LONDON
METROPOLITAN
UNIVERSITY

CS6004ES –Application Development

Coursework – 2 (2024/25)

Final Report

E-Book Web Application

Group Members:

23057846 – Kanagasabai Tharshan

23057852 – Vimalanathan Thiviya

Submission deadline: 24 November 2024

Table of Contents

No table of contents entries found.

Table of Figures

Table of Tables

Abstract

This project focuses on the development and implementation of an E-Book Web Application System designed to modernize the process of buying and selling used books in Sri Lanka. The system provides a user-friendly and efficient platform to connect book sellers and buyers, enhancing the overall experience of online book transactions. Key features include customer registration, book search functionality, secure ordering, and feedback management for customers, as well as book and order management, report generation, and an admin dashboard for administrators.

For customers, the platform offers account creation, order tracking, and personalized experiences through features such as search, feedback submission, and a user-friendly interface. Administrators benefit from tools to manage book details, oversee customer accounts, and generate sales reports. The application prioritizes data security, robust access control, and usability, ensuring a safe and accessible environment for all users.

The system is developed using the ASP.NET MVC framework with C# programming language, leveraging object-oriented design principles such as encapsulation, inheritance, and abstraction to achieve high cohesion and loose coupling. For the database, MySQL was selected due to its open-source nature, speed, scalability, and ability to handle significant transaction volumes efficiently. Front-end technologies such as HTML, CSS, and JavaScript complement the backend for a responsive and dynamic user interface.

This document outlines the planning, design, and development processes employed in creating the application, including system diagrams, the database schema, and a detailed project plan. By addressing the growing demand for an accessible and scalable online platform for books, this project provides a modern solution that enhances customer satisfaction and operational efficiency. The successful implementation of this system could significantly impact the used book retail industry in Sri Lanka, providing a convenient and comprehensive alternative to traditional physical bookstores.

Introduction

In the modern digital era, the rapid evolution of technology has reshaped industries worldwide, including the retail sector. The convenience and accessibility offered by online platforms have transformed the way consumers interact with businesses, and the book retail industry is no exception. With the rise in demand for digital shopping experiences, there is an increasing need for an efficient and user-friendly system that caters to the dynamic requirements of book buyers and sellers. This project presents the E-Book Web Application System, designed to modernize the process of buying and selling used books in Sri Lanka.

The E-Book Web Application System bridges the gap between traditional brick-and-mortar bookstores and the growing demand for online platforms. It serves as a comprehensive solution that enables book buyers and sellers to interact seamlessly. For sellers, the system offers tools to manage book listings with detailed descriptions, pricing, and stock availability. For buyers, it provides an intuitive interface to search for books, place orders, and provide feedback, all while enjoying a personalized and efficient shopping experience.

To ensure accessibility and security, the system incorporates advanced features such as secure user authentication, order tracking, and feedback submission. Administrators are equipped with tools to manage book details, oversee customer accounts, track orders, and generate sales reports. The system also supports multiple payment options, making it inclusive for a wide range of users.

The development of this system leverages ASP.NET MVC with C#, adhering to object-oriented design principles such as encapsulation, abstraction, and inheritance. By combining these advanced methodologies with technologies like HTML, CSS, and JavaScript, the system ensures a robust backend and responsive frontend. A scalable and reliable database structure is achieved using MySQL, which effectively manages user accounts, orders, and book data.

The E-Book Web Application System aims to transform the used book retail market in Sri Lanka by providing an innovative and efficient platform for users. By streamlining operations, improving accessibility, and enhancing customer satisfaction, this system addresses the needs of a growing market. This introduction highlights the system’s purpose and capabilities, laying the foundation for a detailed exploration of its design, implementation, and anticipated impact on the book retail industry.

Aim

The E-Book Web Application System aims to create an efficient, scalable, and user-friendly platform for buying and selling used books online. By leveraging advanced web technologies and secure payment systems, the application enhances accessibility, convenience, and user satisfaction for both buyers and administrators. It provides key functionalities such as robust catalog management, user account creation, order management, and personalized book recommendations.

The system is designed to streamline operations, simplify the book purchasing process, and provide a secure and engaging platform that meets the evolving needs of modern consumers. Ultimately, it seeks to revolutionize the book retail industry by offering a reliable and innovative online alternative to traditional brick-and-mortar bookstores.

Objectives

The objectives of the E-Book Web Application System are as follows.

1. Develop a User-Friendly Platform

Create an intuitive and responsive web application that allows customers to easily browse, search, and purchase books online.

2. Enable Customer Account Management

Provide functionalities for users to register, log in, manage their profiles, and track their orders seamlessly.

3. Implement Robust Book Management Features

Allow administrators to efficiently manage book details, including adding, updating, deleting, and viewing inventory information.

4. Facilitate Secure Transactions

Integrate secure payment gateways to ensure safe and reliable online transactions for customers.

5. Personalize User Experience

Offer personalized book recommendations based on customers' browsing and purchase history to enhance user satisfaction and engagement.

6. Simplify Order Management

Enable customers to place, view, and delete orders, while administrators can oversee and manage all orders effectively.

7. Provide Feedback Mechanisms

Allow customers to leave feedback on books, contributing to a more interactive and customer-focused platform.

8. Generate Comprehensive Reports

Equip administrators with tools to generate reports on sales, orders, and customer activity for business insights and decision-making.

9. Ensure Scalability and Reliability

Design the system to handle increasing user traffic and large-scale operations without compromising performance.

10. Streamline Business Operations

Enhance operational efficiency by bridging the gap between traditional bookstores and modern online retailing, addressing the growing demand for digital book shopping solutions.

Scope

The E-Book Web Application System is designed to cater to the needs of both customers and administrators, providing a comprehensive platform for buying and selling used books online. The scope of this project includes the following aspects:

Customer Features

1. User Registration and Login:

- New customers can register and create accounts.
- Registered users can log in to access additional features such as order management and feedback.

2. Book Search and Browsing

- All users, regardless of registration, can search and browse the available books by categories, authors, or keywords.

3. Order Management

- Logged-in customers can place, view, and delete their orders conveniently.

4. Feedback Submission

- Customers can leave feedback and reviews on books they have purchased, fostering an interactive community.

Administrator Features

1. Book Management

- Administrators can add, edit, delete, and view book details, including descriptions, pricing, stock levels, and availability.

2. Customer Management

- Administrators can view and manage customer accounts, ensuring smooth operations and resolving issues.

3. Order Oversight

- The system allows administrators to track and manage all customer orders, ensuring timely processing and delivery.

4. Report Generation

- Administrators can generate detailed reports on sales, customer activity, and inventory for business insights and decision-making.

5. Admin Dashboard

- A centralized dashboard provides a comprehensive overview of system metrics, including sales data, user activity, and book inventory.

