Group name: ALPHA

Group members:

1. Aktaruzzaman Ridoy (F16040112)
2. Fabiha Tafannum (F16040113)
3. Md. Nazim Uddin Nayim (F16040114)
4. Md. Jahidul Kabir (F16040117)
5. Golam Jilany Rayhan (F16040102)

Final Project:

Project name: STUDENT REGISTRATION AND COURSE SELECTION

STUDENT REGISTRATION AND COURSE SELECTION

INTRODUCTION

This is a course (Software Engineering) to build our own software and we planned to make one named ‘Student Registration System and Course Selection’. It’s basically for any types of probationers. First of all, we chose this software to make it very easy for the students (specially the new students) to register their personal information and second to search for the courses and choose as one's comfort. But to do that, one user must has to log in to the software by inputting a valid and proper e-mail id and password.

PROJECT DESCRIPTION

As we have already implied that our software is made to register student information and course selection, we can expect that, any types of educational institution can use it to do such works in an easy way. It is such a software that a student can be one of the admins by himself entering his mail and password. So it is of course beneficial for a student to change any of his data whenever he wants; he even won't have to ask the main official admin to do so as he can make himself an admin of this software. So we think it can be different from the others.

The major features of our software are given below:

- Any student's information will be saved as database and anytime the admins and even users can also see the data what will be input there.
- Student can Search his/her information by using his/her ID.
- If necessary information can be updated.

- In our software there is an option which is "**show students**" where all the information of students can be seen in a table.

- If a new student wants to select a course after knowing the details of it, there is an option named **"Course"** for this purpose.

Use cases:

- The user enters the mail and password and chooses if the user is student or administrator. If entered details are valid, the user's account becomes available. If it invalid and appropriate message is displayed to the user.

- The User added his/her information, now he/she needs to search it, if he/she put the correct id in search box all the information will be shown in instantly.

- Somehow the user needs to update the information, like recently he has changed his phone number and he wants to update his phone number's information to the registration system in this case he/she can easily update his/her information.

- Now the user wants to know about a specific course that he/she is interested in. So, he/she searches the 'Course' option and can find the courses details and select whatever he likes.

Primary and Secondary Actors

Φ Primary Actors

- Actors who initiate a scenario (use case) causing the system to achieve a goal
- Automated registration system example the “student or admin” is a primary actor.

Φ Secondary Actors

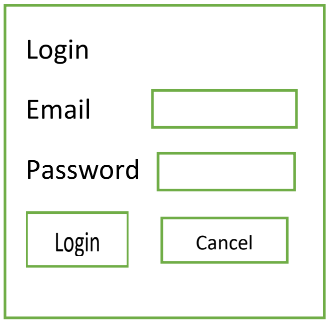
- Actors supporting the system so primary users goals can be completed.
- Automated registration system example the registration “data base” is a secondary actor.

Main Success Scenario

Step Actor Action Description

1 System The login box pops up. The following fields are available

-



- Email
- Password
- Login
- Cancel

The login box provides the Register Option.

2 User User enters Email, Password and press login Option.

-

3 System Login page disappears and Home page pops up.

Scenario Extensions

Step	Actor	Action Description
2a	User types wrong - Email or Password	the popup window says “email or password is wrong “.

Here is our whole project plan epitomized below in two parts-

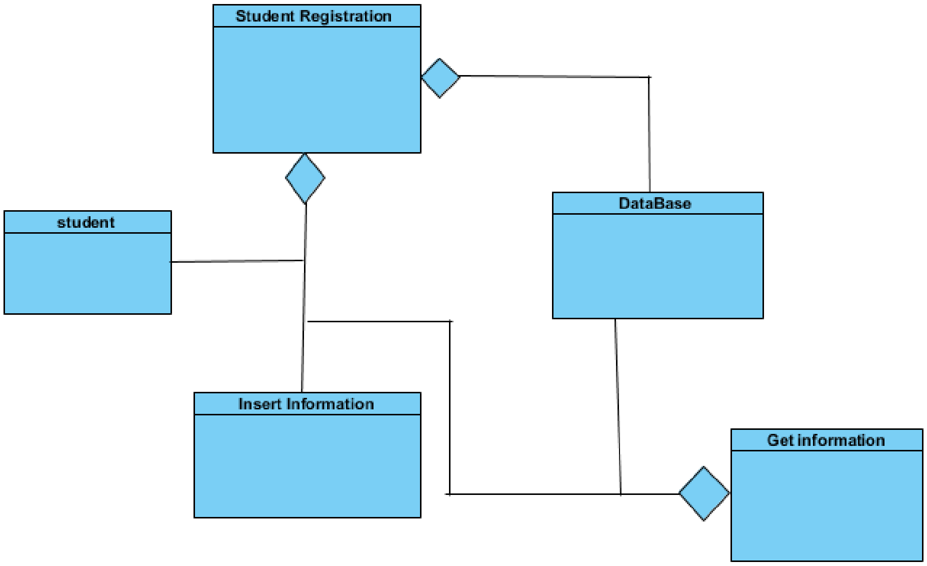
1. According to class diagram
2. According to sequence diagram

According to class diagram:

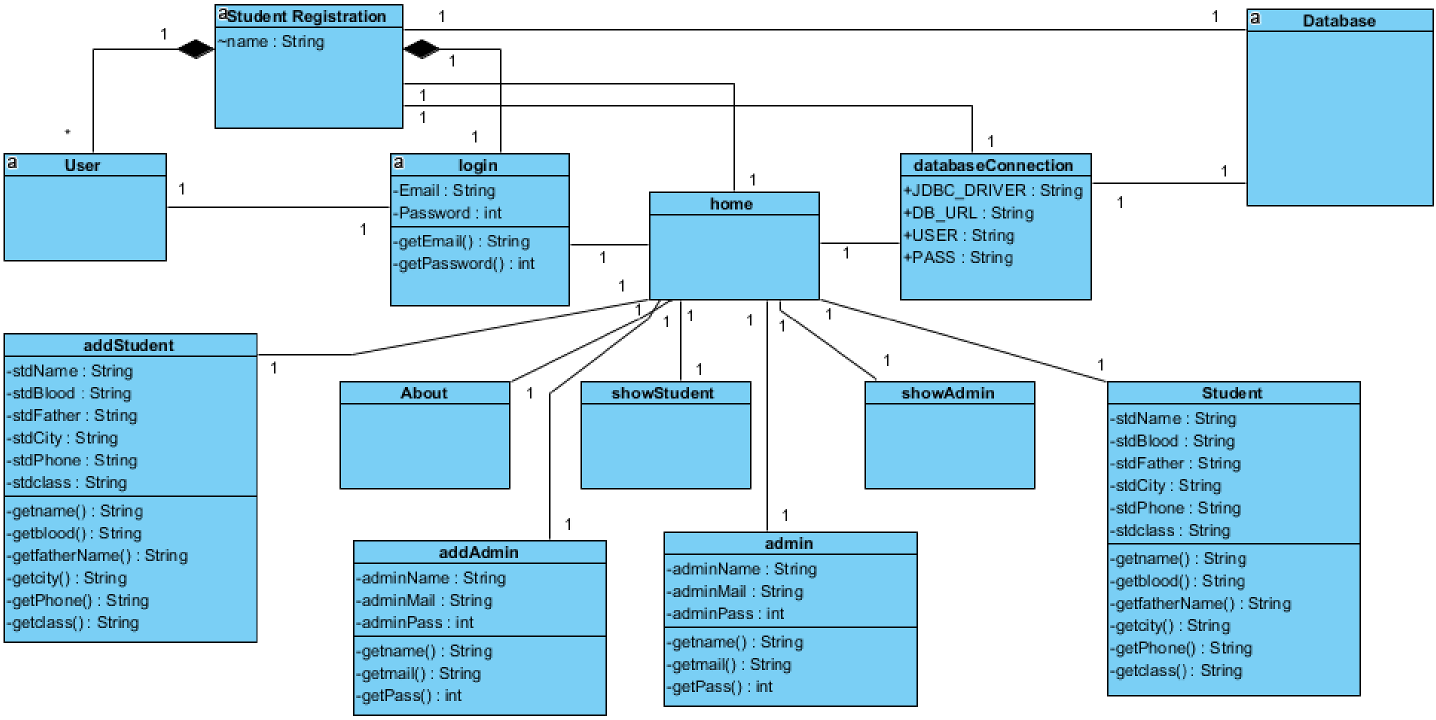
- The opening point of this software ‘Student Registration’, is a String data type.
- At the beginning, a login page will appear containing ‘mail id’ and ‘password’. And for this, we built a class named ‘login’. ‘String’ datatype for ‘email’ and ‘int’ datatype for ‘password’ were used to implement the login system.
- After logging in, a home page will be popped up that is performed in class module named as ‘home’.
- When a user will enter the home page he/she will see nine options that can be checked out. The classes ‘student’, ‘addStudent’, ‘showStudent’, ‘admin’, ’addAdmin’,

'showAdmin', 'CourseDetails', 'ManageCourse', 'ShowCourse' has been created for their functionality.

