Library Information System:

Lito

was designed by Marlen Ibraimov

Table of Contents

First of all, I would like to thank my lecturer Mr.Nurlan Shaydullaev for helping me to acquire some basic knowledge of “Java Programming Language”. At the same time, he gave

me the opportunity to learn something new related to our module like constructors, methods, arrays, JFrames etc. Beside from my lecturer, I like to thank my other classmates for helping to understand the assignment related questions more clearly. They gave their best for completing this report on time. I thank them for their efforts.

INTRODUCTION

This assignment is based on developing an LIS (Library Information System) using “Java

Programming Language".
For that we used GUI (Graphical User Interface) in this development so that it will become more users friendly to interact.
Besides, we also added text files for user's records that are directly linked with this program. It is
so called a heart of this program where all the functions depend on it.

EXPLANATIONS

in this documentation we have given explanations of how to interact successfully with this LIS (Library Information

System). We have explained here step by step so that it will surely help users to become more user friendly with it. Below are our explanations: First Things First:
Before execute this program users need to do some works that it will run properly into their system. First they need to make sure their system is having “JDK”. If they don’t have it then they can download from this below link:
http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html
Depending on their system (Windows 64bit/32bit) they need to download and install. Then they need to add the “JAVA” files to their system “PATH” so that the system can run the program from CMD (Command Prompt). The path will show something like this “C:\Program Files (x86)\Java\jre1.8.0_25\bin;”. Now just add the address besides the current path directory and
save it. The other way they can execute this program in to download the IDE (Integrated Development
Environment) on their system. They can download ECLIPSE or NETBEANS depending on the
windows (32bit/64bit). Below is the link:
The other way they can execute this program in to download the IDE (Integrated Development
Environment) on their system. They can download ECLIPSE or NETBEANS depending on the
windows (32bit/64bit). Below is the link:
IntelliJ IDEA:

https://www.jetbrains.com/idea/

We developed this program using “Intelij IDEA”.

Execution Procedures:

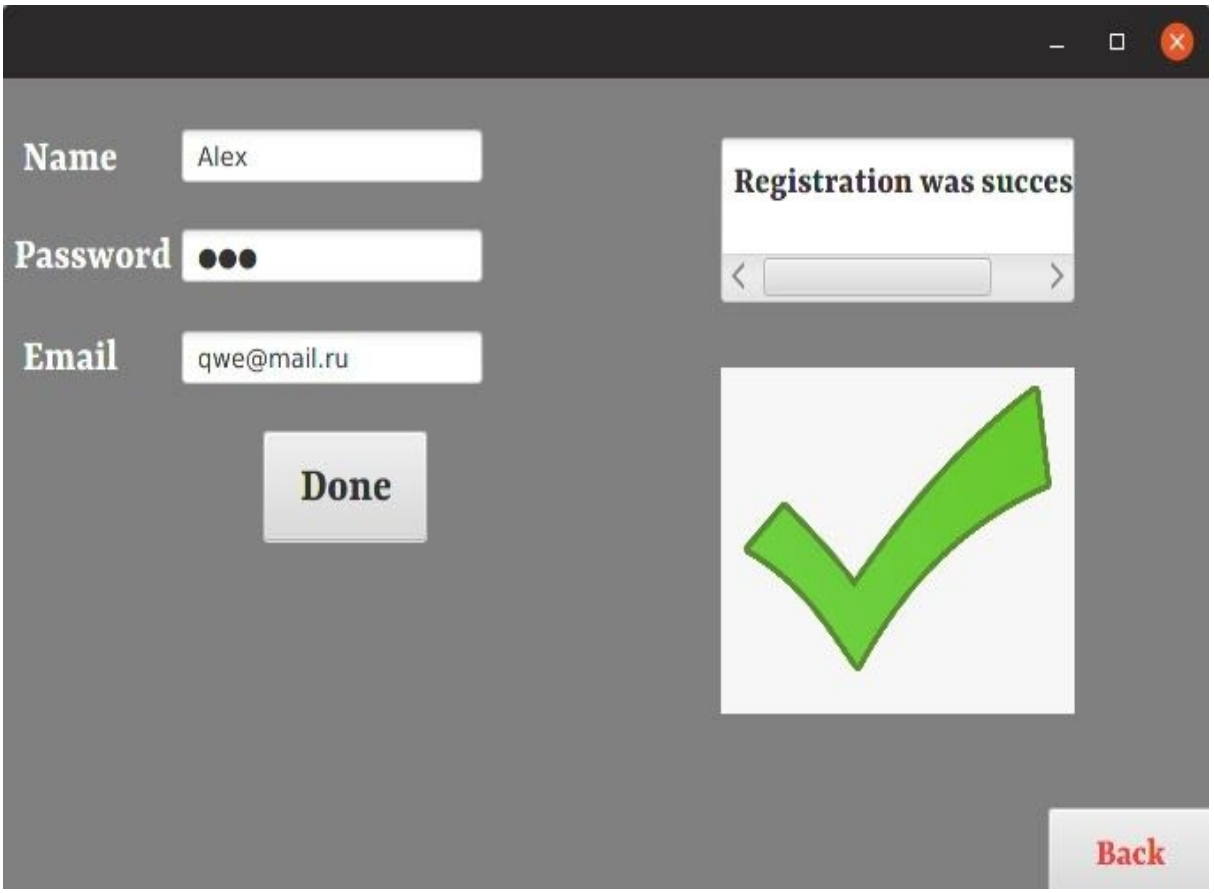
They will see the startup GUI (Graphical User Interface) of this program.

