C:\Users\DELL\Downloads\software%20finallabreport%203nd%20draft%20Anni.docx#_Toc198927669)

[**Conclusion** 54](file:///C:\Users\DELL\Downloads\software%20finallabreport%203nd%20draft%20Anni.docx#_Toc198927670)

[**6.1 Conclusion** 55](#_Toc198927671)

[**6.2 Limitation of the Project** 55](#_Toc198927672)

[**Chapter 7** 56](file:///C:\Users\DELL\Downloads\software%20finallabreport%203nd%20draft%20Anni.docx#_Toc198927673)

[**Bibliography** 56](file:///C:\Users\DELL\Downloads\software%20finallabreport%203nd%20draft%20Anni.docx#_Toc198927674)

[**7.1 Book** 57](#_Toc198927675)

[**7.2 Websites** 57](#_Toc198927676)

Table of Figure

**Figure 2.1: Use case diagram………………………………………………………………...16**

**Figure 3.1: Activity Diagram for Inventory………………………………………………...25**

**Figure 3.2: Swim lane Diagram……………………………………………………………...26**

**Figure 3.3: SequenceDiagram………………………………………………………………..27**

**Figure 3.4: ClassDiagram……………………………………………………………………28**

**Figure 3.5 : Effort Distribution……………………………………………………………...37**

**Figure 3.6: Detailed Effort Distribution…………………………………………………….37**

**Figure 3.7: Gantt Chart……………………………………………………………………...38**

**Figure3.8 : Milestone Timeline……………………………………………………………...38**

**Figure 4.1: Context Level Diagram…………………………………………………………44**

**Figure 4.2: Level1 DFD………………………………………………………………………45**

**Figure 4.3: Level2 DFD………………………………………………………………………46**

**Figure 4.4: ERD………………………………………………………………………………47**

List of Table

**Table 3.1: CRC of Admin……………………………………………………………………20**

**Table 3.2: CRC of User………………………………………………………………………21**

**Table 3.3: CRC of Product…………………………………………………………………..21**

**Table 3.4: CRC of Order…………………………………………………………………….22**

**Table 3.5: CRC of Cart……………………………………………………………………...22**

**Table 3.6: CRC of Payment…………………………………………………………………23**

**Table 3.7: CRC of Inventory………………………………………………………………..23**

**Table 3.8: CRC of Notification……………………………………………………………...24**

**Table 3.2.1: Risk identification ……………………………………………………………..29**

**Table 3.2.2: Risk Analysis…………………………………………………………………...30**

**Table 3.2.3: Risk Planning…………………………………………………………………..31**

**Table 3.3.1: Identifying complexity(Transaction function)……………………………….33**

**Table 3.3.2: Identifying complexity(Data function)……………………………………….33**

**Table 3.3.3: Unadjusted function point contribution……………………………………..34**

**Table 3.3.4: Unadjusted function point contribution……………………………………..35**

**Table 3.3.5: Performance and Environmental Impact……………………………………36**

**Table 3.4.1: Draft cost……………………………………………………………………….39**

**Table 3.4.2: Personnel cost………………………………………………………………….39**

**Table 3.4.3: Software Cost………………………………………………………………….40**

**Table 3.4.4: Other Cost……………………………………………………………………..40**

**Table 3.4.5: Accounts table…………………………………………………………………41**

**Table 4.1: Database Table for users……………………………………………………….48**

**Table 4.2: Database Table for Messages…………………………………………………..48**

**Table 4.3: Database Table for Cart………………………………………………………..48**

**Table 4.4: Database Table for Admins…………………………………………………….48**

**Table 4.5: Database Table for products……………………………………………………49**

**Table 4.6: Database Table for Orders……………………………………………………...49**

**Table 4.7: Database Table for Wishlist…………………………………………………….49**1.1 **Objective**:

# **Chapter 1**

# **Introduction**

In the digital age, online shopping has become a major part of everyday life, allowing people to purchase goods and services from the comfort of their homes. Managing a large inventory efficiently is essential for any e-commerce platform to ensure customer satisfaction and smooth business operations.

This project, *Online Shopping &Inventory Management System*, offers a comprehensive solution for managing inventory in an online retail environment. It automates the tracking and updating of product availability, helping both administrators and customers interact with the system effectively.

The system provides an interactive website where administrators can add, update, or remove products, manage stock levels, and monitor sales performance. Customers can browse products by category, view detailed information, add items to their Wishlist, add items to their cart, leave reviews and complete purchases using secure online payment methods.

Once a purchase is made, the system automatically updates the inventory, reflecting real-time stock levels. It also includes features like order history, review or postdelivery support.

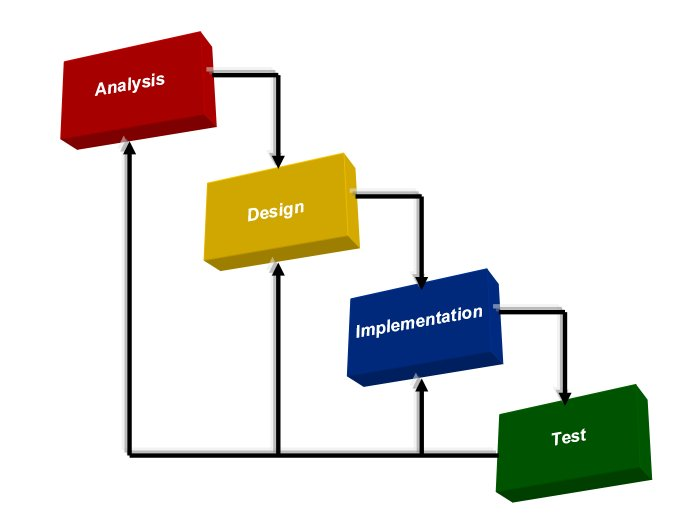
This system simulates a real-world e-commerce platform, offering users a realistic and practical understanding of how online shopping and inventory management work together to create a seamless shopping experience.

* Developing a system that is error free.
* Developing a system that will save time.
* Developing a system which is user-friendly and effective user interface.
* Developing a system that will help to find out the category base system.

# 1.2 **System benefit**:

* Eliminate Human Error
* Faster Transaction
* Accurate Record keeping
* User satisfaction
* Get the idea about a online shopping inventory management system
* A web-based platform for managing products, orders, and inventory in an e-commerce environment.

# **1.3 Software Process model:**

The process model for this software is selected as Waterfall process model.

**1.4 Reasons for Choosing the Waterfall Model?**

**Why we choosen:**

* The project is relatively small in scope.
* The Waterfall model is straightforward, easy to understand, and simple to implement.
* Its process activities are well-defined, clearly separated, and logically organized.
* This model is well-suited for smaller projects where the requirements are stable and well-understood.
* All project requirements have been clearly identified from the outset.
* The underlying technology is familiar and well-understood by the development team.

# **Chapter 2**

# **Requirement Engineering**

**2.1 Requirements Engineering** involves the processes that help determine the business impact of a software system, identify customer needs, and understand how end-users will interact with the software. It is defined by several key activities:

* Gaining a clear understanding of the problems to be solved
* Identifying and evaluating potential solutions
* Specifying a solution that is testable, understandable, maintainable, and aligned with project quality standards

# **2.2User requirement:**

**User:**

1. User will be able to register and log in securely.
2. User will be able to browse and search for products by category.
3. User will be able to view product details, including price, description, and stock availability.
4. User will be able to add products to the shopping cart.
5. User will be able to add products to the shopping Wishlist.
6. User will be able to place orders and make online payments.
7. User will be able to view their order history and review.
8. User will be notified when an order is placed.

**Admin:**

1. Admin will be able to log in securely.
2. Admin will be able to add, update, or delete product listings.
3. Admin will be able to manage product categories.
4. Admin will be able to view and manage customer orders.
5. Admin will be able to view sales reports and inventory levels.
6. Admin will be notified when stock is low or out of stock.
7. Admin will be able to manage user accounts and permissions.

# **2.3 System Requirement:**

**User:**

**1. User will be able to register and log in securely.**  
1.1 User must provide valid registration information (e.g., name, email, password).  
1.2 User must verify their email before logging in (optional feature).  
1.3 Login requires a valid email and password combination.  
1.4 System will display an error message for incorrect login credentials.

**2. User will be able to browse and search for products by category.**  
2.1 Products are organized into categories (e.g., Electronics, Clothing, etc.).  
2.2 User can search for products by name or keyword.  
2.3 User can filter products by price, rating, and availability.

**3. User will be able to view product details.**  
3.1 Product details include name, description, price, availability, and images.  
3.2 User can see product reviews and ratings.  
3.3 System displays an “Out of Stock” message if the product is unavailable.

**4. User will be able to add products to the shopping cart.**  
4.1 User can specify quantity before adding to cart.  
4.2 User must be logged in to add items to the cart.  
4.3 Cart updates total price dynamically based on quantity and product price.

**5. User will be able to add products to the shopping Wishlist.**  
5.1 Wishlist is user-specific and stored in the account.  
5.2 User must be logged in to add or view Wishlist items.  
5.3 User can remove items from the Wishlist anytime.

**6. User will be able to place orders and make online payments.**  
6.1 User must be logged in to place an order.  
6.2 User can choose from multiple payment methods (e.g., credit card, UPI, wallet).  
6.3 Order confirmation and receipt will be sent via email.  
6.4 Payment transactions are secured using encryption protocols.