Technical Scope

1. Technology Stack

- The application is built using the ASP.NET MVC framework with C# for both front-end and back-end development.
- It incorporates HTML, CSS, and JavaScript for an interactive user interface and a seamless browsing experience.

2. Database Management

- The system uses MySQL to manage data such as customer accounts, book details, orders, and feedback efficiently.

3. Security and Scalability

- The platform implements secure authentication mechanisms to protect user data and supports multiple payment options for secure transactions.
- It is designed to handle increasing user traffic and data volumes, ensuring reliability and performance.

4. Cross-Platform Accessibility

- The system is optimized for both desktop and mobile devices, enabling users to access the platform from anywhere.

Limitations

1. Geographic Scope

- Initially, the system targets customers and sellers in Sri Lanka but can be expanded to other regions in the future.

2. Payment Gateways

- The project supports a limited number of payment methods during the initial implementation phase, with plans for future expansion.

This project aims to address the growing demand for online book retailing by providing an efficient, secure, and user-friendly platform that benefits both buyers and administrators.

System Analysis

What is System Analysis?

System Analysis is the process of thoroughly examining and understanding a system to identify its components, functionality, and requirements. It plays a critical role in software development by establishing a clear foundation for designing and implementing an efficient and user-friendly system. The process involves gathering and analyzing user requirements, identifying problems in the current system (if one exists), and defining the scope and objectives of the proposed solution. Key activities in system analysis include requirement gathering, feasibility studies, problem identification, and creating models such as UML diagrams and ER diagrams to visualize the system's structure and data flow. System analysis ensures that the system is feasible, cost-effective, and capable of addressing stakeholder needs while minimizing risks and enhancing overall efficiency.

For the E-Book Web Application System, system analysis is vital for understanding the specific needs of both customers and administrators. It helps define essential functionalities like customer account management, book catalog management, secure order processing, and feedback mechanisms. Additionally, system analysis identifies limitations in traditional book-selling methods and proposes solutions to overcome these challenges using modern web technologies. By conducting a comprehensive system analysis, this project ensures the development of a scalable, secure, and user-centric platform that meets the evolving demands of the book retail industry.

System Overview

The E-Book Web Application System is a comprehensive online platform designed to facilitate the buying and selling of used books, catering to the needs of both customers and administrators. It addresses the limitations of traditional book-selling methods by providing a modern, digital solution that enhances accessibility, convenience, and operational efficiency. The system is developed with a focus on scalability, security, and user experience, leveraging advanced web technologies and object-oriented design principles.

The platform offers distinct functionalities for two primary user groups: customers and administrators. Customers can register and log in to create personalized accounts, allowing them to browse the book catalog, search for specific titles, place orders, track their purchases, and provide feedback on books. For administrators, the system includes tools to manage books, oversee customer accounts, process orders, and generate reports. Additionally, an admin dashboard provides a centralized view of key metrics such as sales data, inventory levels, and user activity.

Built using the ASP.NET MVC framework with C# programming language, the system combines robust back-end functionality with a responsive front-end developed using HTML, CSS, and JavaScript. The database is powered by MySQL, ensuring efficient management of large volumes of data, including book details, user accounts, orders, and feedback. Secure authentication mechanisms protect user data, while the system's design supports multiple payment options, making it accessible to a diverse range of users.

The E-Book Web Application System not only simplifies the process of buying and selling books but also fosters a seamless and enjoyable user experience. Its modular and scalable design ensures that it can accommodate future enhancements, making it a reliable and long-term solution for the evolving needs of the book retail industry. This system provides a significant step forward in transforming traditional bookstores into a digital marketplace, offering a convenient alternative for book enthusiasts and sellers.

Current System

The existing manual book-selling system predominantly relies on physical bookstores, where transactions require buyers to visit the store in person to browse and purchase books. This approach often proves to be time-consuming, as buyers are limited by the store's opening hours and the availability of inventory. Inventory management in this system is handled manually, which can lead to inaccuracies and inefficiencies, such as overstocking or stock shortages. Marketing efforts in the current system are restricted to in-store displays or local advertising, limiting the reach to a broader audience and reducing visibility for potential customers. Customer support is primarily offered face-to-face or through telephone communication, which may not always efficiently address inquiries or resolve issues promptly. Overall, the manual system lacks the convenience, accessibility, and operational efficiency that a digital platform could provide, highlighting the need for an online book-selling solution to enhance the buying and selling experience for both customers and sellers.

Proposed System

The proposed E-Book Web Application System is designed to modernize and digitize the traditional book-selling process, offering a streamlined and efficient platform for buyers and sellers. This web-based application will empower users with functionalities such as user registration, secure login, and personalized accounts to create a tailored and convenient experience. Sellers will be able to list books with detailed descriptions, manage inventory in real-time, and launch promotional campaigns to attract customers. Buyers will have access to advanced search and filtering options to easily find specific titles or categories, personalized recommendations based on their browsing and purchase history, and a secure payment gateway with multiple payment options for convenience.

In addition to simplifying transactions, the system will offer robust features such as order tracking, live chat, and helpdesk services for enhanced customer support. Data analytics tools will enable sellers to gain valuable insights into sales trends and customer behavior, helping them make informed business decisions. The proposed system is designed to be scalable, mobile-friendly, and capable of handling high volumes of traffic, ensuring accessibility and reliability. By addressing the limitations of the current manual process, this system will significantly improve the efficiency, convenience, and overall experience of book retailing for both buyers and sellers.

User Classes and Characteristics

The E-Book Web Application System serves two primary user classes, each with distinct roles and characteristics.

1. Customers

Characteristics:

- **Role:**
  - Customers are individuals who use the platform to search for, purchase, and provide feedback on books.
- **Access:**
  - They are required to register and log in to access personalized features such as order management and feedback submission.
- **Diversity:**
  - Includes a wide range of users, from casual readers to academic professionals, with varying levels of technical proficiency.
- **Key Interactions:**
  - Browse and search for books across categories.
  - Place orders and track their purchases.
  - Save favorite books for future reference.
  - Provide feedback and reviews on purchased books.

Needs and Expectations:

- Easy navigation and a user-friendly interface.
- Secure registration and login processes.
- Personalized book recommendations based on browsing and purchase history.
- Real-time updates on book availability and order status.
- Multiple payment options for convenience.

2. Administrators

Characteristics:

- **Role:**
  - Administrators manage the operational and backend aspects of the system, ensuring smooth functionality and data integrity.
- **Access:**
  - Admins have privileged access to manage books, customers, and orders, as well as generate reports and oversee system activities.
- **Technical Proficiency:**
  - Typically possess a higher level of technical expertise compared to customers, enabling them to handle system configurations and management tasks effectively.

