**22103182 - Aryan Agrawal**

**22103200 – Swapnil**

**Case Study: Online Library Management System Using the Waterfall Model**

**Objective**

To design and implement an Online Library Management System (LMS) using the Waterfall Model, encompassing the phases of software development: Requirement Analysis, System Design, Implementation, Testing, Deployment, and Maintenance.

**Phase 1: Requirement Analysis**

**Functional Requirements**

1. **User Registration and Login**
   * Users can create an account by providing personal details.
   * Users can log in using their credentials to access services.
2. **Search and Browse Books**
   * Search books by title, author, genre, or ISBN.
   * Display availability, book descriptions, and reviews.
3. **Book Borrowing and Returning**
   * Users can borrow books online and receive a due date.
   * Users can return books, and the system updates availability.
4. **Admin Functions**
   * Admins can add, update, or remove books.
   * Manage user accounts and monitor library usage.

**Non-Functional Requirements**

* User-friendly interface for smooth navigation.
* Secure handling of user credentials and transaction data.
* High system reliability with minimal downtime.

**Phase 2: System Design**

**Architectural Design**

* **Three-Tier Architecture:**
  1. **Presentation Layer:** Web or app interface for users and admins.
  2. **Business Logic Layer:** Handles book searches, borrowing, and admin operations.
  3. **Database Layer:** Stores user details, book records, and transaction logs.

**Modules and Flow**

1. **User Module:**
   * Register, Login, Manage Account.
2. **Library Module:**
   * Search Books, View Availability.
3. **Transaction Module:**
   * Borrow Book, Return Book, View Borrowing History.
4. **Admin Module:**
   * Manage Books, Monitor Transactions, Generate Reports.

**ER Diagram**

* **Entities:**
  + User, Book, Transaction, Admin.
* **Relationships:**
  + A User borrows multiple Books.
  + A Book can have multiple Borrowers over time.
  + Admins manage Books and Transactions.

**Phase 3: Implementation**

**Tools and Technologies**

* **Frontend:** HTML, CSS, JavaScript (web interface).
* **Backend:** Python/Java.
* **Database:** MySQL/PostgreSQL for storing records.

**Phase 4: Testing**

**Test Cases**

1. **User Registration and Login:**
   * Verify successful user registration and login with valid inputs.
   * Test incorrect logins and password recovery processes.
2. **Search Functionality:**
   * Check searching by various criteria (title, author, genre).
   * Verify book details and availability are displayed correctly.
3. **Borrowing and Returning Books:**
   * Test the borrowing process for available books.
   * Verify system updates on book return and overdue handling.
4. **Admin Functions:**
   * Validate addition, modification, and removal of books.
   * Test report generation and monitoring of user activities.

**Phase 5: Deployment**

* Deploy the system on a web server or cloud platform (e.g., AWS, Azure).
* Configure the database and ensure secure communication between layers.
* Perform load testing to handle multiple users.

**Phase 6: Maintenance**

* Update the library catalog regularly.
* Monitor database performance and perform backups.
* Address user feedback and add new features, such as eBook integration.

**Solution Benefits**

* **Accessibility:** Allows users to access the library from anywhere.
* **Efficiency:** Simplifies library operations and minimizes manual effort.
* **Reliability:** Ensures accurate record-keeping and secure transactions.
* **Scalability:** Supports growing user bases and book collections.