**7. User will be notified of key activities.**  
8.1 Notification for successful order placement.  
8.2 Notification for shipping and delivery updates.  
8.3 Notification center allows users to read and delete notifications.

**Admin:**

**1. Admin will be able to log in securely.**  
1.1 Admin must be an authorized user.  
1.2 Admin login requires a username and password.  
1.3 System logs admin login times for audit purposes.

**2. Admin will be able to manage product listings.**  
2.1 Admin can add new products with full details.  
2.2 Admin can update existing product information.  
2.3 Admin can delete products from the catalog.  
2.4 Admin can set product visibility (e.g., Active/Inactive).

**3. Admin will be able to manage product categories.**  
3.1 Admin can create new categories.  
3.2 Admin can edit or delete existing categories.  
3.3 Admin can assign products to specific categories.

**4. Admin will be able to view and manage customer orders.**  
4.1 Admin can view order details for all customers.  
4.2 Admin can update order statuses (e.g., mark as shipped).  
4.3 Admin can cancel or refund orders if necessary.

**5. Admin will be notified of inventory and order updates.**  
6.1 Notification when product stock is low.  
6.2 Notification when a new order is placed.  
6.3 Notification when a product is out of stock.  
6.4 Admin can delete or archive notifications.

**6. Admin will be able to manage user accounts.**  
7.1 Admin can view user profiles and order histories.  
7.2 Admin can deactivate or delete user accounts.  
7.3 Admin can reset passwords or modify user permissions.

# 2.4 Functional Requirements

**User:**

1. Register account
2. Login
3. Browse products by category
4. Search products
5. View product details
6. Add products to shopping cart
7. Add products to Wishlist
8. Place orders
9. Make online payments
10. View order history
11. Receive notifications (order placed, shipped, delivered)
12. Logout

**Admin:**

1. Login
2. Add new product listings
3. Update product details
4. Delete product listings
5. Manage product categories (add/edit/delete)
6. View and manage customer orders
7. Update order statuses
8. View sales reports
9. Monitor inventory levels
10. Receive notifications (low stock, new order)
11. Manage user accounts (view, deactivate, delete)
12. Logout

# **2.5 Non-Functional Requirements**

**Performance Requirements:**

1. Product listings should load within 2 seconds per page (max 10 products per page).
2. Search results should be displayed within 3 seconds.
3. Order confirmation and payment processing should be completed within 5 seconds.

**Security:**

1. Each user must have a unique account secured by a password.
2. Passwords must be stored securely using encryption or hashing.
3. All user sessions must use secure protocols (e.g., HTTPS).
4. Users must be authenticated before accessing sensitive features (e.g., checkout, order history).
5. Admin access requires higher-level authentication and authorization.

**Usability:**

1. The system shall have an intuitive and user-friendly interface.
2. The website shall be responsive and accessible on multiple devices (desktop, tablet, mobile).

**Reliability:**

1. The system shall maintain at least 99.5% uptime.
2. Data backups must be performed regularly to prevent data loss.

**Maintainability:**

1. The system shall be designed to allow easy updates to product listings and categories.
2. The codebase shall follow best practices to facilitate troubleshooting and future enhancements.

# **2.6 Use case diagram:**



Figure 2.1: Use case diagram

# **2.7 Use case Text:**

# 2.7.1 For User:

**Use Case Title:** Register Account  
**Actor:** User  
**Description:** User can create a new account by providing valid personal and login information.

**Use Case Title:** Search Products  
**Actor:** User  
**Description:** User can search for specific products using keywords.

**Use Case Title:** View Product Details  
**Actor:** User  
**Description:** User can view detailed information of any selected product.

**Use Case Title:** Login  
**Actor:** User  
**Description:** User can log into the system using a valid email and password.

**Use Case Title:** Add to Cart  
**Actor:** User  
**Description:** User can add products to their shopping cart with specified quantities.

**Use Case Title:** Browse Products  
**Actor:** User  
**Description:** User can view available products categorized by type.

**Use Case Title:** Place Order  
**Actor:** User  
**Description:** User can place an order for products added to the cart.

**Use Case Title:** Add to Wishlist  
**Actor:** User  
**Description:** User can save products to their Wishlist for future purchases.

**Use Case Title:** View Order History  
**Actor:** User  
**Description:** User can view a list of their past orders.

**Use Case Title:** Make Payment  
**Actor:** User  
**Description:** User can complete payment through available online payment methods.

**Use Case Title:** Logout  
**Actor:** User  
**Description:** User can log out of their account securely.

## **2.7.2 For Admin:**

**Use Case Title:** Update Order Status  
**Actor:** Admin  
**Description:** Admin can change the status of orders (e.g., Shipped, Delivered).

**Use Case Title:** Login  
**Actor:** Admin  
**Description:** Admin can log into the system with authorized credentials.

**Use Case Title:** View Reports  
**Actor:** Admin  
**Description:** Admin can access and review sales and performance reports.

**Use Case Title:** Add Product  
**Actor:** Admin  
**Description:** Admin can add new products to the inventory

**Use Case Title:** Monitor Inventory  
**Actor:** Admin  
**Description:** Admin can track stock levels and be alerted when inventory is low.

**Use Case Title:** Update Product  
**Actor:** Admin  
**Description:** Admin can update the details of existing products.

# 3.1 Analysis Modeling:

**Use Case Title:** Logout  
**Actor:** Admin  
**Description:** Admin can log out of the system securely.

**Use Case Title:** View Orders  
**Actor:** Admin  
**Description:** Admin can view all customer orders placed through the system.

**Use Case Title:** Receive Notifications  
**Actor:** Admin  
**Description:** Admin receives alerts for new orders and low stock notifications.

**Use Case Title:** Manage Categories  
**Actor:** Admin  
**Description:** Admin can add, edit, or delete product categories.

**Use Case Title:** Delete Product  
**Actor:** Admin  
**Description:** Admin can remove products from the inventory.

**Use Case Title:** Manage User Accounts  
**Actor:** Admin  
**Description:** Admin can view, update, deactivate, or delete customer accounts.

# **Chapter 3**

# **Analysis**

The online shopping inventory management system serves as a powerful tool for streamlining product, order, and user management. While textual descriptions provide valuable insights, they are not always the most effective way to represent software requirements. Therefore, analysis modeling combines descriptive text with diagrams to illustrate data flow, system functions, and user behaviors in a clear and structured manner. This approach enhances understanding and enables easier validation of the system’s completeness, correctness, and consistency.

In the following sections, various use cases and system scenarios will be illustrated through conventional and object-oriented analysis (OOA) techniques, including the use of Unified Modeling Language (UML), to provide a comprehensive view of the system’s functionality and interactions.

**3.1.1 CRC:**

**3.1.1.1 CRC for Admin**

|  |  |
| --- | --- |
| **Class Name:** Admin | |
| **Class Type:** Role | |
| **Class characteristic:** Tangible, Atomic, Sequential, Persistent, Guarded | |
| **Responsibilities:**   Login   Add, update, delete product   Manage categories   View and manage orders   Monitor inventory   Manage user accounts   Receive system notifications | **Collaborations:**   Product   Category   Order   Inventory   User   Notification |

Table 3.1: CRC of Admin

**3.1.1.2 CRC for User**

|  |  |
| --- | --- |
| **Class Name:** User | |
| **Class Type:** Role | |
| **Class characteristic:** Tangible, Sequential, Persistent, Guarded | |
| **Responsibilities:**   Register account   Login   View and update profile   Browse and search products   Add products to cart/Wishlist   Place order and make payment   View order history and track order   Receive notifications | **Collaborations:**   Product   Cart   Payment  Order   Notification |

Table 3.2: CRC of User

**3.1.1.3 CRC for Product**

|  |  |
| --- | --- |
| **Class Name:**Product | |
| **Class Type:**Entity | |
| **Class characteristic:**Persistent, Indexed, Categorized | |
| **Responsibilities:**   Store product details (name, price, stock, description)   Provide details for browsing, search, and order   Update stock levels after order | **Collaborations:**   Admin   Order   Inventory   Category |

Table 3.3: CRC of Product

**3.1.1.4 CRC for Order**

|  |  |
| --- | --- |
| **Class Name:**Order | |
| **Class Type:**Entity | |
| **Class characteristic:**Sequential, Persistent, Trackable | |
| **Responsibilities:**   Record user orders   Track order status (placed, shipped, delivered)   Link payments to specific orders   Notify user on status changes | **Collaborations:**   User   Product   Payment   Notification | |

Table 3.4: CRC of Order

**3.1.1.5 CRC for Cart**

|  |  |
| --- | --- |
| **Class Name:**Cart | |
| **Class Type:**Entity | |
| **Class characteristic:**Temporary, User-specific | |
| **Responsibilities:**   Hold selected products until checkout   Update quantities or remove products   Calculate subtotal before placing order | **Collaborations:**   User   Product | |

Table 3.5: CRC of Cart

**3.1.1.6 CRC for Payment**

|  |  |
| --- | --- |
| **Class Name:**Payment | |
| **Class Type:**Properties | |
| **Class characteristic:**Secure, Guarded, Transactional | |
| **Responsibilities:**   Process user payments securely   Validate transaction success   Link payment to order | **Collaborations:**   User   Order | |