1.MENU

2. Sign Up

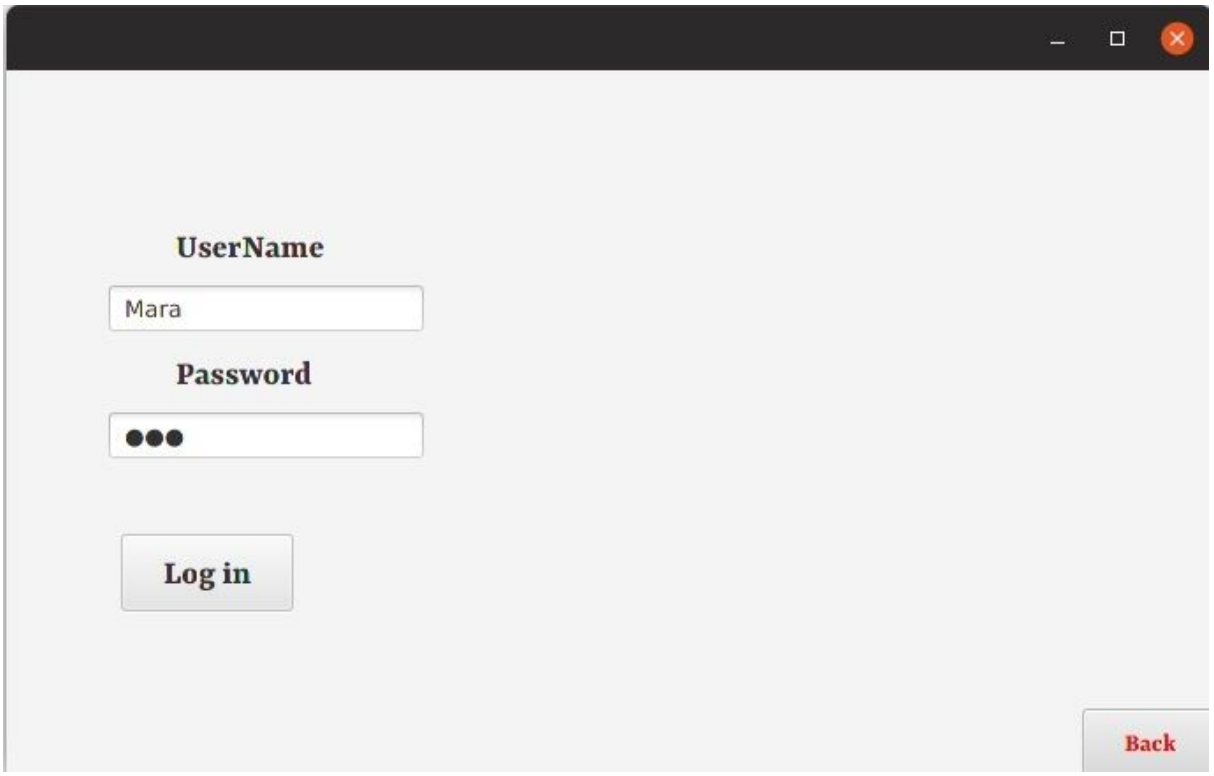
Click “Sign up” to open Registration Form

User need to fill the fields.