Key Responsibilities:

- Add, update, and delete book details in the catalog.
- Manage customer accounts, including resolving account-related issues.
- Monitor and process customer orders to ensure timely fulfillment.
- Generate and analyze reports on sales, inventory, and user activity.
- Oversee feedback and address any customer complaints or concerns.

Needs and Expectations:

- A robust dashboard with an overview of system metrics and activities.
- Efficient tools for managing books, orders, and customer accounts.
- Real-time notifications for critical updates (e.g., low stock levels, pending orders).
- Advanced data analytics to inform business decisions and strategies.

User Class Comparison

Future / Functionality	Customers	Administrators
Registration / Login	Required for personalization	Required for system access
Book Browsing/Search	Yes	No
Manage Books	No	Yes
Place/Manage Orders	Yes	Oversee all orders
Feedback Submission	Yes	Monitor and respond
Generate Reports	No	Yes
View Dashboard	No	Yes

Requirement Analysis

The Requirement Analysis phase is a critical step in the development of the E-Book Web Application System. It identifies and documents the functional and non-functional requirements of the system, ensuring it meets the needs of all stakeholders effectively. These requirements serve as the foundation for designing and implementing the system.

1. Functional Requirements

Customer Requirements:

1. **Registration and Login:**
   - Customers must be able to register by providing details such as name, email, and password.
   - Secure login functionality must be available for registered customers.
2. **Book Browsing and Searching:**
   - Users should be able to browse books by categories such as genre, author, or price.
   - A search feature should allow users to find specific books by keywords or filters.
3. **Order Management:**
   - Customers must be able to add books to a cart and place orders securely.
   - Users should have the ability to view and delete their orders.

4. **Feedback System:**
    - Customers should be able to provide feedback and reviews on books they have purchased.
5. **Personalized Recommendations:**
    - Display book recommendations based on the user’s browsing and purchase history.

Administrator Requirements:

1. **Book Management:**
    - Admins should be able to add, update, delete, and view book details such as title, author, price, stock, and description.
2. **Customer Management:**
    - Admins must have access to manage customer accounts, including resolving issues or deleting inactive accounts.
3. **Order Management:**
    - Admins should be able to view, process, and manage all customer orders.
4. **Report Generation:**
    - The system should allow admins to generate detailed reports on sales, inventory, and user activity.
5. **Dashboard:**
    - Provide a dashboard with a summary of sales, inventory levels, pending orders, and system metrics.

2. Non-Functional Requirements

Performance Requirements:

- The system must be able to handle at least 500 concurrent users without performance degradation.
- Page load times should not exceed 3 seconds under normal conditions.

Security Requirements:

- All user data must be securely stored in an encrypted format.
- Implement role-based access control to ensure data privacy for customers and administrators.
- Ensure secure payment processing using trusted payment gateways.

Scalability Requirements:

- The system must be designed to accommodate future growth, including an increase in user base and book listings.

Usability Requirements:

- The interface must be user-friendly and intuitive for users with varying levels of technical proficiency.
- The application must be compatible with both desktop and mobile devices.

Reliability Requirements:

- The system should ensure 99.9% uptime to guarantee availability for users.
- Backup mechanisms must be implemented to prevent data loss in case of system failures.

Maintainability Requirements:

- The codebase must follow modular design principles to facilitate updates and maintenance.
- Detailed documentation should be provided for developers to understand and modify the system.

Compliance Requirements:

- The application must comply with data protection regulations, such as GDPR, to ensure user privacy.

Prioritization of Requirements

Requirment	Priority
Secure Registration/Login	High
Book Browsing/Search	High
Order placement and Tracking	High
Feedback Submission	Medium
Book and Customer Management	High
Report Generation	Medium
Personalized Recommendations	Medium
Scalability	High
Security	High

The Requirement Analysis ensures that the E-Book Web Application System aligns with user needs and technical constraints, providing a comprehensive guide for the subsequent design and development phases.

Resource Requirement

The development of the E-Book Web Application System requires a combination of hardware, software, and human resources. These resources are essential to ensure the successful design, implementation, and deployment of the system.

1. Hardware Requirements

Development Environment:

- **Developer Machines:**
  - Minimum Specifications:
    - Processor: Intel Core i5 or equivalent
    - RAM: 8 GB
    - Storage: 250 GB SSD
    - Operating System: Windows 10 or higher
- **Server:**
  - Specifications for hosting the web application:
    - Processor: Quad-Core (Intel Xeon or equivalent)
    - RAM: 16 GB
    - Storage: 500 GB SSD
    - Operating System: Windows Server 2019 or Linux (CentOS/Ubuntu)
    - Network: High-speed internet connection (1 Gbps)

Client Devices:

- Desktop, laptop, tablet, or smartphone with internet access and a modern web browser (e.g., Chrome, Firefox, Safari).

2. Software Requirements

Development Tools:

- **Integrated Development Environment (IDE):**
  - Visual Studio 2019 or higher (Community/Enterprise Edition) for ASP.NET MVC development.
- **Database Management System (DBMS):**
  - MySQL for managing the system's data.

- **Version Control:**
    - Git for source code management and collaboration.
- **Design Tools:**
    - Draw.io or Lucidchart for creating UML diagrams and ER diagrams.
    - Figma or Adobe XD for UI/UX design.

Technologies and Frameworks:

- **Backend Framework:**
    - ASP.NET MVC (C#) for server-side development.
- **Frontend Technologies:**
    - HTML, CSS, JavaScript for creating the user interface.
    - Bootstrap for responsive design.
- **Payment Integration:**
    - APIs for secure payment gateways (e.g., PayPal, Stripe).
- **Analytics Tools:**
    - Google Analytics for tracking user activity and sales trends.

Testing Tools:

- Selenium for automated UI testing.
- Postman for testing APIs.
- NUnit for unit testing.

3. Human Resources

Development Team:

- **Project Manager:**
    - Responsible for overseeing the project timeline, resources, and deliverables.
- **Backend Developer:**
    - Focuses on implementing server-side logic using ASP.NET MVC and managing the database.
- **Frontend Developer:**
    - Designs and implements the user interface using HTML, CSS, JavaScript, and Bootstrap.
- **Database Administrator:**
    - Manages the database schema, data integrity, and backup processes.

- **Quality Assurance (QA) Tester:**
    - Conducts functional, usability, and performance testing to ensure the system meets requirements.

Other Roles:

- **UI/UX Designer:**
    - Designs an intuitive and responsive user interface.
- **System Analyst:**
    - Analyzes user requirements and ensures the system aligns with business needs.
- **Technical Support:**
    - Provides post-deployment support to maintain system performance and resolve user issues.