Table 3.6: CRC of Payment

**3.1.1.7 CRC for Inventory**

|  |  |
| --- | --- |
| **Class Name:**Inventory | |
| **Class Type:**Properties | |
| **Class characteristic:**Persistent, Monitored | |
| **Responsibilities:**   Track product stock levels   Notify admin on low stock | **Collaborations:**   Admin   Product | |

Table 3.7: CRC of Inventory

**3.1.1.8 CRC for Notification**

|  |  |
| --- | --- |
| **Class Name:**Notification | |
| **Class Type:**Entity | |
| **Class characteristic:**Event-driven, Transient | |
| **Responsibilities:**   Notify users/admin of relevant events (order status, low stock, etc.)   Allow users to delete notifications | **Collaborations:**   User   Admin   Order   Inventory | |

Table 3.8: CRC of Notification **3.1.2 Activity Diagram:**

**Activity Diagram for Inventory:**

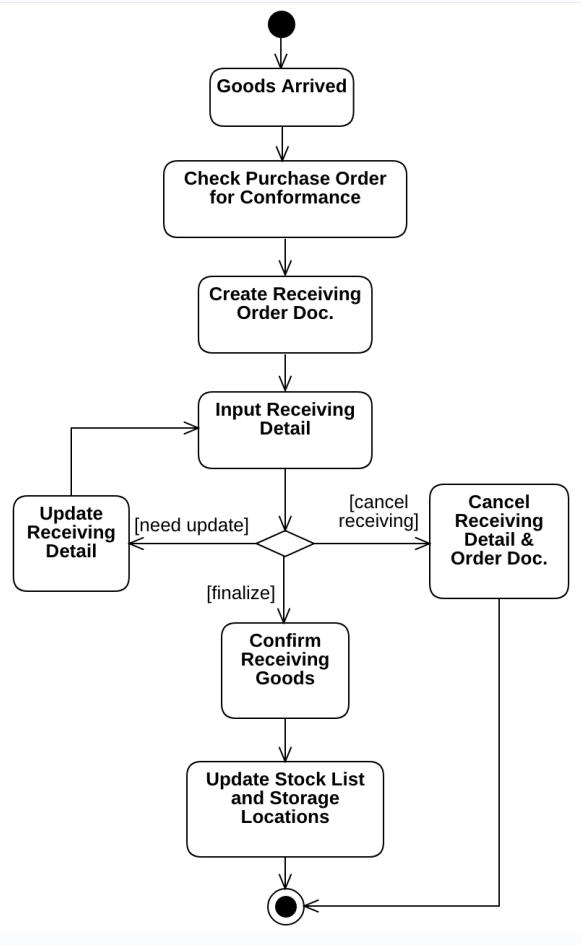
****

Figure 3.1: Activity Diagram for Inventory

# **3.1.3 Swim Lane Diagram:**

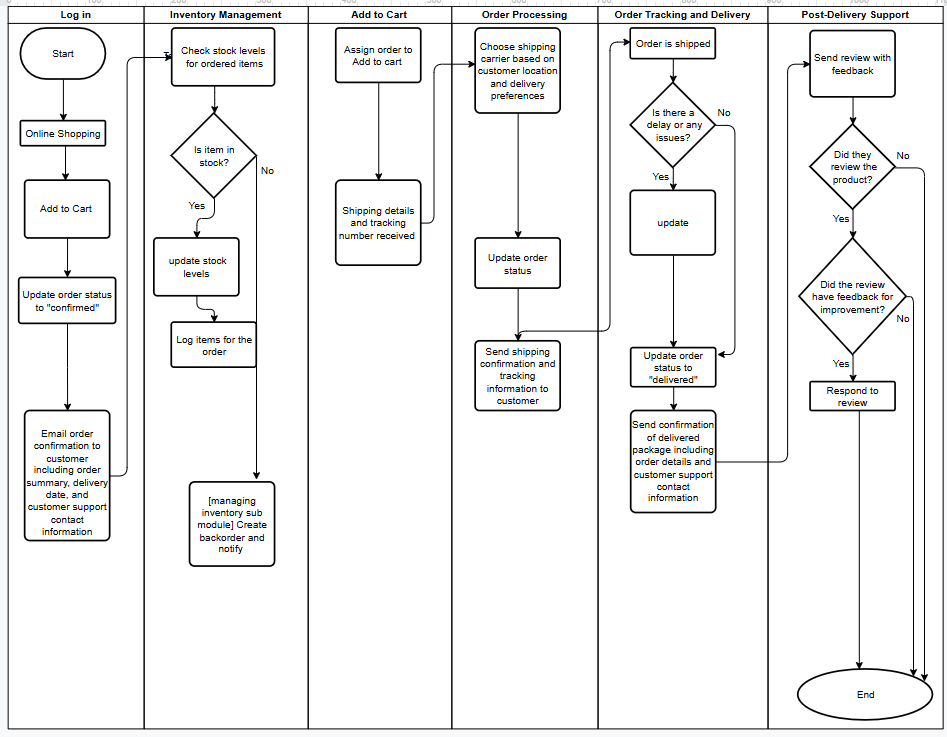
****

Figure 3.2: Swim lane Diagram

# **3.1.4 Sequence diagram:**

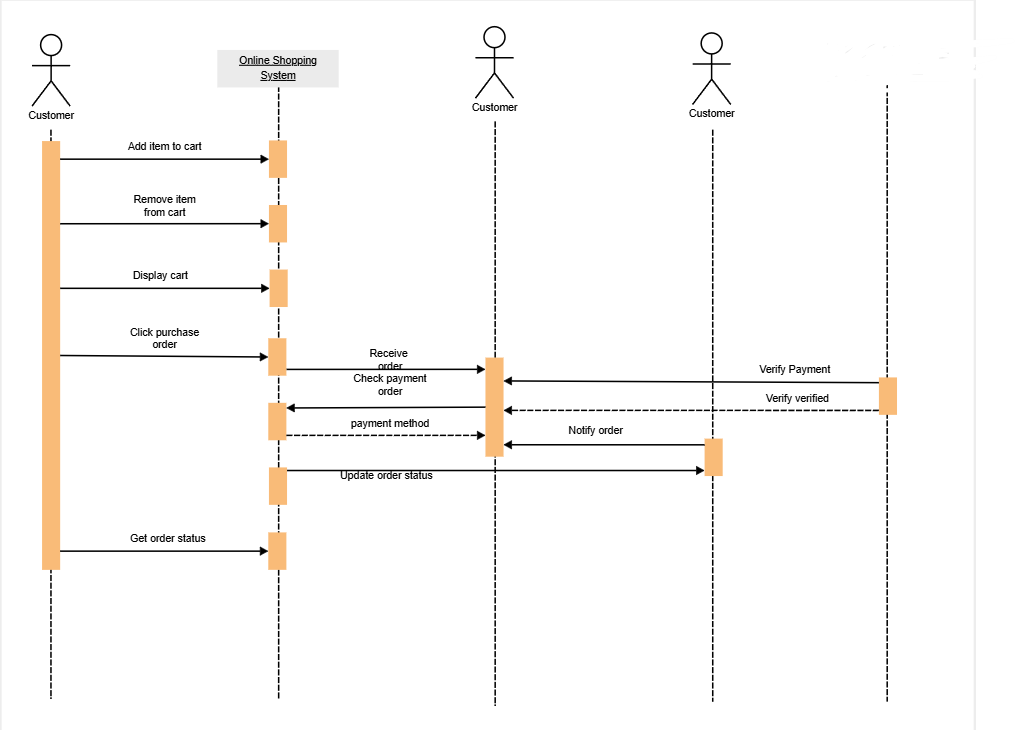


Figure 3.3: SequenceDiagram

# 3.1.5 Class Diagram:

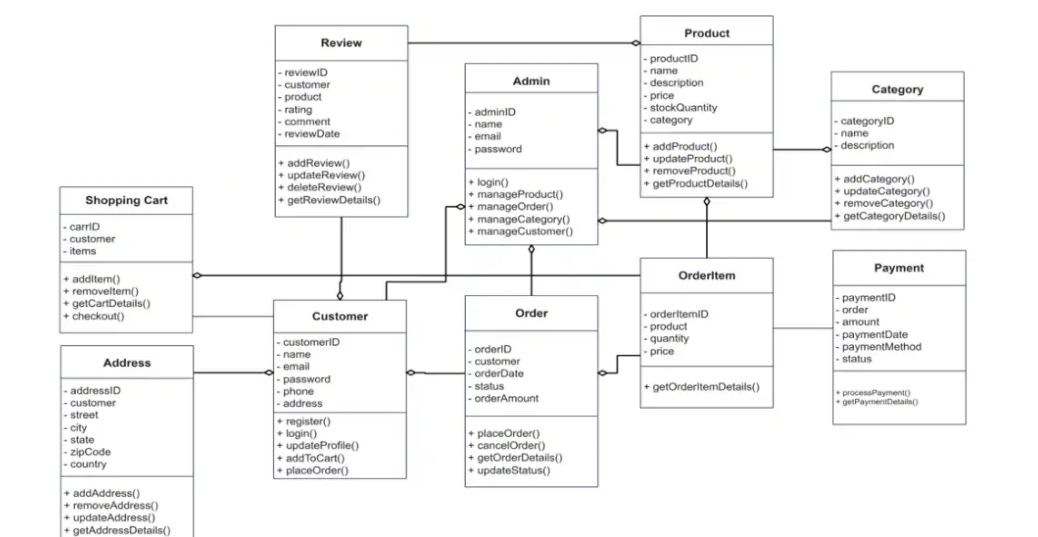


Figure 3.4: ClassDiagram

# **3.2 Project Management**

Software project management is the art and science of planning and leading software projects. It is a sub-discipline of project management in which software projects are planned, monitored and controlled.

