****

**Green University of Bangladesh**

**Department of Computer Science and Engineering (CSE)**

**Faculty of Sciences and Engineering**

**Semester: (Summer, Year:2021), B.Sc. in CSE (Day)**

**Course Title: Object Oriented Programming Lab**

**Course Code: CSE 202**

**Section: DA**

**Lab Project Name: Student Information Collection Using Java/Java Swing**

**Student Details**

|  |  |  |
| --- | --- | --- |
|  | **Name** | **ID** |
| **1.** | Jahid Hasan Santo | 201002463 |
| **2.** | Dyana Dipa Gomes | 201002480 |

**Submission Date : 05/09/2021**

**Course Teacher’s Name : Arnab Kanti Tarafder**

**[For Teachers use only: Don’t Write Anything inside this box]**

|  |
| --- |
| **Lab Project Status**  **Marks: ………………………………… Signature: .....................**  **Comments: .............................................. Date: ..............................** |

Table of Contents

[Chapter 1: Introduction 3](#_Toc80980720)

[**1.1 Introduction** 3](#_Toc80980721)

[**1.2 Design Goals/Objective** 3](#_Toc80980722)

[Chapter 2: Design of the project 3](#_Toc80980723)

[**2.1 Design** 3](#_Toc80980724)

[**2.2.1 Heading panel** 3](#_Toc80980725)

[**2.2.2 Promotion panel** 4](#_Toc80980726)

[**2.2.3 Info panel** 4](#_Toc80980727)

[**2.2.4 Button panel** 4](#_Toc80980728)

[**2.2.5 Table panel** 4](#_Toc80980729)

[Chapter 3: Implementation 4](#_Toc80980730)

[**3.1 Container:** 4](#_Toc80980731)

[**3.2 JFrame:** 4](#_Toc80980732)

[**3.3 JPanel:** 5](#_Toc80980733)

[**3.4 JLabel:** 5](#_Toc80980734)

[**3.5 JTextField & JTextArea:** 5](#_Toc80980735)

[**3.6 JPasswordField:** 5](#_Toc80980736)

[**3.7 JButton:** 5](#_Toc80980737)

[**3.8 JScrollPane:** 5](#_Toc80980738)

[**3.9 ActionListener:** 6](#_Toc80980739)

[**3.10 MouseListener:** 6](#_Toc80980740)

[Chapter 4: Performance Evaluation 6](#_Toc80980741)

[**4.1 Simulation procedure** 6](#_Toc80980742)

[**Add Button:** 7](#_Toc80980743)

[**Clear Button:** 7](#_Toc80980744)

[**Update Button:** 8](#_Toc80980745)

[**Print Button:** 8](#_Toc80980746)

[**Delete Button:** 9](#_Toc80980747)

[Chapter 5: Conclusion 9](#_Toc80980748)

[**5.1 Scope of future work** 9](#_Toc80980749)

[**5.2 Practical implications** 9](#_Toc80980750)

# Chapter 1: Introduction

## **1.1 Introduction**

We have created a project named Student Information which is a window-based application. Here we used Java Swing to create this project. By this project we can store student information, can update it, delete it etc.

## **1.2 Design Goals/Objective**

This project we made that will help a institute to manage the data more easier and will give better clarity. That will be very helpful for administration. In future if we want to handle huge amount of data then we can use this project by developing it.

# Chapter 2: Design of the project

## **2.1 Design**

## **2.2.1 Heading panel**

This is the first panel of our Frame and here we putted our heading or title of the project that is student information.

## **2.2.2 Promotion panel**

This panel is for promoting. That means if any institution wants to promotion any person then they can do it by their information by putting in this section. Here we gave our information.

## **2.2.3 Info panel**

This panel is for one of the important work of our project that is taking information. Here we created section that will take Student name, ID, email and other related data from user.

## **2.2.4 Button panel**

So this is another important panel. In this panel we have putted 5 buttons they are add, update, print, delete, clear. By pressing this buttons we can perform actions that we want for the data.

## **2.2.5 Table panel**

So this the last panel. In this panel we can see the listed data of the individual student or person that we collected from the user.

# Chapter 3: Implementation

This project is very easy to develop and implement as we have done this project by a very simple code. Now in this section I will discuss about the main components what we used to implement this project

## **3.1 Container:**

Container can be described as a special component that can hold collection of components such as frame, panel, label, text-field etc. So to hold all components that we have to use in our project at first we need a container that’s why added a container.

## **3.2 JFrame:**

As we know JFrame works like a main window where we can cad components like button, panel, text-field etc. as our need. So we extended JFrame class in our source code.

## **3.3 JPanel:**

In our project we need to take information, need to set a heading, need to see the data list. That’s why for better management and we used panels in our project. We have created heading panel, promotion panel, info panel, button panel, table panel in which we have contained different components as per our need.