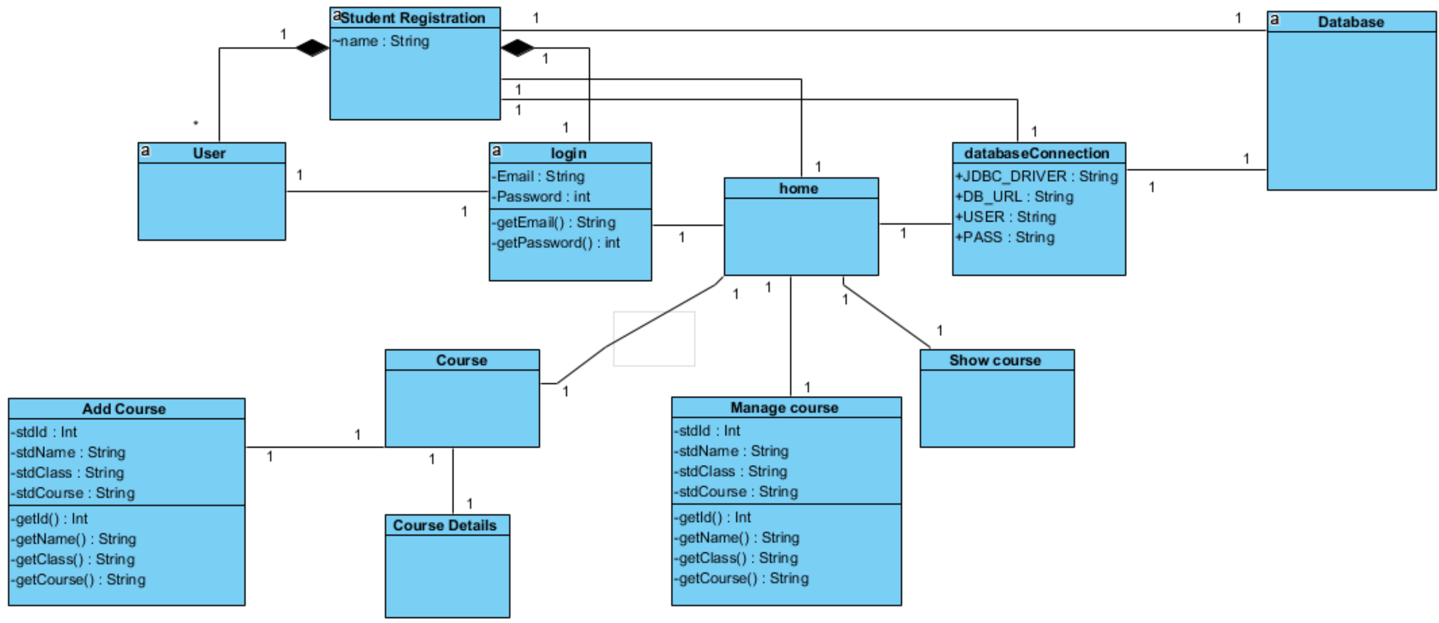
- Among the nine modules above, for 'CourseDetails', it has another package called 'Add Course'. Some other sub modules are also there to describe the courses.
- Into the 'Student' and 'addStudent' classes, six variables 'stdName', 'stdBlood', 'stdFather', 'stdCity', 'stdPhone' and 'stdclass' were implemented with 'string' data type.
- For the admin panel 'adminName', 'adminMail' and 'adminPass' are the variables of both 'admin' and 'addAdmin' classes. Here also the variables are all 'string' data type.
- The class 'databaseConnection' has member functions JDBC_DRIVER, DB_URL, USER and PASS and all four of these are 'string' type.



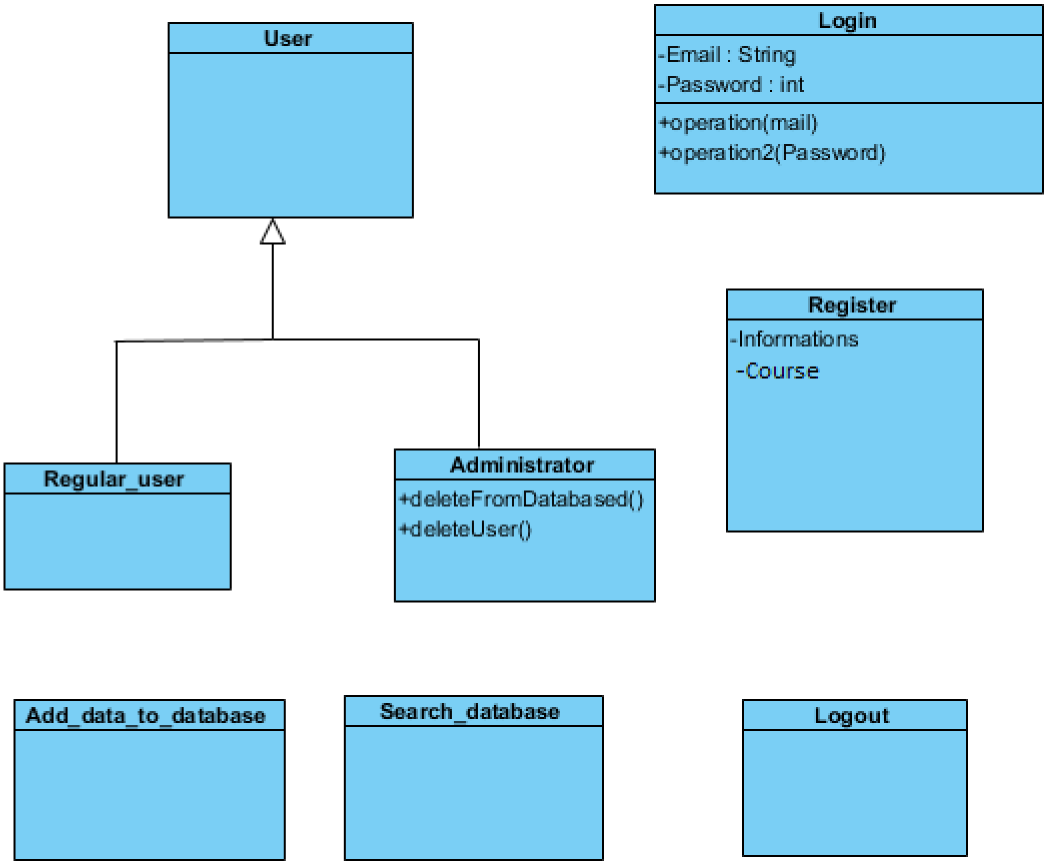
UML Class Diagram 1.1: Basic idea about our Project.



UML Class Diagram 1.2: About information Registration.



UML Class Diagram 1.3: About Course Registration.