**3.2.1 Risk identification:**

|  |  |
| --- | --- |
| **Risk type** | **Possible risks** |
| 1. Technology | 1.1Security of the system  1.2Reusable software components contain defects that mean they cannot be reused as planned. |
| 2. People | 2.1Key staff is ill and unavailable at critical times.  2.2 Required training for staff is not available. |
| 3. organizational | 3.1Organizational financial problems force reductions in the project budget. |
| 4. Requirement | 4.1Changes to requirements that require major design rework are proposed.  4.2 Customers fail to understand the impact of requirements changes. |
| 5. Estimation | 5.1 The time required to develop the software is underestimated.  5.2 The rate of defect repair is underestimated. |

Table 3.2.1: Risk identification

**3.2.2 Risk Analysis:**

|  |  |  |
| --- | --- | --- |
| **Risk** | **Probability** | **Effects** |
| Security of the system(1.1) | High | Serious |
| Reusable software components contain defects that mean they cannot be reused as planned.(1.2) | Moderate | Serious |
| Key staff are ill and unavailable at critical times.(2.1) | Moderate | serious |
| Required training for staff is not available.(2.2) | Moderate | Tolerable |
| Organizational financial problems force reductions in the project budget.(3.1) | Low | Catastrophic |
| Changes to requirements that require major design rework are proposed. (4.1) | Moderate | Serious |
| Customers fail to understand the impact of requirements changes. (4.2) | Moderate | Tolerable |
| The time required to develop the software is underestimated.(5.1) | High | Serious |
| The rate of defect repair is underestimated. (5.2) | Moderate | Tolerable |

Table 3.2.2: Risk Analysis

**3.2.3 Risk Planning:**

|  |  |
| --- | --- |
| **Risk** | **Strategy** |
| Security | Investigate the possible security leaks and measurements |
| Organizational financial  Problems | Prepare briefing documents for senior management showing how the project is making a very important contribution to the goals of business and presenting reasons why cuts to the project budget would not be cost-effective. |
| Requirements problem | Alerts customer to potential difficulties and possibility of delays; investigate buying in component. |
| Staff illness | Reorganize them so that there is more overlap work and people therefore understand each other jobs. |
| Defective component | Replace defective potential component with bought in component of know reliability. |
| Requirements changes | Derive traceability information to access requirements change impact; maximizing information hiding in the design. |
| Underestimated development time | Investigate buying in component; investigate use of a program generator. |

Table 3.2.3: Risk Planning

**3.2.4 Risk monitoring:**

* A re-planning of the project occurs. New task schedule and milestones are defined. Staffs work on their assigned jobs within the new timeframe.
* In order to prevent this from happening, the software will develop with the end user in mind.
* The user-interface will design in a way to make use of the program convenient and pleasurable.
* Meetings (formal and informal) will be held with the stakeholders regularly. This insures that the product we are producing solves a problem.
* The development cost of the software may increase by 20%.Consult with the System Analyst during the system analysis, design and testing phase of the software project
* Proper coding grammar is followed to make sure that the codes are easily understandable and reusable.
* Cost and Time will increase. Project will be modified. Everything will be at where it all started.

**3.3 Project planning and scheduling**

**3.3.1 Function Point Estimation:**

**3.3.1.1 Identifying complexity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Transition function** | **Fields/ file involve** | **FTRs** | **DETs** |
| 1. Register Account(EI-External Input) | **Fields –** Name, Email, Password, Confirm Password, Submit  **File Name -** Users | 1 | 6 |
| 2. Login(EI) | **Fields -** Email, Password, Login Button  **File Name -** Users | 1 | 4 |
| 3. Add to Cart(EI) | **Fields –** Product ID, Quantity, Price, User ID  **File Name –** Cart, Products, Users | 3 | 4 |
| 4. Place Order(EI) | **Fields –** Address, Payment Method, Total, User ID, Submit  **File Name –** Orders, Cart, Users | 3 | 5 |
| 5. Logout(EI) | **Fields –** Logout Trigger  **File Name –** Users | 1 | 1 |
| 6. Make Payment(EO-External Output) | **Fields-** Card Number, Expiry, CVV, Amount  **File Name –** Orders, Payment Gateway | 3 | 4 |
| 7. Notifications(EO) | **Fields-** Message Text, Notification Type, Timestamp  **File Name –**Users, Orders | 2 | 3 |
| 8. View Order History(EO) | **Fields-** Order ID, Product Names, Dates, Status  **File Name –** Users, Orders | 2 | 4 |
| 9. Product Recommendations(EO) | **Fields-** Suggested Products, Product ID, Name, Image  **File Name –** Products, Orders, Wishlist | 3 | 4 |
| 10. Search Products(EQ-External Query) | **Fields-** Search Text, Filters (category, price), Submit  **File Name –** Products | 1 | 4 |
| 11. Track Order(EQ) | **Fields-** Order ID, Tracking ID, Status  **File Name –**Orders | 1 | 3 |
| 12. Browse Products(EQ) | **Fields-** Product List, Images, Price, Pagination  **File Name –** Products, Categories | 2 | 4 |

Table 3.3.1: Identifying complexity(Transaction function)

|  |  |  |  |
| --- | --- | --- | --- |
| **Data function** | **Fields/File involve** | **RETs** | **DETs** |
| 1. Users(ILF-Internal Logical File) | **Fields-** ID, Name, Email, Password | 1 | 4 |
| 2. Products(ILF) | **Fields-** ID, Name, Price, Description, Image\_01, Image\_02, Image\_03 | 1 | 7 |
| 3. Orders(ILF) | **Fields-** ID, User ID, Address, Total, Method, Payment Status, Date, Products | 1 | 8 |
| 4. Wishlist(ILF) | **Fields-** ID, User ID, Product ID, Product Name, Price, Image | 1 | 6 |
| 5. Payment Gateway APIs(EIF-External Interface File) | **Fields-** Transaction ID, Amount, Date, Status | 1 | 4 |
| **6.** Shipping Integration APIs(EIF) | **Fields-** Tracking Number, Status, Estimated Delivery Date | 1 | 3 |

Table 3.3.2: Identifying complexity(Data function)

**3.3.1.2 Unadjusted function point contribution**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Transition function** | **FTRs** | **DETs** | **Complexity** | **UFP** |
| 1. Register Account(EI-External Input) | 1 | 6 | Low | 3 |
| 2. Login(EI) | 1 | 4 | Low | 3 |
| 3. Add to Cart(EI) | 3 | 4 | Avg | 4 |
| 4. Place Order(EI) | 3 | 5 | High | 6 |
| 5. Logout(EI) | 1 | 1 | low | 3 |
| 6. Make Payment(EO-External Output) | 3 | 4 | Low | 4 |
| 7. Notifications(EO) | 2 | 3 | Low | 4 |
| 8. View Order History(EO) | 2 | 4 | Low | 4 |
| 9. Product Recommendations(EO) | 3 | 4 | Low | 4 |
| 10. Search Products(EQ-External Query) | 1 | 4 | Low | 3 |
| 11. Track Order(EQ) | 1 | 3 | Low | 3 |
| 12. Browse Products(EQ) | 2 | 4 | Low | 3 |
| Total |  | | | 44 |

Table 3.3.3: Unadjusted function point contribution

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data function** | **RETs** | **DETs** | **Complexity** | | **UFP** |
| 1. Users(ILF-Internal Logical File) | 1 | 4 | Low | | 7 |
| 2. Products(ILF) | 1 | 7 | Low | | 7 |
| 3. Orders(ILF) | 1 | 8 | Low | | 7 |
| 4. Wishlist(ILF) | 1 | 6 | Low | | 7 |
| 5. Payment Gateway APIs(EIF-External Interface File) | 1 | 4 | Low | | 5 |
| **6.** Shipping Integration APIs(EIF) | 1 | 3 | Low | | **5** |
| Total |  | | | 38 | |

