Library Management System

Muhammad Farooq Sheikh

Object Oriented Programming

Introduction

The **Library Management System** is a software designed for managing all the functionality of a library by controlling different operations on the data and managing the records in a library. This makes easy to maintain the records in the database/File.
The problem to manage and maintain all the functionality of a library manually has given rise to demand for a system that manages and controls the library functions on a software. As by performing these tasks manually can be very difficult, time taking, and error prone. To overcome this problem, the **Library Management System** came into existence. This system automates all the tasks that are to be performed by the librarian of a library in an advanced, easier, and efficient way using the software.
The desired system is developed in Java, using JavaFx which helps to implement a Graphical User-Interface (GUI) environment together with the Database/File. The system is made user-friendly so that everyone can access it easily. The software comprises of five main elements:

1. Login-In and Log-Out module to access the system by a librarian or a student.
2. Inserting or removing records of the books and their borrowers by librarian in the Database.
3. Searching records of the books and their borrowers by librarian in the Database.
4. Report Generation module that provides the list of borrowed books and available books by librarian and student.
5. Issuing and returning a book from the library by the student

Running a library isn't as easy as it sounds. Libraries are constantly changing. Without a Library Management system, librarians are required to do everything manually. This takes time and doesn't help in maintaining decorum of a library. They must keep up with technology to be efficient. They need a better way of organizing their data so they can make more informed decisions that will benefit their organization. Libraries are knowledge hubs, and they are very useful in building an individual as they are subsystems of some supersystem (education, research, or social service). A library management system enhances the efficiency of both the librarians and the library users. It enables users to easily keep track of the books issued, reissued, and those not returned. A library management system is the most significant measure to improve the efficiency of a library. As technology advances, it is important to update the system according to the latest trends and technologies. With this project, we want the comfort of the library users as well as the librarians. Instead of doing everything manually, use of a management system helps maintain efficiency.

Project Environment

For developing a user-friendly software, we need to work together with Java language such that JavaFx and Database/File to result in an attractive and easier system to operate. The language, software, and hardware which we will be using to develop the system are as following:

- **Language:**
  - Java
  - JavaFx
  - CSS
- **Hardware Requirements:**
  - **Processor:** Intel Pentium 4 or above.
  - **Hard Disk:** 20 GB

- **RAM:** 1 GB
- **Keyboard**
- **Mouse**
- **Monitor**

➢ **Software Requirements:**

- **Operating Systems:** Windows XP, Windows Vista, Windows 7, Windows 8, or Windows 10
- **Programming Language:** Java, JavaFx, CSS
- **Integrated Development Environment:** IntelliJ Idea
- **Database Management System:** MySQL WorkBench

Project Details

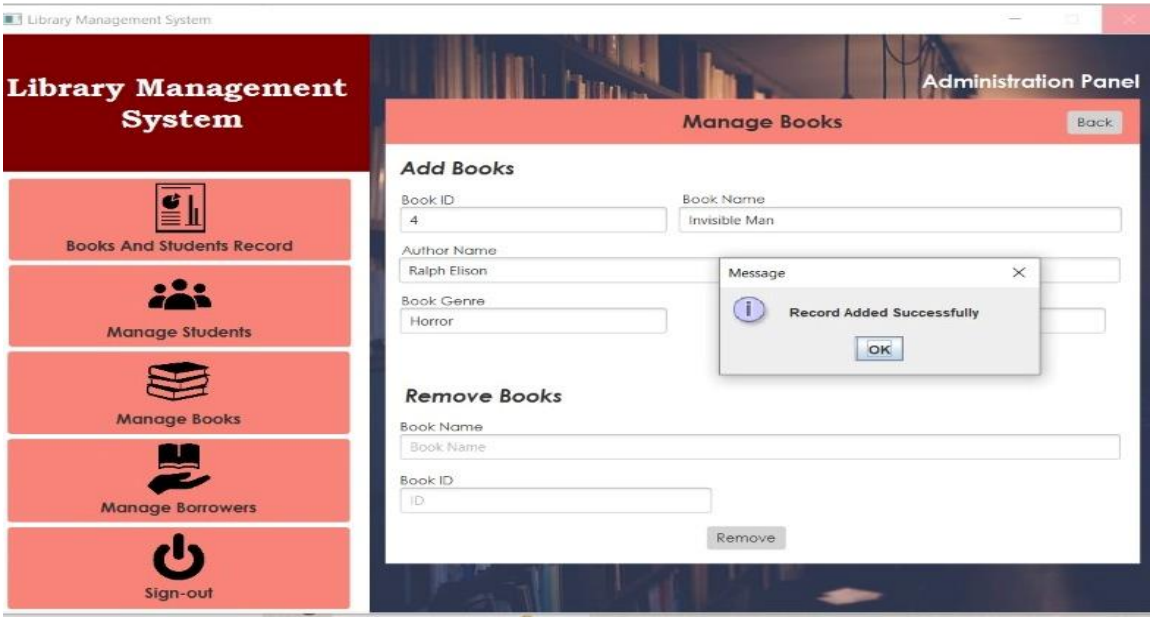
The main functionality of the Library Management System project is given here. The main concepts of the system are discussed and explained here.

