****

**American International University-Bangladesh (AIUB)**

Department of Computer Science

Faculty of Science & Technology (FST)

**Coffee Shop Billing Management System**

**Semester: Fall 25-26**

|  |  |  |
| --- | --- | --- |
| ***Group: 07*** | | ***Section: N*** |
| **SL** | **Student Name** | **Student ID** |
| 1 | FUAD HASAN | 23-51947-2 |
| 2 | MD. TOFAYEL HOSSAIN | 23-51928-2 |
| 3 | SARKER LAMIA YEAMIN | 23-51926-2 |
| 4 | NAFISA ANJUM TITHI | 23-51723-2 |

**Table of Contents**

[1. Project Proposal 0](#_17wy973q88ow)3

[1.1 Background to the Problem](#_esff8883yxju) 03

[1.2 Selection of Process Model 0](#_mb0fhcj88zlk)4

[2. Software Requirements Specification (SRS) / PRD 0](#_z5ea9qqxht6v)5

[2.1 Scopes and Features 0](#_esff8883yxju)5

[2.2 User Story Table 0](#_mb0fhcj88zlk)7

[2.3 Requirements Traceability Matrix 1](#_honhxrbwotn9)4

[2.3.1 Functional Requirements 1](#_ir5phrbjxrc6)5

[2.3.2 Non-Functional Requirements 1](#_vdse4tbaw3ra)6

[3. Software Design 1](#_z5ea9qqxht6v)8

[3.1 System Deisgn 1](#_esff8883yxju)8

[3.2 UI Deisgn using Figma 2](#_esff8883yxju)0

[4. Git Workflow 2](#_z5ea9qqxht6v)7

[5. Software Testing 2](#_z5ea9qqxht6v)9

[6. Conclusion 47](#_z5ea9qqxht6v)

1. **PROJECT PROPOSAL**
   1. **Background to the Problem**

In the modern food and beverage industry, coffee shops have become essential social and business hubs. However, many small and medium-sized coffee shops in our region continue to rely on outdated manual methods for managing their daily operations. The current scenario in most of these establishments involves cashiers using basic calculators and traditional cash registers to process customer orders, while shop owners maintain inventory records in handwritten notebooks or simple spreadsheets. This primitive approach creates numerous operational challenges that directly impact business growth and customer satisfaction.  
  
During peak business hours, particularly morning rushes and evening hours, the manual order-taking and billing process becomes extremely slow. Cashiers must manually write down each item, calculate prices, apply taxes, and compute the total amount, which often results in long queues of frustrated customers waiting to be served. This inefficiency not only reduces the number of customers served per hour but also negatively affects the overall customer experience, potentially driving loyal patrons to competing establishments that offer faster service.

The manual system is highly prone to human error at multiple stages. Simple arithmetic mistakes in billing can lead to financial losses either through undercharging or customer disputes over overcharging. Inventory tracking errors are even more problematic – when staff forget to record each sale, the recorded stock levels become inaccurate, leading to situations where popular items unexpectedly run out during busy periods. Conversely, over-ordering of less popular items results in wastage and tied-up capital. These errors compound over time, creating significant discrepancies between actual and recorded data.  
  
Perhaps the most critical limitation of manual operations is the complete absence of business intelligence. Shop owners operate without access to vital statistics that could inform strategic decisions. They cannot easily determine which products generate the most profit, identify slow-moving items that should be discontinued, recognize their busiest days and hours for optimal staff scheduling, or track sales trends over time. The lack of a centralized dashboard means owners must rely on gut feelings rather than concrete data when making purchasing decisions, planning promotions, or evaluating business performance.  
  
Financial transparency and accountability also suffer under manual systems. Cash handling without proper digital records increases the risk of theft and makes it difficult to track daily revenue accurately. During audits or tax calculations, the absence of systematic digital records creates additional workload and potential compliance issues.

To address these fundamental challenges, our group proposes the development of a comprehensive "Coffee Shop Management System." This software solution will digitize and automate core coffee shop operations including order processing, inventory management, sales tracking, and reporting. By implementing this system, coffee shops can significantly reduce service time, eliminate calculation errors, maintain real-time inventory accuracy, and gain valuable business insights through automated reporting and analytics. The system will provide role-based access for administrators, cashiers, inventory managers, and customers, ensuring that each user type has appropriate tools and permissions to perform their tasks efficiently.  
  
The implementation of this system represents a crucial step toward modernizing small and medium coffee shop operations, enabling them to compete effectively with larger chains while maintaining their unique character and customer relationships. The scalability of the solution ensures that it can grow with the business, adding new features and capabilities as the coffee shop expands to multiple locations or introduces additional services.

* 1. **Selection of Process Model**

For our Coffee Shop Management System project, we have chosen the Scrum process model as our software development methodology. Scrum is an Agile framework that is well-suited for projects that require flexibility, collaboration, and a focus on customer satisfaction through iterative and incremental development. The Scrum process can be visualized through a series of phases that include the Pre-game, Development (Game), and Post-game phases.

Fig 01: Scrum Process Model

In the Pre-game phase, planning and architecture are the primary focus. During planning, the system being developed is defined, and a Product Backlog list is created, containing all the known requirements. These requirements are prioritized, and the effort needed for their implementation is estimated. The Product Backlog is constantly updated with new and more detailed items, as well as with more accurate estimations and new priority orders. Planning also includes the definition of the project team, tools, and other resources, risk assessment and controlling issues, training needs, and verification management approval. Architecture involves planning the high-level design of the system, including the architecture, based on the current items in the Product Backlog. If the project involves enhancing an existing system, the changes needed for implementing the Backlog items are identified along with the problems they may cause. A design review meeting is held to go over the proposals for the implementation, and decisions are made based on this review.  
  
The Development (Game) phase is treated as a "black box" where the unpredictable is expected. The system is developed in Sprints, which are iterative cycles where the functionality is developed or enhanced to produce new increments. Each Sprint includes the traditional phases of software development: requirements, analysis, design, evolution, and delivery phases. One Sprint is planned to last from one week to four weeks, not exceeding one month.

The Post-game phase is entered when an agreement has been made, such as when the requirements are completed. In this case, no more items and issues can be found, nor can any new ones be invented. The system is now ready for release, and the preparation for this is done during the post-game phase, including tasks such as integration, system testing, and documentation.

The Scrum process involves specific roles and responsibilities. The Scrum Master is responsible for ensuring that the project is carried through according to the practices, values, and rules of Scrum and that it progresses as planned. The Scrum Master interacts with the project team as well as with the customer and the management during the project. The Product Owner is officially responsible for the project, managing, controlling, and making visible the Product Backlog list. The Scrum Team is the project team that has the authority to decide on the necessary actions and to organize itself in order to achieve the goals of each Sprint. The customer participates in the tasks related to product Backlog items for the system being developed or enhanced. Management is in charge of final decision-making, along with the agreements, standards, and conventions to be followed in the project. Management also participates in the setting of goals and requirements.  
  
Scrum practices include the Product Backlog & Sprint, where the team adapts to changing environmental variables using Sprint Planning Meetings, Sprint Backlog, and Daily Scrum meetings. Effort Estimation & Sprint Backlog involve selecting Product Backlog items to be implemented in the next Sprint during the Sprint Planning meeting. The Sprint Planning meeting is a two-phase meeting organized by the Scrum Master to decide upon the goals and the functionality of the next Sprint. Daily Scrum meetings are organized to keep track of the progress of the Scrum Team continuously and serve as planning meetings for what has been done since the last meeting and what is to be done before the next one. The Sprint Review meeting is held on the last day of the Sprint, where the Scrum Team and the Scrum Master present the results of the Sprint to the management, customers, users, and the Product Owner in an informal meeting.  
  
We chose Scrum because of its adaptability, which allows us to respond to changing requirements, a common occurrence in software development. The framework's emphasis on collaboration ensures that all team members are aligned and can quickly address any issues that arise. By involving the customer in the process and delivering working software at the end of each Sprint, we can ensure that the final product meets their needs and expectations. The focus on delivering a potentially shippable product at the end of each Sprint helps to maintain a high standard of quality throughout the development process. Scrum's structured approach to development helps to maximize efficiency by minimizing wasted effort and ensuring that the team is always working on the most valuable features. The clear roles and responsibilities, along with regular reviews and meetings, ensure that everyone involved in the project has a clear understanding of progress and any challenges. By choosing Scrum, we aim to create a Coffee Shop Management System that is not only functional and efficient but also aligns closely with the needs of its users, emphasizing flexibility, collaboration, and customer feedback to deliver a high-quality product that meets the specific challenges faced by coffee shops in managing their operations.

# **SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS) / PRODUCT REQUIREMENTS DOCUMENT (PRD)**

* 1. **Scopes and Features**

The Coffee Shop Management System is designed to be a comprehensive digital solution that addresses the operational challenges faced by small to medium-sized coffee shops. The system scope encompasses the entire lifecycle of coffee shop operations, from inventory procurement to final sales reporting, while providing role-based access to different user categories.

**System Scope:** The system will operate as a centralized web-based application accessible from multiple devices including desktop computers, tablets, and smartphones. It will support single-location coffee shops with the potential for future multi-location expansion. The scope includes managing product catalogs (coffee, tea, pastries, merchandise), processing customer orders through multiple payment methods, tracking inventory levels in real-time, maintaining customer profiles, generating comprehensive business reports, and providing audit trails for all transactions.

The system explicitly excludes features such as advanced accounting (which would integrate with external accounting software), supply chain management beyond basic supplier information, and customer-facing mobile applications (though the web interface will be mobile-responsive). The initial release will focus on core functionalities essential for daily operations, with future enhancements planned based on user feedback.  
  