## **3.4 JLabel:**

We know that JLabel is used for adding text and images in the frame so used it to set text in our frame. It will help a user to know what type of information where to input in the frame that the reason of using JLabel in our frame.

## **3.5 JTextField & JTextArea:**

We used JTextField in our info panel to write the data inputted by the user. And in promotion panel we used Text area as we needed multiple line text.

## **3.6 JPasswordField:**

Before get into our main frame we added a user name and password for security purpose. So we used JPasswordField here because this component is specialized for password entry. And also we set echo character for JPasswordField.

## **3.7 JButton:**

After taking information we have to perform action. So we used buttons for perform action. After pressing the one button it will perform action accordance to it.

## **3.8 JScrollPane:**

As we set a size of our table panel so when we take much information from user it is not possible to list the data in the visible screen size. That’s why we have to use a scrollbar in our screen size. So we used it in table panel as it gives a scrollable view of a component.

## **3.9 ActionListener:**

As we know ActionListener handles all the actions such as when the user clicks on a component. Mostly, action listeners are used for JButtons. So we used it for perform action.

## **3.10 MouseListener:**

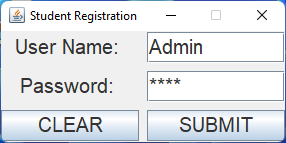
When we need to update information or delete a row from the list, we have select the simply by clicking the mouse. So to track mouse moves or mouse drags we use MouseListener.

# Chapter 4: Performance Evaluation

## **4.1 Simulation procedure**

This project is very user friendly because we have created this project with a very simple code. In this project a lot can be done in a few things. As it is quite easy to implement it will helpful for a user to maintain. By simulating this project it doesn’t create any pressure on our device as a result it not hamper our device speed.

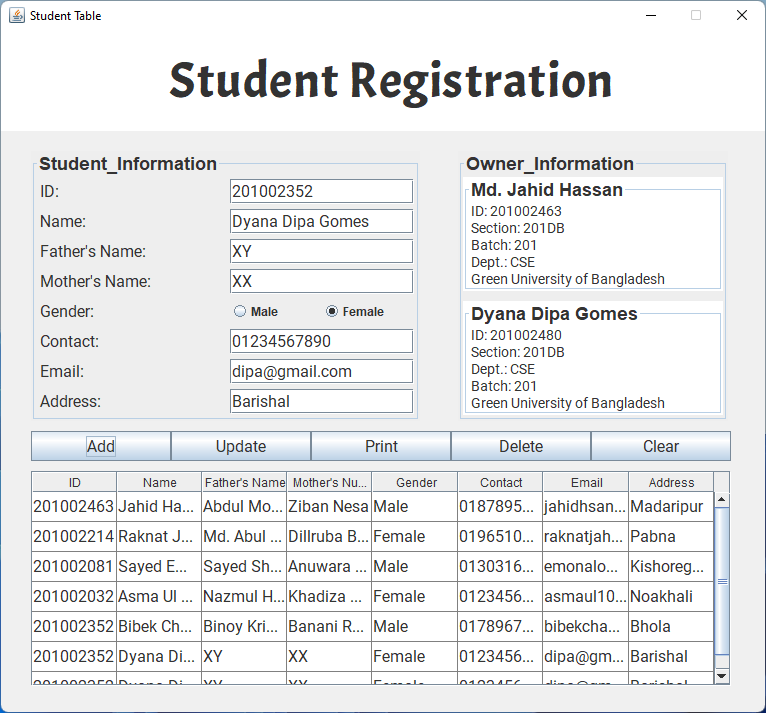
After creating this this project, we exported it in a .Jar file that’s why user can easily open it and can simulate it.