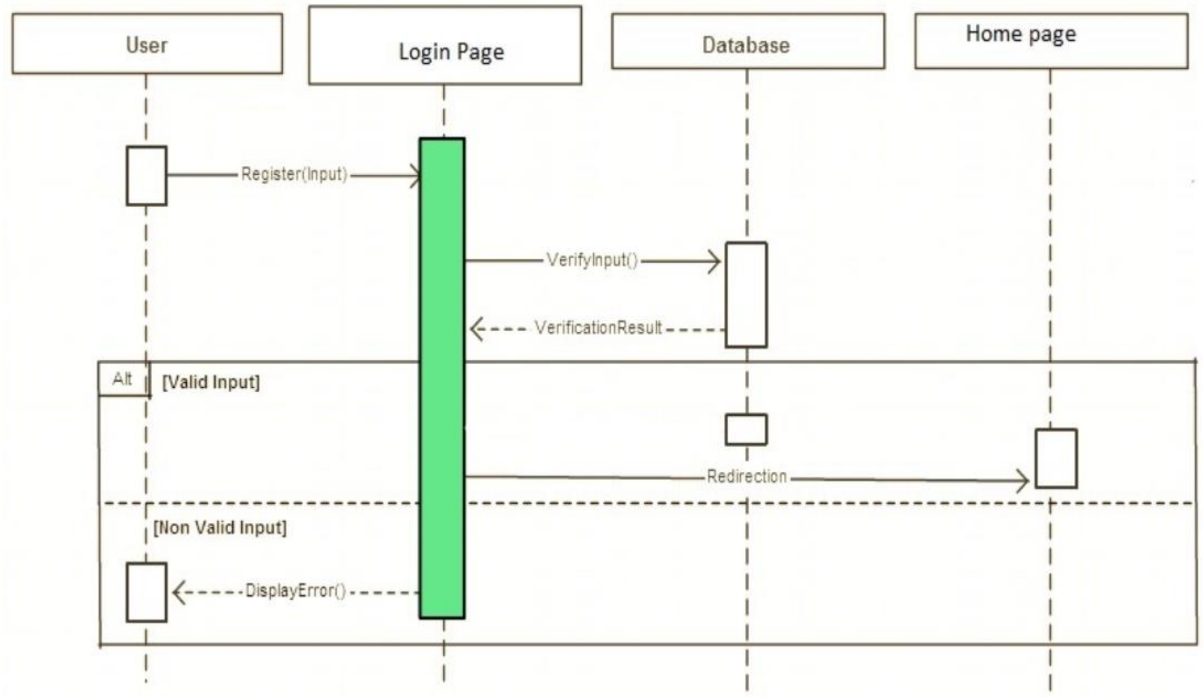
UML Class Diagram 1.4: About How it works.

According to sequence diagram

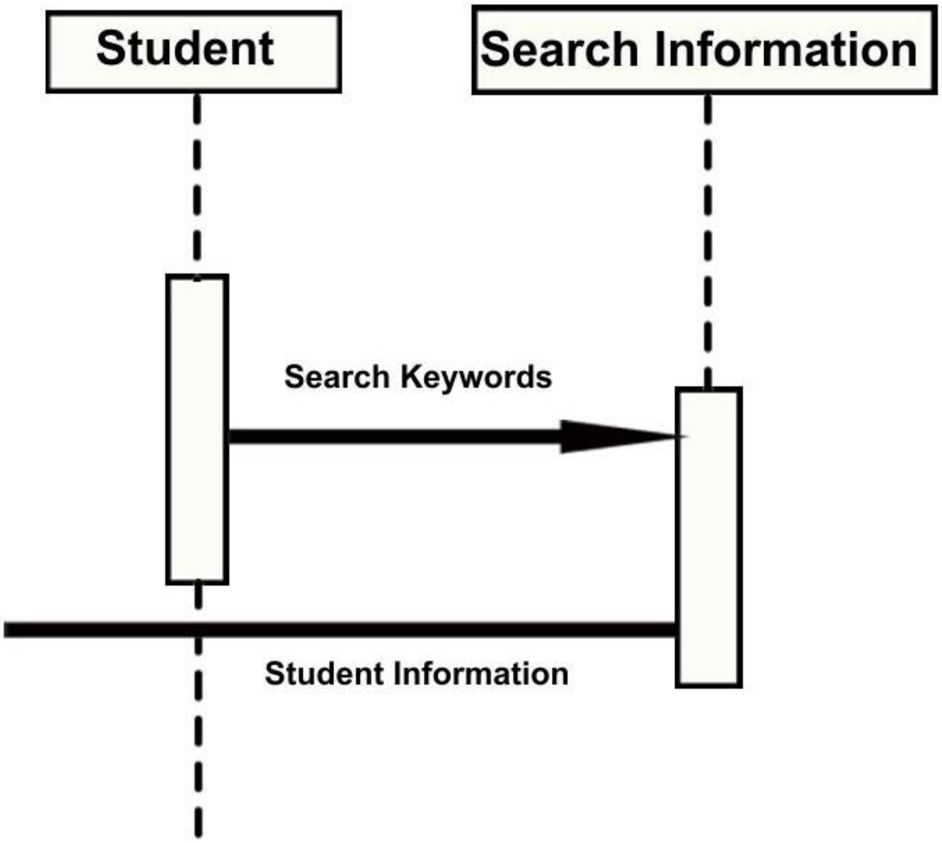
- At the commencement of starting work in our designed software, in the login page the user enters the mail and password and gets access to enter into next page.
- If entered details are valid, the user's account becomes available. If it's invalid then appropriate message for this displays to the user.
- After logging in, a home page appears that contains functions for students, admin and courses.
    - Add Students, Operation and Show Students (for Students).
    - Add Admin, Operation and Show Admin (for Admin).
    - Course, Manage Course and Show Course.

- Next the user chooses whether he/she will be student or administrator and click on 'Add Student' or 'Add Admin' button according to that. A table is there to insert his/her information as requested and submit it then.
- If all the information are entered, a confirmation message show up to ensure the submission.

- Now the user needs to search own information. He/she has to enter into ‘Operation’ and put the correct id in search box there, all the information will then show up instantly.
- If user wants to update any data that has already been inserted, he/she just have to insert the new data in the correct box and click on update button. Here’s also the back option to go back further.
- All the information of the user that have registered will come up altogether in a sequenced table if user wants to visit ‘show students’ or ‘Show Admin’ options.
- Users also have that chance to select a course about which he/she wants to know the details.
- Course can be changed by entering into the ‘Manage Course’ option.
- Short keys are there on the top which will help users to go to home page or to logout or to know about our app in a shortcut way with the switches-
    - ctrl+A ➔ Home
    - ctrl+L ➔ Logout
    - ctrl+S ➔ About
- Last of all, the class called ‘databaseConnection’ has been created to connect our project with database with the help of the functions of this class.