**Key Features:**

1. User Authentication and Role Management:

* Secure login system with username and password for all user roles
* Role-based access control (RBAC) separating permissions for Admin, Cashier, Inventory Manager, and Customer
* Password reset functionality via OTP sent to registered mobile number and email
* Profile management allowing users to view and update their personal information

2. Admin Dashboard and Control:

* Centralized admin panel with overview of daily sales, inventory status, and staff activity
* Complete product lifecycle management (add, edit, delete products with pricing details)
* Category management for organizing products (e.g., Hot Beverages, Cold Beverages, Snacks)
* Staff account management (create, modify, deactivate cashier and inventory manager accounts)
* Customer data management with viewing and search capabilities
* Access to all bill records and transaction history
* System configuration settings including tax rates and discount rules

3. Cashier Operations Module:

* Streamlined order processing interface for quick customer service
* Product search functionality to rapidly locate items in the catalog
* Shopping cart system to compile multiple items in a single order
* Automatic bill calculation including taxes and discounts
* Multiple payment method support (Cash, Card)
* Digital receipt generation and printing capability
* Order history viewing for reference and reprints
* Customer lookup to associate orders with registered customers

4. Inventory Management:

* Real-time stock level tracking for all products
* Stock count functionality for physical inventory verification
* Automated stock level updates upon sale completion
* Low stock alerts when items fall below defined thresholds
* Stock history tracking to monitor consumption patterns
* Product-wise inventory reports showing current quantities and valuation

5. Customer Self-Service:

* Customer registration with mobile and email verification
* Personal profile management view
* Ability to browse product catalog with current prices
* Shopping cart functionality for order preparation
* Order placement with multiple payment options
* Order history tracking to view past purchases
* Receipt viewing and download capability

6. Reporting and Analytics:

* Daily sales reports showing total revenue, transaction count, and popular items
* Monthly and yearly sales summaries for trend analysis
* Inventory turnover reports to identify fast and slow-moving products
* Profit analysis reports comparing revenue against cost
* Peak hours identification to optimize staff scheduling
* Tax calculation reports for compliance purposes

7. System Integration and Data Management:

* Secure database storage for all business data
* Data backup and recovery mechanisms
* Export functionality for reports in PDF and Excel formats
* Audit trail logging all critical operations for accountability
* Mobile-responsive design for access on various devices

8. Security and Compliance:

* Data encryption for sensitive information (passwords, payment details)
* Session management with automatic timeout for security
* Input validation to prevent SQL injection and cross-site scripting
* Role-based permissions ensuring users only access authorized functions
* Compliance with data protection regulations through proper user consent management

**2.2 User Story Table**

The user stories below capture the functional requirements from the perspective of each system actor, following the standard format: "As a [user role], I want to [action], so that [benefit]." Each story includes clear acceptance criteria for implementation and testing.

|  |  |  |  |
| --- | --- | --- | --- |
| **As a/an** | **I want to** | **So that** | **Acceptance Criteria** |
| **Admin** | Create a new cashier profile | I can enroll staff in the system | 1. System sends OTP to cashier's mobile and confirmation email.  2. Account is activated only after OTP verification.  3. Default password must be changed on first login. |
| Create a new inventory manager profile | I can assign inventory responsibilities to staff | 1. OTP verification required via mobile and email.  2. Role is clearly marked as "Inventory Manager".  3. User receives role-specific permissions automatically. |
| Login with my username and password | The system can authenticate me and I can trust it | 1. After successful login, admin dashboard loads within 3 seconds.  2. Failed login attempts are logged.  3. Account locks after 5 consecutive failed attempts. |
| View my own profile details | I can see information provided during registration | 1. Information displayed in read-only table format.  2. All fields from registration form are visible.  3. Edit profile button is clearly accessible. |
| Update my information | I can replace incorrect information with correct data | 1. Clicking edit transforms fields to editable mode.  2. Changes require password confirmation.  3. Update notification appears after successful save. |
| Change my password | My account can be more secure | 1. Must input old password and new password twice.  2. New password must meet complexity requirements.  3. Success notification appears, and user is logged out. |
| Request a new password | I do not permanently lose access if I forget my password | 1. OTP sent to registered mobile and email.  2. OTP expires after 10 minutes.  3. Can set new password only after OTP verification. |
| Delete a cashier account | The cashier cannot access the system anymore | 1. Confirmation popup shows cashier details.  2. Must type "CONFIRM" to proceed.  3. System logs the deletion with admin ID and timestamp. |
| Delete an inventory manager account | I can remove staff who left the organization | 1. Shows popup with staff details.  2. Requires secondary confirmation.  3. Archived account data remains in system logs. |
| Add a new product | I can introduce new items to the menu | 1. Must input product name, price, category, and initial stock.  2. Success notification appears.  3. Product immediately appears in active product list. |
| Edit product information | All users see updated product details | 1. Can modify name, price, category, and description.  2. Changes reflect in real-time across all modules.  3. Price changes logged with admin ID for audit. |
| Delete a product | The product does not appear in any search | 1. Product removed from active list but remains in archive.  2. Cannot delete if stock quantity > 0.  3. Shows confirmation popup before deletion. |
| View complete product list | I can see all products with their current status | 1. Table shows all products with stock levels and prices.  2. Include filter by category and stock status.  3. Shows total inventory value at bottom. |
| Generate daily sales report | I can know about daily profit and performance | 1. Report shows total sales, transaction count, and popular items.  2. Can filter by date range.  3. Exportable to PDF format. |
| Generate monthly/yearly sales report | I can analyze long-term business trends | 1. Shows monthly comparison charts.  2. Includes profit margins and growth percentages.  3. Identifies best and worst performing products. |
| Manage product categories | I can organize products logically | 1. Can add, edit, and delete categories.  2. System prevents deletion of categories containing active products.  3. Category changes reflect immediately in product listings. |
| **Cashier** | Login with my username and password | The system can authenticate me | 1. Redirects to cashier dashboard after login.  2. Shows only cashier-relevant functions.  3. Session times out after 30 minutes of inactivity. |
| View my profile details | I can verify my employment information | 1. Displays name, ID, joining date, and contact details.  2. Edit button allows limited field updates.  3. Shows my daily transaction count and totals. |
| Update my contact information | I can keep my details current | 1. Can edit phone number and email.  2. Requires OTP verification for new contact info.  3. Success message confirms update. |
| Change my password | My account remains secure | 1. Follows same security rules as admin.  2. Cannot reuse last 3 passwords.  3. Logs password change event. |
| Process a customer order quickly | I can serve customers efficiently during rush hours | 1. Search products by name or category.  2. Add multiple items to cart with quantity.  3. Automatic price calculation including taxes. |
| Apply discounts to an order | I can honor promotional offers | 1. Can apply percentage or fixed amount discounts.  2. Requires admin PIN for discounts > 20%.  3. Discount reason is logged. |
| Handle cash payments | I can complete transactions | 1. System calculates change amount automatically.  2. Prints receipt with "PAID BY CASH" marking.  3. Updates cash drawer balance. |
| Handle card payments | I can process digital payments | 1. Integrates with payment gateway for card processing.  2. Marks order as "PAID BY CARD".  3. Generates digital payment confirmation. |
| Print receipts | Customers have proof of purchase | 1. Receipt prints within 5 seconds.  2. Includes shop details, order items, total, and date/time.  3. Can reprint last transaction if needed. |
| Search for products | I can quickly find items for customers | 1. Real-time search as user types.  2. Shows product availability status.  3. Displays product price and code. |
| View my daily sales summary | I can track my performance | 1. Shows total sales, transaction count, and average order value.  2. Compares with previous day's performance.  3. Updates in real-time as new sales occur. |
| **Inventory Manager** | Login with my credentials | I can access inventory functions | 1. Dashboard shows low-stock alerts prominently.  2. Quick access buttons for stock count and updates.  3. Shows inventory valuation summary. |
| Count physical stock | I can verify actual stock levels | 1. Generate stock count sheets by category.  2. Input physical counts for each product.  3. System highlights discrepancies between recorded and actual. |
| Update stock levels | I can correct inventory records | 1. Can add or subtract stock with reason code.  2. Requires note for adjustments > 10% of stock.  3. Changes reflect immediately in product availability. |
| Receive new stock shipments | I can add purchased inventory | 1. Select supplier and enter invoice number.  2. Add multiple products with quantities and costs.  3. Updates stock levels and calculates new inventory value. |
| Set low stock thresholds | I get alerts before items run out | 1. Define minimum quantity for each product.  2. System sends daily email alert for items below threshold.  3. Threshold value appears in product details. |
| View stock movement history | I can track inventory changes | 1. Shows all stock in/out transactions with dates.  2. Can filter by date range and product.  3. Identifies patterns in stock consumption. |
| Generate inventory reports | I can plan purchasing decisions | 1. Shows current stock levels, values, and turnover rates.  2. Identifies dead stock (no movement in 30 days).  3. Suggests reorder quantities based on consumption. |
| Manage supplier information | I can maintain purchasing contacts | 1. Add/edit supplier details (name, contact, products supplied).  2. Link suppliers to specific products.  3. View purchase history per supplier. |
| **Customer** | Create a new customer profile | I can enroll myself in the system | 1. OTP sent to mobile and confirmation email.  2. Profile is active immediately after verification.  3. Receive welcome email with loyalty program details. |
| Login to my account | I can access personalized features | 1. Dashboard shows order history and loyalty points.  2. Can view and repeat past orders.  3. Shows personalized recommendations based on order history. |
| View and update my profile | I can keep my information current | 1. Can edit name, phone, email, and delivery address.  2. Change history is maintained.  3. Requires password confirmation for email changes. |
| Browse the product catalog | I can see available items and prices | 1. Products organized by category with images.  2. Shows real-time availability status.  3. Can view product descriptions and nutritional info. |
| Search for specific products | I can quickly find what I want | 1. Search by product name or category.  2. Shows results with price and availability.  3. Recent searches are saved for convenience. |
| Add products to my cart | I can compile my order | 1. Cart persists across sessions for 7 days.  2. Shows running total with estimated tax.  3. Can modify quantities or remove items easily. |
| Place an order | I can get my selected items | 1. Order confirmation email sent immediately.  2. Shows estimated preparation time.  3. Order appears in "My Orders" with status tracking. |
| Make payment online | I can complete purchase digitally | 1. Integrates with secure payment gateway.  2. Supports credit/debit cards and mobile banking.  3. Payment confirmation shows transaction ID. |
| View my order history | I can track past purchases | 1. Shows all orders with date, items, and total.  2. Can reorder previous orders with one click.  3. Download receipts for past orders. |
| Rate and review products | I can share my experience | 1. Can rate products 1-5 stars.  2. Write text reviews with character limit.  3. Reviews appear after admin moderation. |

