Software Requirements and Design Document

for

REVAMP: A Smart Marketplace

Prepared by Muhammad Ali, Moaz Farooq, Abdullah Azeem

National University of Computer and Emerging Sciences

October 6, 2024

Table of Contents

1. Introduction

1.1 Purpose

The project, "Revamp: A Smart Marketplace," aims to create a marketplace application that facilitates buying, renting, and servicing items. It provides a streamlined, user-friendly platform for customers, retailers, and service providers.

1.2 Product Scope

This application is a one-stop solution for customers to browse and purchase items, rent products, or request services like refurbishment and repair. Retailers and service providers can list their offerings while administrators ensure seamless operations.

1.3 Title

REVAMP: A Smart Marketplace

1.4 Objectives

- *Provide a unified platform for diverse marketplace interactions.*
- *Improve transparency between customers and service providers.*
- *Streamline the process of renting, buying, and repairing items.*
- *Offer real-time tracking, order management, and complaint resolution.*

1.5 Problem Statement

Existing marketplaces fail to seamlessly integrate buying, renting, and repairing services into one platform. This results in inefficiencies and limited options for consumers. REVAMP addresses these issues by providing a centralized solution, ensuring a streamlined and user-friendly experience.

2. Overall Description

2.1 Product Perspective

The "REVAMP: A Smart Marketplace" application is a ***new, self-contained product*** *designed to address gaps in current online marketplaces. Unlike existing solutions, which often focus solely on buying and selling, REVAMP incorporates renting and repairing services to provide a comprehensive solution.*

The product is not a follow-on member of an existing system but rather an innovative approach to merging diverse functionalities into one cohesive platform. It includes the following key interfaces and integrations:

- ***Customer Interface****: Allows users to browse items, place orders, and request services.*
- ***Retailer Interface****: Enables retailers to list items, manage inventory, and offer discounts.*
- ***Evaluator/Service Provider Interface****: Facilitates evaluation, repair, and refurbishment services.*
- ***Admin Interface****: Maintains system integrity, resolves complaints, and ensures compliance with business rules.*

This system functions independently, while its modular design allows potential integration with external payment gateways, shipment tracking APIs, and third-party authentication services.

2.2 Product Functions

The product must support the following major functions, categorized by user roles:

1. Customer Functions

- ***Account Management****: Register, login, and manage user profiles.*
- ***Browsing and Searching****: Explore products, filter items, and view details.*
- ***Order Management****: Place orders for purchase or rental.*
- ***Service Requests****: Submit requests for evaluation, repair, or refurbishment.*
- ***Payment****: Securely process payments via multiple methods (credit card, cash on delivery, etc.).*
- ***Feedback and Reviews****: Provide ratings and reviews for products and services.*
- ***Order Tracking****: Track the status and location of orders.*

2. Retailer Functions

- ***Inventory Management****: Add, update, or remove items from listings.*
- ***Discount Offers****: Create and manage discounts or promotional offers.*
- ***Order Fulfillment****: Handle orders and update status as processed or delivered.*

3. Service Provider (Evaluator/Refurbisher) Functions

- ***Service Listing****: Offer evaluation, repair, or refurbishment services.*

- ***Request Handling****: Accept or reject service requests based on availability and requirements.*
- ***Service Management****: Track and update service progress.*

4. Admin Functions

- ***System Monitoring****: Oversee all platform operations, including user activities.*
- ***Complaint Resolution****: Manage and resolve customer complaints.*
- ***Policy Enforcement****: Ensure compliance with business rules and security policies.*

2.3 List of Use Cases

- Register Revamp Account
- Rent/Order Item
- Request Service
- Browse Items
- Track Item
- Make Complaint
- Handle Shipment
- Make Payment
- Give Rating/Review
- Evaluate Item
- Accept Service Request
- List Item
- Offer Discounts
- Resolve Complaints

2.4 Extended Use Cases

2.5 Use Case Diagram

3. Other Nonfunctional Requirements

3.1 Performance Requirements

Response Time

- *The system should respond to user actions (e.g., navigating pages, clicking buttons) within 1-2 seconds.*

Data Processing

- *Data operations, such as adding or retrieving items from the database, should complete within 2-3 seconds.*

System Uptime

- *The application should function reliably during user interaction with minimal crashes or errors, targeting 95% uptime during testing and usage.*

Real-Time Updates

- *Updates such as status changes for orders or inventory adjustments should reflect within 5 seconds.*

3.2 Safety Requirements

Data Loss Prevention

- *The application should implement basic safeguards to prevent data loss during crashes or power failures, such as:*
  - *Periodic autosaving of critical data (e.g., active orders or user progress).*
  - *Database backups scheduled at the end of each day.*

Error Handling

- *The application must handle user input errors gracefully, providing clear feedback (e.g., invalid credentials or incorrect item selection) to prevent unintended actions or data corruption.*

Unauthorized Access

- *The system must restrict unauthorized users from accessing sensitive operations or data by:*
  - *Validating user credentials during login.*
  - *Implementing role-based access for features (e.g., administrative actions limited to authorized users).*