Table 3.3.4: Unadjusted function point contribution

**3.3.1.3 Performance and Environmental Impact**



Table 3.3.5: Performance and Environmental Impact



**= (82\*1) = 82**

**= (38+44) = 82**

**= 82\*15.5**

**= 1271 person hours / 6 hours**

**=211.833 person days / 16 days**

**=13.24 person months / 5 persons**

**=2.65 months for 5 persons**

**Approximately 3 Months required for 5 persons to finish the project**

**3.3.1.4 Effort Distribution:**

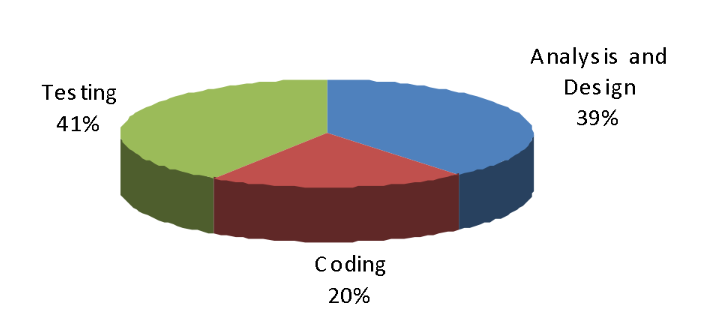
****

Figure 3.5 : Effort Distribution

**3.3.1.5 Detailed Effort Distribution**

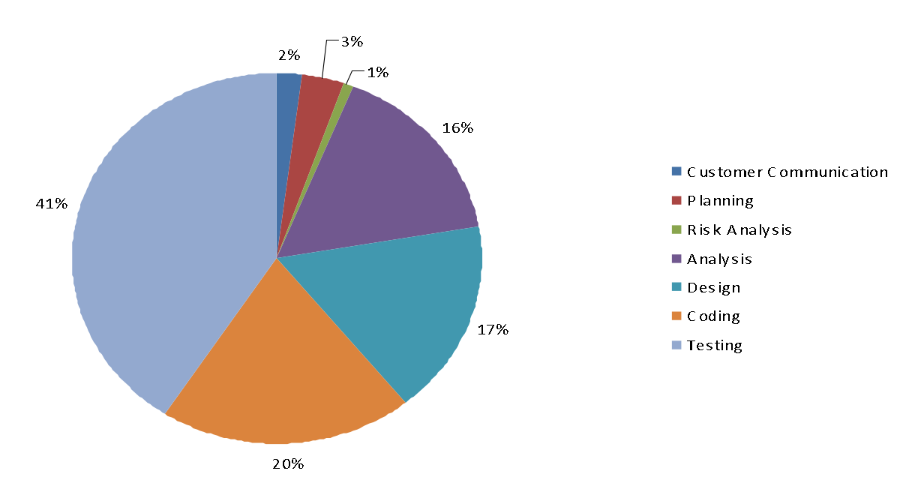
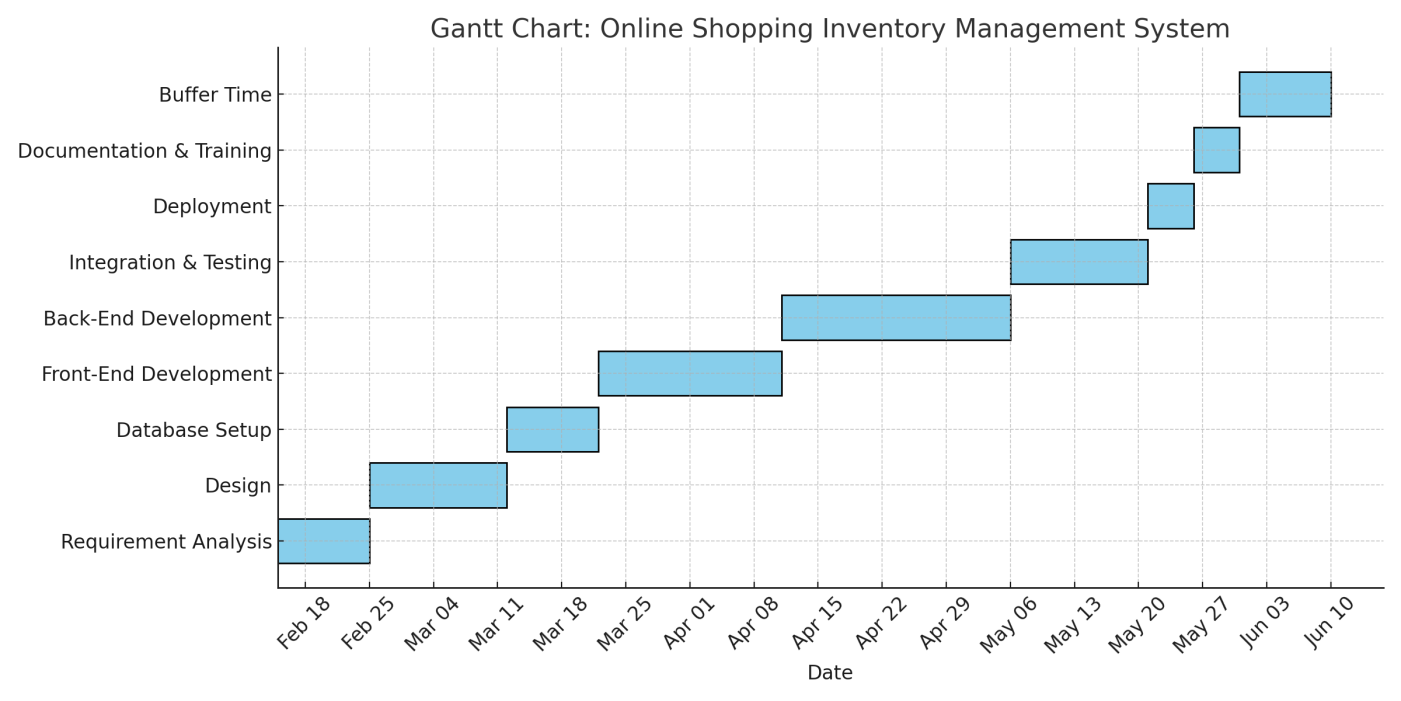
****

Figure 3.6: Detailed Effort Distribution

**3.3.1.6 Project Gantt Chart**

****

May 12 May 18 May22

Figure 3.7: Gantt Chart

**3.3.1.7 Project Milestone Timeline:**

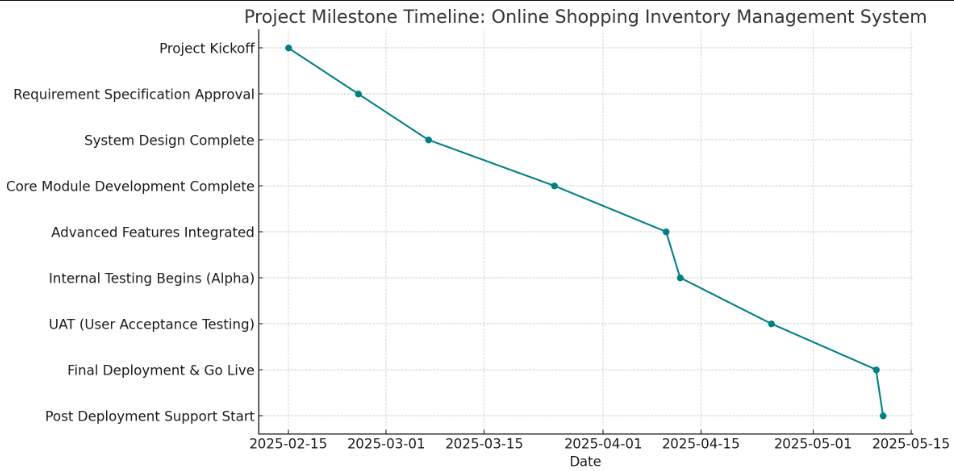
****

Figure3.8 : Milestone Timeline

**3.4 Project Estimation**

**3.4.1 Personnel cost**

Number of days in a year = 365

Number of government holidays in a year =42

Number of weekly holidays in a year =131

Total number of working days to develop the project =365-(42+131) =192 days

Total number of working days per months to develop the project =192/12 =16 days

Organization working hours per day = 6 hours

Organization working hours per month=16\*6= 96 hours

**Budget Estimation**

|  |  |
| --- | --- |
| **Component** | **Estimated Cost (USD)** |
| Personnel | $20,000 |
| Hosting Services | $1,500 |
| Tools and Licenses | $1,000 |
| Miscellaneous | $500 |
| **Total** | **$23,000** |

Table 3.4.1: Draft cost

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **No. of Members** | **Months** | **Salary** |
| System Analyst | 1 | 1 | 20,000.00 |
| Senior Developer | 1 | 1 | 15,000.00 |
| Web/Graphics Designer | 1 | 1 | 15,000.00 |
| Customer Communicator | 1 | 1 | 10,000.00 |
| Tester | 1 | 1 | 10,000.00 |
| **Total** | | | **70,000.00** |

Table 3.4.2: Personnel cost

**3.4.2 Hardware cost:**

The percentage of year is: 1/15 = 6.67% = 0.0667

The depreciation cost of Computer is = (50,000 \* 0.0667) = 3,335.00

The depreciation cost of Scanner is = (2000 \* 0.0667) = 134.40

The depreciation cost of Printer is = (4000 \* 0.0667) = 266.80

**3.4.3 Software Cost**

|  |  |
| --- | --- |
| Microsoft Office 2010 | 80.00 |
| Microsoft Windows 7 | 100.00 |
| Xampp Sarver | 55.00 |

Table 3.4.3: Software Cost

**3.4.4 Other Cost**

|  |  |
| --- | --- |
| Furniture | 6,000.00 |
| Electricity bill | 600.00 |
| House Rent | 4,000.00 |
| Vehicle Rent | 500.00 |
| Extra | 1,000.00 |

Table 3.4.4: Other Cost

**3.4.5 Accounts table**

|  |  |
| --- | --- |
| Particulars | TK |
| Salary-   * System Analyst * Senior Developer * Web/Graphics Designer * Risk Analyzer * Customer communication * Tester | 20,000.00  15,000.00  15,000.00  10,000.00  10,000.00 |
|  | 70,000.00 /= |
| Hardware Cost –   * Computer * Scanner * Printer | 3,335.00  134.40  266.80 |
|  | 3,736.20 /= |
| Other Costs-   * Furniture * House Rent * Electricity bill * Vehicle Rent * Extra | 6,000.00  4,000.00  600.00  500.00  1,000.00 |
|  | 12,100.00 /= |
| **Total cost** | **85,836.20 /=** |

Table 3.4.5: Accounts table

# 3.5 System Quality Management for Online Shopping Inventory Management System

Agreement on quality requirements and clear communication of what constitutes “quality” are essential for ensuring that the final product meets user expectations. In the context of the Online Shopping Inventory Management System, understanding and formally defining the various aspects of software quality are critical.