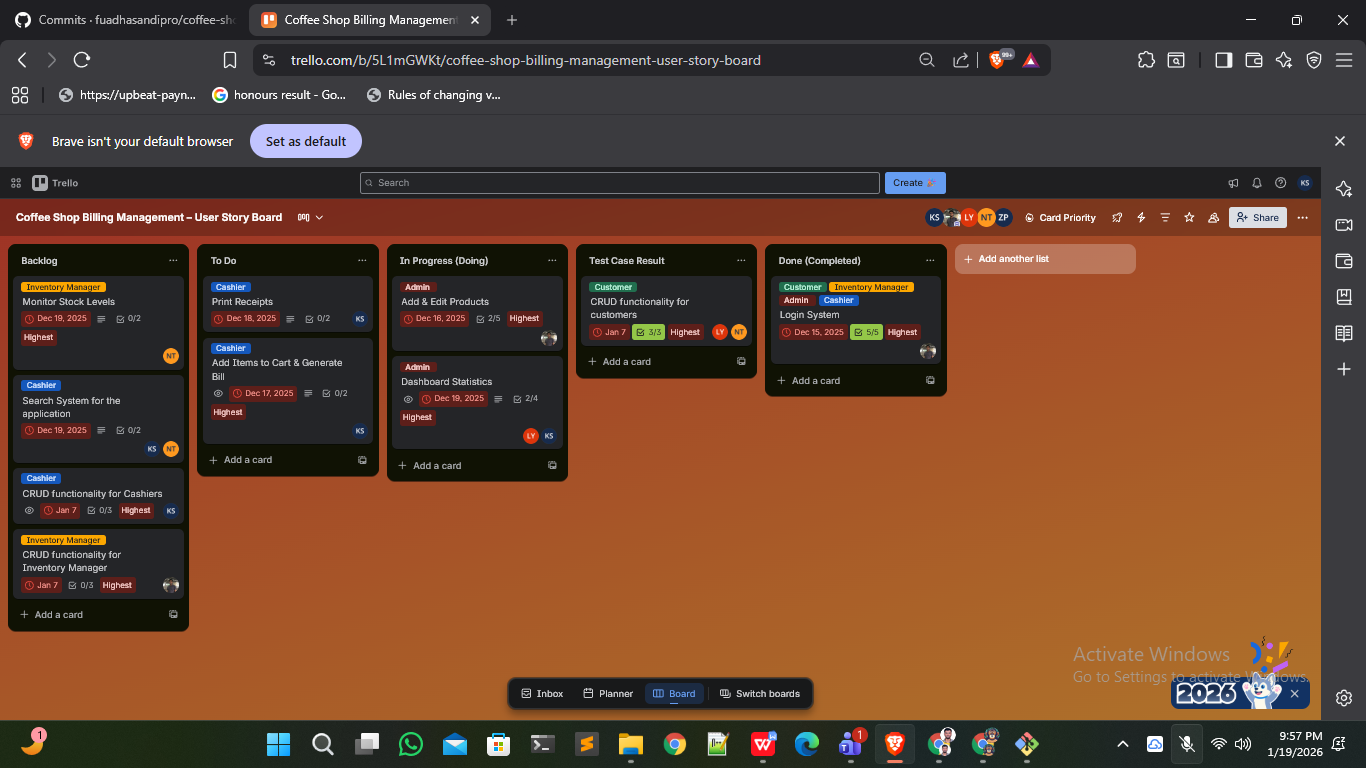


Figure 01: Shows the user story board using Trello

Trello Link: <https://trello.com/invite/b/691b2ea874ee6ef54881fb85/ATTI361a777cc177d8e62a28a4cdddf6f5cf7E1741E0/coffee-shop-billing-management-user-story-board>

* 1. **Requirements Traceability Matrix**

Functional requirements define the specific behaviors and functions that the Coffee Shop Management System must perform. Each requirement is uniquely identified and mapped to corresponding user stories to ensure complete coverage.

* + 1. Functional Requirements

|  |  |  |
| --- | --- | --- |
| ****FR Category**** | ****Requirement / Description**** | ****Priority**** |
| User Management | The system shall allow new users to register and must verify accounts by sending OTP to mobile and confirmation email. | High |
| User Management | The system shall enforce four distinct roles (Admin, Cashier, Inventory Manager, Customer) with specific access permissions. | High |
| Product Management | The Admin shall be able to add, edit, delete, and view products with unique details like name, price, category, and stock. | High |
| Sales & Orders | The Cashier shall be able to create orders where the system automatically calculates totals, taxes, and change amount. | High |
| Sales & Orders | The system shall support Cash and Card payments, including integration with a secure payment gateway. | High |
| Sales & Orders | Upon payment completion, the system shall generate a printable and downloadable receipt with transaction details. | High |
| Inventory Management | The system shall automatically update stock levels immediately when a sale is completed. | High |
| Search & Navigation | The system shall provide real-time search functionality allowing users to find products by name or category. | High |
| Security & Audit | The system shall maintain a strictly immutable audit trail logging all logins, stock adjustments, and critical data changes. | High |
| Security & Audit | The system shall send a single-use OTP to the registered mobile/email for password reset requests (expires in 10 minutes). | High |
| Security & Audit | The system shall automatically lock user accounts after 5 consecutive failed login attempts. | High |
| Product Management | The system shall prevent the deletion of product categories if they currently contain active products. | Medium |
| Sales & Orders | The system shall require an Admin PIN to authorize any discount that exceeds 20% of the order value. | Medium |
| Inventory Management | The system shall send daily email alerts to the Inventory Manager when product stock falls below the minimum threshold. | Medium |
| Inventory Management | The Inventory Manager shall be able to perform physical stock counts and reconcile discrepancies with reason notes. | Medium |
| Customer Features | Customers shall be able to manage their profiles and view their order history to easily repeat previous orders. | Medium |
| Customer Features | Customers shall be able to place orders online, receiving an estimated preparation time and confirmation email. | Medium |
| Reporting | The system shall generate sales, inventory, and profit reports that are exportable to PDF and Excel formats. | Medium |
| Inventory Management | The system shall allow the management of supplier information and link suppliers to specific products. | Low |

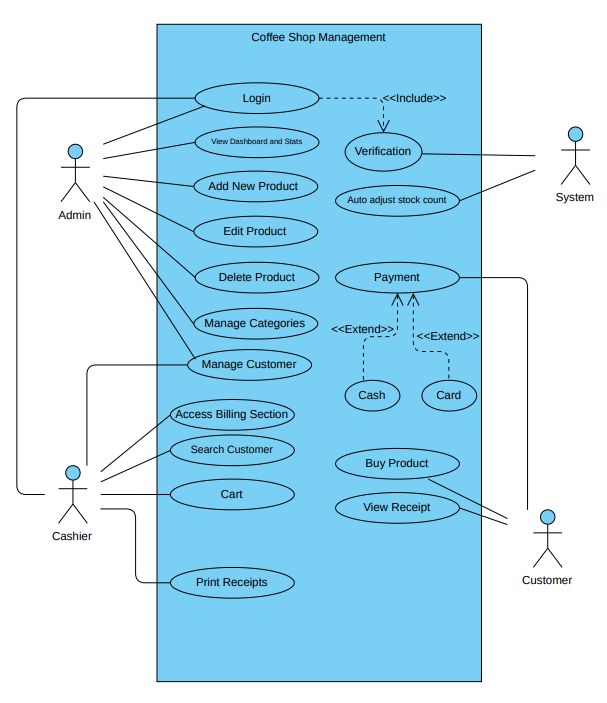
* + 1. Non-Functional Requirements

|  |  |
| --- | --- |
| **NFR Category** | **Requirement / Description** |
| Performance | The system should generate a bill within 2–3 seconds after items are added. |
| Performance | It should handle multiple billing operations at the same time without slowing down. |
| Performance | Price calculation and tax computation must be fast and accurate. |
| Reliability | The system should work correctly during busy hours without crashing. |
| Reliability | Billing data must not be lost in case of power failure or system restart. |
| Reliability | The system should maintain correct records of all sales transactions. |
| Reliability | The system should save each transaction automatically as it is completed to ensure no data is lost during unexpected failures. |
| Scalability | The system should support adding new coffee items or accessories easily. |
| Scalability | It should handle an increase in customers or sales volume without performance issues. |
| Security | Only authorized staff should be allowed to access billing and sales data. |
| Security | User login (admin/cashier) should be protected with password authentication. |
| Security | Sales and customer data should be stored securely. |
| Security | The system should automatically log out a user after a period of inactivity to prevent unauthorized access. |
| Usability | The interface should be simple and easy to use for cashiers. |
| Usability | New staff should be able to learn the system in less than one hour. |
| Usability | Buttons, menus, and item names should be clearly visible and readable. |
| Usability | The system should allow quick item selection using search or category buttons (coffee, snacks, accessories) to speed up billing. |
| Usability | The system should display a clear bill preview showing item name, quantity, price, tax, and total before final payment. |
| Usability | The system should support easy correction of mistakes, such as editing item quantity or removing an item before generating the final bill. |

# SOFTWARE DESIGN

* 1. **System Design:**

The system design for the Coffee Shop Management System follows a three-tier architecture comprising Presentation Layer, Business Logic Layer, and Data Access Layer. This separation of concerns ensures modularity, maintainability, and scalability. The design incorporates industry-standard UML diagrams to visualize system structure and behavior.