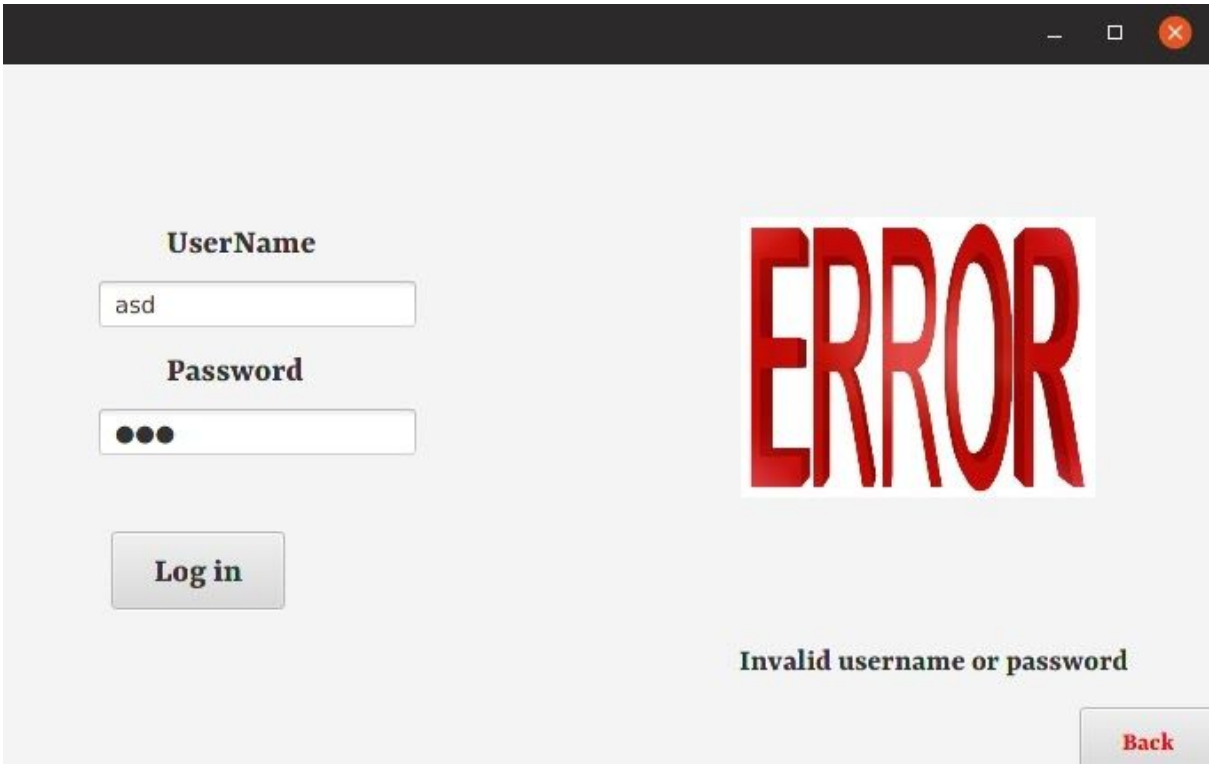


Sending data to database by using the button “Done”.