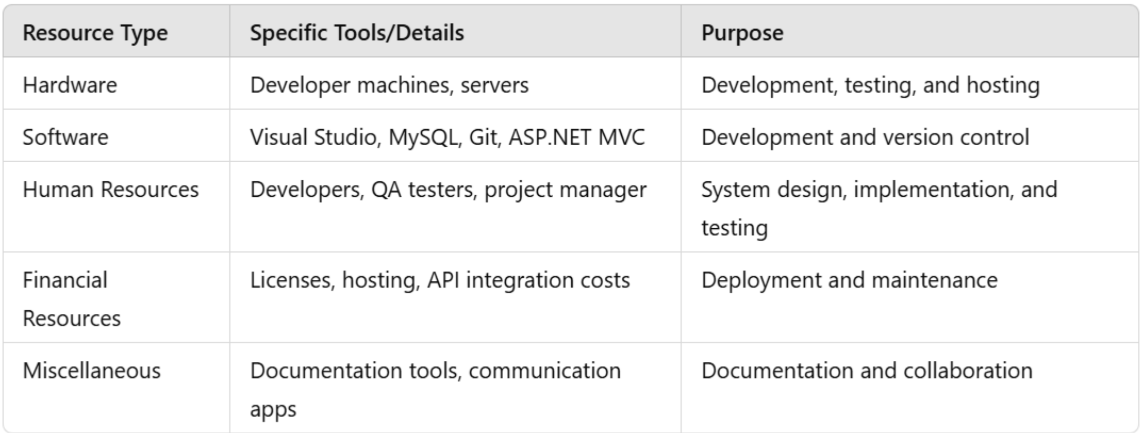
4. Financial Resources

- **Licenses:**
  - Visual Studio (if using Enterprise Edition) and any premium database tools.
- **Hosting Costs:**
  - Cloud-based server hosting for deployment (e.g., Azure, AWS, or Google Cloud).
- **API Costs:**
  - Fees associated with payment gateway integrations like Stripe or PayPal.
- **Miscellaneous:**
  - Budget for software subscriptions, design tools, and testing utilities.

5. Other Resources

- **Documentation Tools:**
  - Microsoft Word and Excel for creating reports and system documentation.
- **Communication Tools:**
  - Microsoft Teams, Slack, or Zoom for team collaboration.
- **Issue Tracking Tools:**
  - Jira or Trello for managing tasks, bugs, and project milestones.

Summary of Resource Requirements



Resource Type	Specific Tools/Details	Purpose
Hardware	Developer machines, servers	Development, testing, and hosting
Software	Visual Studio, MySQL, Git, ASP.NET MVC	Development and version control
Human Resources	Developers, QA testers, project manager	System design, implementation, and testing
Financial Resources	Licenses, hosting, API integration costs	Deployment and maintenance
Miscellaneous	Documentation tools, communication apps	Documentation and collaboration

These resources are critical to ensure the **E-Book Web Application System** is developed, tested, and deployed successfully, meeting both technical and business objectives.

What is System Methodology?

System Methodology refers to the structured approach or framework used to plan, design, develop, implement, and maintain a system. It provides a clear roadmap for managing and organizing the various stages of system development. System methodology ensures that the development process is efficient, consistent, and aligned with the project goals, addressing both technical and business requirements.

1. **Structured Process:**
   A systematic approach to breaking down the system development lifecycle into manageable phases.
2. **Defined Roles:**
   Clarifies responsibilities among team members, such as developers, designers, testers, and project managers.
3. **Clear Deliverables:**
   Outlines the expected outputs at each phase, such as diagrams, prototypes, code, and documentation.
4. **Flexibility and Adaptability:**
   Depending on the chosen methodology, allows the project to adapt to changing requirements or unforeseen challenges.

Types of System Methodologies

There are several methodologies, each suited to different types of projects:

1. **Waterfall Methodology:**

- A linear and sequential approach where each phase (e.g., requirements, design, implementation) is completed before moving to the next.
- Best for projects with well-defined requirements and little expected change.

2. **Agile Methodology:**
    - An iterative and incremental approach that emphasizes flexibility, collaboration, and customer feedback.
    - Suitable for projects with evolving requirements and a need for quick delivery.
3. **Spiral Methodology:**
    - Combines iterative development with risk assessment, focusing on refining the system through repeated cycles.
    - Ideal for complex projects with high levels of uncertainty.
4. **Prototyping:**
    - Focuses on building a prototype to gather user feedback and refine requirements.
    - Useful for projects where requirements are not clearly defined at the outset.
5. **RAD (Rapid Application Development):**
    - Prioritizes speed and flexibility, with a focus on developing functional components quickly.
    - Best for projects with a tight timeline.
6. **Scrum (Subset of Agile):**
    - Involves iterative development cycles called sprints, with a focus on collaboration and incremental delivery.

Importance of System Methodology

- **Efficiency:** Ensures resources are used effectively by following a well-defined process.
- **Consistency:** Provides a standardized approach, making it easier to manage and track progress.
- **Risk Management:** Identifies and addresses potential risks early in the development process.

- **Quality Assurance:** Promotes thorough testing and validation to deliver a reliable and functional system.
- **User Satisfaction:** Involves stakeholders throughout the development process, ensuring the system meets their needs.

Waterfall Methodology

The **Waterfall Methodology** is a linear and sequential approach to system development where each phase of the project is completed before moving on to the next. It follows a structured process that typically includes distinct phases: requirements gathering, system design, implementation, testing, deployment, and maintenance. In this methodology, the output of one phase acts as the input for the next, ensuring that progress flows in one direction, like a waterfall. Waterfall methodology is best suited for projects with well-defined requirements that are unlikely to change during development. It provides a clear roadmap, making it easier to manage and document each phase. However, its rigid structure can make it challenging to accommodate changes once a phase is completed. Despite this limitation, the Waterfall Methodology is often used in projects where deliverables and timelines are fixed, providing a systematic framework for achieving well-documented and predictable outcomes.

Agile Methodology

The **Agile Methodology** is an iterative and flexible approach to software development that emphasizes collaboration, customer feedback, and adaptability. Unlike traditional linear models, Agile breaks the project into smaller, manageable iterations or "sprints," each delivering a functional increment of the system. This methodology encourages constant communication among team members and stakeholders, allowing for continuous refinement of requirements and adjustments to changes. Agile prioritizes delivering value to users quickly by focusing on high-priority features first. It promotes teamwork, testing, and regular feedback to ensure the product meets evolving user needs. Agile is especially suitable for projects with dynamic requirements, where flexibility and incremental progress are essential. By fostering a customer-centric and adaptive development process, Agile improves efficiency, reduces risks, and enhances the quality of the final product.