External File Validation

- *Any files uploaded or downloaded by the system (e.g., receipts or reports) should be scanned for potential threats and verified to ensure integrity.*

Regulatory Compliance

- *The system should comply with basic data protection guidelines, ensuring that personal information (e.g., user credentials or order details) is not exposed or mishandled.*
- *Avoid storing sensitive data like passwords in plain text.*

User Safety Notifications

- *Provide warnings or confirmations before executing critical actions, such as deleting items, placing orders, or making payments, to prevent accidental operations.*

3.3 Security Requirements

User Authentication

- *Users must log in with valid credentials (username and password) to access the system.*
- *Passwords should be stored securely using hashing techniques.*

Access Control

- *Features are accessible based on user roles:*
    - *Customers: Access to browse items and place orders.*
    - *Retailers: Access to manage inventory and sales.*
    - *Administrators: Access to system management and complaint handling.*

Data Privacy

- *User data (e.g., email addresses, phone numbers) must be encrypted when stored and during transmission.*

Secure Data Transmission

- *All communication between the client and the database must use a secure protocol to prevent unauthorized access.*

Session Management

- *Sessions should automatically expire after inactivity.*
- *Logging out should invalidate active sessions.*

Error Handling

- *Error messages should not expose sensitive details, such as database queries or system paths.*

Data Retention and Deletion

- *User data should only be kept as long as necessary for system functionality.*
- *Users should be able to request account deletion to remove their data.*

3.4 Software Quality Attributes

1. ***Usability***
   - *The system must provide an intuitive interface that can be used without extensive training.*
   - *User feedback and error messages must be clear and helpful to ensure smooth interaction.*
2. ***Reliability***
   - *The application should maintain consistent performance during user interactions, with minimal crashes or errors.*
3. ***Maintainability***
   - *The codebase must be modular and well-documented to facilitate updates and bug fixes.*
4. ***Flexibility***
   - *The system should allow for easy addition of new features (e.g., additional payment methods or inventory management tools).*
5. ***Testability***
   - *All critical features should be testable, with automated tests covering at least 80% of the core functionality.*
6. ***Portability***
   - *The application should run on any platform that supports JavaFX, such as Windows, macOS, and Linux.*
7. ***Robustness***
   - *The system must handle invalid input gracefully and recover from errors without compromising data integrity.*

3.5 Business Rules

1. ***Role-Based Access***
   - *Customers can browse items, place orders, and provide reviews.*
   - *Retailers can manage inventory, add discounts, and list products.*
   - *Administrators handle complaints and oversee system integrity.*
2. ***Data Integrity***
   - *Customers can only modify their own orders or reviews.*
   - *Retailers cannot alter customer reviews.*
3. ***Transaction Policies***
   - *Payments must be processed before confirming an order.*
   - *Refunds or cancellations should follow predefined conditions set by administrators.*
4. ***Service Request Handling***
   - *Evaluators or refurbishers must approve service requests before they are confirmed.*
5. ***Item Availability***
   - *Orders or rentals can only proceed if items are in stock or available for the requested period.*

3.6 Operating Environment

1. ***Hardware Requirements***
   - *Minimum of 4 GB RAM and dual-core processor for smooth operation.*
   - *Screen resolution of 1024x768 or higher for optimal display.*
2. ***Operating Systems***
   - *Windows 10 or later.*
   - *macOS 10.15 (Catalina) or later.*
   - *Any Linux distribution with JavaFX runtime support.*
3. ***Software Dependencies***
   - *Java Development Kit (JDK) version 11 or higher.*
   - *JavaFX runtime libraries.*
   - *Relational database (e.g., MySQL or SQLite) for backend data storage.*
4. ***Network Requirements***
   - *Internet connection required for certain features, such as OTP validation and real-time tracking.*

3.7 User Interfaces

1. ***Login and Signup Pages***
   - *Fields for username and password, with clear error messages for invalid input.*
   - *A "Forgot Password" link for account recovery.*
2. ***Customer Dashboard***
   - *Displays browsing categories, order history, and recommendations.*
   - *Clear buttons for actions like placing orders and requesting services.*
3. ***Retailer Dashboard***
   - *Sections for managing inventory, viewing sales data, and applying discounts.*
   - *Simple forms for listing new items with fields for description, price, and stock quantity.*
4. ***Service Provider Interface***
   - *Displays pending service requests with options to accept or reject.*
   - *Forms for updating service progress and uploading reports.*
5. ***Admin Panel***
   - *Overview of user activities, complaint management, and system monitoring.*
   - *Tools for resolving disputes and ensuring data consistency.*
6. ***Standard Buttons and Features***
   - *Consistent use of buttons like "Save," "Cancel," and "Submit."*
   - *Keyboard shortcuts for navigation and frequently used actions.*
   - *Error messages and confirmation dialogs to prevent unintended actions.*

4. Domain Model

5. System Sequence Diagram

6. Sequence Diagram

7. Class Diagram

8. Component Diagram

9. Package Diagram

10. Deployment Diagram