Library Management System

Table of Contents

1. Introduction

The Library Management System is a Java-based application that allows clients to interact with a server to manage a library's book inventory. The system consists of several components, including a graphical user interface (ClientGUI), client-side socket communication (ClientSock), server-side data handling (ServerData), and server-side socket management (ServerSock).

2. ClientGUI

The `ClientGUI` class represents the graphical user interface for the library management system. It includes functionality for refreshing the book list, adding books, borrowing books, and returning books. The GUI displays a table with information about each book, such as ID, title, author, price, and associated student ID.

Methods
- `showGUI()`: Initializes and displays the GUI components.
- `refreshTable()`: Refreshes the book table by retrieving the latest data from the server.
- `updateTable(ArrayList<Product> books)`: Updates the table with a new set of books.
- `showAddBookDialog()`: Displays a dialog for adding a new book.
- `showBorrowBookDialog()`: Displays a dialog for borrowing a book.
- `showReturnBookDialog()`: Displays a dialog for returning a book.
- `main(String[] args)`: Entry point for running the GUI.

3. ClientSock

The `ClientSock` class handles client-side socket communication. It provides methods for retrieving all books, adding books, borrowing books, and returning books from the server.

Methods
- `getAllBooks()`: Requests and retrieves all books from the server.
- `addBook(Product book)`: Sends a request to the server to add a new book.
- `borrowBook(int studentId, int bookId)`: Sends a request to the server to borrow a book.
- `returnBook(int bookId)`: Sends a request to the server to return a book.

4. ClientRequest

The `ClientRequest` class represents requests made by the client to the server. It includes different request types such as getting all books, adding a book, borrowing a book, and returning a book.

5. Product

The `Product` class represents a book in the library. It includes attributes such as ID, title, author, price, and associated student ID.

6. ServerData

The `ServerData` class handles server-side data management. It interacts with a MySQL database to perform operations such as retrieving all books, adding books, borrowing books, returning books, and getting the student ID associated with a book.

7. ServerSock

The `ServerSock` class manages server-side socket connections. It listens for incoming client connections and spawns a new `ClientHandler` thread for each connected client.

Methods
- `startServer()`: Starts the server and continuously accepts incoming client connections.

8. MainServer

The `MainServer` class serves as the entry point for starting the server.

Methods
- `main(String[] args)`: Entry point for starting the server.

This documentation provides an overview of the key components and their functionalities in the Library Management System. Specific details about classes, methods, and interactions can be found in the corresponding sections above.

README

Overview:

The Library Management System is a simple client-server application for managing a library's book inventory.
It allows users to perform various operations such as viewing all books, adding new books, borrowing books, and returning books.

Components

Client: The client-side application allows users to interact with the library system through a graphical user interface (GUI).

Server: The server-side application manages the book data and handles client requests.

Technologies Used

Java: The project is implemented using Java programming language.

Swing: The GUI is created using Java Swing.

MySQL: The server stores book data in a MySQL database.

Setup Instructions

Server Setup:

Ensure you have MySQL installed on your system.

Create a MySQL database named "library" with a table named "books" using the provided SQL script (library.sql).

Update the MySQL connection details in the ServerData class.

Client Setup

Open the project in an IDE (Integrated Development Environment) such as IntelliJ/Eclipse.

Update the server address and port in the MainClient class.

Running the Application

Start the server by running the MainServer class.

Run the MainClient class to start the client application.

The GUI will appear, allowing you to interact with the Library Management System.

Usage

View All Books: Click the "Refresh" button to view all books in the library.

Add Book: Click the "Add Book" button to add a new book to the library.

Borrow Book: Click the "Borrow Book" button to borrow a book by providing the book ID and student ID.

Return Book: Click the "Return Book" button to return a borrowed book by providing the book ID.