  
Figure 02: Use Case Diagram

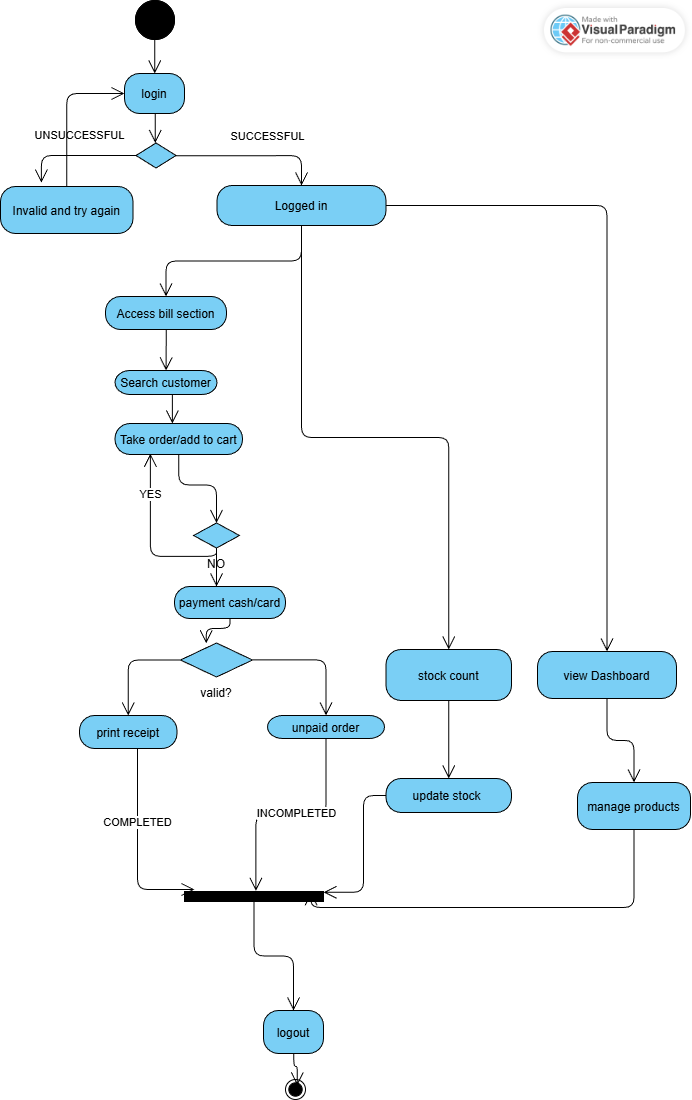


Figure 03: Activity Diagram

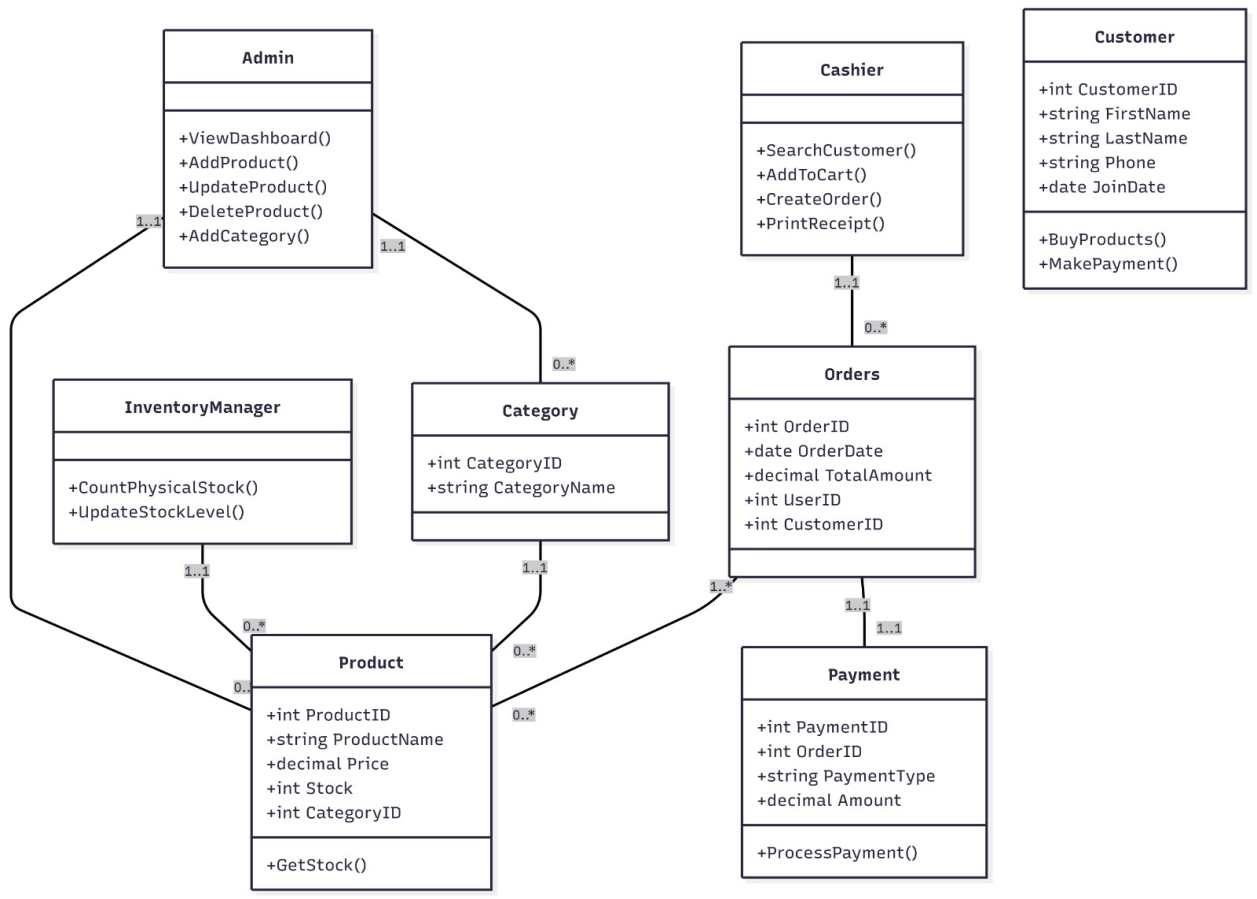


Figure 04: Class Diagram

* 1. **UI Deisgn using Figma**

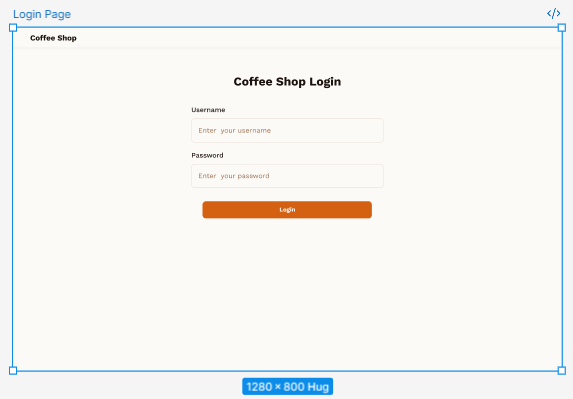


Figure 05: Login Page Figma Design

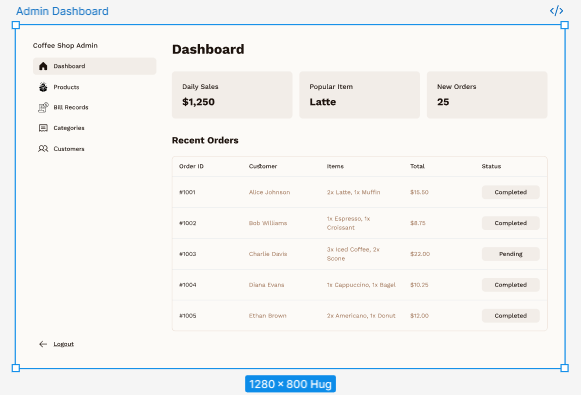


Figure 06:Admin Dashboard Page Figma Design

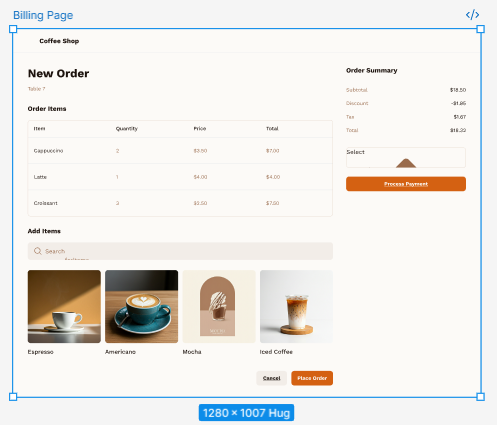


Figure 07: Billing Page Figma Design

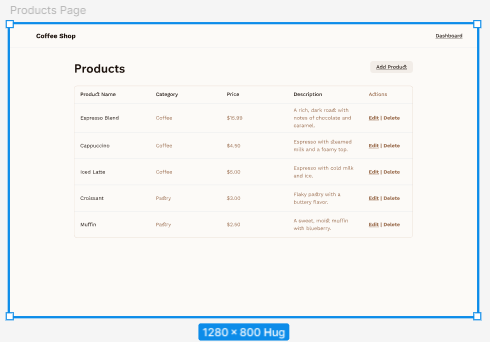


Figure 08: Products Page Figma Design

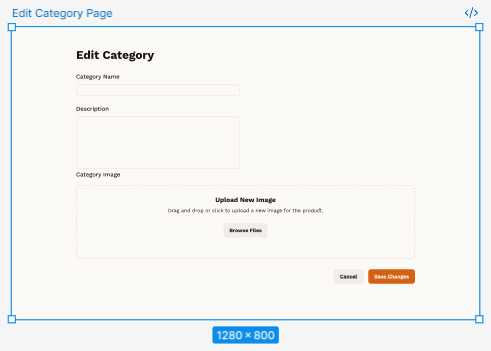


Figure 09: Edit Category Page Figma Design

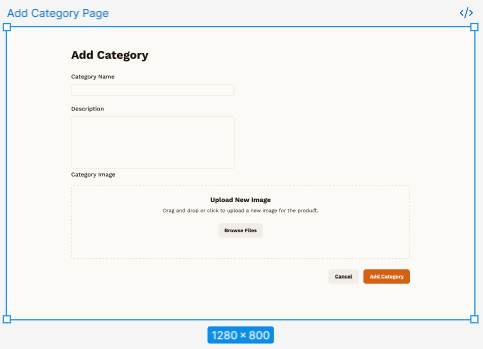


Figure 10: Add Category Page Figma Design

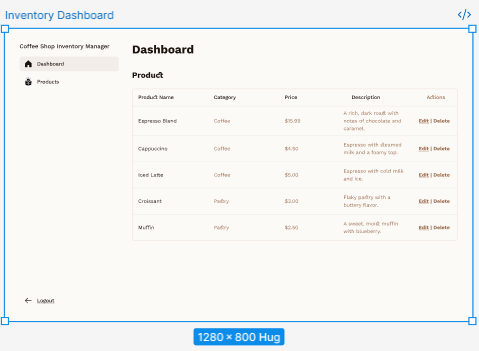


Figure 11: Inventory Dashboard Page Figma Design

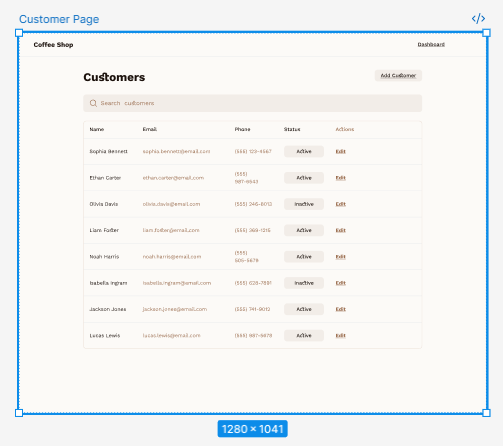


Figure 12: Customer Page Figma Design

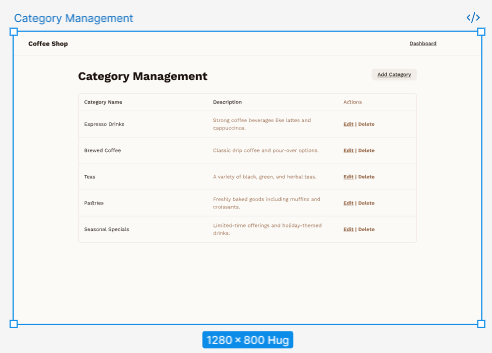


Figure 13: Category Management Page Figma Design

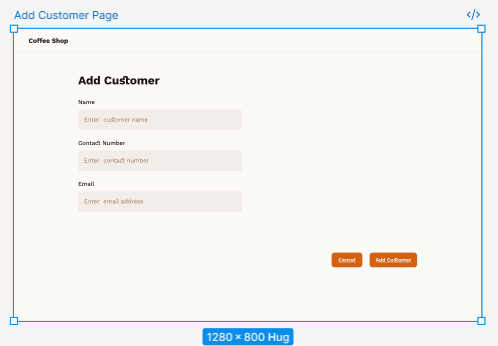


Figure 14: Add Customer Page Figma Design



Figure 15: Edit Cusstomer Page Figma Design

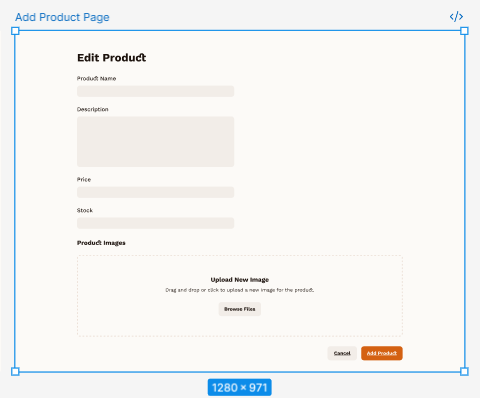


Figure 16: Add Product Page Figma Design

Figma Link: <https://www.figma.com/design/0I3NfNevdPhDGeD25Q1UTs/Coffee-Shop-Management?node-id=3-2239&t=oBXOJNNiyrNl4XK1-1>

