STUDENTS’ RESULTS PROCESSING SYSTEM

BY LINDA KUMAH

PROJECT REPORT

AMALITECH JAVA ONLINE COURSE

Submitted on 25th March, 2021

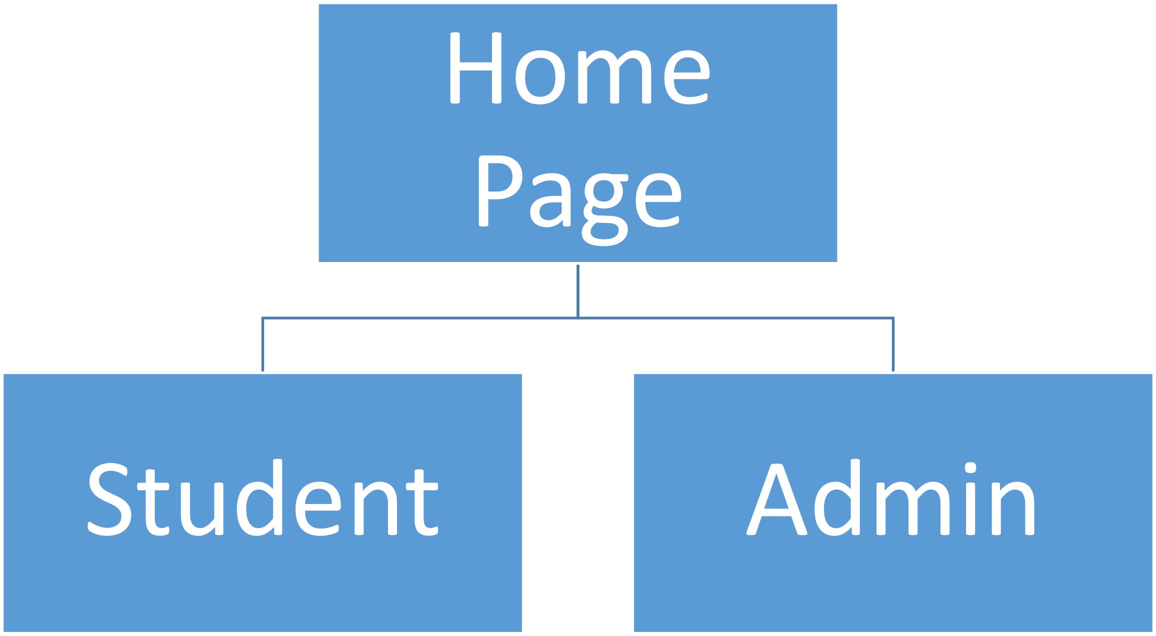
ABSTRACT

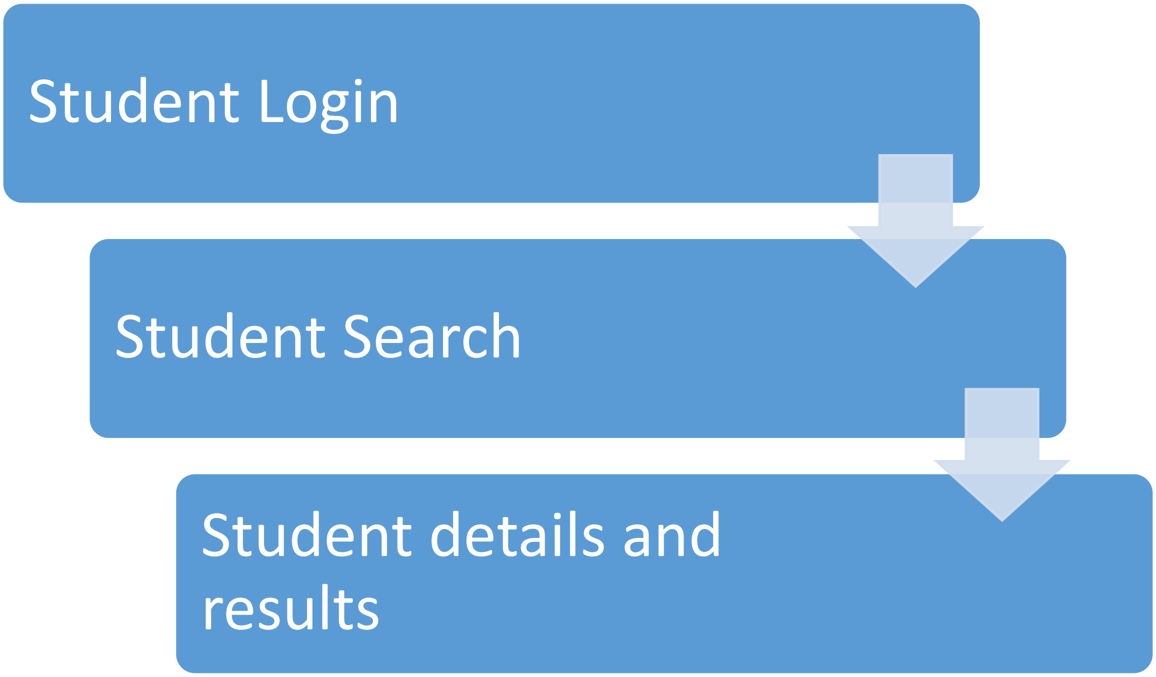
This project uses Java Programming Language to develop a system that stores the scores results and registration details of Students. The software is aimed for Schools to store the Students details and results. It allows the administrator to store huge amount of data, records and information and also, allows the students to enter their login details and have access to their results. Various controls have been put in place to provide User friendliness and also, provide a high level of security. The overall processing system for entering Students' results is flexible, easy and requires less time.