Classes:

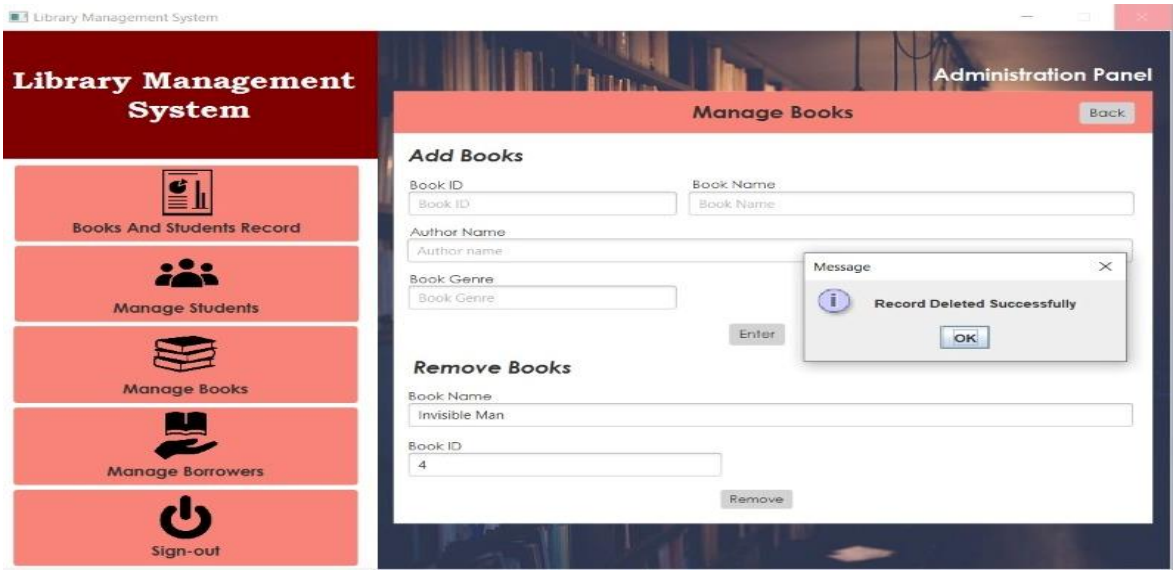
There are multiple classes which are used in this project. All these classes perform unique tasks. Here are all the classes performing main tasks of this application.

- **Controller class** is a Java class which performs basic operations like switching scenes and action handling. Multiple methods are used in this class as shown below:
  1. Switching methods are used to perform event handling. Scenes switch using these methods. The buttons are declared in this class which when interacted perform scene switches. For example: switchToAdminPanel() and switchToStudentPanel() etc.