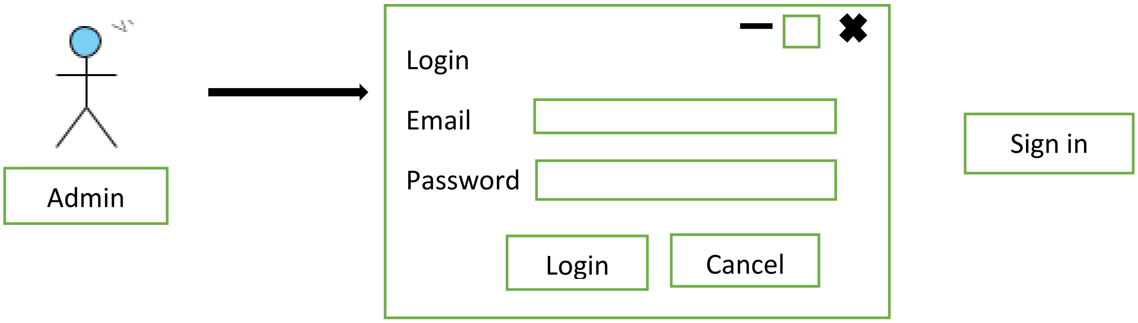


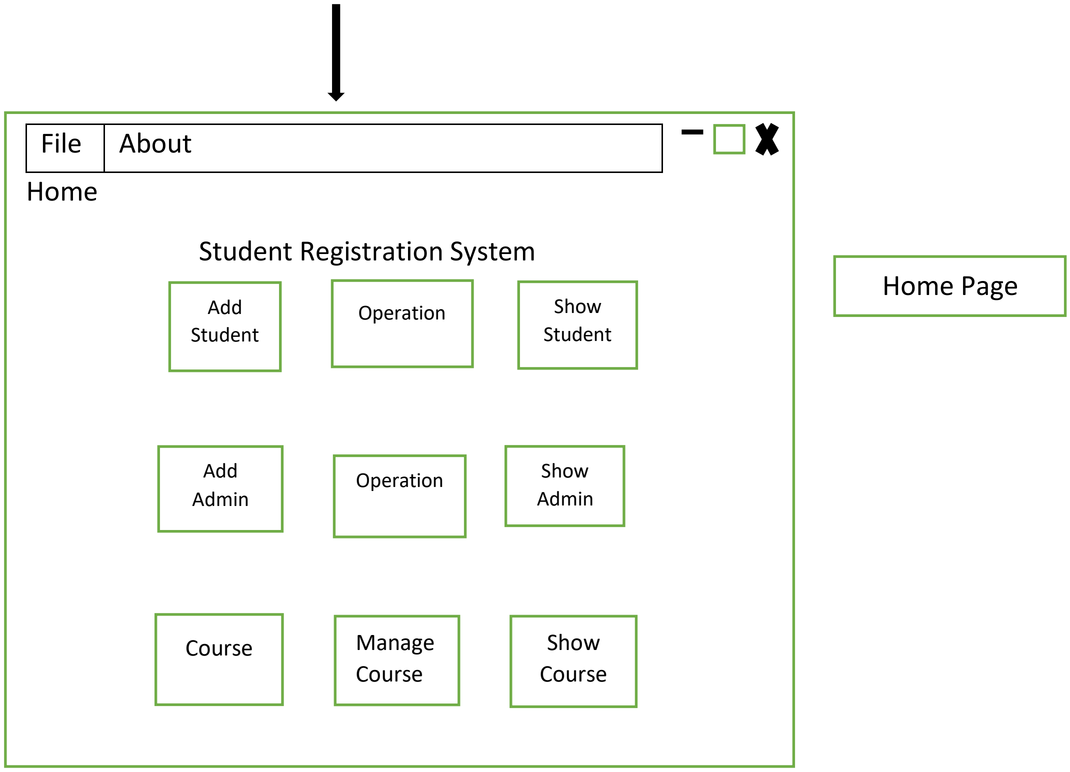
UML sequence diagram 2.1: First user case



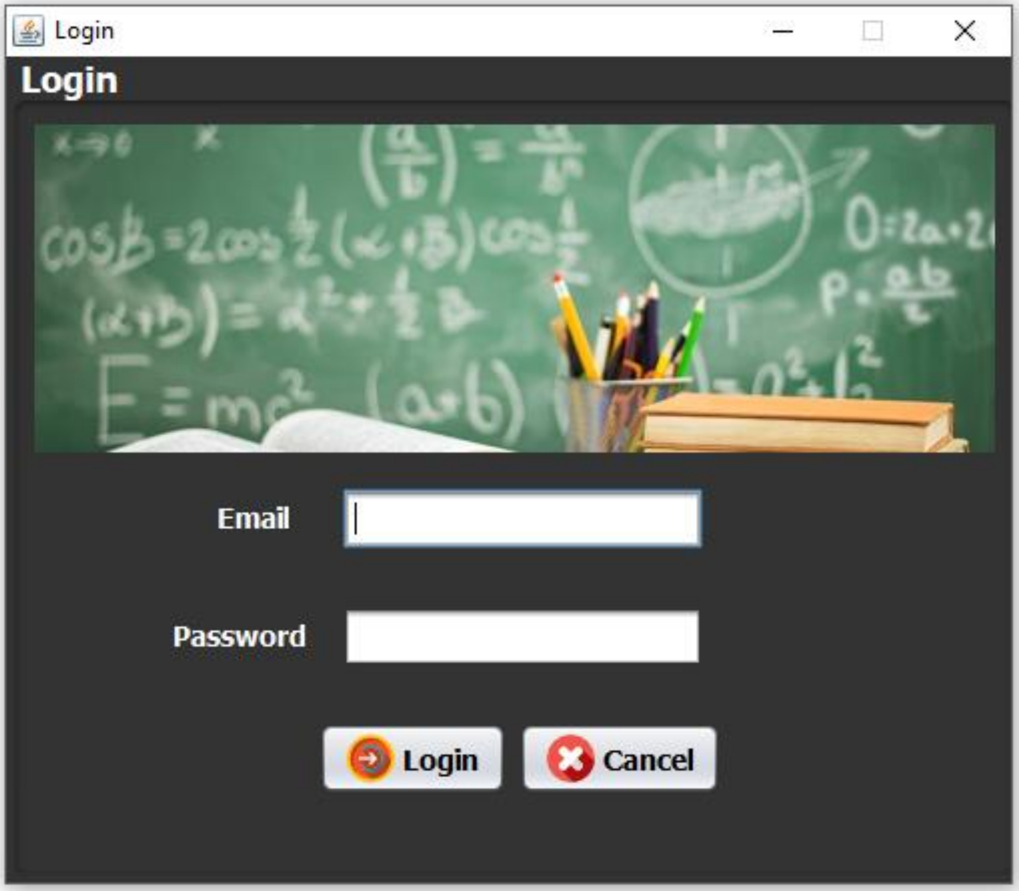
UML sequence diagram 2.2: Second user case