INTRODUCTION

Most Schools and colleges need a system that manages the results of their students but how safe and secured are the data that are entered into the School's results processing system? In this project work, systems have been put in place to ensure that results entered are safe. Also, the user interface is made very friendly which makes its usage easier and less time-consuming to use.

FLOW OF APPLICATION

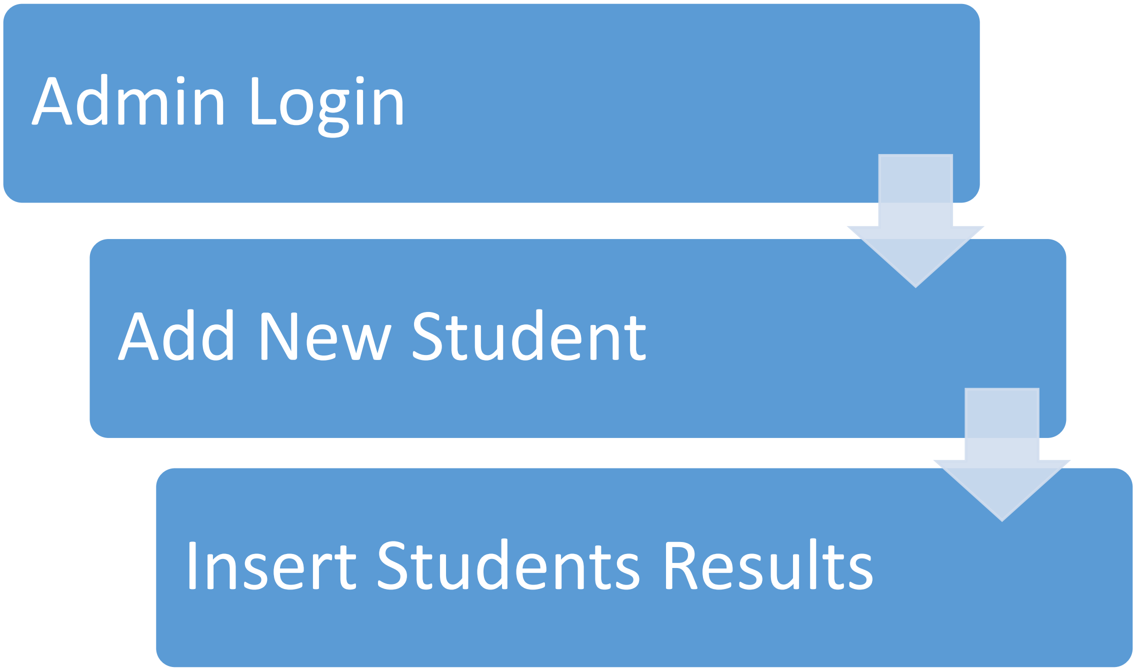




Student Login

Student Search

Student details and results



Admin Login

Add New Student

Insert Students Results

METHODS

This project uses Java GUI to create the interfaces and uses Java programming language to write the source codes. Some of the properties of the Java GUI that were used includes the following:

1. JPanel which is a lightweight container for organizing the components in the interface.
2. Absolute Layout for positioning the components at the absolute wanted position.
3. JLabel as a display area for short text such as Name, Course Name, Gender, etc.
4. JTextfield which is a lightweight component for editing a single line of text.
5. JButton for implementing push buttons.
6. JPasswordfield for text in which the original characters do not show.
7. JTable for data in tabular form.
8. JCombobox for an editable field that allows the user to select from the drop-down list. This was used for Gender, Course Name and Branch Name.