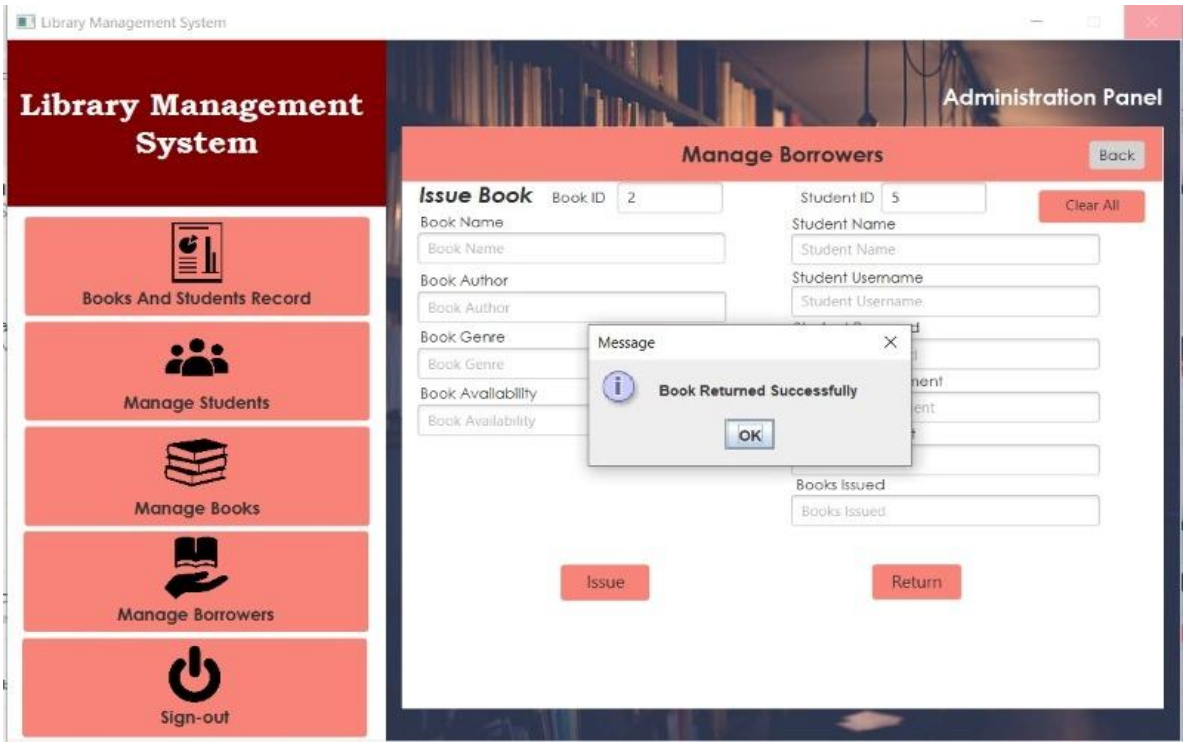
  2. ManageAddBook();this method is used to add books to database via admin. Admin has the authority to add books into database as shown in this method.



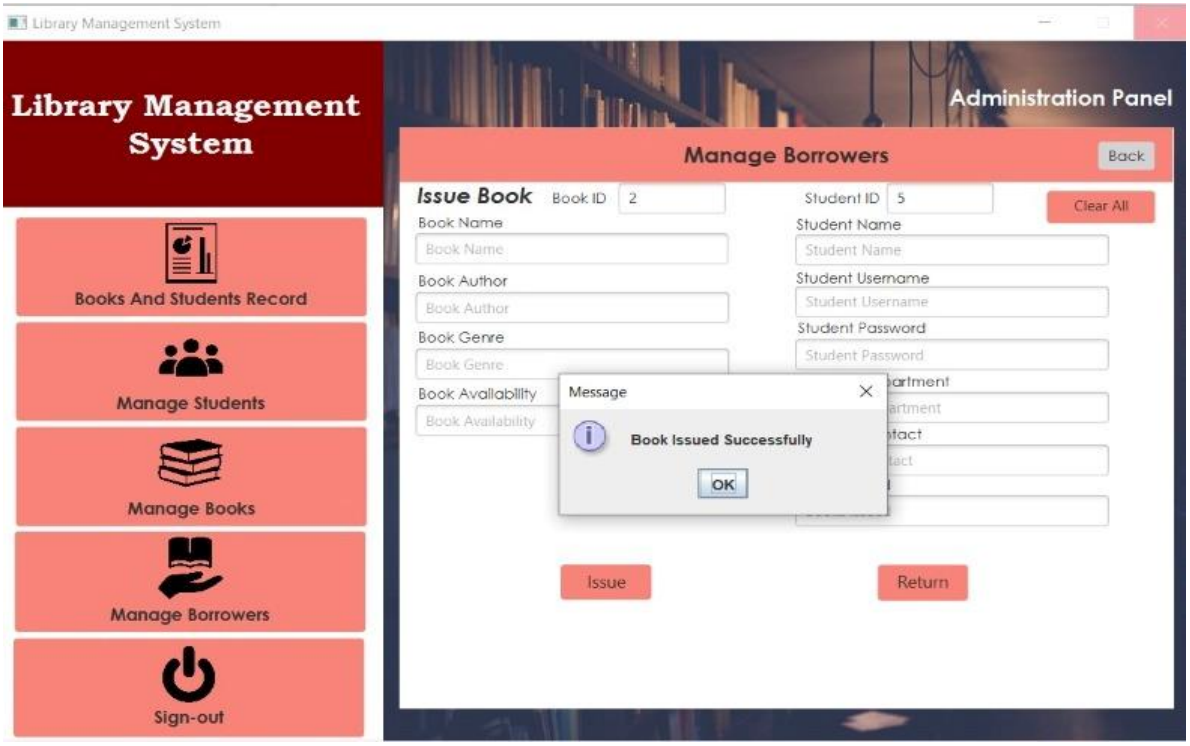
3. ManageDeleteBook(); this method is used to delete books form database via admin. Admin has the authority to delete books from database as shown in this method.