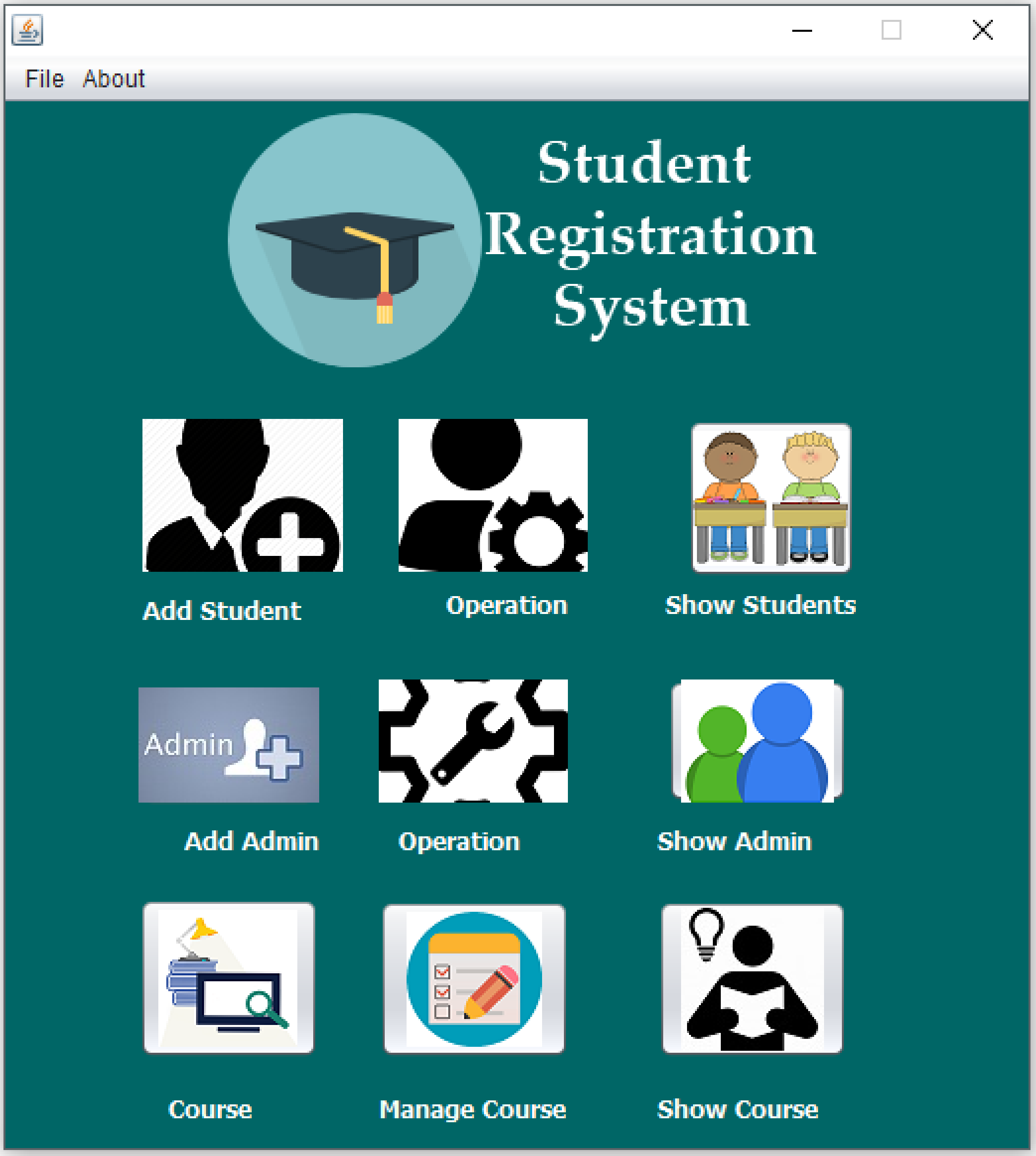
SOFTWARE’S DEMO

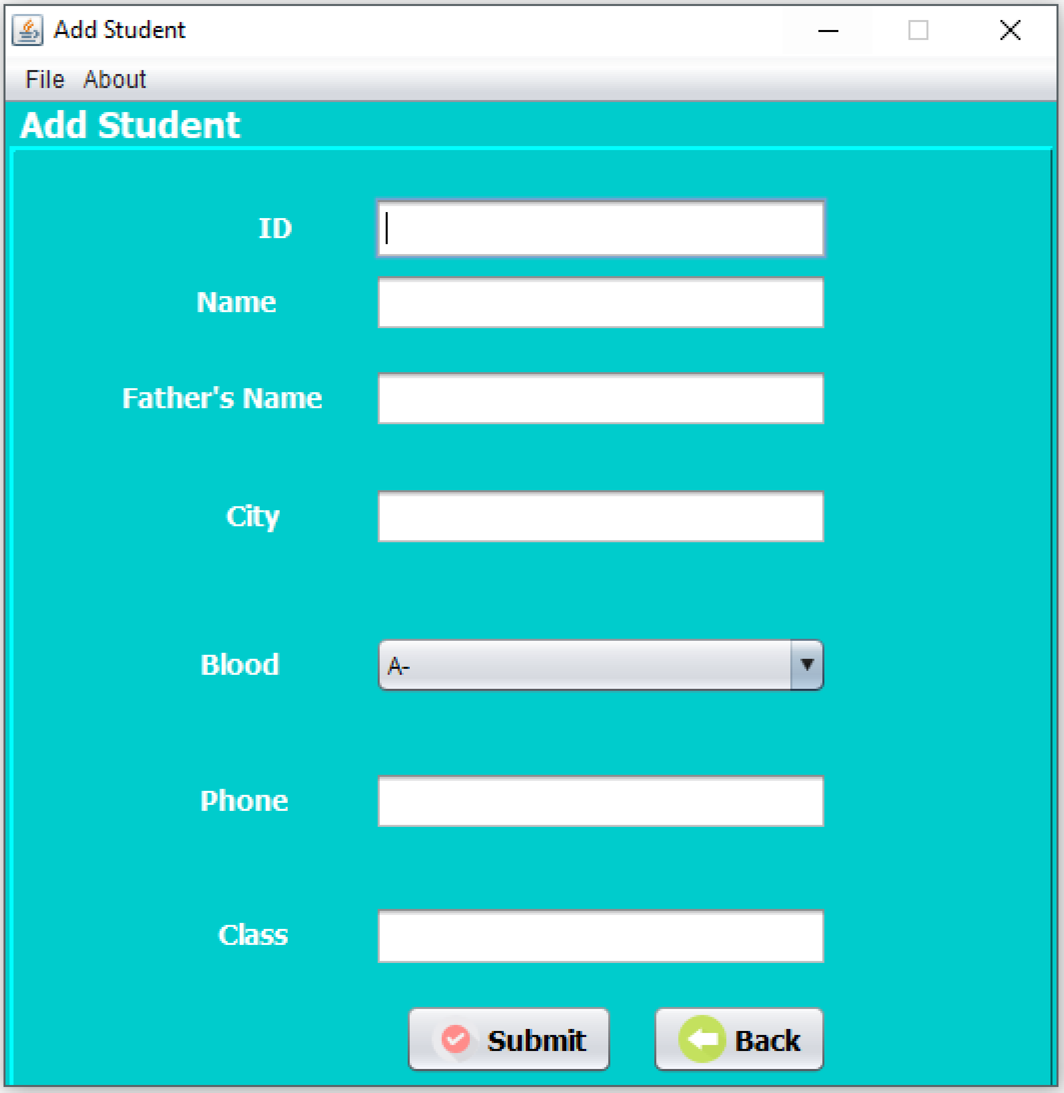




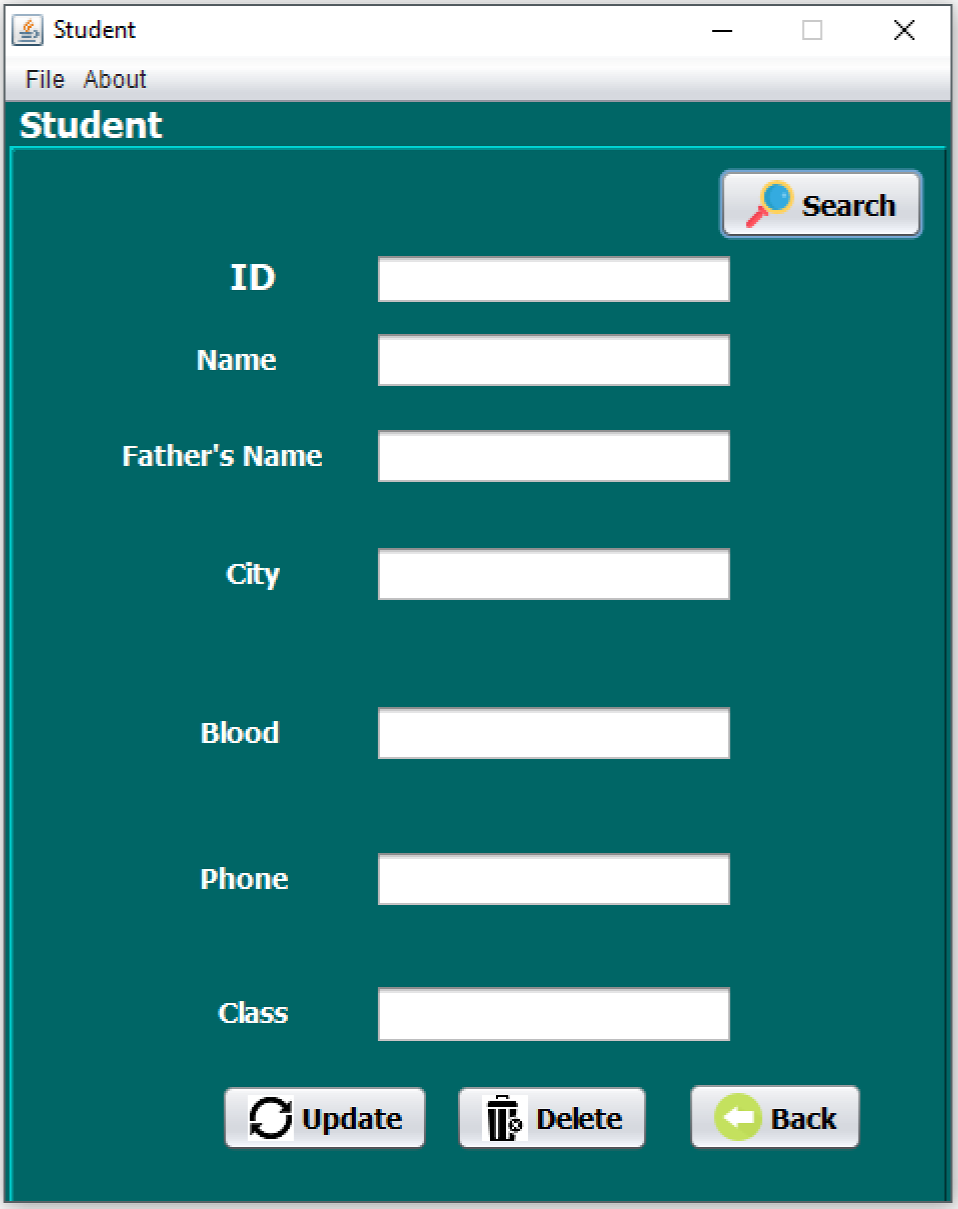
The features of our software is given below with the screenshots:

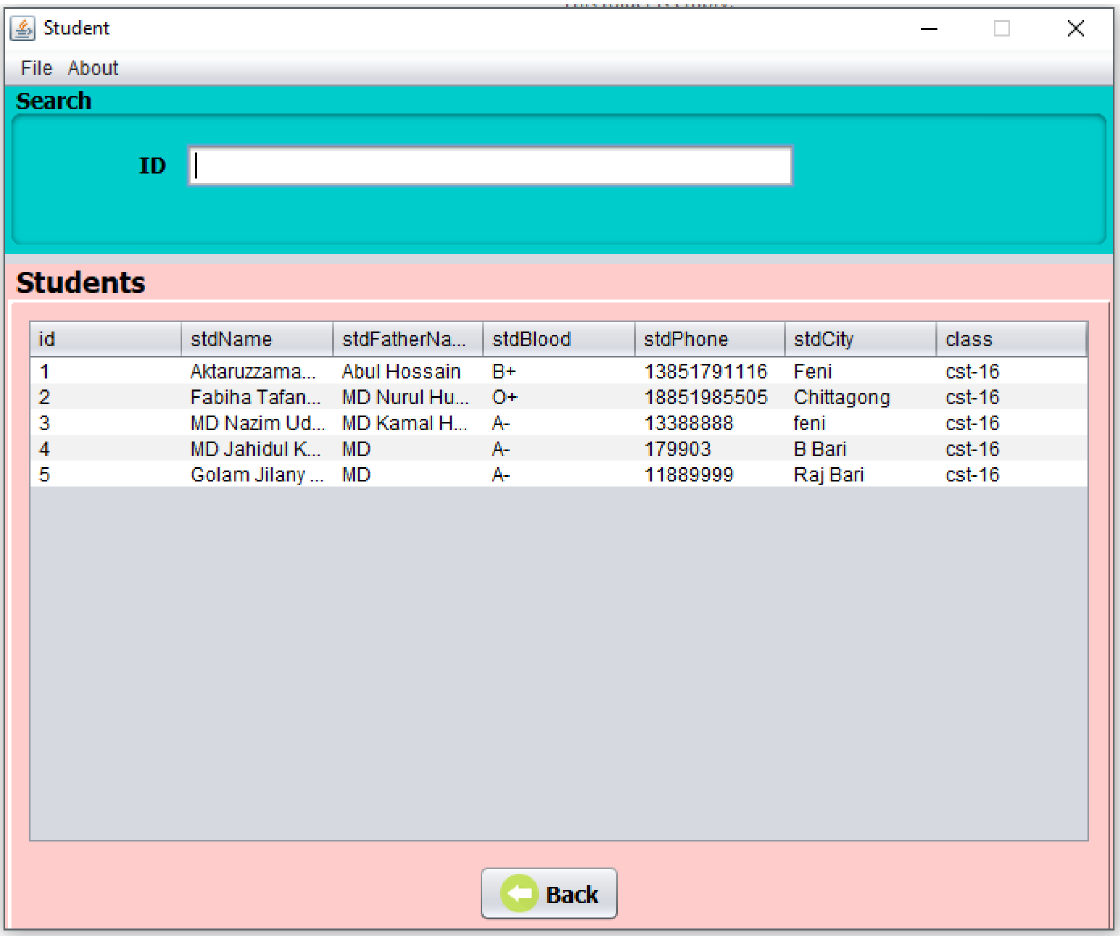

Login
Login
Email
Password
Login
Cancel


File
About
Student Registration System
Add Student
Operation
Show Students
Admin
Add Admin
Operation
Show Admin
Course
Manage Course
Show Course

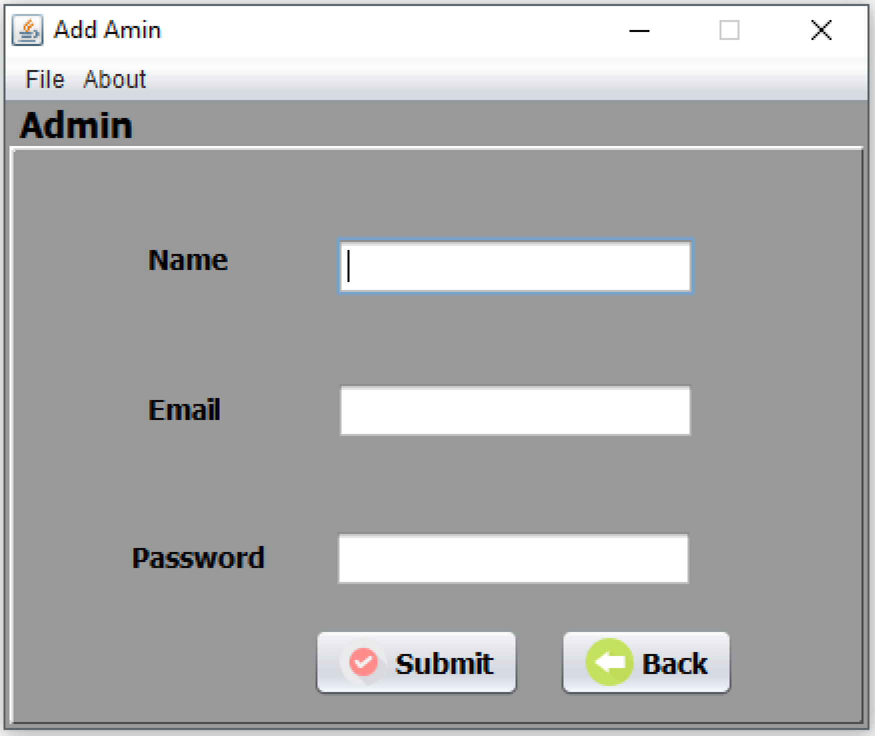

Add Student
File
About
Add Student
ID
Name
Father's Name
City
Phone
Class
Submit
Back