This project uses SQLite to create the database that was used for the application. The database created consists of four tables as described below.

Table 1 below is the "User" Table which contains the Username and password of the Administrator as shown below.

Fields	**Inputs**
Username	Linda
Password	12

Table 2 below is the "StudentUser" Table which contains the Username and password of the student as shown below.

Fields	**Inputs**
Username	Lina
Password	12

Table 3 below is the "InsertResults" Table which contain the details for registering a new student as shown below.

Fields	**Data Types**
ID	INTEGER
Roll Number	INTEGER
History	INTEGER
Computer	INTEGER
Communication	INTEGER
Hardware	INTEGER
Data Science	INTEGER

Table 4 below is the "Admin" table which contains the details for entering a new Students' results as shown below.

Fields	Data Types
ID	INTEGER
Course Name	TEXT
Branch Name	TEXT
Roll Number	INTEGER
Name	TEXT
Gender	TEXT
Mother’s Name	TEXT

In the AdminHome.java source code, the code below was used to save the new details entered by the Administrator for registering a new student to the table. If the Administrator clicks on “Create”, the new details are added to the table. Also, in the InsertTheResults.java source code, the same code was used to save the new results entered by the Administrator to the table.

```
public void RefreshTable() {
                try {
                                 String query = "select * from InsertResult";
                                 PreparedStatement ps =
connection.prepareStatement(query);
                                 ResultSet rs = ps.executeQuery();

                                 table.setModel(DbUtils.resultSetToTableModel(rs));
                                 rs.close();
                                 ps.close();

                } catch (Exception e1) {
                                 e1.printStackTrace();
                          }
            }
```