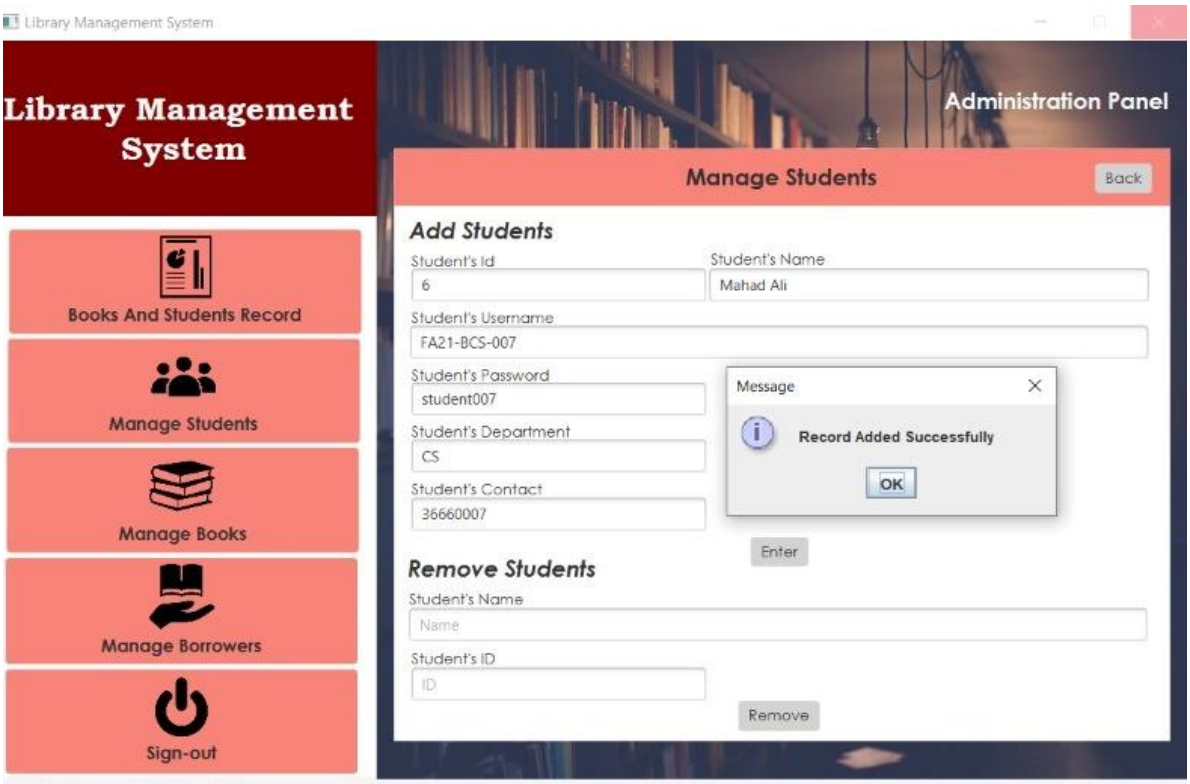
4. AdReturnBook(); this method is used to return the books which are issued by the student. The book is re-entered into the database when this method performs its functionality.



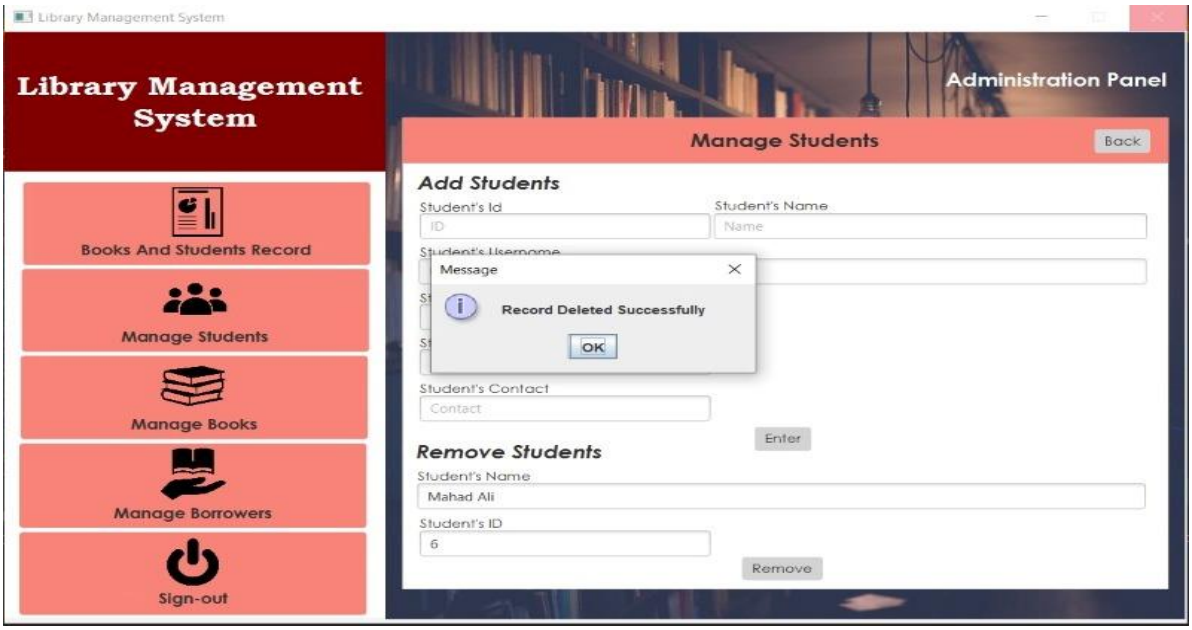
5. AdIssueBook(); this method is used to allow a student to issue book from the system. The book which is issued is removed from the database and is given issue status when this method performs its functionality.