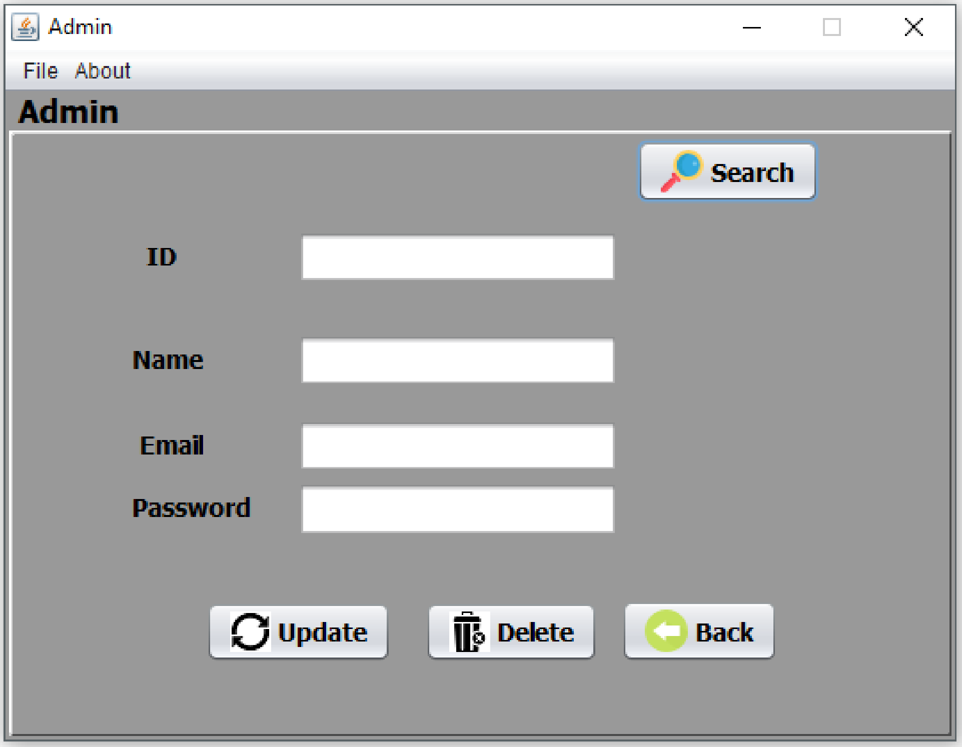
Blood A-

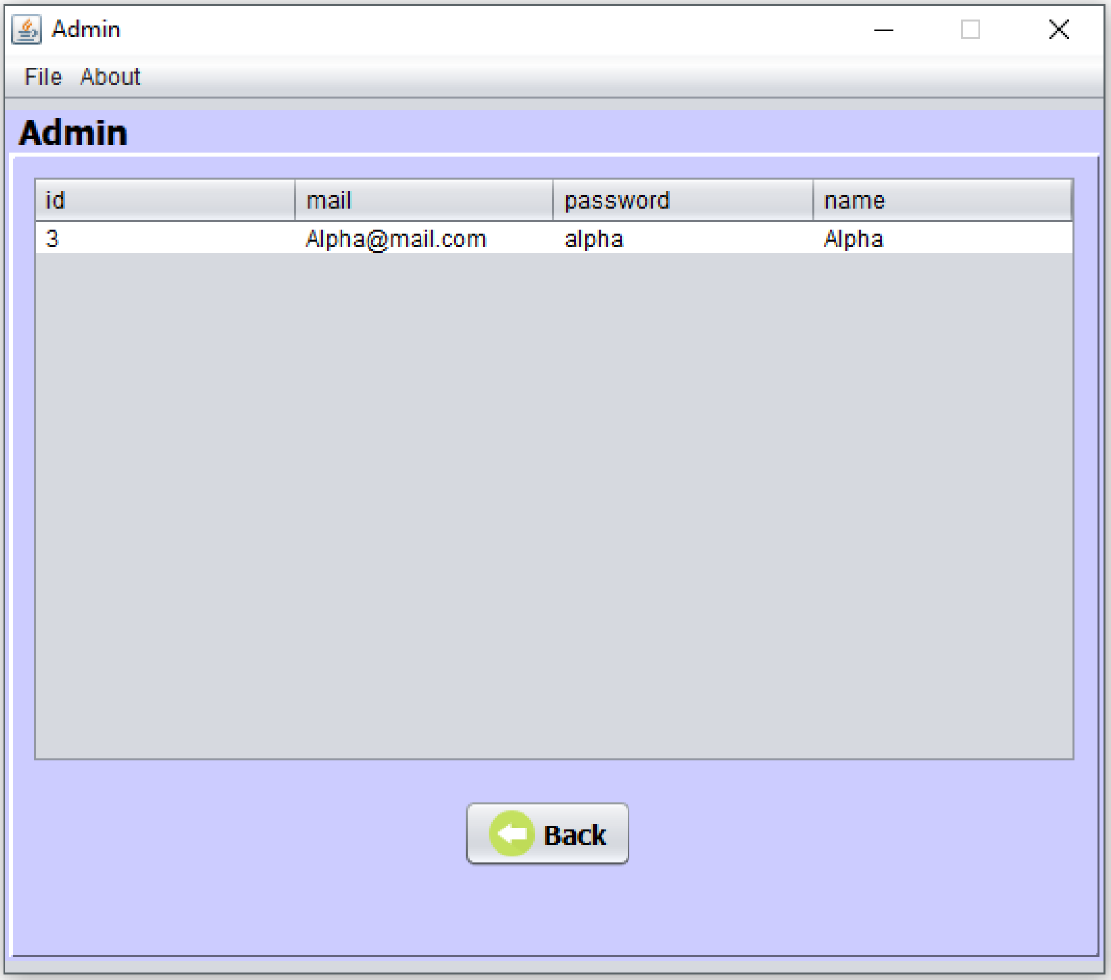

Student
File
About
Student
Search
ID
Name
Father's Name
City
Blood
Phone
Class
Update
Delete
Back



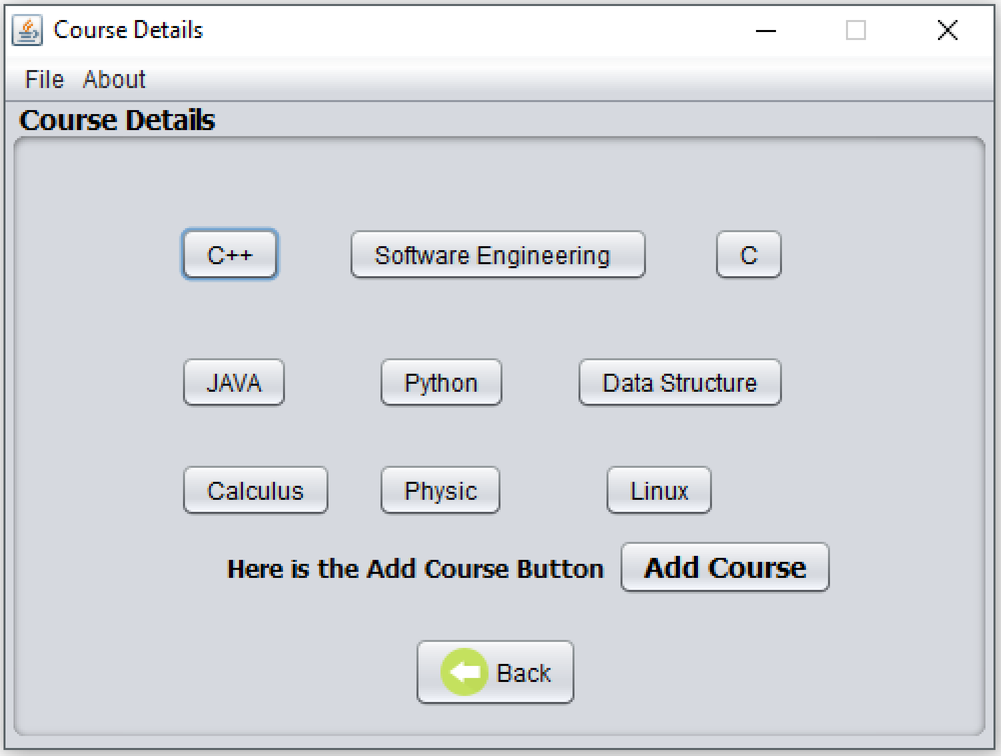
id	stdName	stdFatherNa...	stdBlood	stdPhone	stdCity	class
1	Aktaruzzama...	Abul Hossain	B+	13851791116	Feni	cst-16
2	Fabiha Tafan...	MD Nurul Hu...	O+	18851985505	Chittagong	cst-16
3	MD Nazim Ud...	MD Kamal H...	A-	13388888	feni	cst-16
4	MD Jahidul K...	MD	A-	179903	B Bari	cst-16
5	Golam Jilany ...	MD	A-	11889999	Raj Bari	cst-16