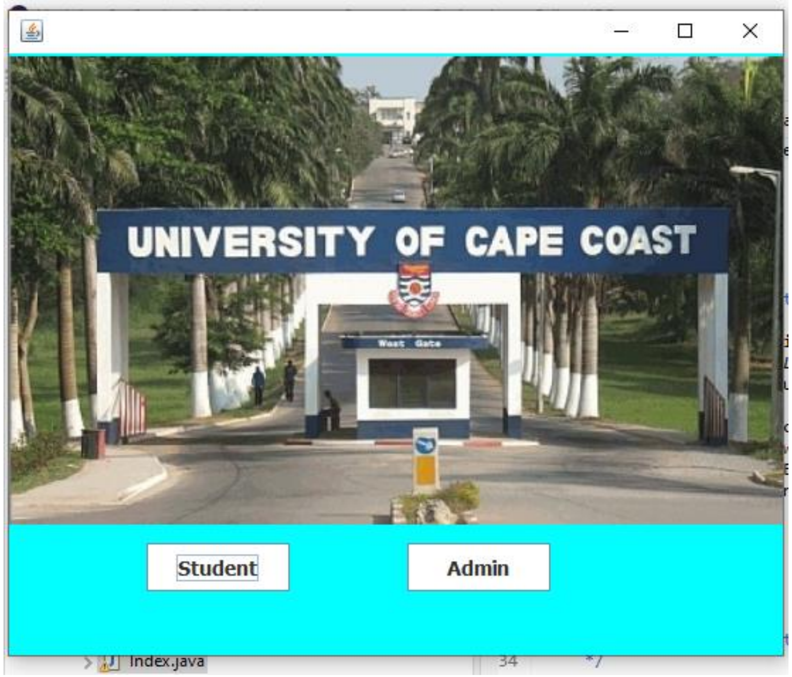
USER REQUIREMENTS

The Requirements for the Application are shown below

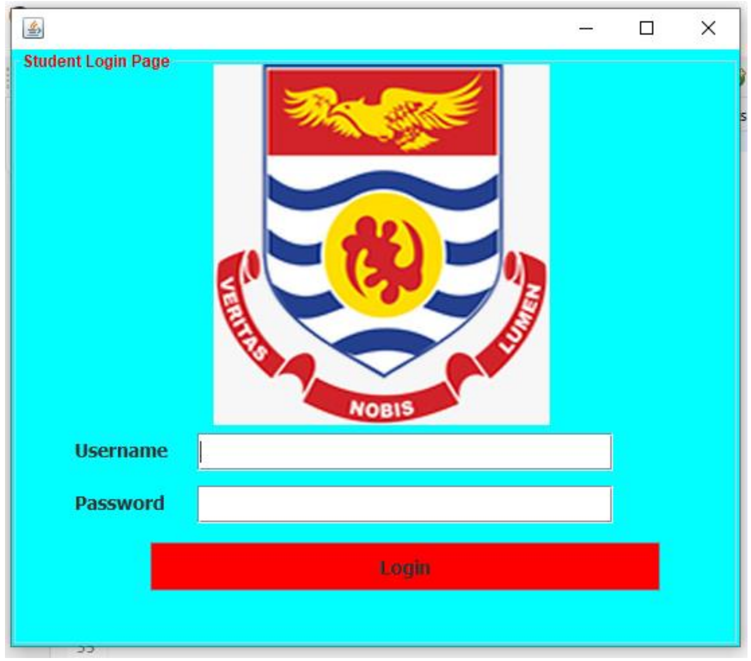
1. Inputs
    - Student login (Username and password)
    - Student search (Roll number)
    - Admin login (Username and password)
2. Outputs
    - Students’ details and results

How To Run the Program

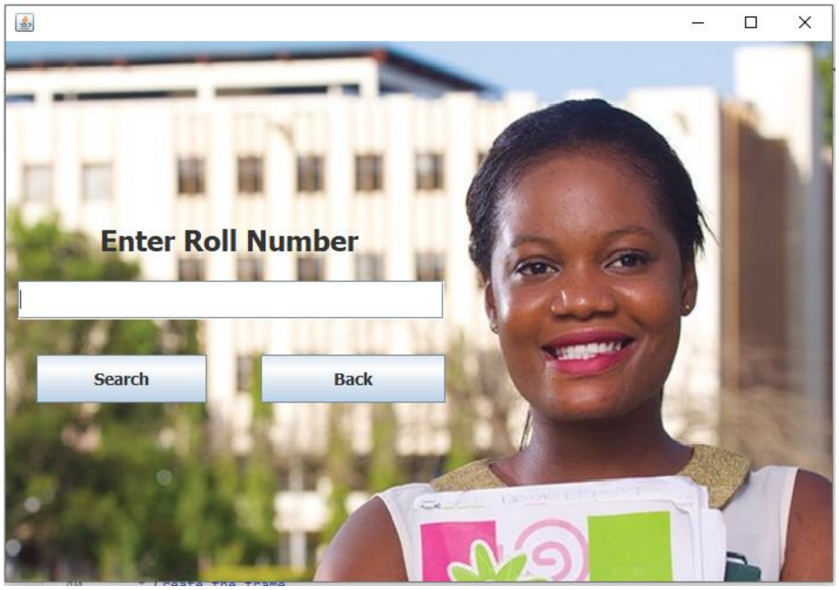
1. Create a database of tables 2 and 3 discussed above using SQLite.
2. Start from the Project.java class and run it. The interface below appears.



3. A click on the Student button takes you to the StudentLogin.java class and the interface below appears.

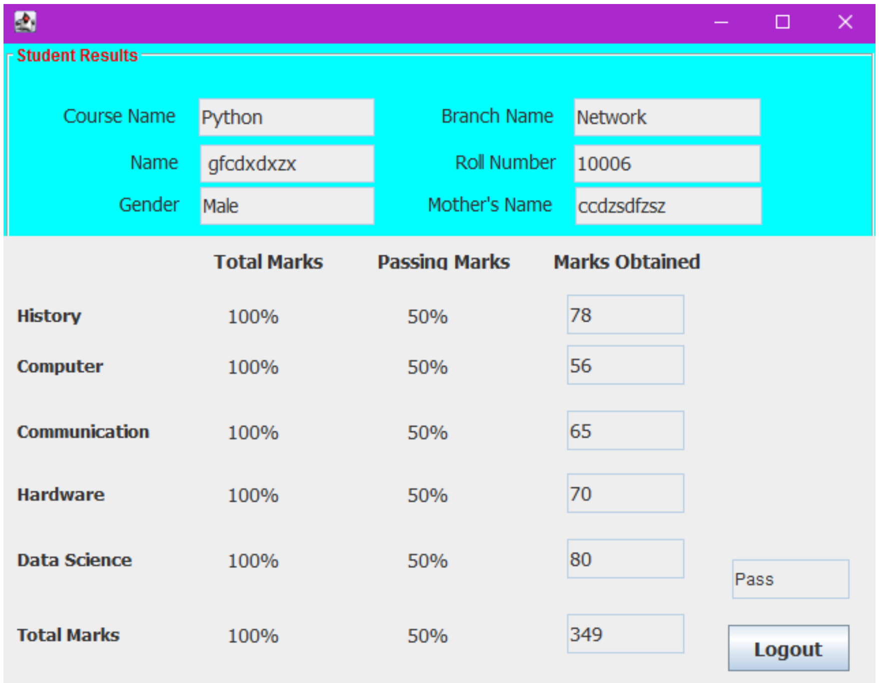


4. A student then enters his/her password and username as shown above and clicks on Login. This takes you to the StudentIndex.java class and the interface below pops up.