The quality characteristics of the system—such as reliability, usability, security, and performance—must be explicitly stated in the software requirements. These characteristics serve as the foundation for defining measurable quality metrics and setting the acceptance criteria for validating the product.

**SQA Methods Used**

* Peer code reviews
* CI pipeline via GitHub Actions
* ISO 9126 alignment for usability and maintainability

**Testing Strategies**

* **Unit Testing:** Jest (frontend), Mocha (backend)
* **Integration Testing:** Postman, Super test
* **System Testing:** Full process validation (e.g., add → purchase → report)
* **Acceptance Testing:** Final approval from end-users
* **Regression Testing:** Automated CI-based

**Justification for Testing Methods**

* High modularity → Easy unit tests
* Automation → Efficient iteration
* Regression tests → Reliable updates
* CI pipelines → Continuous delivery confidence

**Key Software Quality Management Processes Applied:**

* **Quality Assurance Process:** Ensures that all activities involved in system development and maintenance are performed with the objective of delivering a quality product.
* **Verification Process**: Confirms that the system correctly implements specific functions and logic as per the design specifications.
* **Validation Process:** Validates that the system fulfills the user requirements and meets the expectations in a real-world environment.
* **Review Process**: Includes regular code, design, and requirement reviews to detect and correct defects early.
* **Audit Process**: Periodic evaluations are conducted to ensure adherence to quality standards and development protocols.

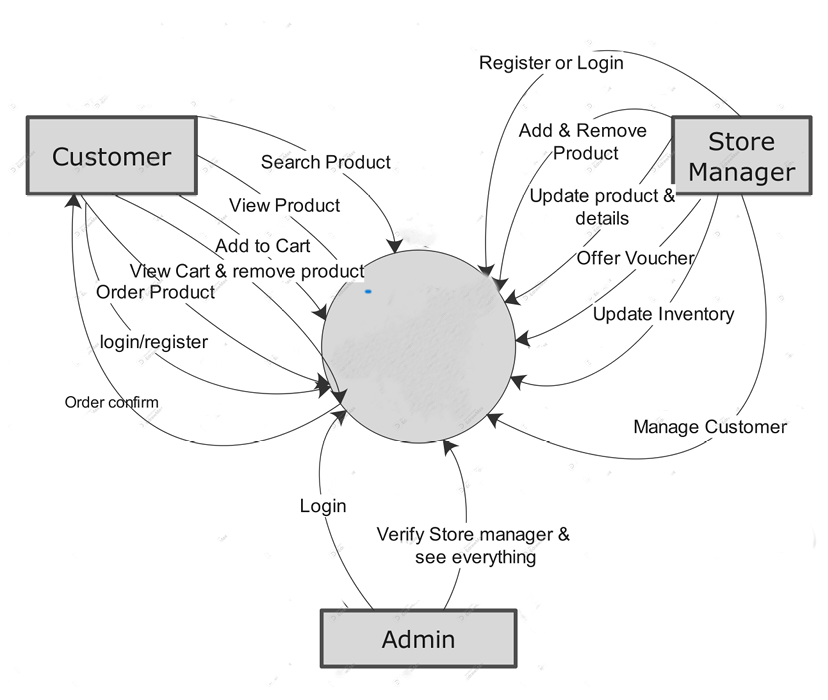
By adhering to these structured Software Quality Management practices, the Online Shopping Inventory Management System aims to deliver a reliable, maintainable, and user-centric software product.