1. **GIT WORKFLOW**

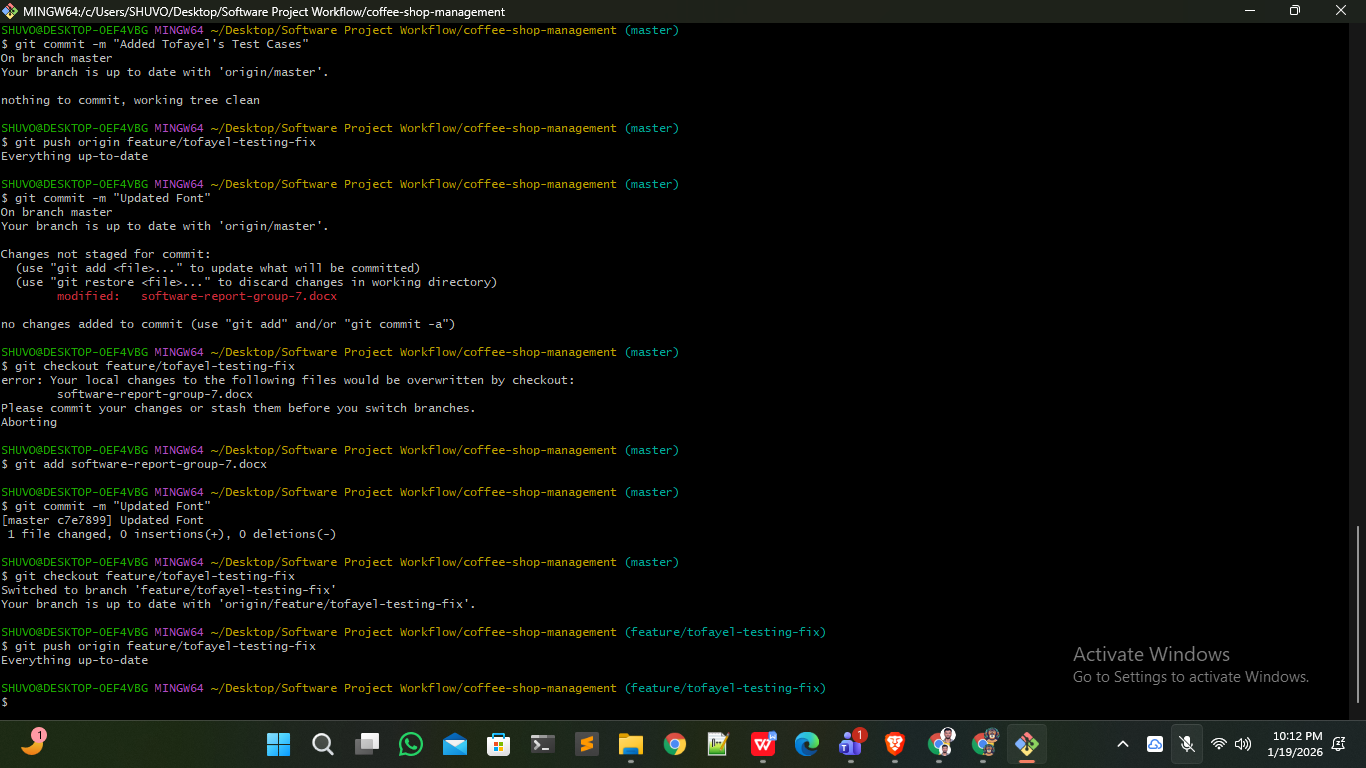


Figure 17: Tofayel’s Github workflow

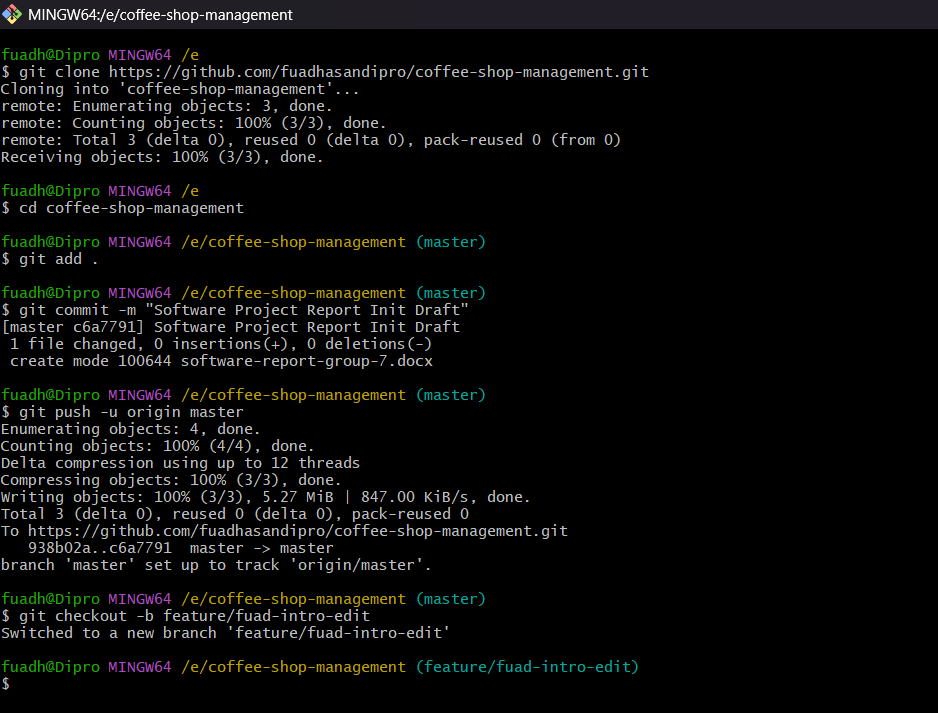


Figure 18: Dipro’s Github workflow

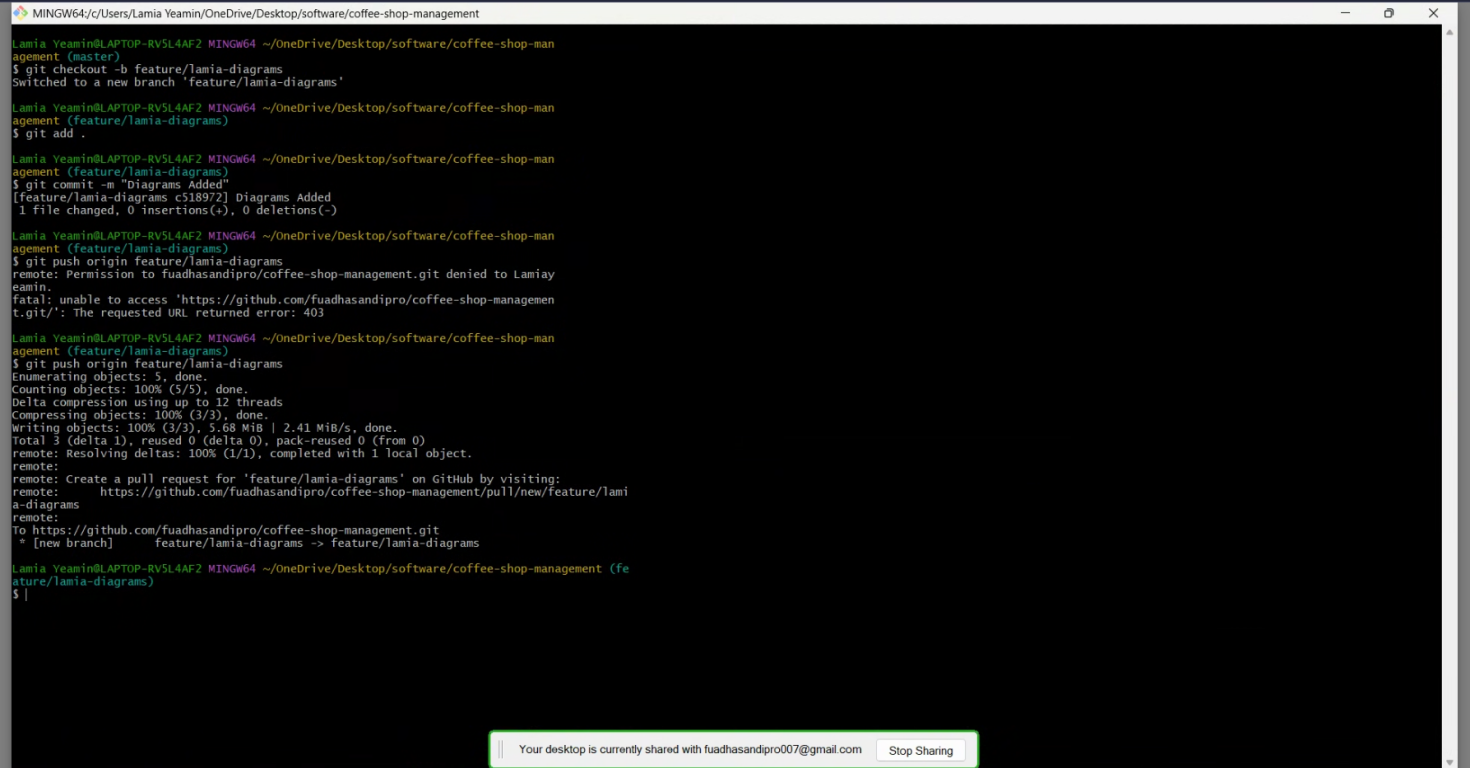


Figure 19: Lamia’s Github workflow

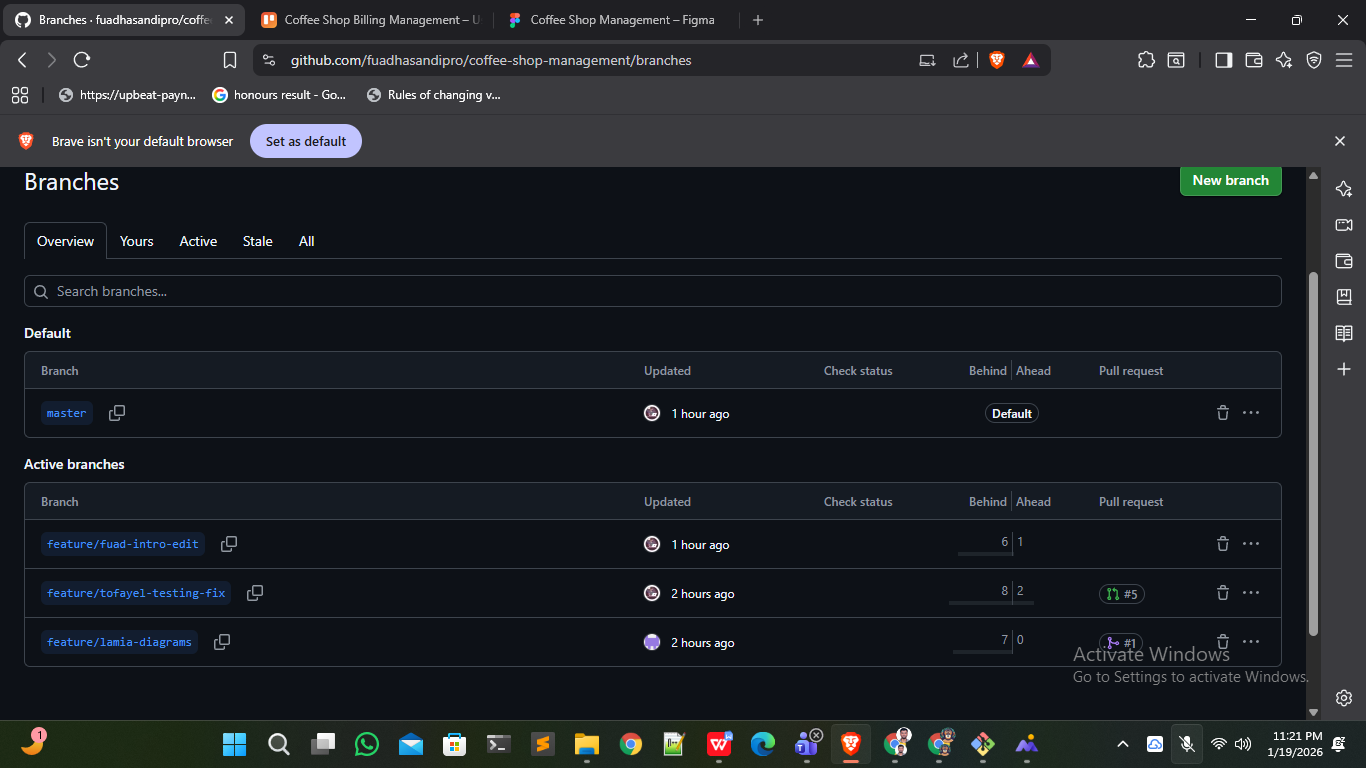


Figure 20: Github branches

Github Repository Link: <https://github.com/fuadhasandipro/coffee-shop-management>

# SOFTWARE TESTING We implemented a comprehensive testing strategy combining manual testing, automated unit tests, and integration testing. For this report, we present 20 detailed manual test cases covering critical functionalities across all user roles. Each test case follows the standard template provided, ensuring thorough validation of requirements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Fuad Hasan** | |
| **Test Case ID:** TC\_001 | | | **Test Designed date:** 2/1/2026 | |
| **Test Priority:** High | | | **Test Executed by:** Fuad Hasan | |
| **Module Name: User Authentication** | | | **Test Execution date:** 3/1/2026 | |
| **Test Title: Verify successful login for Admin role** | | |  | |
| **Description:** Test that Admin can login with valid credentials and access admin dashboard | | |  | |
| **Precondition:** Admin account exists with username "admin\_aiub" and password | | |  | |
| **Dependencies:** Database connection active; Admin account is active | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1. Navigate to system URL  2. Enter username: admin\_aiub  3. Enter valid password  4. Click Login button | Username: admin\_aiub  Password: Admin@1234 | Admin dashboard loads within 3 seconds; dashboard shows admin-specific menu options (Products, Staff, Reports) | Dashboard loaded in 2.1 seconds; all admin menus visible and accessible | **Pass** |