Spiral Methodology

The **Spiral Methodology** is a risk-driven approach to software development that combines elements of iterative and sequential models. It emphasizes continuous refinement through multiple development cycles, or "spirals," each consisting of four key phases: planning, risk analysis, development, and evaluation. At each iteration, the system is progressively developed by refining the design, implementing features, and conducting thorough testing. This methodology is particularly well-suited for large, complex projects with high levels of uncertainty, as it allows for the identification and mitigation of risks early in the development process. The Spiral Methodology promotes flexibility, enabling teams to incorporate feedback and adapt to changes at any stage. By focusing on risk assessment and iterative improvement, this model ensures that the final product meets user needs while minimizing potential challenges during development.

System Methodology for E-Book Web Application System

The development of the **E-Book Web Application System** employs the **Agile Methodology**, a flexible and iterative approach that is well-suited for modern web application projects with evolving requirements. Agile focuses on delivering value incrementally through continuous collaboration, customer feedback, and adaptability. This methodology ensures the system is developed efficiently while addressing the dynamic needs of both customers and administrators.

Why Agile Methodology?

Agile was selected as the methodology for this project because:

1. **Incremental Delivery:** The system can be developed and deployed in smaller, functional increments, ensuring faster delivery of core features.
2. **Stakeholder Involvement:** Regular feedback from stakeholders ensures the system aligns with their expectations and requirements.
3. **Flexibility:** Agile allows the team to adapt to changes in requirements or priorities throughout the project lifecycle.
4. **Quality Assurance:** Frequent testing during each iteration helps identify and resolve issues early, resulting in a more reliable system.

Phases of Agile Methodology for E-Book Web Application System

1. **Requirement Gathering and Analysis:**
    - Stakeholders, including customers and administrators, collaborate with the development team to define functional and non-functional requirements.
    - Features such as user registration, book management, order processing, and feedback mechanisms are prioritized.
2. **Planning:**
    - The project is divided into sprints, with each sprint focusing on specific modules or functionalities (e.g., user management, order tracking).
    - A backlog of tasks is created, detailing the work to be completed in each sprint.
3. **Design:**
    - System diagrams such as use case diagrams, class diagrams, and ER diagrams are created to visualize the system architecture.
    - Wireframes and prototypes are designed to provide a blueprint for the user interface and user experience.
4. **Development:**
    - Each sprint involves the development of specific features, such as secure login, book search functionality, and personalized recommendations.
    - The backend is developed using **ASP.NET MVC with C#**, while the frontend is implemented using **HTML, CSS, JavaScript, and Bootstrap**.
    - The database is designed and managed using **MySQL** to store and retrieve system data efficiently.
5. **Testing:**
    - Continuous testing is conducted during each sprint, including unit testing, integration testing, and user acceptance testing.
    - Tools like Selenium (for automated UI testing) and Postman (for API testing) are utilized to ensure system functionality and reliability.
6. **Deployment:**
    - At the end of each sprint, functional modules are deployed for review and feedback.
    - The final product is deployed on a cloud-based server (e.g., Azure or AWS) for public access.
7. **Maintenance and Updates:**

    - Post-deployment, the system is monitored for bugs, performance issues, and user feedback.
    - Regular updates are made to improve features and address new requirements.

Benefits of Agile Methodology for This Project

- **Faster Delivery:** High-priority features, such as user registration and book search, are developed and delivered early.
- **Continuous Improvement:** Stakeholder feedback ensures the system evolves to meet user needs effectively.
- **Flexibility:** Changes to requirements can be accommodated without disrupting the overall development process.
- **Risk Mitigation:** Frequent testing reduces the likelihood of major defects or failures.

UML Diagram

The complete design process used throughout the project's development cycle is described in this section. Several UML (Unified Modeling Language) diagrams were made to depict the application's overall architecture and logical flow once the requirements gathering and feasibility analysis were finished. Use case diagrams, UML diagrams, database diagrams, and sequence diagrams are among the crucial diagrams included in this area. These designs are essential for upcoming improvements and act as the application's blueprint. (geeksforgeeks, geeksforgeeks, 2024)

Use case Diagram

The features that users and administrators can access within the e-book web application are displayed in the use-case diagram. Registering, logging in, ordering books, managing orders, searching for books, and leaving comments are all available to customers. Admins can access the admin dashboard, create reports, manage orders, customer data, and book details. The system's capabilities and user-application interactions are succinctly summarized in this diagram, which aids in the creation and understanding of key features.

Class Diagram

The class diagram lists the main classes—Customer, Admin, Book, and Order—as well as their properties and functions in the e-book web application. Order placing, user registration, login, and management responsibilities are handled by the customer and admin classes. While the Order class maintains order details, the Book class handles book-related tasks like searching. Relationships demonstrate how classes interact, guaranteeing system coherence and full capability for effective e-book administration.

Entity-Relationship Diagram

The entity-relationship diagram (ERD) depicts the database layout for the e-book web application, featuring entities such as Customer, Admin, Book, Order, and Feedback. It highlights primary keys, attributes, and foreign keys, outlining relationships like customersplacing orders and giving feedback, admins overseeing customers, books, and orders. Theserelationships establish a well-defined data model crucial for database design and ensuring effective data management within the system.

Deployment Diagram for E-Book Web Application System

A **Deployment Diagram** represents the physical architecture of a system, showing how components are deployed across servers, clients, and networks. Below is a detailed description of the deployment diagram for the **E-Book Web Application System**:

Description of the Deployment Architecture

1. **Client Tier:**
    - Represents end-user devices, such as desktops, laptops, tablets, or smartphones.
    - Users access the web application through a web browser (e.g., Chrome, Firefox, Safari).
    - Interacts with the application using the user interface rendered by the frontend technologies (HTML, CSS, JavaScript).
2. **Application Tier:**
    - Hosted on a web server, responsible for handling the business logic of the application.
    - Developed using the **ASP.NET MVC Framework**, this layer processes requests from the client, interacts with the database, and returns appropriate responses.
    - Includes modules for user authentication, book catalog management, order processing, and feedback handling.
3. **Database Tier:**
    - A dedicated database server stores and manages data for the application.
    - Uses **MySQL** for efficient handling of user data, book details, order records, and feedback.
    - Supports secure data storage with encryption for sensitive information such as passwords and transaction details.
4. **Network Infrastructure:**
    - The client tier communicates with the application tier via the **Internet** using secure HTTP/HTTPS protocols.
    - The application tier interacts with the database tier over a secure connection to ensure data integrity.