4.1 Data Flow Diagram:

# **Chapter 4**

# **Designing**

# 4.1.1Context Level Diagram:

****

Online shop & inventory management system

Figure 4.1: Context Level Diagram

# **4.1.2Level 1 DFD:**

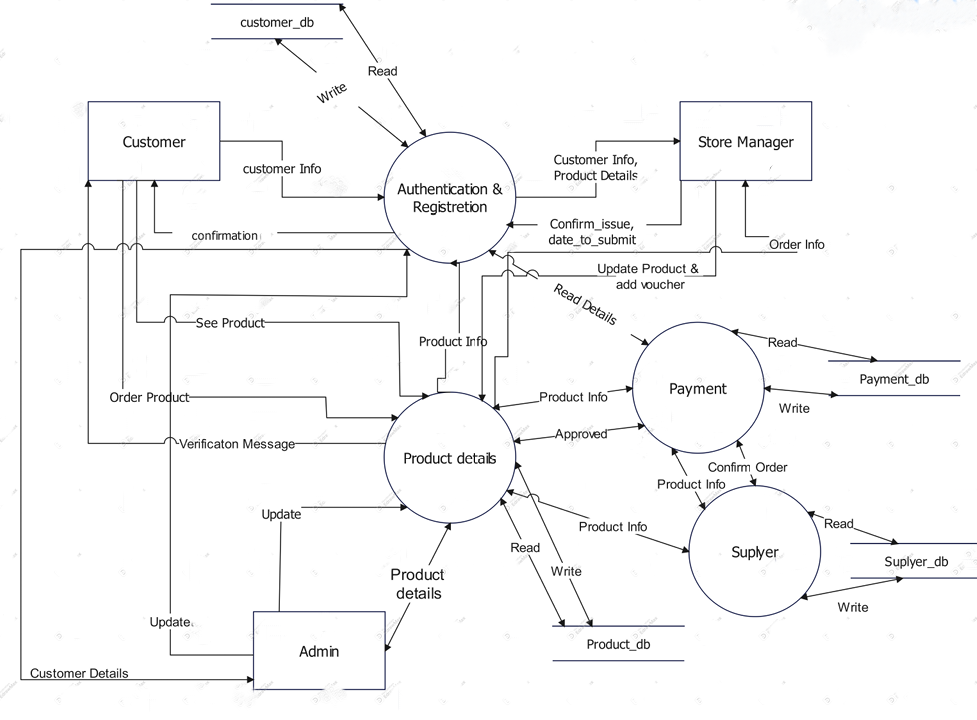
****

Figure 4.2: Level1 DFD

# **4.1.3 Level 2 DFD:**



Figure 4.3: Level2 DFD

# 4.2 ERD:

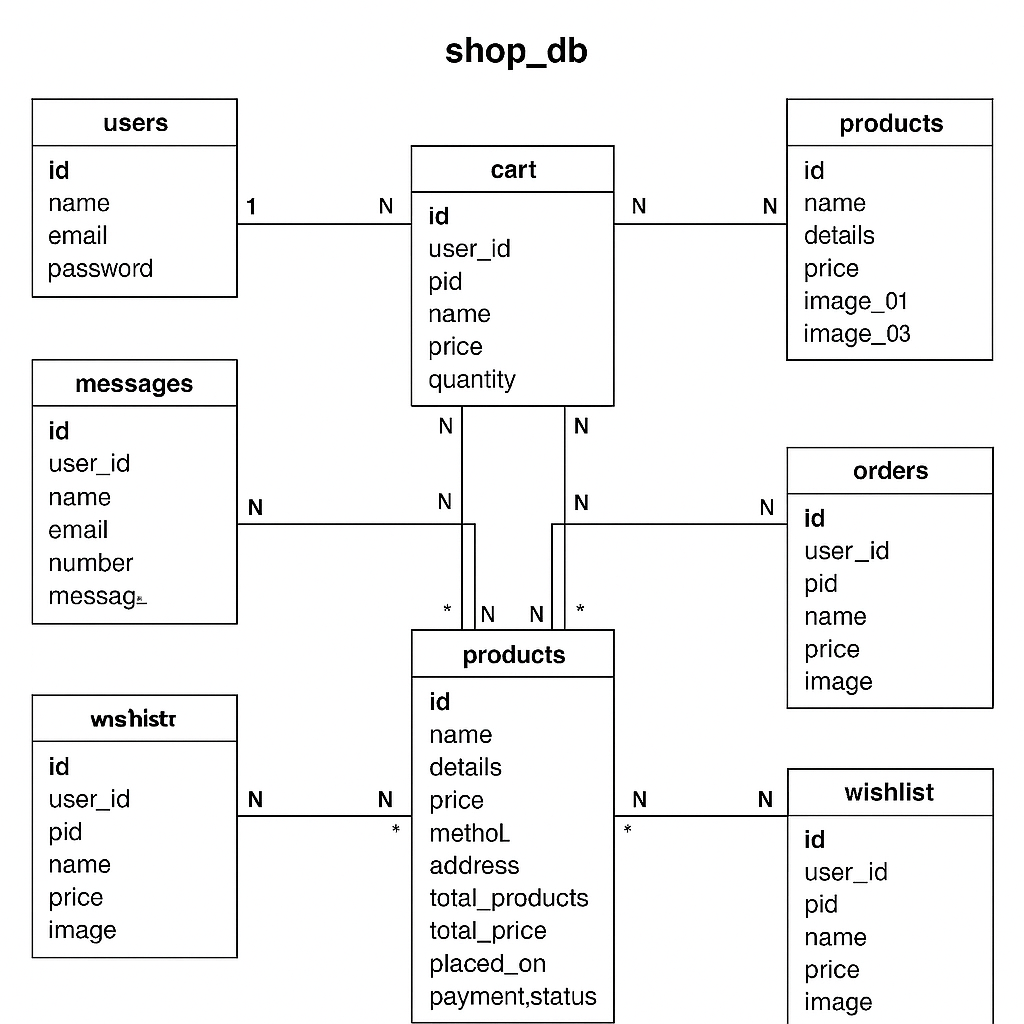


Figure 4.4 : ERD

# 4.3 Database Field Design:

**4.3.1 Database Table for users:**

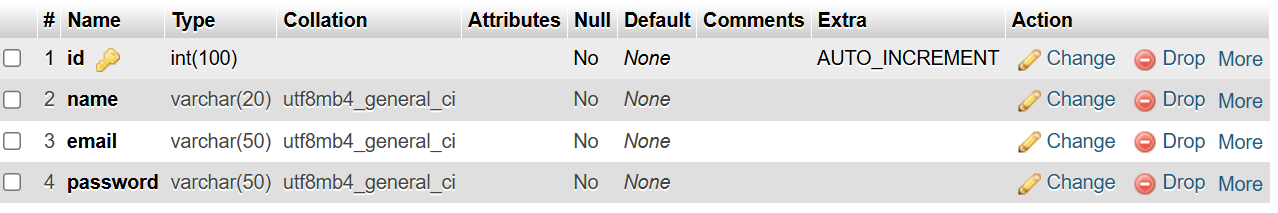
****

Table 4.1: Database Table for users

**4.3.2 Database Table for Messages:**

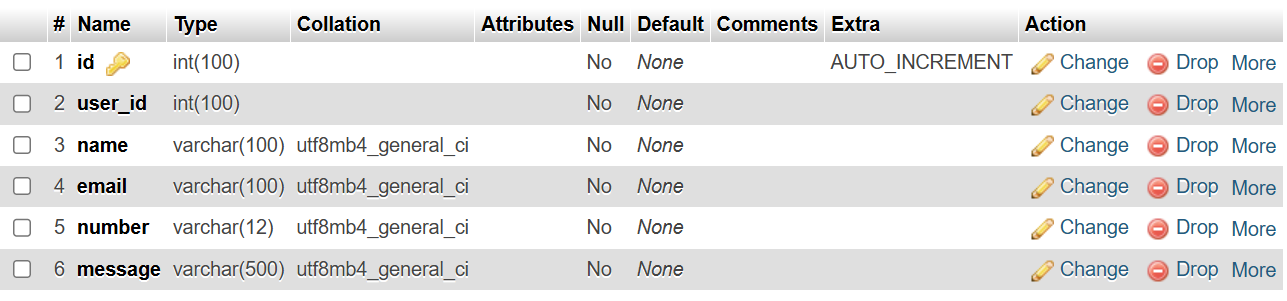
****

Table 4.2: Database Table for Messages

**4.3.3 Database Table for Cart:**

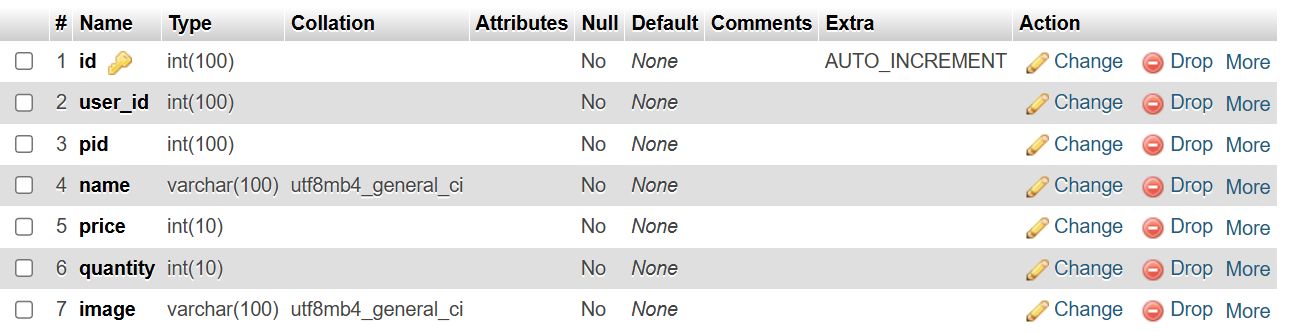


Table 4.3: Database Table for Cart

**4.3.4 Database Table for Admins:**

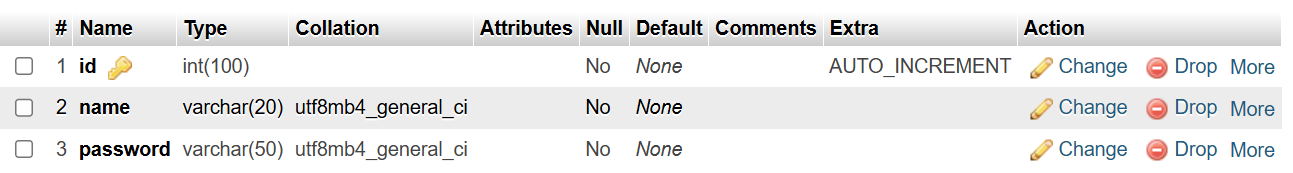


Table 4.4: Database Table for Admins

**4.3.5Database Table for Products:**

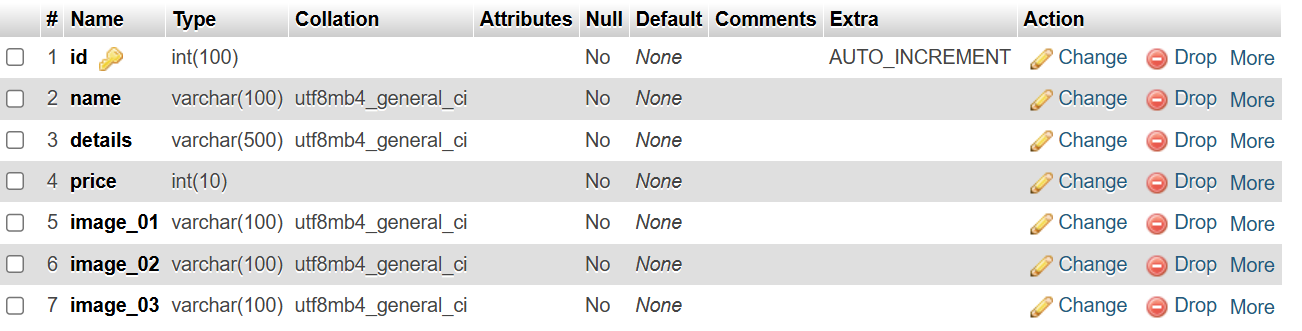
****

Table 4.5: Database Table for products

**4.3.6Database Table for Orders:**

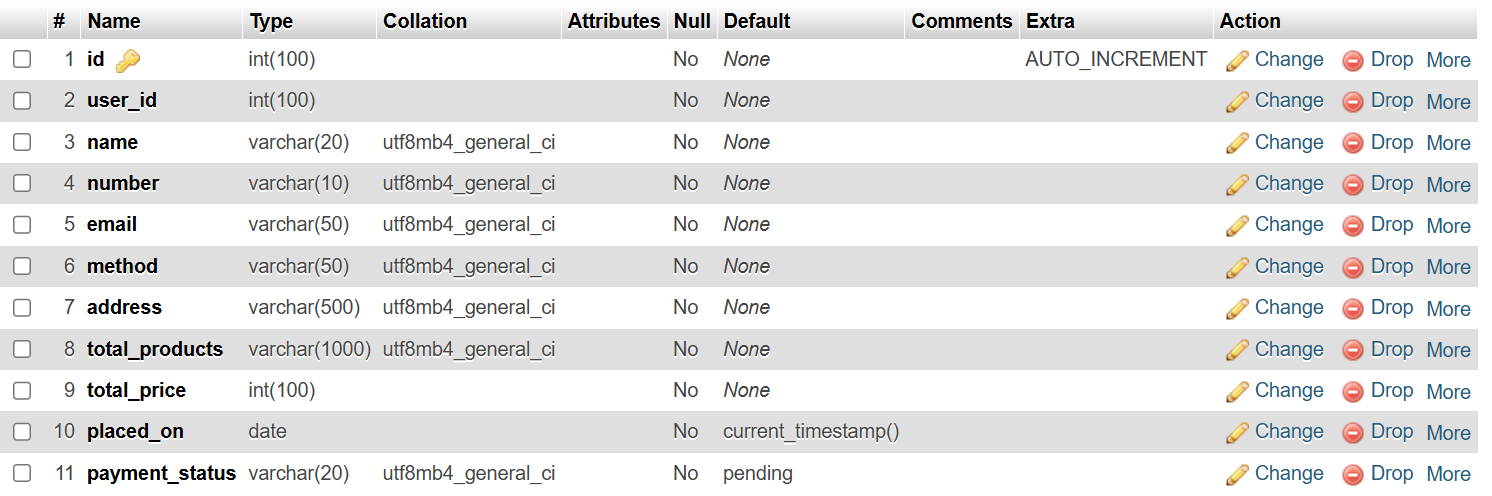


Table 4.6: Database Table for Orders

**4.3.7Database Table for Wishlist:**

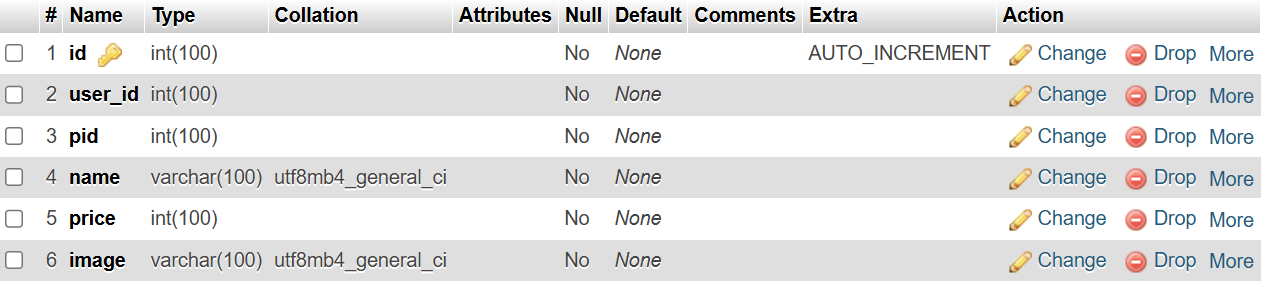


Table 4.7: Database Table for Wishlist

**Chapter5**

**Testing**

**5.1 Testing:**

**Chapter 5**

**Testing**

Testing is a structured and systematic process that can be planned in advance to ensure the reliability and functionality of a system. It typically starts at the module level and progresses through the integration of the complete computer-based system. No software development process is complete without thorough testing, as it plays a crucial role in the system’s success.

The primary objective of testing is to identify errors and verify that the system behaves as expected. A well-designed test case is one that effectively uncovers potential defects that may not have been previously detected. Ultimately, testing aims to improve software quality by ensuring that all components function correctly and meet the specified requirements.

**5.2 Testing Technique:**

**Test Case 1: Guest User Browsing and Adding to Cart**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case ID** | | 001 | **Test case Description** | | Verify that a guest user can browse products and successfully add them to the shopping cart. | | |  | |  |
| **Created by** | | Fahmida\_Anni | **Reviewed by** | | Jawad | | | **Version** | | 11 |
| **Tester’s Name** | | Fahmida\_Anni | **Date tested** | | 18-05-2025 | **Test case(pass/fail)** | | | | Pass |
| **Test scenario:** Guest user can browse products and successfully add them to the shopping cart. | | | | | | | | | | |
| **SL** | **Step details** | | | **Expected result** | | | **Actual result** | | **Status** | |
|  | Go to product listing page | | | refer to "View Product" use case in Image1 | | | Do | | Pass | |
|  | Browse or search products | | | Products appear as per criteria | | | Do | | Pass | |
|  | Select a product | | | Product details are shown | | | Do | | Pass | |
|  | Click “Add to cart” | | | Product is added to cart | | | Do | | Pass | |
|  | Check if product is in cart | | | Product appears with correct info | | | Do | | Pass | |
|  | Open cart page | | | Cart displays item correctly | | | Do | | Pass | |

**Test case 2: Registered User Placing an Order**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case ID** | | 002 | **Test case Description** | | Verify that a registered user can log in, add products to the cart, and successfully place an order. | | |  | |  |
| **Created by** | | Fahmida\_Anni | **Reviewed by** | | Jawad | | **Version** | | | 11 |
| **Tester’s Name** | | Fahmida\_Anni | **Date tested** | | 18-05-2025 | | **Test case(pass/fail/not)** | | | Pass |
| **Test scenario:** Registered User Placing an Order | | | | | | | | | | |
| **SL** | **Step details** | | | **Expected result** | | **Actual result** | | | **Status** | |
|  | Log in as a registered user | | | User is authenticated and logged in | | Do | | | Pass | |
|  | Browse and add products to cart | | | Selected products appear in cart | | Do | | | Pass | |
|  | Go to checkout page | | | Checkout page is displayed | | Do | | | Pass | |
|  | Enter shipping and payment info | | | Info is accepted and validated | | Do | | | Pass | |
|  | Confirm the order | | | |  | | --- | |  |  |  | | --- | | Order is placed successfully | | | Do | | | Pass | |
|  | Check for confirmation | | | Order confirmation is shown or emailed | | Do | | | Pass | |

**Test case 3: Store Manager Updating Product Inventory**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case ID** | | 003 | **Test case Description** | | Verify that a store manager can successfully update the inventory levels of a product. | | | |  |  |
| **Created by** | | Fahmida\_Anni | **Reviewed by** | | Jawad | | **Version** | | | 11 |