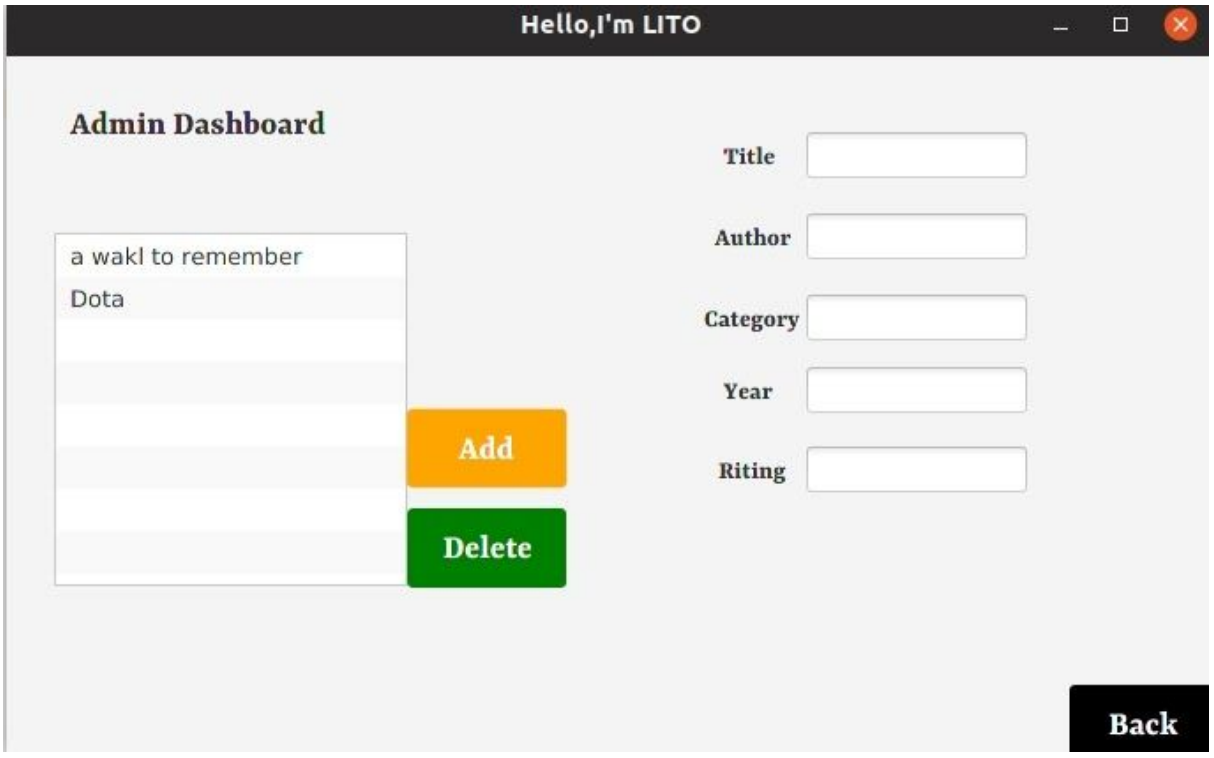
Click “Sign up” to open Registration Form



Finding user from DataBase;



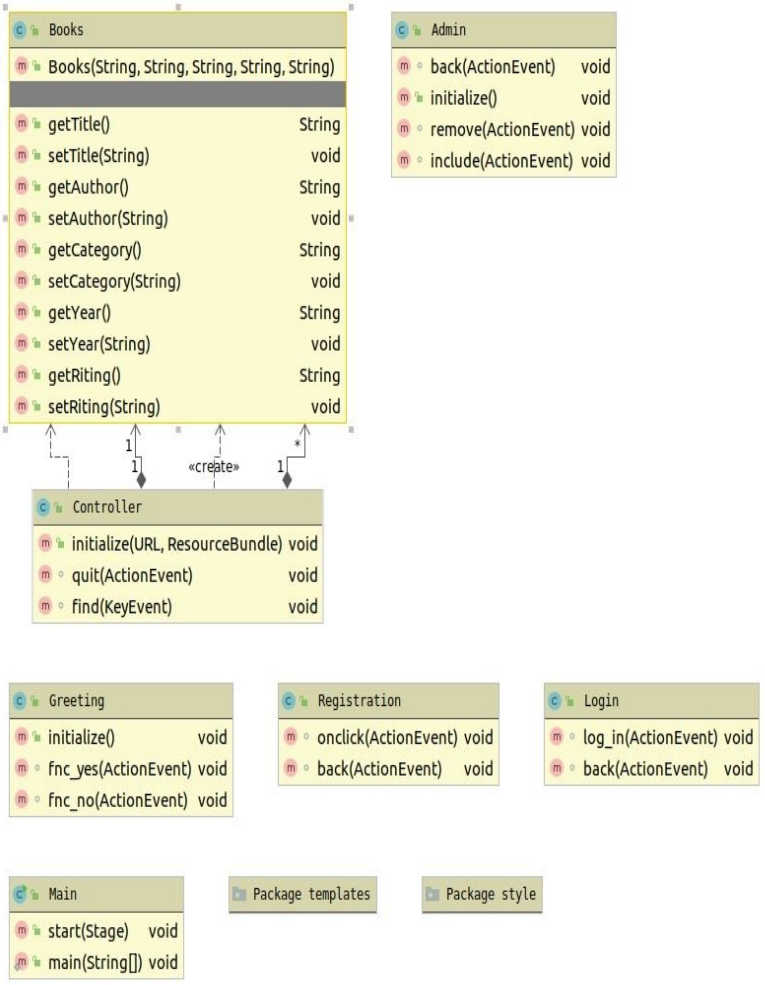
if no that user, it will throw error ;