5. The student now enters his/her roll number and click on Search. If a correct roll number is entered, it takes you to the StudentHome.java class and the interface below shows up. The interface below displays the Student’s details and results. If an incorrect roll

number is entered, the statement “Incorrect Roll Number” pops up. A click on the Logout button takes you back to the StudentLogin.java class that requires the Student to enter his/her Username and Password. The passing mark is 50%. If all the marks for the subjects indicated are equal to or above 50, the button at the bottom right corner indicates ”Pass”. Otherwise, it indicates “Fail”.



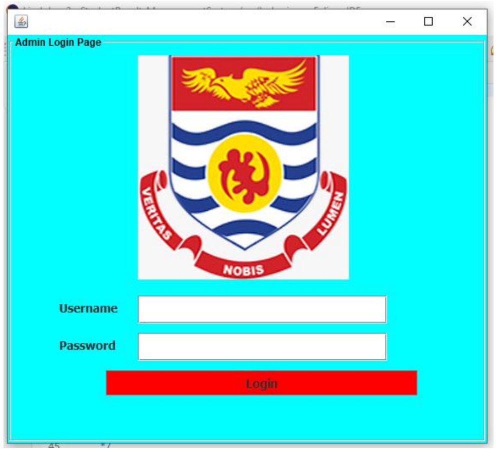
Student Results

Course Name: Python
Branch Name: Network
Name: gfcdxdxzx
Roll Number: 10006
Gender: Male
Mother's Name: ccdzsdfzsz

	Total Marks	Passing Marks	Marks Obtained
History	100%	50%	78
Computer	100%	50%	56
Communication	100%	50%	65
Hardware	100%	50%	70
Data Science	100%	50%	80
Total Marks	100%	50%	349

Pass

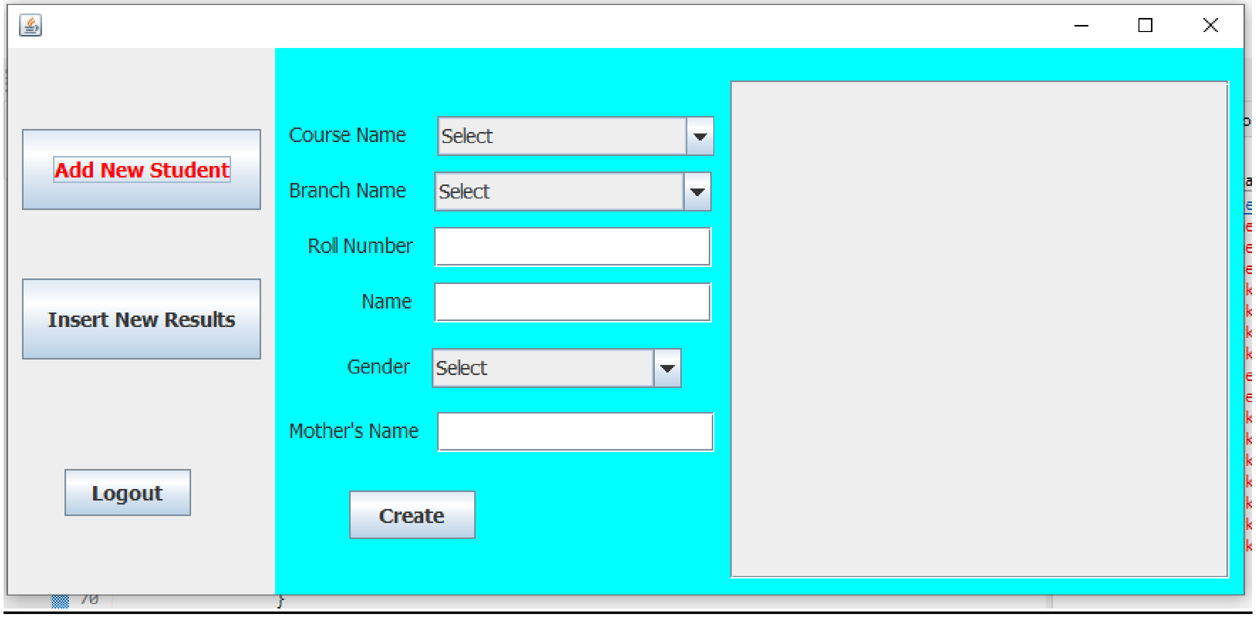
6. When you run the Project.java class, a click on the Admin button takes you to the Index.java class which requires the Administrator to enter his/her login details as shown in Table 1 above. The interface is shown below.