Components in the Deployment Diagram

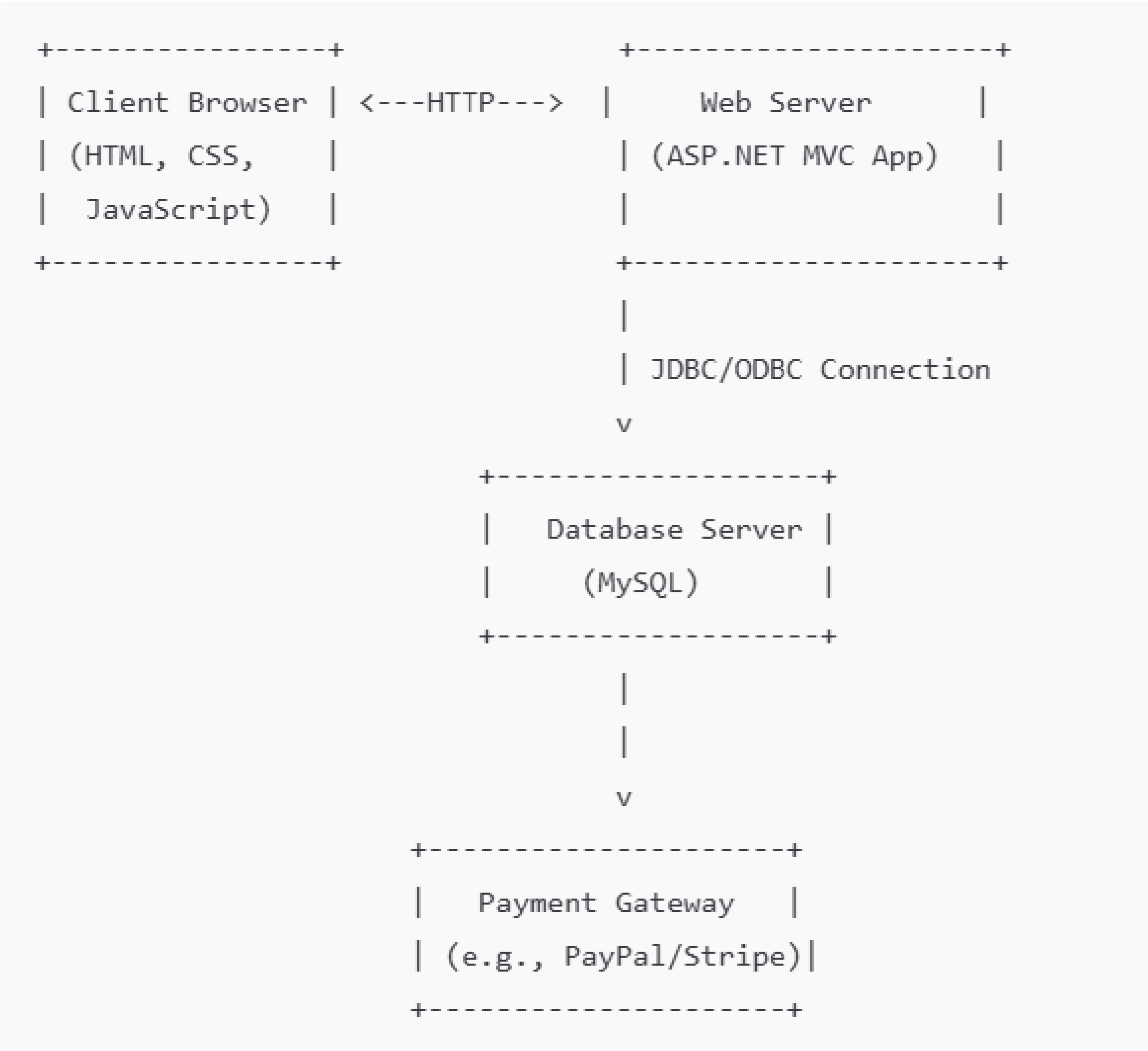
1. **Client Devices:**
   - Devices include browsers that send HTTP requests to the web server.
2. **Web Server:**
   - Hosts the **ASP.NET MVC Application** and processes incoming client requests.
   - Handles business logic and renders dynamic content for the users.
3. **Database Server:**
   - Manages all persistent data required for the application.
   - Ensures secure storage of user credentials, book inventory, order details, and feedback.
4. **Payment Gateway (External Service):**
   - Handles online payments, ensuring secure transactions for users.
   - Interacts with the application tier to validate and process payments.

Deployment Diagram Overview

Here is how the components are connected:

1. **Client Device (Browser):**
   - Sends requests via HTTP/HTTPS to the **Web Server**.
2. **Web Server:**
   - Processes requests using the **ASP.NET MVC application**.
   - Retrieves or updates data by interacting with the **Database Server**.
3. **Database Server:**
   - Stores user, book, order, and feedback data in **MySQL**.
4. **Payment Gateway (Optional):**
   - Processes payments securely via API integration.

Diagram Representation (Conceptual)



```
+----------------+                +---------------------+
| Client Browser | <---HTTP--->  |     Web Server      |
| (HTML, CSS,    |                | (ASP.NET MVC App)   |
|  JavaScript)   |                |                     |
+----------------+                +---------------------+
                                  |
                                  | JDBC/ODBC Connection
                                  v
                          +-------------------+
                          |   Database Server |
                          |     (MySQL)       |
                          +-------------------+
                                  |
                                  |
                                  v
                      +---------------------+
                      |   Payment Gateway   |
                      | (e.g., PayPal/Stripe)|
                      +---------------------+
```

Advantages of the Deployment Architecture

1. **Scalability:**
   - The system can handle increased traffic by scaling the web server and database server independently.
2. **Security:**
   - Sensitive data is transmitted using HTTPS and stored securely in the database.
3. **Reliability:**
   - The modular structure allows each tier to function independently, ensuring minimal downtime.
4. **Performance:**
   - Efficient data management and caching mechanisms improve system responsiveness.

This deployment architecture ensures a robust, secure, and scalable solution for the **E-Book Web Application System**.

Activity Diagram for E-Book Web Application System

An **Activity Diagram** represents the workflow of the system, showing the sequence of activities performed by users and the system itself. Below is an explanation of the activities involved in the **E-Book Web Application System**, including customer and administrator actions.

Key Activities in the System

1. **For Customers:**
   - Register an account and log in.
   - Search for books and view details.
   - Add books to the cart and place an order.
   - Track and manage orders (view/delete).
   - Provide feedback on purchased books.
2. **For Administrators:**
   - Log in to the admin dashboard.
   - Manage book inventory (add/update/delete books).
   - View and process customer orders.