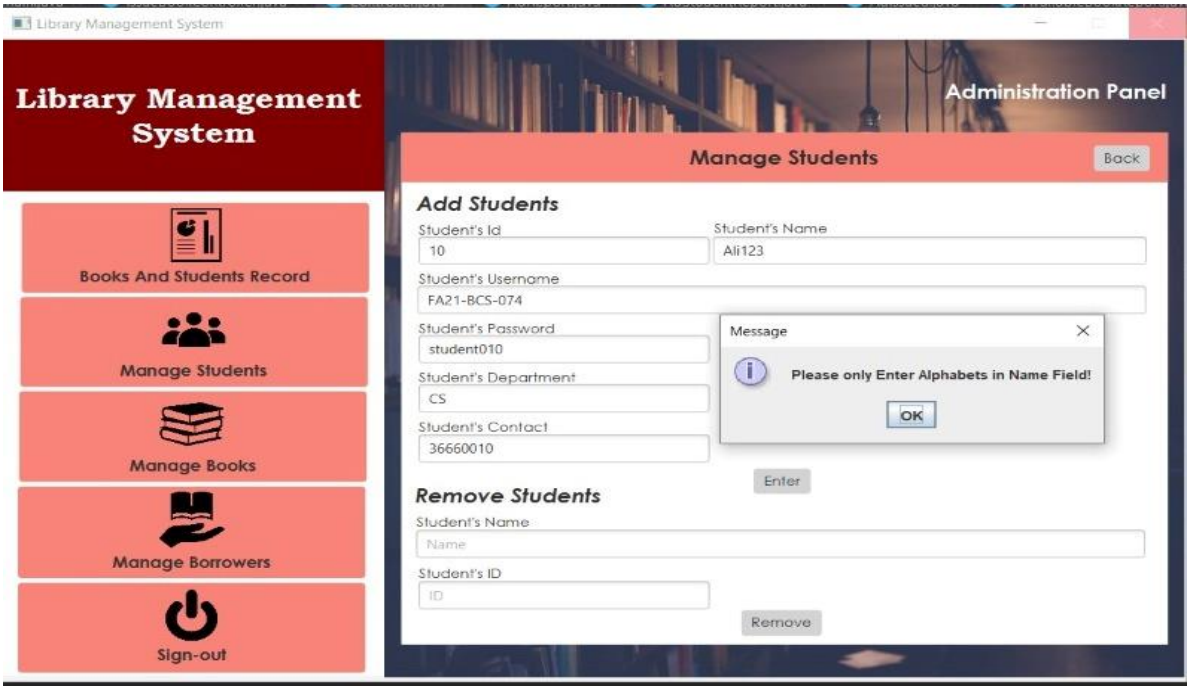
6. ManageAddStudent(); this method is used to add students into database via admin. Admin has the authority to enter students into the database as shown in this method.