# 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Fuad Hasan** | |
| **Test Case ID:** TC\_002 | | | **Test Designed date:** 2/1/2026 | |
| **Test Priority:** High | | | **Test Executed by:** Fuad Hasan | |
| **Module Name: User Authentication** | | | **Test Execution date:** 3/1/2026 | |
| **Test Title: Verify account lockout after 5 failed login attempts** | | |  | |
| **Description:** Test security feature that locks account after consecutive failed logins | | |  | |
| **Precondition:** Cashier account exists and is currently unlocked | | |  | |
| **Dependencies:** Failed attempt counter is reset | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1. Navigate to login page  2. Enter correct username: cashier\_01  3. Enter wrong password 5 times  4. On 6th attempt, enter correct password | Username: cashier\_01  Wrong Passwords: 123, abc, pass, wrong, test | After 5th failure, message shows "Account locked. Contact Admin." 6th attempt with correct password still shows lockout message | Account locked after 5th attempt; admin panel showed lockout status; unlock required admin intervention | **Pass** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Fuad Hasan** | |
| **Test Case ID:** TC\_003 | | | **Test Designed date:** 2/1/2026 | |
| **Test Priority:** High | | | **Test Executed by:** Fuad Hasan | |
| **Module Name:** Product Management | | | **Test Execution date:** 3/1/2026 | |
| **Test Title:** Verify Admin can add new product | | |  | |
| **Description:** Test product creation with all required fields | | |  | |
| **Precondition:** Admin is logged in; product category "Hot Beverages" exists | | |  | |
| **Dependencies:** Category must exist before product creation | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1. Click "Add Product" button  2. Enter Product Name: Cappuccino Large  3. Select Category: Hot Beverages  4. Enter Price: 250  5. Enter Initial Stock: 50  6. Click Save | Product: Cappuccino Large  Category: Hot Beverages  Price: 250 TK  Stock: 50 | Success message "Product added successfully"; product appears in product list; stock shows 50 | Product added; appeared in list; stock count accurate; addition logged in audit trail | **Pass** |

# 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Fuad Hasan** | |
| **Test Case ID:** TC\_004 | | | **Test Designed date:** 2/1/2026 | |
| **Test Priority:** Medium | | | **Test Executed by:** Fuad Hasan | |
| **Module Name:** Product Management | | | **Test Execution date:** 3/1/2026 | |
| **Test Title: Verify product edit updates reflect immediately** | | |  | |
| **Description:** Test that price changes are immediately visible to cashiers | | |  | |
| **Precondition:** Product "Latte" exists with price 200; Admin and Cashier both logged in on separate sessions | | |  | |
| **Dependencies:** Real-time update mechanism active | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1. Admin changes Latte price from 200 to 220  2. Cashier searches for Latte in order screen | Product: Latte  Old Price: 200  New Price: 220 | Cashier sees updated price 220 within 2 seconds of Admin's change | Price updated in cashier view in 1.3 seconds; no page refresh required | **Pass** |