| **Tester’s Name** | | Fahmida\_Anni | **Date tested** | | 18-05-2025 | | **Test case(pass/fail/not)** | | | Pass |
| **Test scenario:** Store Manager Updating Product Inventory | | | | | | | | | | |
| **SL** | **Step details** | | | **Expected result** | | **Actual result** | | **Status** | | |
|  | Log in as store manager | | | Store manager is authenticated | | Do | | Pass | | |
|  | Go to inventory section | | | Inventory management interface is displayed | | Do | | Pass | | |
|  | Select a product | | | |  | | --- | |  |  |  | | --- | | Product details are shown | | | Do | | Pass | | |
|  | Update stock quantity | | | New quantity is entered | | Do | | Pass | | |
|  | Save changes | | | System confirms update | | Do | | Pass | | |
|  | Verify update | | | Inventory level is updated in the database | | Do | | Pass | | |

**Test Case 4: System Handling Out-of-Stock Items During Order Placement**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case ID** | | 004 | **Test case Description** | | Verify how the system handles a scenario where a user tries to order a product that is out of stock. | |  | |  |
| **Created by** | | Fahmida\_Anni | **Reviewed by** | | Jawad | | **Version** | | 11 |
| **Tester’s Name** | | Fahmida\_Anni | **Date tested** | | 18-05-2025 | | **Test case(pass/fail/not)** | | Pass |
| **Test scenario:** System Handling Out-of-Stock Items During Order Placement | | | | | | | | | |
| **SL** | **Step details** | | | **Expected result** | | **Actual result** | | **Status** | |
|  | Set product stock to zero | | | Product is marked as out of stock | | Do | | Pass | |
|  | Try adding the out-of-stock product to cart | | | System blocks addition or shows out-of-stock message | | Do | | Pass | |
|  | Try to checkout with the item | | | System prevents checkout or highlights the issue | | Do | | Pass | |

# **6.1 Conclusion**

# **Chapter 6**

# **Conclusion**

My project is a humble effort aimed at fulfilling the essential needs of an Online Shopping Inventory Management System. Several user-friendly and efficient coding practices have been adopted to enhance the functionality and usability of the system. This software package is expected to serve as a powerful tool in meeting the various requirements of users in an online shopping environment.

The primary objective of software planning in this project is to provide a solid framework that enables the project manager to make realistic estimations within a limited timeframe. These estimations are intended to be refined and updated as the development process advances, ensuring accuracy and adaptability throughout the project lifecycle.

Although I could not implement every planned feature due to time constraints, I have made a dedicated effort to ensure the system is functional, stable, and user-centric. As my experience in software development grows, I aim to continuously enhance and expand this system in every possible way. I hope this software proves beneficial to its users and provides a comprehensive understanding of how an online shopping and inventory system operates in a real-world scenario.

# **6.2 Limitation of the Project**

One of the primary limitations of this project is the constraint of time. The time allocated for the development of this Online Shopping Inventory Management System was very limited. While attempting to follow standard software engineering practices, it became challenging to complete the entire system within the short timeframe. As a result, the overall scope of the project had to be reduced.

Within this limited period, it was difficult to fully understand the underlying problems, gather comprehensive requirements, and build a fully-featured system. This time limitation significantly impacted the ability to explore advanced features and optimize the system to its full potential. Additionally, several challenges were encountered during the data collection phase, which further hindered the development process. Despite these limitations, every effort was made to ensure the system is functional and provides a solid foundation for future enhancements.

.

# **Chapter 7**

# **Bibliography**

# **7.1 Book**

1. Pressman, Roger S. Software Engineering: A Practitioner’s Approach. 5th edition. Boston: McGraw Hill, 2004.

2. Kendall, E. Kendall. System Analysis and Design.6th edition.

New Delhi: Prentice Hall.

# **7.2 Websites**

* <https://www.google.com.bd>
* http://www.php.net/
* <http://php.net/manual/en/book.pdo.php>
* <https://ellislab.com/codeigniter>
* <http://www.tutorialspoint.com/>
* <https://www.youtube.com/>
* <http://www.w3school.com/>
* http://www.stackoverflow.com/
* <http://www.codeacademy.com/>
* <https://www.visual-paradigm.com/>
* <https://chatgpt.com/>