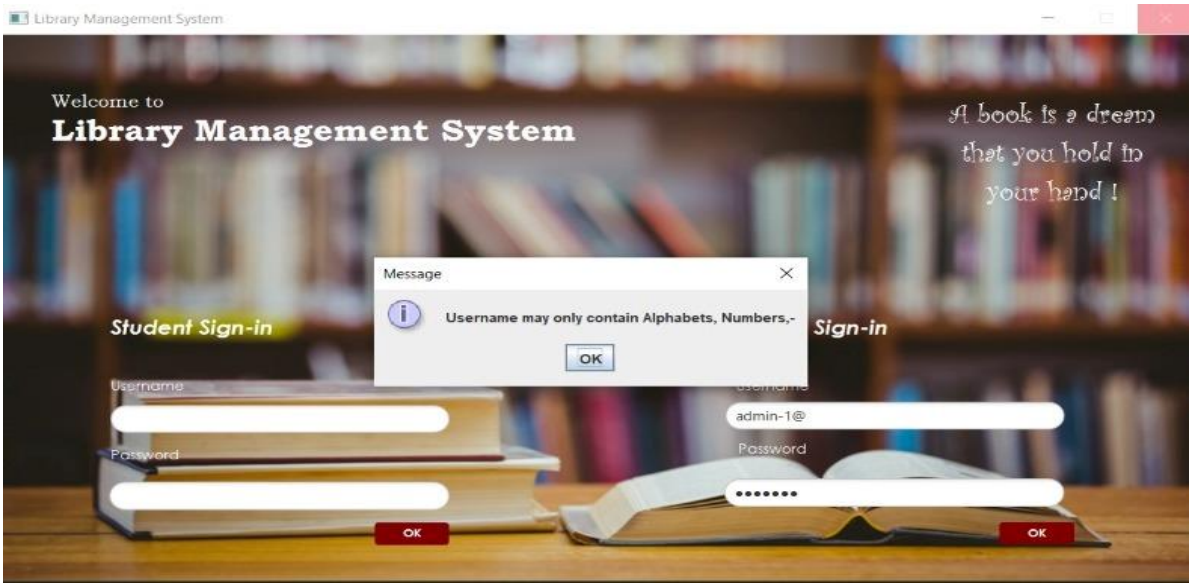
7. ManageDeleteStudent (); this method is used to delete students form database via admin. Admin has the authority to delete students into database as shown in this method.



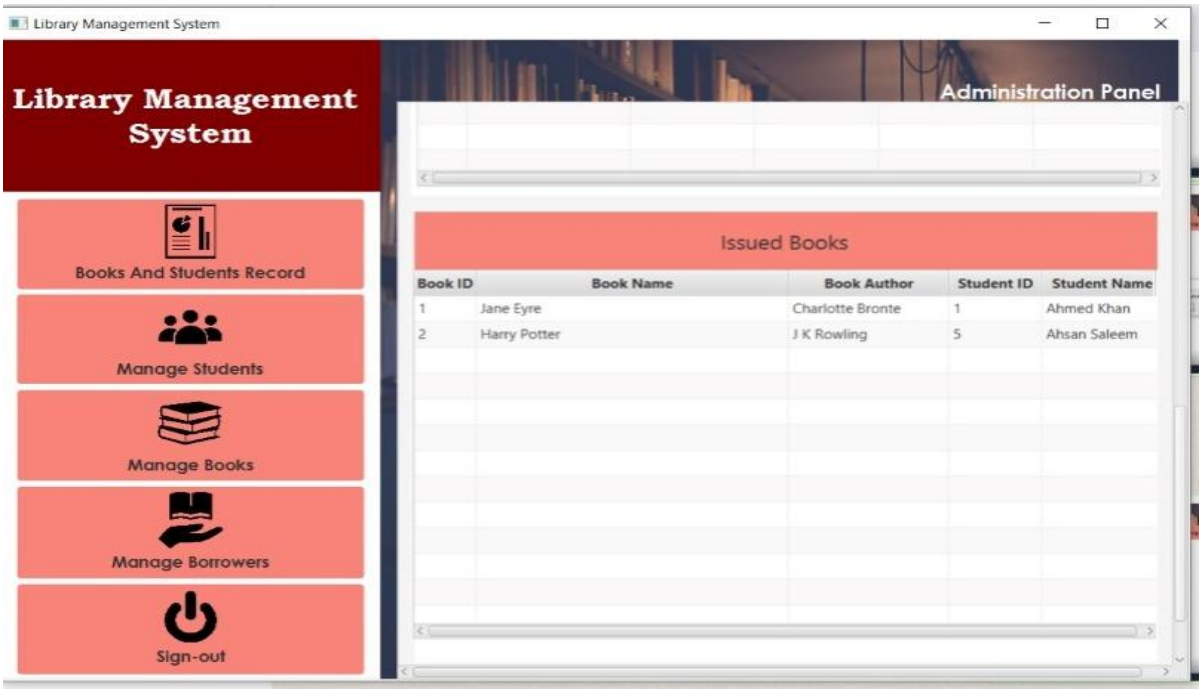
8. Some methods in this class are used to give validations while entering data into the system. For example: isValidContact(); method is used for entering contact information of the student. It can be integer type only and can add only 8 digits, not less than 8 or more than 8 digits are allowed. Error occurs if validation is violated. isValidName(); method is used for entering name of student it should contain uppercase or lowercase characters only, otherwise error alert is displayed.