# 

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | | | **Test Designed by: Fuad Hasan** | | | |
| **Test Case ID:** TC\_005 | | | | | **Test Designed date:** 2/1/2026 | | | |
| **Test Priority:** High | | | | | **Test Executed by:** Fuad Hasan | | | |
| **Module Name:** Order Processing | | | | | **Test Execution date:** 3/1/2026 | | | |
| **Test Title: Verify cashier can process complete order with cash payment** | | | | |  | | | |
| **Description:** End-to-end test of order creation, payment, and receipt generation | | | | |  | | | |
| **Precondition:** Cashier logged in; products Espresso (150 TK) and Croissant (120 TK) in stock | | | | |  | | | |
| **Dependencies:** Printer connected and configured; cash drawer float set | | | | |  | | | |
| **Test Steps** | **Test Data** | | **Expected Results** | | **Actual Results** | | **Status** | |
| 1. Search and add Espresso to cart  2. Search and add Croissant to cart  3. Verify total shows 270 TK  4. Click Proceed to Payment  5. Select Cash payment  6. Enter Amount Tendered: 300  7. Click Complete Payment  8. Verify change shows 30 TK  9. Confirm receipt prints | | Items: Espresso (150), Croissant (120)  Tendered: 300 | | Order completes successfully; receipt prints with correct items, totals, and change; inventory reduced by 1 each for Espresso and Croissant | | Order processed in 45 seconds; receipt accurate; inventory updated; cash drawer balance increased by 270 TK | | **Pass** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Tofayel Hossain** | |
| **Test Case ID:** TC\_006 | | | **Test Designed date:** 14/10/2025 | |
| **Test Priority:** High | | | **Test Executed by:** Tofayel Hossain | |
| **Module Name:** Order Processing | | | **Test Execution date:** 20/10/2025 | |
| **Test Title:** Verify card payment integration | | |  | |
| **Description:** Test payment gateway integration for card transactions | | |  | |
| **Precondition:** Cashier logged in; order ready for payment; Stripe test mode configured | | |  | |
| **Dependencies:** Internet connection; payment gateway credentials valid | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1. Create order with items totaling 500 TK   2. Select Card payment  3. Enter test card number: 4242…  4. Enter expiry: 12/25, CVV: 123  5. Click Process Payment  6. Wait for gateway response | Order Total: 500 TK Test  Card: 4242 4242 4242 4242 | Payment successful; order marked as "Paid by Card"; receipt shows card payment confirmation code | Payment processed in 8 seconds; Stripe returned success code; order completed; no inventory issues | **Pass** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Tofayel Hossain** | |
| **Test Case ID:** TC\_007 | | | **Test Designed date:** 16/10/2025 | |
| **Test Priority:** Medium | | | **Test Executed by:** Tofayel Hossain | |
| **Module Name:** Inventory Management | | | **Test Execution date:** 22/10/2025 | |
| **Test Title:** Verify inventory update on sales | | |  | |
| **Description:** Test automatic stock reduction when order is completed | | |  | |
| **Precondition:** Product "Muffin" has stock quantity 30; Inventory Manager logged in | | |  | |
| **Dependencies:** Inventory tracking enabled | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1. Cashier creates and completes order with 3 Muffins 2. Inventory Manager checks stock for Muffin | Initial Stock: 30  Sold: 3 | New stock shows 27 (30-3); update timestamp matches order completion time | Stock correctly reduced to 27; timestamp accurate; audit log shows reason as "Sale #1023" | **Pass** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Tofayel Hossain** | |
| **Test Case ID:** TC\_008 | | | **Test Designed date:** 16/10/2025 | |
| **Test Priority:** Medium | | | **Test Executed by:** Tofayel Hossain | |
| **Module Name:** Inventory Management | | | **Test Execution date:** 22/10/2025 | |
| **Test Title:** Verify low stock alert generation | | |  | |
| **Description:** Test alert when stock falls below defined threshold | | |  | |
| **Precondition:** Product "Green Tea" threshold set to 10; current stock 11 | | |  | |
| **Dependencies:** Alert system configured; email service active | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1. Cashier sells 2 Green Tea (stock becomes 9) 2. Check Inventory Manager dashboard next day   3. Verify email received at inventory manager's address | Product: Green Tea Threshold: 10 Initial Stock: 11 Sold: 2 | Green Tea appears in Low Stock Alerts list; email sent with subject "Low Stock Alert: Green Tea" | Alert appeared on dashboard; email received at 2 AM daily alert time; contained correct product and current stock level | **Pass** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Tofayel Hossain** | |
| **Test Case ID:** TC\_009 | | | **Test Designed date:** 11/10/2025 | |
| **Test Priority:** High | | | **Test Executed by:** Tofayel Hossain | |
| **Module Name:** User Management | | | **Test Execution date:** 17/10/2025 | |
| **Test Title:** Verify Admin can create cashier account | | |  | |
| **Description:** Test staff enrollment with OTP verification | | |  | |
| **Precondition:** Admin logged in; new cashier details available | | |  | |
| **Dependencies:** SMS gateway and email service configured | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1. Navigate to Staff Management 2. Click "Add Cashier" 3. Enter Name: Sarah, Email: sarah@coffee.com, Mobile: 01711223344 4. Click Create 5. Check SMS and Email for OTP 6. Enter OTP to activate account | Name: Sarah Email: sarah@coffee.com Mobile: 01711223344 | OTP received via SMS and email; account activates after OTP; login credentials sent to email | OTP arrived within 1 minute both channels; account activated successfully; welcome email sent | **Pass** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Tofayel Hossain** | |
| **Test Case ID:** TC\_010 | | | **Test Designed date:** 11/10/2025 | |
| **Test Priority:** Medium | | | **Test Executed by:** Tofayel Hossain | |
| **Module Name:** User Management | | | **Test Execution date:** 17/10/2025 | |
| **Test Title:** Verify password reset OTP functionality | | |  | |
| **Description:** Test OTP-based password recovery for locked account | | |  | |
| **Precondition:** Customer account exists with mobile 01887654321 | | |  | |
| **Dependencies:** Customer knows registered mobile number | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1. Click "Forgot Password" on login 2. Enter mobile: 01887654321 3. Request OTP 4. Enter received OTP 5. Set new password: Customer@2026 6. Login with new password | Mobile: 01887654321 New Password: Customer@2026 | OTP received; password changes successfully; can login with new password; old password no longer works | OTP received in 45 seconds; password reset successful; login worked; old password rejected | **Pass** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Sarker Lamia Yeamin** | |
| **Test Case ID:** TC\_011 | | | **Test Designed date:** 2/1/2026 | |
| **Test Priority:** High | | | **Test Executed by:** Sarker Lamia Yeamin | |
| **Module Name:** Reporting | | | **Test Execution date:** 3/1/2026 | |
| **Test Title: Verify daily sales report accuracy** | | |  | |
| **Description:** Test report generation matches actual transaction | | |  | |
| **Precondition:** Admin logged in; 15 transaction completed today totaling 8,750 TK | | |  | |
| **Dependencies:** Transactions logged in database | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1.Navigate to Reports → DailySales  2. Select today's date  3. Click Generate Report  4. Verify total and transaction count | Expected Total: 8,750 TK  Expected Transactions: 15 | Report shows total 8,750 TK and 15 transactions; lists all products sold with quantities | Report accurate to the taka; all 15 transactions listed; product breakdown correct; generation took 4 seconds | **Pass** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Sarker Lamia Yeamin** | |
| **Test Case ID:** TC\_012 | | | **Test Designed date:** 2/1/2026 | |
| **Test Priority:** Medium | | | **Test Executed by:** Sarker Lamia Yeamin | |
| **Module Name:** Customer Portal | | | **Test Execution date:** 3/1/2026 | |
| **Test Title: Verify customer shopping cart persistence** | | |  | |
| **Description:** Test cart contents saved across sessions | | |  | |
| **Precondition:** Customer logged in; added items to cart | | |  | |
| **Dependencies:** Session management configured for 7-day persistence | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1. Customer adds 2 Cappuccino and 1 Brownie to cart  2. Logout and close browser  3. Reopen browser and login after 3 days  4. Check cart contents | Items: 2x Cappuccino,  1x Brownie | Cart shows same items (2 Cappuccino, 1 Brownie) with quantities preserved | Cart persisted correctly; items and quantities maintained; total calculated accurately | **Pass** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Sarker Lamia Yeamin** | |
| **Test Case ID:** TC\_013 | | | **Test Designed date:** 2/1/2026 | |
| **Test Priority:** High | | | **Test Executed by:** Sarker Lamia Yeamin | |
| **Module Name:** Security | | | **Test Execution date:** 3/1/2026 | |
| **Test Title: Verify role-based access control** | | |  | |
| **Description:** Test that Cashier cannot access Admin functions | | |  | |
| **Precondition:** Cashier logged in; knows direct URL to Admin panel | | |  | |
