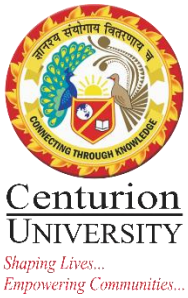
Centurion University of Technology & Management

Bhubaneswar



Software Requirements Specification

Project: Ebook Management System — Advance Java Web Project (Servlet, JSP, JDBC, MySQL)

Supervisor

Mr. Rakesh kumar Ray

By

Vikas Das (230301120326)

Shrabanee Routray (230301120295)

Vasetti Dixita (230301120299)

Satyapriya Das (230301120308)

Jyoti Anurag (230301120003)

Table of Contents

- 5.4 Security

---

1. Introduction

1.1 Project Overview

The Ebook Management System is a web-based application developed as a B.Tech project. It is designed using **Java**, **JSP**, **Servlets**, and **MySQL** to create a dynamic platform for users to browse and interact with a digital book collection. This document outlines the initial phase, which focuses on the design and implementation of the application's home page, as demonstrated in the tutorial video.

1.2 Purpose

The primary purpose of this project is to provide a centralized and user-friendly platform for managing and displaying e-books. It aims to deliver a seamless user experience for discovering books, viewing their details, and (in future phases) purchasing them. For a B.Tech student, this project serves as a practical application of web development skills using the Java technology stack.

1.3 Scope

The scope of the project as covered in the video is the front-end development of the home page. This includes:

- Designing a responsive layout using **Bootstrap**.
- Creating a navigation system for the application.
- Displaying books in categorized sections such as **Recent**, **New**, and **Old**.
- Implementing interactive elements like buttons and icons using **Font Awesome**.

The project's future scope, mentioned in the video, includes backend development for user registration, login, and book management functionalities.

1.4 Target Audience

The primary users of this application are general consumers interested in browsing and purchasing e-books. The secondary audience includes system administrators who will manage the book inventory and user data.

---

2. Overall Description

2.1 Product Perspective

This system is a standalone web application that will operate independently. It is designed to be accessible through any modern web browser. The application follows a client-server architecture, where the user's browser is the client and the Java web application running on a server (like Apache Tomcat) is the server, which interacts with the MySQL database.

2.2 Tools and Technologies

The development of this project relies on a specific set of tools and technologies identified in the video: | Technology | Version/Type | Purpose | Timestamp | | :--- | :--- | :--- | :--- | | **IDE** | Eclipse | Code development, debugging, and project management. | | | **Database** | MySQL Workbench | Storing and managing all application data. | | | **Build Tool** | Maven | Managing project dependencies and build lifecycle. | | | **Frontend** | Bootstrap 4 | Creating a responsive and visually appealing UI. | | | **Icons** | Font Awesome | Providing scalable vector icons for UI elements. | | | **Backend**| Java (JDK 8+), JSP, Servlets | Core application logic and server-side processing. | |

2.3 User Classes and Characteristics

- **General User/Guest:** Any visitor to the site. They can browse books, use the search functionality, and view book details.
- **Registered User (Future):** A user who has created an account. They will have additional privileges like adding books to a cart and making purchases.
- **Administrator (Future):** A privileged user responsible for adding new books, updating existing book information, and managing user accounts.

2.4 Operating Environment

- **Server-Side:** The application will be deployed on a web server that supports Java Servlets and JSP, such as **Apache Tomcat**. It requires a Java Runtime Environment (JRE) and a MySQL database server.
- **Client-Side:** The application will be accessible on any device with a modern web browser (e.g., Chrome, Firefox, Safari, Edge) and an internet connection.

2.5 Design and Implementation Constraints

- The project must be developed using the specified Java technology stack.
- The initial design must be responsive and adapt to different screen sizes, from mobile phones to desktops.
- The video uses a procedural approach to build the UI, starting with basic HTML structure and progressively adding styling and components.

---

3. System Features and Requirements

3.1 Home Page Module

This is the core module developed in the video tutorial.

3.1.1 Header and Navigation Bar

- **Description:** The header provides branding and primary navigation for the site.
- **Requirements:**
    - Must contain the "E-Book" logo/brand name.
    - Must include a search bar for finding books.
    - Must have navigation links: **Home**, **Recent Book**, **New Book**, and **Old Book**.
    - Must feature **Login** and **Register** buttons for user authentication.
    - The background color and text styling are customized for a unique look.

3.1.2 Main Banner / Hero Section

- **Description:** A large, visually engaging section below the header to attract user attention.
- **Requirements:**
    - Must display a background image related to books or reading.
    - Must prominently display the text "**Book Management System**" in the center.

3.1.3 Book Display Sections

- **Description:** The main content area where books are showcased in different categories. The video implements three such sections.
- **Requirements:**
    - Each section (**Recent Book**, **New Book**, **Old Book**) must have a clear heading.
    - Books are displayed in a grid layout (four books per row).
    - Each book is presented as a "card" containing:
        - Book cover image.
        - Book Title (e.g., "Java Programming").

        - Author Name.
        - Price.
        - Action buttons: "**Add Cart**" with an icon, "**View Details**", and "**Order Now**".
    - Each section concludes with a "**View All**" button to see more books in that category.
    - A hover effect is applied to the book cards for better user interaction.

3.1.4 Footer Section

- **Description:** The section at the bottom of the page containing supplementary information.
- **Requirements:**
    - Must have a distinct background color matching the header's theme.
    - Must display copyright and developer information (e.g., "Designed and Developed by [Author]").

3.2 Future Scope: User and Book Management

The video explicitly states that backend functionality will be covered in subsequent parts. This includes:

- **User Registration:** A form to create a new user account.
- **User Login:** A mechanism for users to sign in.
- **Dynamic Data:** Populating the book cards with data fetched from the MySQL database instead of the current static placeholder content.

---

4. External Interface Requirements

4.1 User Interface (UI)

- The UI must be clean, intuitive, and easy to navigate.
- The design must be responsive, ensuring a consistent experience across desktops, tablets, and mobile devices.
- Visual feedback (e.g., hover effects on buttons and links) must be provided for user actions.

4.2 Software Interfaces

- **Database:** The application will interface with a **MySQL** database via JDBC (Java Database Connectivity) to perform CRUD (Create, Read, Update, Delete) operations on book and user data.
- **Web Browser:** The application will generate HTML, CSS, and JavaScript that is compatible with all major modern web browsers.

---

5. Non-Functional Requirements

5.1 Performance

- The home page should load within 3-5 seconds on a standard internet connection.
- Images should be optimized for the web to reduce load times.

5.2 Usability

- The navigation should be logical and predictable.
- All interactive elements (buttons, links) must be clearly identifiable and functional.
- The color scheme and typography should ensure readability.

5.3 Reliability

- The application should be available 24/7, with minimal downtime.
- Links should not be broken, and all page elements should render correctly.

5.4 Security

- (Future Scope) User passwords must be hashed before being stored in the database.
- (Future Scope) The application should be protected against common web vulnerabilities like SQL Injection and Cross-Site Scripting (XSS).

Thank You !