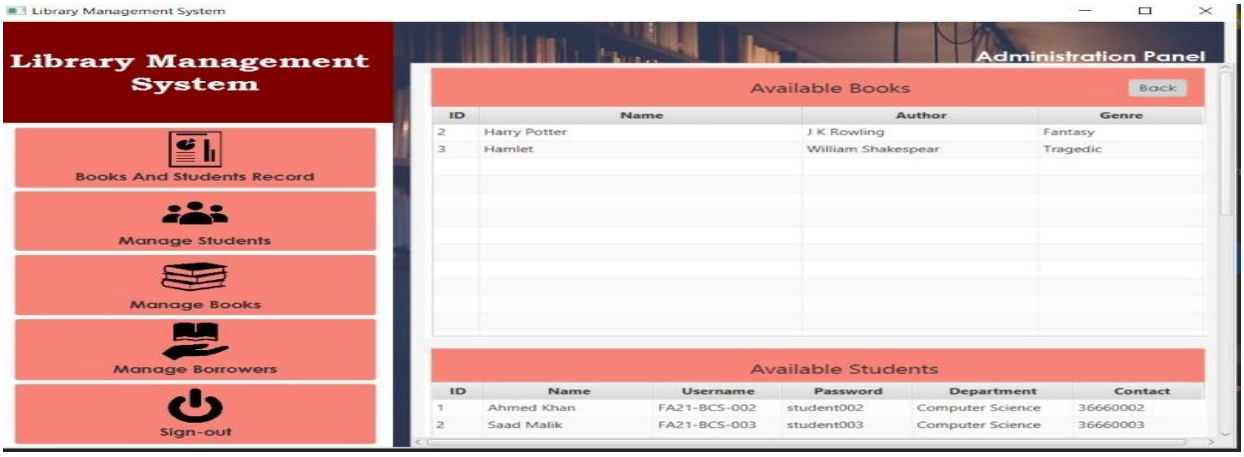
Another example: isValidUsername(); method is used for entering the username of the student. It can be characters, integers and hyphens only. If the conditions are violated error occurs.



