---

# Library Management System Documentation

## 1. Main Class (`Main`)

The `Main` class serves as the entry point for the library management system. It includes the main method where the program starts execution. It initializes the `Library` object and provides a graphical user interface (GUI) for user interaction, specifically the `Login` window.

## 2. Login Class (`Login`)

The `Login` class is responsible for creating the login interface. It includes text fields and buttons for both students and librarians to enter their credentials. Upon successful login, it opens either the `Student_home` or `Librarian_home` interface. Additionally, it allows the user to sign up as a new student.

## 3. Librarian_home Class (`Librarian_home`)

The `Librarian_home` class represents the interface for librarians after a successful login. It provides options to add a book, remove a book, view books, and exit the librarian interface. Each option is associated with a specific action, such as opening a book table or performing book addition/removal.

## 4. Student_home Class (`Student_home`)

The `Student_home` class represents the interface for students after a successful login. It provides options to borrow books, return books, view books, and exit the student interface. Similar to the `Librarian_home` class, each option is associated with specific actions.

## 5. BorrowBookGui Class

The `BorrowBookGui` class creates a GUI for students to borrow books. It includes a text field for entering the book title and a button to perform the borrowing action. The class communicates with the `Library` and `Student` classes to execute the borrowing process.

## 6. ReturnBookGui Class

The `ReturnBookGui` class creates a GUI for students to return books. It includes text fields for entering the book title, author, and ISBN, along with a button to perform the return action. Similar to `BorrowBookGui`, it communicates with the `Library` and `Student` classes.

## 7. AddBookGui Class

The `AddBookGui` class creates a GUI for librarians to add new books to the library. It includes text fields for entering the book title, author, and ISBN, along with a button to add the book to the library. It communicates with the `Library` class to execute the book addition process.

## 8. Library Class

The `Library` class serves as a central component of the library management system. It manages lists of books, students, librarians, and users. Key functionalities include adding and removing books, finding books by ISBN or title, and handling user authentication for students and librarians.

Methods:

- `addabook(Book book)`: Adds a book to the library.
- `remove_book(Book book)`: Removes a book from the library.
- `addStudent(Student student)`: Adds a student to the library.
- `addLibrarian(Librarian librarian)`: Adds a librarian to the library.
- `findStudentByUsernameAndPassword(String username, String password)`: Finds a student by username and password.

- `findLibrarianByUsernameAndPassword(String username, String password)`: Finds a librarian by username and password.

- `findBookByISBN(int isbn)`: Finds a book by ISBN.

- `findBookByTitle(String title)`: Finds a book by title.

- `display()`: Displays information about all books in the library.

- `displayStudents()`: Displays information about all students in the library.

- `displayLibrarians()`: Displays information about all librarians in the library.

## 9. Librarian Class

The `Librarian` class extends the `User` class and represents a librarian in the library management system. It provides methods for adding and removing books from the library. Additionally, it includes a method to get the librarian's username.

Constructors:

- `Librarian(String user_id, String password)`: Constructs a librarian with the specified user ID and password.

- `Librarian(String User_name, String email_id, int phone_number, String user_id, String password)`: Constructs a librarian with additional details.

Methods:

- `add_a_new_Book(Library library, Book b)`: Adds a new book to the library.

- `remove_book(Library library, Book b)`: Removes a book from the library.

- `getUsername()`: Returns the username of the librarian.

## 10. Book Class

The `Book` class represents a book in the library. It includes information such as the title, author, and ISBN. The class overrides the `equals` method for comparing books.

Constructor:

- `Book(String title, String author, int isbn)`: Constructs a book with the specified title, author, and ISBN.

Methods:

- `getTitle()`: Returns the title of the book.
- `getAuthor()`: Returns the author of the book.
- `getIsbn()`: Returns the ISBN of the book.
- `equals(Object obj)`: Overrides the equals method for comparing books.

## 11. BookTable Class

The `BookTable` class creates a graphical interface for displaying the list of books in a table. It utilizes Java Swing components and includes an "Exit" button to close the table.

Methods:

- `book_table_fun(List<Book> books)`: Initializes and sets up the book table.
- `Display(List<Book> books)`: Populates the book table with book information.
- `showtable()`: Displays the book table with an exit button.

## 12. Student Class

The `Student` class extends the `User` class and represents a student in the library management system. It includes a list of borrowed books and methods for borrowing and returning books.

Constructors:

- `Student(String user_id, String password)`: Constructs a student with the specified user ID and password.

- `Student(String User_name, String email_id, int phone_number, String user_id, String password)`: Constructs a student with additional details.

Methods:

- `getBorrowed_books()`: Returns the list of borrowed books.
- `Borrow_book(Library library, Book b)`: Borrows a book from the library.
- `Return_book(Library library, Book b)`: Returns a borrowed book to the library.
- `Display_Borrowed_Books()`: Displays information about borrowed books.

## 13. RemoveBookGui Class

The `RemoveBookGui` class creates a graphical interface for librarians to remove a book from the library. It includes a text field for entering the ISBN and a button to perform the removal.

Constructors:

- `RemoveBookGui(Library library)`: Constructs a `RemoveBookGui` with a reference to the library.

Methods:

- `setGui_()`: Sets up the graphical user interface for removing a book.
- `removeBook()`: Removes a book from the library based on the entered ISBN.

## 14. StudentSignUpGUI Class

The `StudentSignUpGUI` class creates a graphical interface for students to sign up with the library. It includes text fields for entering personal information and a button to complete the sign-up process.

Constructors:

- `StudentSignUpGUI(Library library)`: Constructs a `StudentSignUpGUI` with a reference to the library.

Methods:

- `signUpStudent()`: Completes the student sign-up process.

## 15. User Class

The `User` class represents a general user in the library management system. It includes common attributes such as user ID, username, password, email, and phone number.

Constructors:

- `User(String user_id, String password)`: Constructs a user with the specified user ID and password.
- `User(String User_name, String email_id, int phone_number, String user_id, String password)`: Constructs a user with additional details.

Methods:

- Getter and setter methods for user attributes.

This comprehensive documentation provides insights into each class's purpose, constructors, methods, and usage within the library management system. It also includes information about key functionalities, error handling, and the utilization of Java Swing for GUI components. Adjustments and enhancements can be made based on specific requirements.