Firstly We have to input UserName & Password to access the form=>

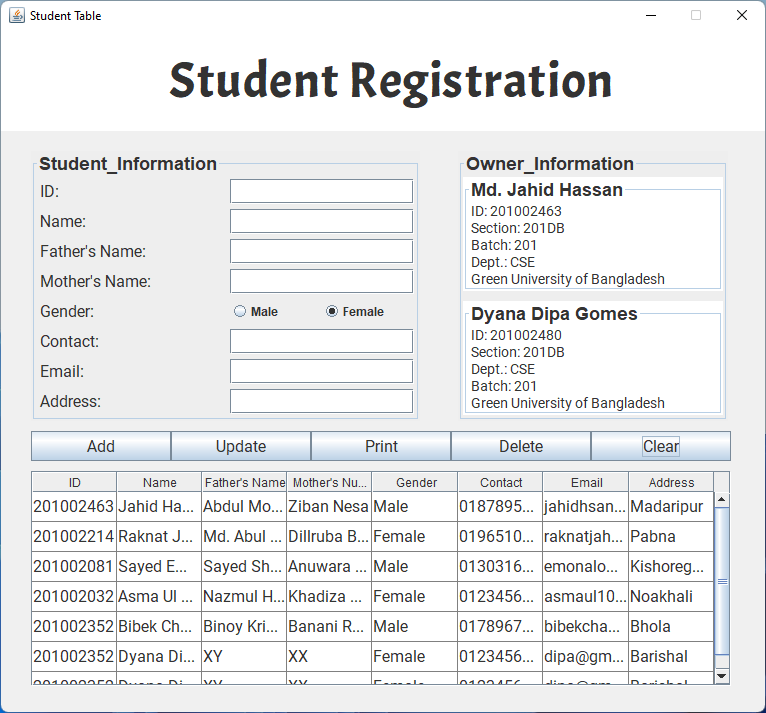
User Name: Admin Password: 2468

After hit the Submit button we will get the Registration Form. Here is some action on Registration Form.

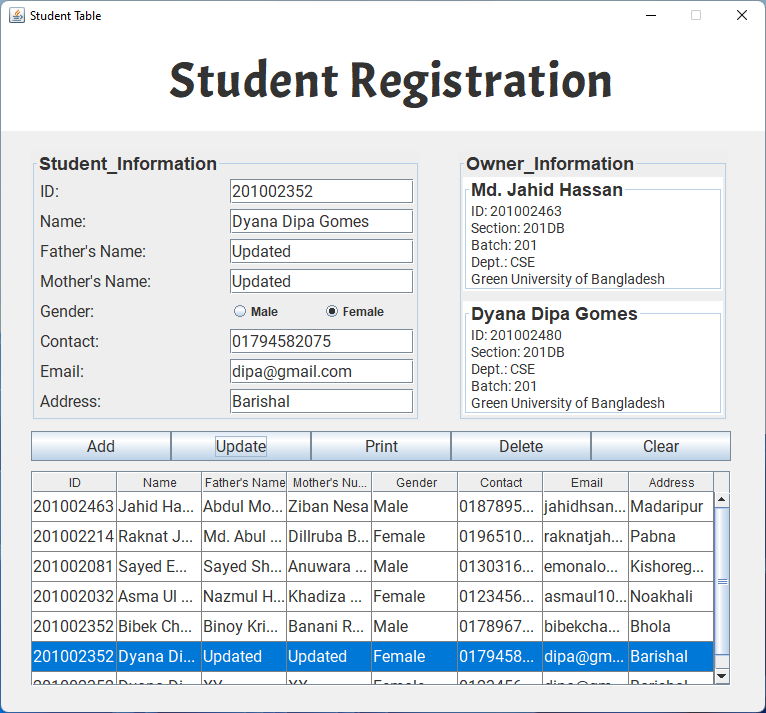
### **Add Button:**

****After Add ActionButton=>

### **Clear Button:**

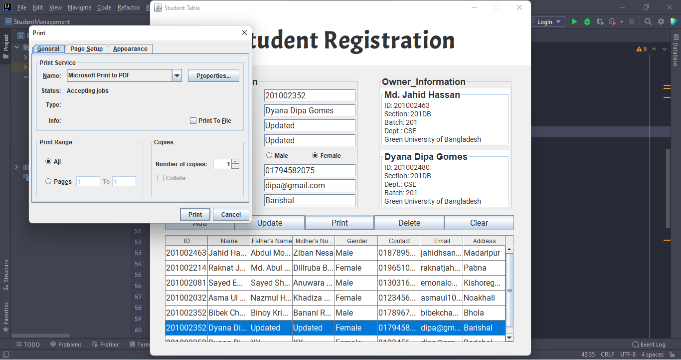
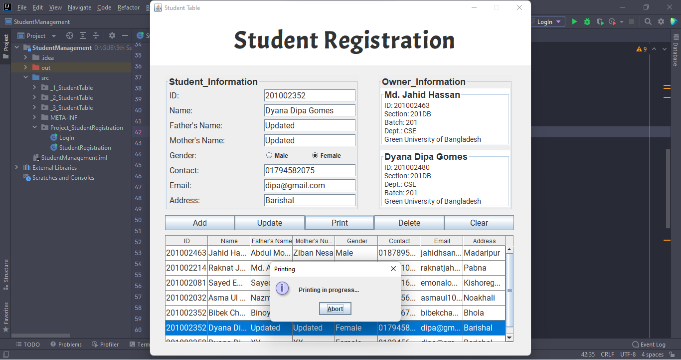
After Clear ActionButton=>

