Library Information System
Lito

Ravshan Khamidov com-18
Ala-Too International University

Table of Contents

INTRODUCTION

This program is based on a _Library System_ using "Java Programming Language". For that we used **JavaFX** in this development so that it will become easy with graphical interface another word with **Scene Builder** program.

Besides, I also studied MySQL for storing data in Database.

EXPLANATIONS

In this documentation we have given explanations of how to interact successfully with this program "Heart of Math". We have explained here s by step so that it will surely help users to become more user friendly with Below are our explanations:

First Things First: Before execute this program, users need to do some wo that it will run properly into their system. First, they need to make sure th system is having "JDK". If they don't have it then they can download from below link:

https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html

Depending on their system (Windows 64bit/32bit) they need to download install. Then they need to add the "JAVA" files to their system "PATH" so t the system can run the program from CMD (Command Prompt). The path show something like this "`Program Files\Java\jdk1.6.0_02`.".
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 downlo IntelliJ IDEA or Eclipse depending on the windows (32bit/64bit). Below is th link:

IntelliJ IDEA:

https://www.jetbrains.com/ru-ru/idea/download/
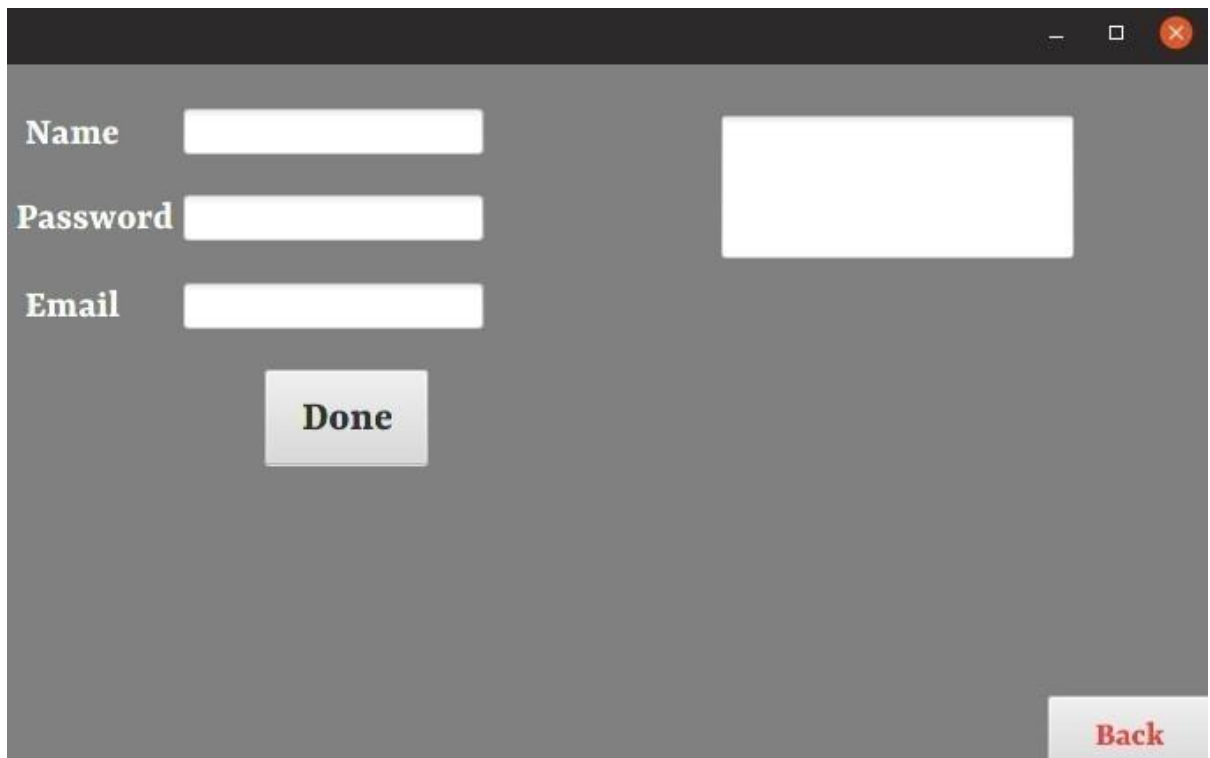
Eclipse:

http://www.eclipse.org/downloads/

We developed this program using "IntelliJ IDEA".

Execution Procedures:

When user executes this program, it will show this:

There is my main menu, as you can see there are two options .

After clicking Sign up you will see this:

It is my registration form to creating new user that will have opportunity t
my program

If the records satisfy requirements(if no user with same name and
password) ,record will be saved into database system. In another case it
throw error.

Clicking Log in

Then you will see new window:

We are logged as user Mara

There are books inside my system and user can go to library and take boo
one week.

Also you can log in as admin:

We are logged as Admin user

Admin can add books and delete books from system. It's more useful.

OBJECT ORIENTTED EXPLANATION

In object-oriented programming, for example, an object is a self-contained entity that consists of both data and procedures to manipulate the data. I other way, object oriented is the software engineering concept where it is represented using the "OBJECTS".Below are the objected oriented sample used in this "Java Programming Language".

Dependency:
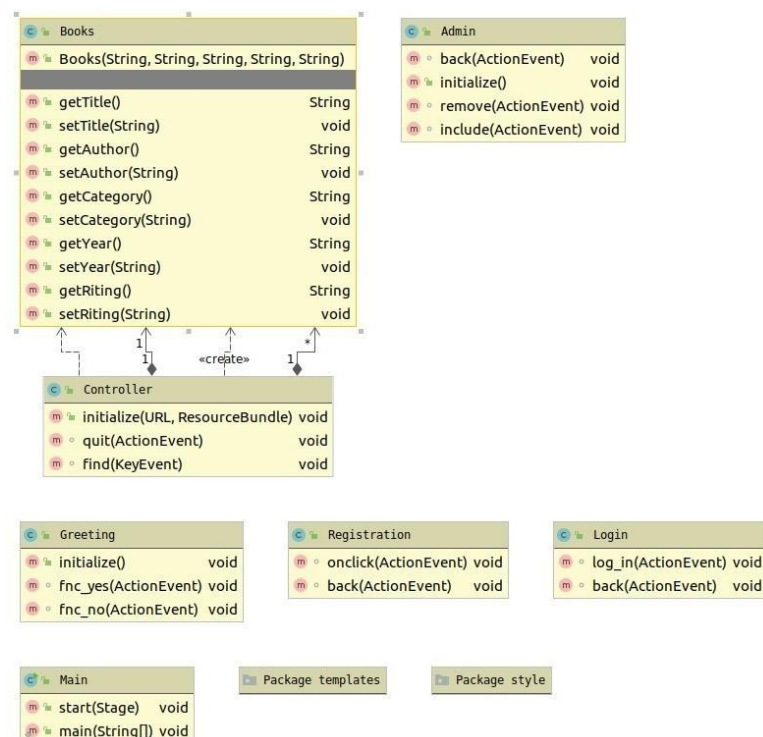
As you can see there are dependencies. I did notfused a lot o

```java
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( 2), rs.getString( 3), rs.getString( 4), rs.getString( 5), rs.getString( 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( 2), rs_ru.getString( 3), rs_ru.getString( 4), rs_ru.getString( 5), rs_ru.getString( 6));
            oblist_2.addAll(books1);

        }
```

The initialize method is called after all @FXML then it create Books instan
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 Book class with own setter,getter and Book constructor.

ASSUMPTION

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

Reference

Github: https://github.com/rockyxam

JavaFX: https://openjfx.io/
Intellij: IDEA: https://www.jetbrains.com/idea/