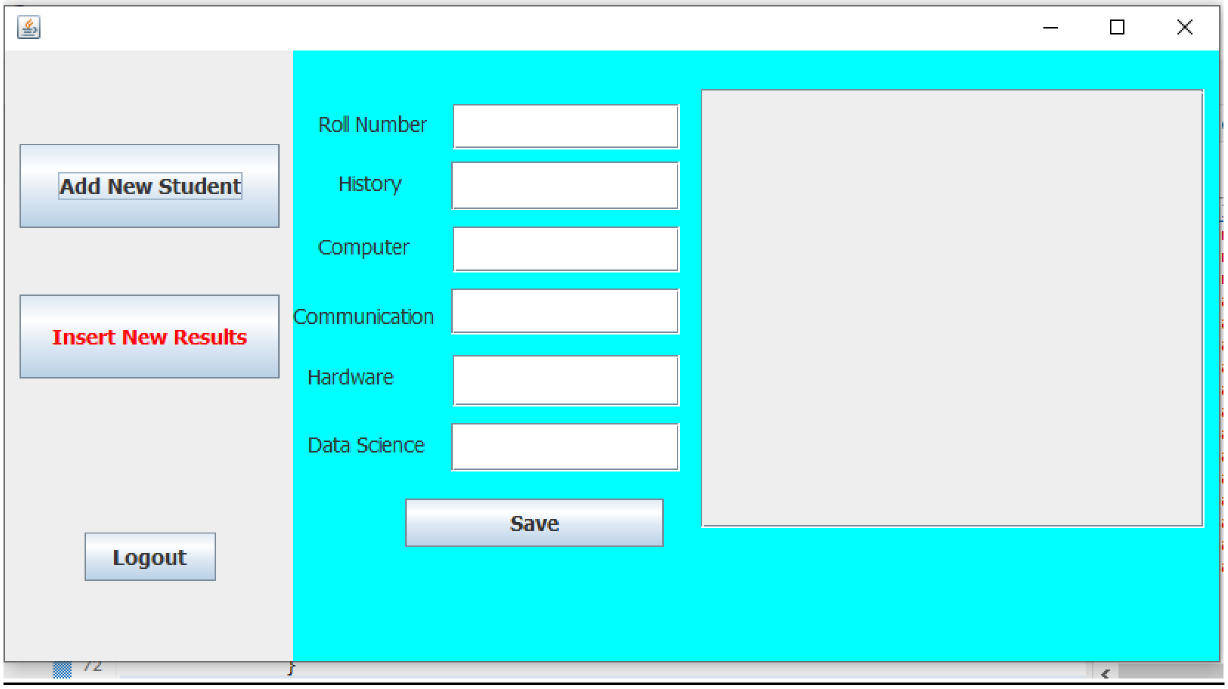
Admin Login Page

7. When the Admin enters his/her Username and Password and clicks on Login, it takes you to the AdminHome.java class where the Administrator can register a new student.

The interface is as shown below. When the Administrator enters the details of the new Student and clicks on create, the new details is added unto the database and it will be added to the table created beside the details. A click on the Logout button takes you back the Index.java class where the Administrator's Username and Password is requested.



8. From the Interface above, the Administrator can click on "Insert New Results" and it will take you to the InsertTheResults.java class where the Administrator can enter the Roll Number of the Student and the new results as shown in the interface below. When the Administrator enters the Roll Number of the Student and the new results and clicks on save, the new results are added unto the database and it will be added to the table created beside the details. A click on the Logout button takes you back to the Index.java class where the Administrator's Username and Password is requested.



CONCLUSION

The Student Results Processing System Application is a very effective and efficient GUI-based component. This software is well tested and it works properly to meet the user requirements described above. It currently gives Students' details and results. However, further enhancements such as results printing, aggregate calculation, etc. can be incorporated into the system.