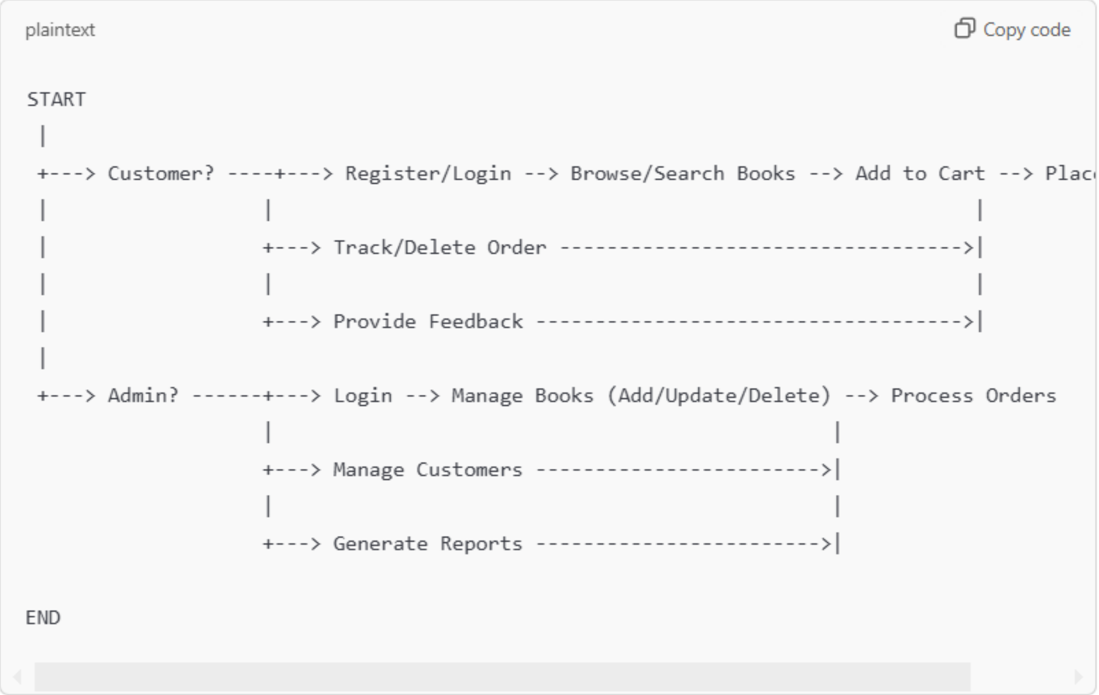
- Manage customer accounts.
- Generate and view sales and user activity reports.

Activity Diagram Description

1. **Start:** The user accesses the system via a web browser.
2. **Decision:**
   - If the user is a **customer**, they are directed to the customer workflows.
   - If the user is an **administrator**, they are directed to the admin workflows.
3. **Customer Workflow:**
   - Register or log in.
   - Browse or search for books.
   - Add books to the cart.
   - Place an order.
   - Track or delete orders.
   - Provide feedback on purchased books.
4. **Admin Workflow:**
   - Log in to the admin dashboard.
   - Manage book inventory (add, update, delete).
   - Process customer orders.
   - Manage customer accounts.
   - Generate and view reports.
5. **End:** Once tasks are completed, the user logs out.

Diagram Representation (Conceptual)

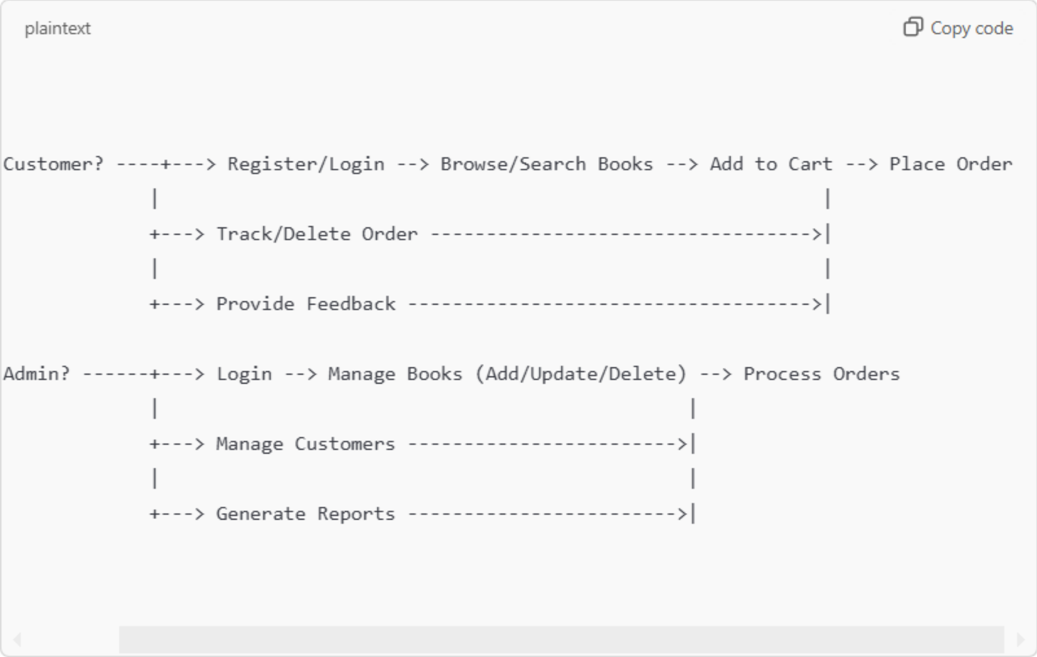
Diagram Representation (Conceptual)



```
START
 |
 +---> Customer? ----+---> Register/Login --> Browse/Search Books --> Add to Cart --> Plac
 |                  |                                                           |
 |                  +---> Track/Delete Order ---------------------------------->|
 |                  |                                                           |
 |                  +---> Provide Feedback ------------------------------------>|
 |
 +---> Admin? ------+---> Login --> Manage Books (Add/Update/Delete) --> Process Orders
                    |                                               |
                    +---> Manage Customers ------------------------>|
                    |                                               |
                    +---> Generate Reports ------------------------>|

END
```

Diagram Representation (Conceptual)



```
Customer? ----+---> Register/Login --> Browse/Search Books --> Add to Cart --> Place Order
             |                                                           |
             +---> Track/Delete Order ---------------------------------->|
             |                                                           |
             +---> Provide Feedback ------------------------------------>|

Admin? ------+---> Login --> Manage Books (Add/Update/Delete) --> Process Orders
             |                                               |
             +---> Manage Customers ------------------------>|
             |                                               |
             +---> Generate Reports ------------------------>|
```

Components of the Activity Diagram

1. **Start and End Nodes:** Represent the initiation and termination of the workflows.
2. **Activities:**
   - Customer and admin tasks such as login, searching for books, managing orders, and generating reports.
3. **Decision Points:**
   - Direct users to appropriate workflows based on their roles.
4. **Transitions:**
   - Show the flow between different activities and decisions.

Benefits of the Activity Diagram

1. **Clarity:** Provides a clear visual representation of the system's workflows.
2. **Role-Specific View:** Distinguishes between customer and administrator actions.
3. **Optimization:** Helps identify potential bottlenecks or redundant processes in the system.