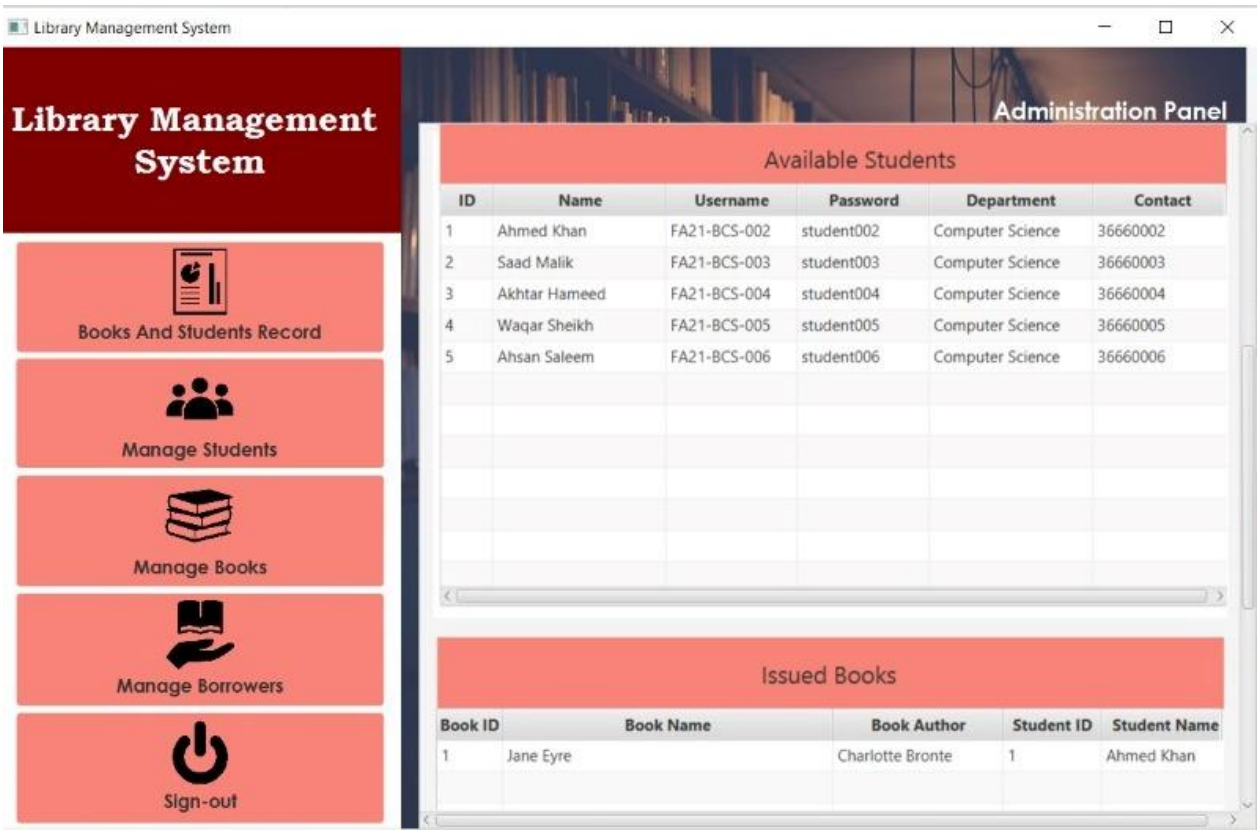
- **DbConnections** is a Java class which is used to perform basic functionality of establishing a connection with the database. This connection is then used in different methods for generating queries from the database. This class contains one method connectingToDatabase().
- **DbUtils** is a Java class which has multiple methods performing different task related to database. Here are the methods present in this class.
    1. changeScene(); this method is used to shift to new scenes. This method is being used in multiple places in this program. This method takes the path of the fxml file (GUI design file) which is used to switch into new scenes.
    2. changeSceneEnter(); this method gives Key action to the application. This method allows the user and admin of the system to perform some actions by using keys i.e. ENTER key.
    3. studentPanel(); this method is used to perform login functionality of the student panel of the system. The username and passwords of students are available in the database which are retrieved. If the username and password is not correct as mentioned in the database an error alert is displayed.
    4. adminPanel(); this method is used to perform login functionality of the admin panel of the system. The username and passwords of admin are available in the database which are retrieved. If the username and password is not correct as mentioned in the database an error alert is displayed.
- **IssueBookController** is a Java class which performs functionality on how books are issued from the system. The methods in this class are given below
    1. searchBook(); this method is used to display whether a book is available or not. It takes the ID of the book present in the database and if the book is available it displays it's availability otherwise displays an alert (Book not available).
    2. searchStudent(); this method is used to display whether a student is in the database record or not. It takes the ID of the student present in the database and if the student is in the record it displays the data of the student otherwise an alert is generated.



- **AvailableBookReport** is a Java class which performs the functionality of the books available in the Database. The books in the database are shown on the using TableView. There is one main method in this class which is performing the class's functionality.
  ObservableList<AddReport>; this method is used to add data from the tables of books from the database. This methods is used for books. This class is linked to a class which will be discussed afterwards named AddReport as specified in the ObservableList <parameters>.
  ObservableList acts as an ArrayList. It's functionality is very similar to that of ArrayList. The constructor of the addReport class is called in this method which provides the parameters to enter for this method.
  Once the parameters are added it stores the data in the form of TableView containing TableColumns as declared above the method body in this class.
  TableView is used to store the data from the database in the form of rows and columns which are displayed on the GUI (Graphical User Interface). TableColumns are used to declare columns for the TableView. The rows in the table are given by using ObservableList one by one.
  This forms a table structure on the GUI which increases readability and effectiveness of the GUI.



- **AddReport** is a Java class which gives a structure of the elements accessed by ObservableList in different classes. This class contains multiple methods i.e. Set and Get methods used for private variables in the class. A constructor is created in this class which gives the parameters that will be followed by the ObservableList methods used in different classes.
- **AddStudentReport** is is a Java class which performs the functionality of the students present in the Database. The students in the database are shown on the using TableView. There is one main method in this class which is performing the class's functionality.
  ObservableList<AddReport>; this method is used to add data from the tables of students from the database. This methods is used for books. This class is linked to a class which will be discussed afterwards named AddReport as specified in the ObservableList <parameters>.
  This class performs same functionality as that of the AddBookReport. Same table structure is generated in this class aswell.



- **Main** is an application class which is available by default in the JavaFx project. This class contains the Main method which is the most essential method of the system. It runs the main application. The first fxml file is loaded first and the rest code as executed in the given order.
- **FXML** are used to design the GUI. There are multiple FXML classes in the system each for different system. They are made using tool SceneBuilder.
- **CSS** are used for styling the GUI. Multiple features are performed using the CSS Files i.e. Button styling etc.