

|  |  |
| --- | --- |
| Students name: | Diana Shakirova, Issa Nurzhanov, Qadyrsha Dariga |
| Group name: | CS-2313 |
| Teacher name: | Aibatbek Aigerim |
| Project type: | Group |
| Number of words |  |

**Library System Project Report**

**1. Topic and Objective**

**Topic: Library System**

**Objective:** Objective: The objective of this project is to develop a comprehensive library management system that streamlines various library operations including book cataloging, user account management, book checkout and book return tracking. The purpose of the system is to improve the efficiency, organization and convenience of users in the library environment.

**Tasks:**

1. Design and implement a database schema to store information about books, users, loan records and other relevant objects.

2. Develop a user-friendly interface for librarians to perform CRUD (create, read, update, delete) operations on books, users and borrowed records.

3. Implement authentication and authorization mechanisms to ensure secure access to the system.

4. Implement user account management features, including registration and login.

5. Develop functions for tracking the issuance and return of books.

**2. Database Connection and Usage**

The project will use the PostgreSQL relational database management system to store and manage data related to books, users, borrowed records, etc. The database connection will be established using appropriate libraries or frameworks in the selected JDBC programming language for Java.

Example object:

Book entity: {ID, title, author, publication year, status}

User Entity: {id, name, age, password}

**3. Project Structure**

The project will follow a structured organization to ensure maintainability, scalability, and readability of the code base. A typical project structure might include directories containing source code, configuration files, documentation, and tests. Additionally, version control with Git and hosting the project on GitHub will make collaboration and code management easier.

**4. Functionality**

**The library system will offer the following functionalities:**

- Book management:

- Add new books to the system.

- Update existing book information.

- Remove books from the library catalog.

- Availability of books.

- Can check up borrowed books.

- User management:

- Authentication of users to log into the system.

- Singing up for new users.

- Manage user profiles and account information.

- Borrowing and returning:

- Take books from the library.

- Return the books you rented.

- Authentication and authorization:

- Secure user authentication.

- Role-based access control for different types of users (librarians, members).

- User interface:

- Develop a user-friendly interface for easy navigation and interaction with the system.

By performing these functions, a library system aims to provide an efficient and organized platform for managing library resources and effectively serving library patrons.

**Create Table for Entities in Database:**

SQL scripts are provided to create tables for each entity with appropriate fields and constraints.

GitHub Repository:

The project files are uploaded to a GitHub repository for version control and collaboration.

Frontend Usage:

Console Interface: Users can interact with the system through a command-line interface, enabling functionalities like adding books, registering users, borrowing and returning books.

Postman (for Web-Services): API endpoints are provided for external integration, allowing functionalities like adding books, registering users, and retrieving borrowing history.

**Technical Requirements:**

**OOP Principles:** Encapsulation and Abstraction are utilized to encapsulate data and operations within classes and to abstract away complex functionalities.

**JDBC Functionality:** JDBC is extensively used for database connectivity, including executing SQL queries and handling transactions.

**SOLID Principles:**

**Single Responsibility Principle:** Each class has a single responsibility, such as managing database operations, representing entities, or handling user interactions.

**Open/Closed Principle:** The system is designed to be open for extension but closed for modification, achieved through interfaces and abstraction.

Design Pattern:

**DAO Pattern (Data Access Object):** Separates data access logic from business logic, enhancing modularity and maintainability.

**GUI/Web Integration:**

If applicable, the Java project can be integrated with a GUI or web interface for enhanced user experience.

**Conclusion**

The System Library Management System developed in Java, connected with PostgreSQL, offers a comprehensive solution for managing library resources. By adhering to OOP principles, utilizing JDBC functionality, applying SOLID principles, and employing design patterns, the project demonstrates a robust and scalable architecture suitable for library management applications. Further enhancements can be made to integrate graphical user interfaces or web interfaces to enhance usability.