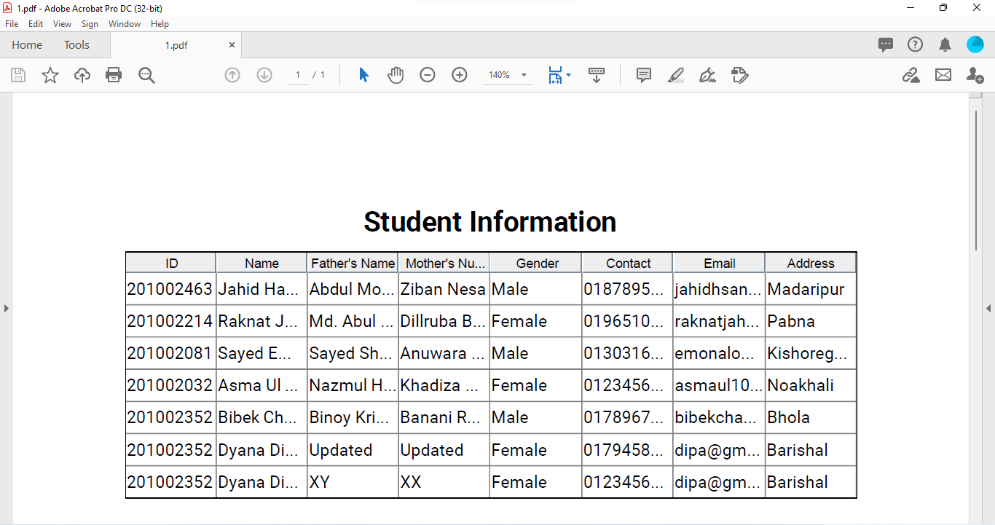
### **Update Button:**

After Update ActionButton=>

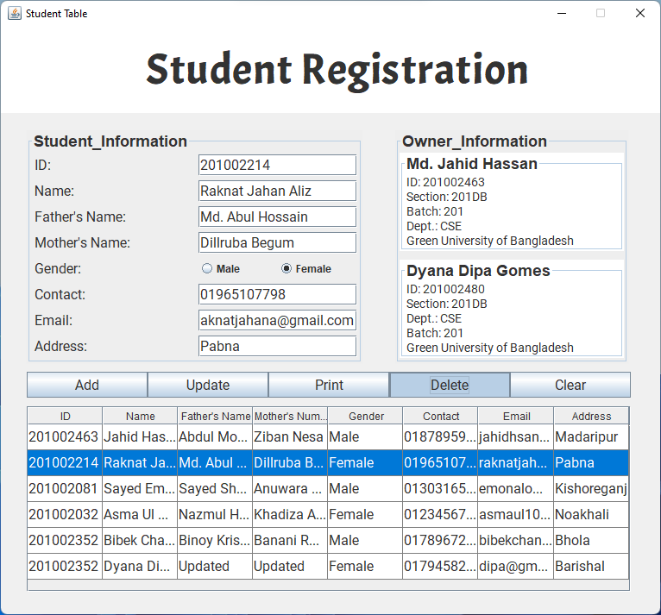
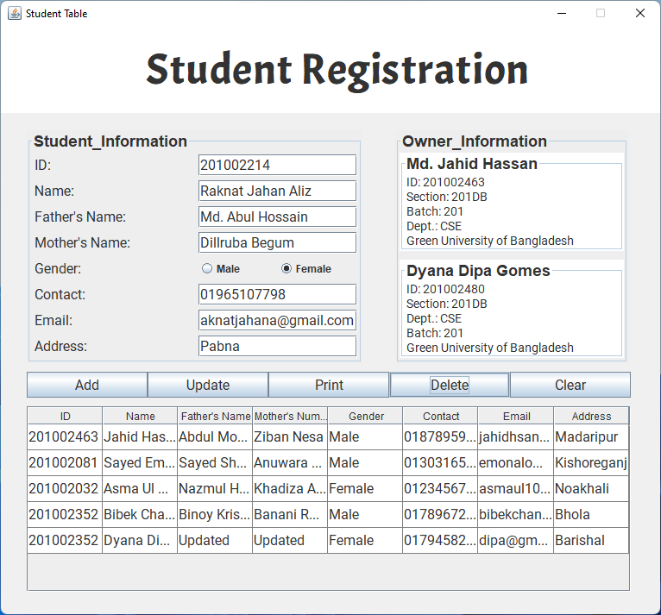
### **Print Button:**

After Print ActionButton=>





### **Delete Button:**

After Delete ActionButton=>

# Chapter 5: Conclusion

## **5.1 Scope of future work**

In future we if we want to do work with a huge range of data then we can do it by developing this project or we can create an advanced project we can say software that can handle a huge data by the concept of our project.

## **5.2 Practical implications**

This project is very useful for educational institutions. By this application admissions department can track prospective students during the application and enrollment process.

In future if we develop it or make any advanced project by the concept of it we can use it huge data entry projects, in survey