This activity diagram ensures that the workflows for both customers and administrators are well-defined, supporting the efficient development and operation of the **E-Book Web Application System**.

Collaborating Tools and Version Control System in E-Book Web Application System

Effective collaboration and version control are essential for the successful development of the **E-Book Web Application System**. These tools streamline communication, task management, and code integration, ensuring the project progresses efficiently and without conflicts.

1. Collaboration Tools

a. Microsoft Teams or Slack

- **Purpose:**
  Facilitates real-time communication among team members, enabling quick resolution of issues and sharing updates.
- **Features:**
  - Channels for organized discussions on specific topics (e.g., database design, frontend development).
  - File sharing for quick access to documentation and diagrams.
  - Video conferencing for team meetings and project reviews.

b. Trello or Jira

- **Purpose:**
  Tracks tasks and progress throughout the project lifecycle.
- **Features:**
  - Kanban boards for visualizing workflows, such as tasks "To Do," "In Progress," and "Done."
  - Sprint management for organizing tasks in Agile iterations.
  - Integration with version control systems like GitHub for automated progress updates.

c. Google Drive or OneDrive

- **Purpose:**
  Centralized storage for project documentation, diagrams, and reports.
- **Features:**
  - Collaborative editing for real-time updates to shared documents.
  - Easy access and organization of project resources.

d. Draw.io or Lucidchart

- **Purpose:**
  Used for creating diagrams such as use case diagrams, class diagrams, ER diagrams, and activity diagrams.
- **Features:**
  - Cloud-based design tools for team collaboration on system models.
  - Export options for integrating diagrams into documentation.

2. Version Control System

a. Git

- **Purpose:**
  Tracks changes to the codebase and manages contributions from multiple developers.
- **Features:**
  - **Branching and Merging:** Enables developers to work on individual features without interfering with the main codebase. Features can be merged back once complete.
  - **Change Tracking:** Maintains a history of code changes, allowing the team to revert to earlier versions if needed.
  - **Conflict Resolution:** Handles code conflicts when multiple developers work on the same file.

b. GitHub

- **Purpose:**
  Provides a cloud-based platform for hosting the project's Git repositories and enables collaboration among developers.
- **Features:**
  - **Repository Management:** Centralizes the project's codebase, ensuring all team members work with the latest version of the code.
  - **Pull Requests:** Facilitates code reviews by allowing team members to review and approve changes before merging them into the main branch.
  - **Issues and Discussions:** Tracks bugs, feature requests, and project discussions.
  - **CI/CD Integration:** Automates testing and deployment pipelines to maintain quality and speed up delivery.

Benefits of Using Collaboration Tools and Version Control

1. **Enhanced Team Communication:**
   Tools like Microsoft Teams and Trello keep all team members informed about project status and tasks.

2. **Task Management:**
   Jira and Trello ensure tasks are clearly defined and tracked, enabling timely completion of sprints.
3. **Version Control:**
   Git and GitHub prevent code conflicts and ensure seamless integration of contributions from multiple developers.
4. **Real-Time Collaboration:**
   Cloud-based tools like Google Drive and Draw.io allow multiple team members to work on documentation and designs simultaneously.
5. **Quality Assurance:**
   GitHub's pull requests and issue tracking ensure code quality and help manage bugs effectively.

Configuration and Run Manual

SQL Server Configuration

Open SQL Server Management Studio (SSMS).

Connect to the SQL Server instance where your database is hosted.

Create a new database or use an existing one for your E-Book web application.

Visual Studio Configuration

Open Visual Studio 2022 and load your E-Book project.

Configure database connections in the server explorer or through Visual Studio's database tools.

Ensure all required dependencies and packages are installed and up to date.

Running the Application

Build the E-Book project to ensure there are no build errors.

Set the startup project and start debugging or run the application directly.

Access the application through a web browser using the specified URL (usually localhost).

Database Design
All Tables

Admin Table

Books Table

Customers Table

Feedback Table

Orders Details Table

Orders Table

Interface and Functions

Login page

Register page

Home Page

Admin Dashboard

Manage Books

Manage Customers

Manage Users

Manage Orders

Manage Order Details

Manage Feedbacks

Manage Reports

Customer Dashboard

All Books Page

View and Purchase Book

Checkout

Order Details

Feedback

Coding and Implementation

Login page

Register page

Dashboard

Search, sort and pagination

Create API code

Update API code

View API code

Delete API Code

Order

Order now code

Visual Studio 2022 Interface and Project Files in solution explorer

Solution Explorer

Server Explorer

Critical Appraisals

Team Confirmation

Component Clarification

Own reflection of own experience (Kanagasabai Tharshan)

Own reflection of own experience (Vimalanathan Thiviya)

Test Plan for E-Book

Test Case

Test Case for the E-Book system

Test Case 01

Test case 02

Test Case 03

Test Case 04

Test Case 05

Test Case 06

Test Case 07

Test Case 08

Test Case 09

Test Case 10

Test Case 11

Test Case 12

Test Case 13

Test Case 14

Future Work

Conclusion

References

Amazon.com 2024. [ONLINE] Available at: https://sell.amazon.com/learn/what-is-ecommerce [Accessed 10 May 2024].

Asp.net MVC Entity Framework - javatpoint (2023) www.javatpoint.com. [ONLINE] Available at: https://www.javatpoint.com/asp-net-mvc-entity-framework [Accessed 10 May 2024].

Microsoft. 2023. ASP.NET MVC Pattern | .NET. [ONLINE] Available at: https://dotnet.microsoft.com/apps/aspnet/mvc. [Accessed 10 May 2024].

Mohammed, K. N. and Syam Chambrelin, K. (2020) “An analytical approach in usage of agile methodologies in construction industries – A case study,” Materials today: proceedings, 33, pp. 475–479. doi: 10.1016/j.matpr.2020.05.045.

solankimayank Follow Improve (2022) What is systems design - learn system design, GeeksforGeeks. [ONLINE] Available at: https://www.geeksforgeeks.org/what-is-system- design-learn-system-design/ [Accessed 14 May 2024].

The 7 principles of privacy by design (2024) Onetrust.com. [ONLINE]Available at: https://www.onetrust.com/blog/principles-of-privacy-by-design/ [Accessed 14 May 2024].

Triple D. 2023. The MVC-architecture - Triple D. [ONLINE] Available at:https://www.tripled.io/08/11/2017/mvc-architecture/. [Accessed 16 May 2024]

What is a UML diagram? (2024) https://miro.com/. [ONLINE] Available at:https://miro.com/diagramming/what-is-a-uml-diagram/ [Accessed 17 May 2024].

What is Agile methodology in project management? (2024) Wrike.com. [ONLINE] Available at: https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/ [Accessed 18 May 2024].