| **Dependencies:** RBAC middleware active | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1. Cashier tries to access /admin/staff-management directly via URL 2. Observe system response | Direct URL: https://coffeeshop.aiub.edu/admin/staff-man agement | Access denied; redirect to cashier dashboard with "Unauthorized access" message; HTTP 403 logged | Redirected immediately; error message displayed; no access to admin functions; logged in security audit | **Pass** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Sarker Lamia Yeamin** | |
| **Test Case ID:** TC\_014 | | | **Test Designed date:** 2/1/2026 | |
| **Test Priority:** High | | | **Test Executed by:** Sarker Lamia Yeamin | |
| **Module Name:** Product Search | | | **Test Execution date:** 3/1/2026 | |
| **Test Title: Verify real-time product search performance** | | |  | |
| **Description:** Test search speed with 500 products in catalogue | | |  | |
| **Precondition:** System loaded with 500 test products across 10 categories | | |  | |
| **Dependencies:** Database indexing optimized | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1. Cashier types "lat" in search bar 2. Measure time for results to appear  3. Verify results include items with "lat" in name | Search Term: "lat" Catalogue Size: 500 products | Results appear within 1 second;  includes Latte, Matcha Latte, etc. | Search returned 12 matching products in 0.7 seconds;  all relevant items included | **Pass** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Sarker Lamia Yeamin** | |
| **Test Case ID:** TC\_015 | | | **Test Designed date:** 2/1/2026 | |
| **Test Priority:** Medium | | | **Test Executed by:** Sarker Lamia Yeamin | |
| **Module Name:** Order Processing | | | **Test Execution date:** 3/1/2026 | |
| **Test Title: Verify discount validation rules** | | |  | |
| **Description:** Test that discounts >20% require admin PIN | | |  | |
| **Precondition:** Cashier logged in; order total: 1,000 TK | | |  | |
| **Dependencies:** Discount rules configured | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1. Apply 25% discount to order  2. System prompts for Admin PIN  3. Enter correct Admin PIN: 9876 4. Complete order | Order Total: 1,000 TK Discount: 25% (250 TK) Admin PIN: 9876 | PIN accepted; discount applied; order completes with total 750 TK;  discount logged with admin approval | PIN prompt appeared; correct PIN accepted; order completed; audit log shows "Discount 25% approved by Admin (ID: 1)" | **Pass** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Nafisa Anjum Tithi** | |
| **Test Case ID:** TC\_016 | | | **Test Designed date:** 2/1/2026 | |
| **Test Priority:** High | | | **Test Executed by:** Nafisa Anjum Tithi | |
| **Module Name:** Database Integrity | | | **Test Execution date:** 3/1/2026 | |
| **Test Title: Verify referential integrity on product deletion** | | |  | |
| **Description:** Test that product with orders cannot be deleted | | |  | |
| **Precondition:** Product "Sandwich" exists; has historical orders; Admin logged in | | |  | |
| **Dependencies:** Foreign key constraints active | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1. Admin attempts to delete "Sandwich" product  2. Observe system response | Product: Sandwich with existing order history | Deletion blocked; error message "Cannot delete product with existing orders"; product remains in list | Deletion prevented; appropriate error shown; referential integrity maintained; no orphan records created | **Pass** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Nafisa Anjum Tithi** | |
| **Test Case ID:** TC\_017 | | | **Test Designed date:** 2/1/2026 | |
| **Test Priority:** Low | | | **Test Executed by:** Nafisa Anjum Tithi | |
| **Module Name:** Receipt Printing | | | **Test Execution date:** 3/1/2026 | |
| **Test Title: Verify receipt reprint functionality** | | |  | |
| **Description:** Test ability to reprint recent transaction receipt | | |  | |
| **Precondition:** Cashier logged in; transaction #1050 completed 10 minutes ago | | |  | |
| **Dependencies:** Printer has paper; transaction in recent history buffer | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1. Access transaction history 2. Locate transaction #1050 3. Click "Reprint Receipt"  4. Verify receipt prints correctly | Transaction ID: 1050  Age: 10 minutes | Receipt prints identical to original; header shows "REPRINT" mark; all items and totals match original | Reprint successful; content matched original; clearly marked as reprint; printed within 3 seconds | **Pass** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Nafisa Anjum Tithi** | |
| **Test Case ID:** TC\_018 | | | **Test Designed date:** 23/10/2025 | |
| **Test Priority: High** | | | **Test Executed by:** Nafisa Anjum Tithi | |
| **Module Name:** Data Backup | | | **Test Execution date:** 29/10/2025 | |
| **Test Title:** Verify automatic daily backup execution | | |  | |
| **Description:** Test backup runs at 2 AM and is recoverable | | |  | |
| **Precondition:** System has 1 week of transaction data | | |  | |
| **Dependencies:** Backup script scheduled; storage space available | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1. Manually trigger backup process (simulating 2 AM schedule) 2. Verify backup file created 3. Test restore process on separate test database | Database size: ~50MB transactions | Backup file created with timestamp; restore completes successfully; restored data matches original | Backup generated in 3 minutes; file size 45MB; restore test successful; data integrity verified | **Pass** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Nafisa Anjum Tithi** | |
| **Test Case ID:** TC\_019 | | | **Test Designed date:** 22/10/2025 | |
| **Test Priority:** Medium | | | **Test Executed by:** Nafisa Anjum Tithi | |
| **Module Name:** Mobile Responsiveness | | | **Test Execution date:** 28/10/2025 | |
| **Test Title:** Verify mobile checkout flow | | |  | |
| **Description:** Test complete order process on smartphone | | |  | |
| **Precondition:** Customer using mobile browser (iPhone 12); logged in | | |  | |
| **Dependencies:** Mobile-responsive CSS active | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1. Browse products on mobile 2. Add items to cart 3. Proceed to checkout 4. Complete payment 5. Verify order confirmation page layout | Device: iPhone 12 Browser: Safari Items: 2 | All buttons tappable; form fields usable; layout adjusts to screen; no horizontal scrolling needed | UI adapted perfectly; buttons sized correctly; checkout completed successfully; receipt email formatted for mobile | **Pass** |
| **Project Name: Coffee Shop Management System** | | | **Test Designed by: Nafisa Anjum Tithi** | |
| **Test Case ID:** TC\_020 | | | **Test Designed date:** 22/10/2025 | |
| **Test Priority:** Medium | | | **Test Executed by:** Nafisa Anjum Tithi | |
| **Module Name:** Inventory Management | | | **Test Execution date:** 28/10/2025 | |
| **Test Title:**  Verify stock receipt workflow for shipment | | |  | |
| **Description:** Test multi-product stock addition process | | |  | |
| **Precondition:** Inventory Manager logged in; shipment received with 5 product types | | |  | |
| **Dependencies:** Supplier record exists in system | | |  | |
| **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| 1. Initiate "Receive Shipment" 2. Select supplier: "ABC Coffee Suppliers" 3. Enter invoice number: INV-2025-1847 4. Add 5 products with quantities 5. Confirm receipt 6. Verify stock levels updated | Supplier: ABC Coffee Suppliers Invoice: INV-2025-1847 Products: 5 SKUs Total Items: 150 units | All 5 products stock increased; inventory value updated; shipment logged; can view in stock history | All quantities added correctly; total inventory value increased; shipment record created; history shows receipt date and supplier | **Pass** |

# 6. CONCLUSION

The Coffee Shop Management System successfully addresses the operational challenges faced by small and medium-sized coffee shops through comprehensive digitization of core business processes. Our team delivered a robust, role-based platform that streamlines order processing, automates inventory tracking, and provides data-driven business insights. The Agile Scrum methodology enabled iterative development with continuous feedback, ensuring the final product aligns with real-world coffee shop needs. Through meticulous requirement analysis, detailed system design, and extensive testing with 20 comprehensive test cases, we validated all functional and non-functional requirements. The system's intuitive interfaces for Admin, Cashier, Inventory Manager, and Customer roles ensure usability across varying technical skill levels. Key achievements include real-time inventory updates, automated low-stock alerts, multi-payment support, and comprehensive reporting. The Git workflow facilitated seamless collaboration among team members, maintaining code quality and project organization. While developed as an academic project, the system's scalable architecture and modular design provide a foundation for commercial deployment. Future enhancements could include customer loyalty programs, supplier portals, and multi-location support. This project demonstrates how technology can transform traditional businesses, enabling coffee shop owners to make informed decisions, reduce operational costs, and enhance customer satisfaction in an increasingly competitive market.