If are logged as admin you will see admin interface ;

If you are logged as user you will see user interface where you can see books ;

OBJECT ORIENTED EXPLANATION



Books and Controller have depency to each other.

```
public void initialize(URL url, ResourceBundle rb) {
    try {
        Books books;
        Books books1;
        String myUrl = "jdbc:mysql://localhost/lito";
        Connection conn = DriverManager.getConnection(myUrl,  user: "username",  password: "password");
        String get_request = "SELECT * FROM books";
        Statement statement = conn.createStatement();
        ResultSet rs = statement.executeQuery(get_request);
        choicebox.getItems().add("EN");
        choicebox.getItems().add("RU");
        choicebox.getSelectionModel().selectedIndexProperty().addListener(new ChangeListener<Number>() {
            @Override
            public void changed(ObservableValue<? extends Number> observableValue, Number number, Number number2) {
                String choice = choicebox.getItems().get((Integer) number2);
                if (choice.equals("EN")) {
                    en_content.setVisible(true);
                    ru_content.setVisible(false);
                }
                if (choice.equals("RU")) {
                    en_content.setVisible(false);
                    ru_content.setVisible(true);
                }

            }
        });

        while (rs.next()) {
            books = new Books(rs.getString( i: 2), rs.getString( i: 3), rs.getString( i: 4), rs.getString( i: 5), rs.getString( i: 6));

            oblist.addAll(books);

        }
        get_request = "SELECT * FROM ru_books";
        ResultSet rs_ru = statement.executeQuery(get_request);

        while (rs_ru.next()) {

            books1 = new Books(rs_ru.getString( i: 2), rs_ru.getString( i: 3), rs_ru.getString( i: 4), rs_ru.getString( i: 5), rs_ru.getString( i: 6));
            oblist_2.addAll(books1);


        }
```

The initialize method is called after all @FXML then it create Books instances which will be inside ObservableList that we need to implement by FXCollections.*observableArrayList*(); Then I will put all of that into TabbleView (JavaFX) to represent.

Also there was used Listener for track behavior of choicebox.
In order to use DB we:

1. Establishing a connection
2. Create a statement
3. Execute the query

```java
public class Books {

    String title, author, category, year, riting;

    public Books(String title, String author, String category, String year, String riting) {
        this.title = title;
        this.author = author;
        this.category = category;
        this.year = year;
        this.riting = riting;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) { this.title = title; }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) { this.author = author; }

    public String getCategory() { return category; }

    public void setCategory(String category) { this.category = category; }

    public String getYear() { return year; }

    public void setYear(String year) { this.year = year; }

    public String getRiting() { return riting; }

    public void setRiting(String riting) { this.riting = riting; }
}
```

My class Books with setter getter and own constructor.

ASSUMPTION

I tried to make this app more useful for myself but i will fix some functionality and add flexible oporrtunities for users not only admins.
For instance people should have a possibility to add own favorites books and order books that not inside in the Library;

Thanks for attention :)

Reference

Github: https://github.com/Marlen1703/qwe.git
JavaFX:https://openjfx.io/
Intellij IDEA: https://www.jetbrains.com/idea/