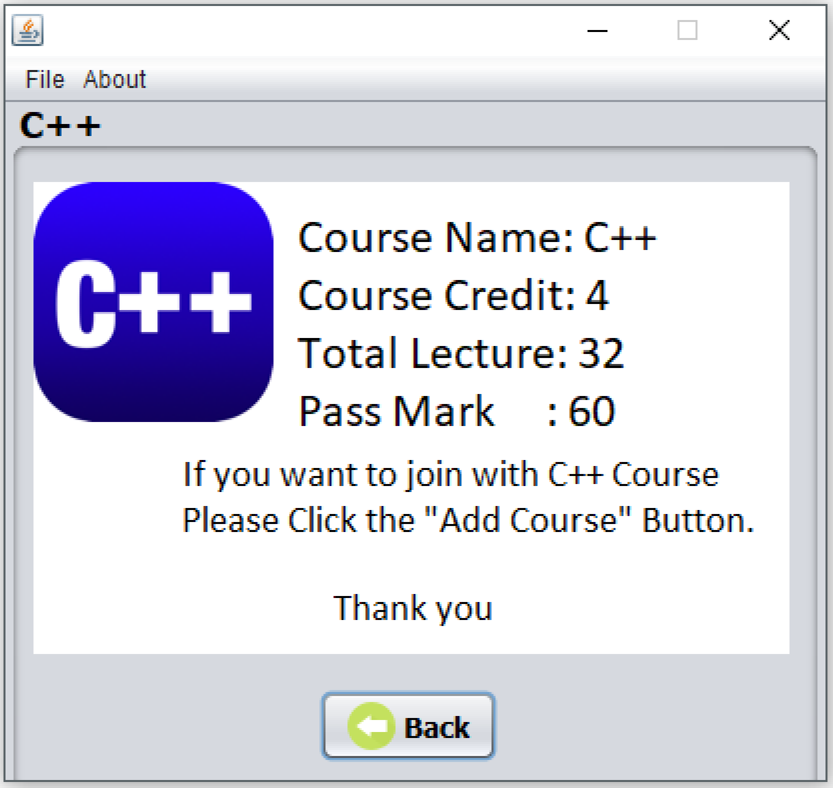

Add Amin
File
About
Admin
Name
Email
Password
Submit
Back


Admin
File
About
Admin
Search
ID
Name
Email
Password
Update
Delete
Back



id	mail	password	name
3	Alpha@mail.com	alpha	Alpha

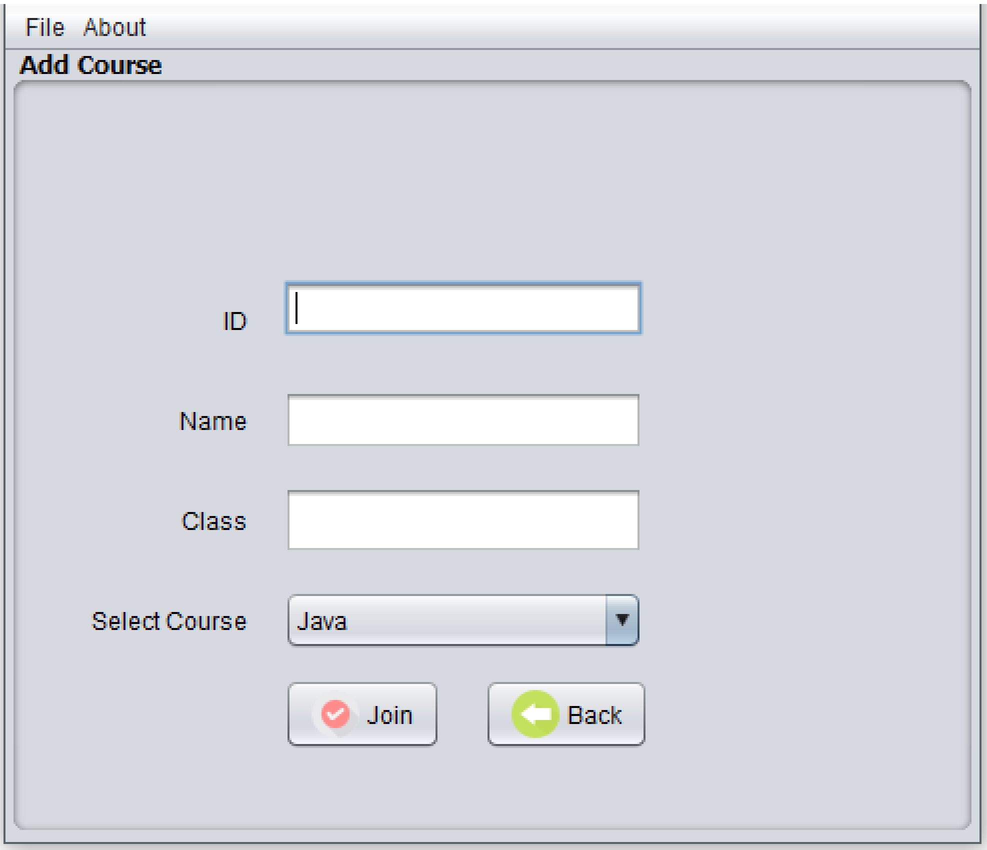

Course Details
File
About
Course Details
C++
Software Engineering
C
JAVA
Python
Data Structure
Calculus
Physic
Linux
Here is the Add Course Button
Add Course
Back



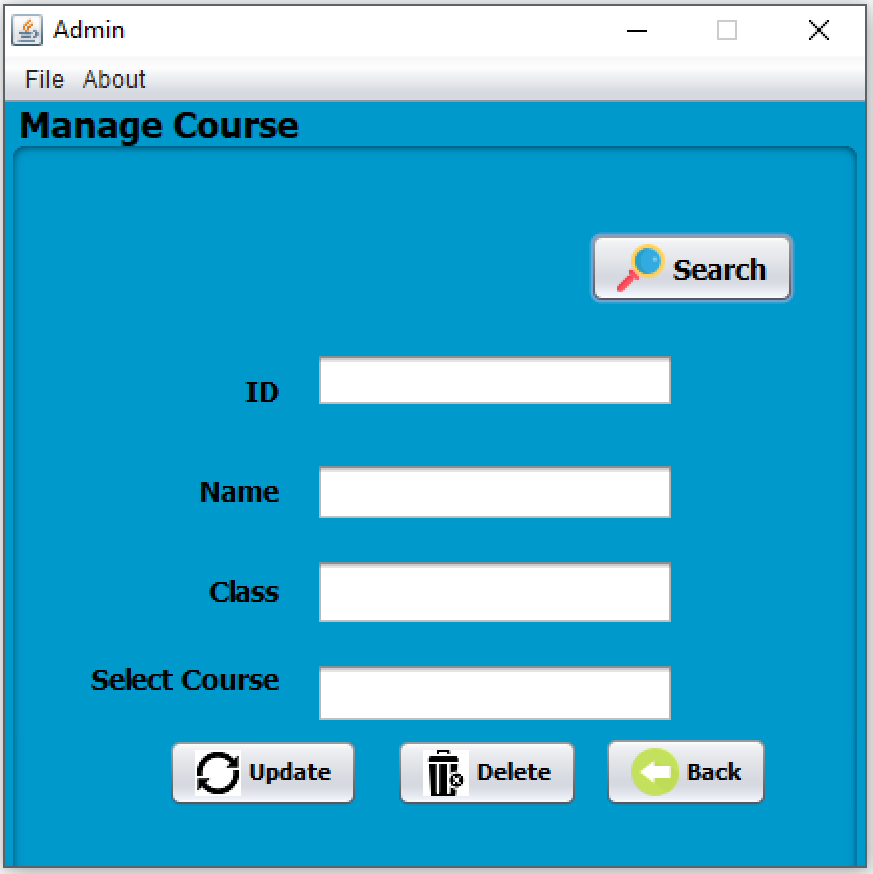
Course Name: C++
Course Credit: 4
Total Lecture: 32
Pass Mark : 60

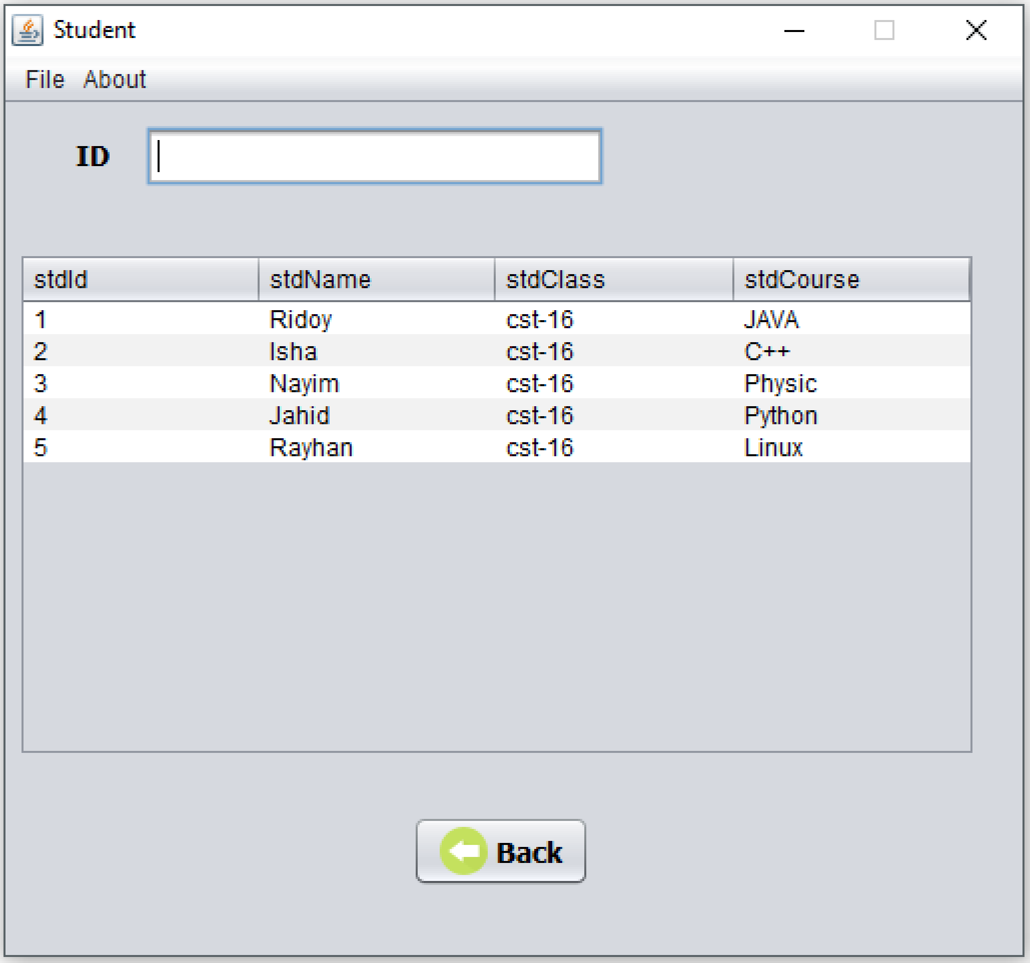
If you want to join with C++ Course
Please Click the "Add Course" Button.

Thank you

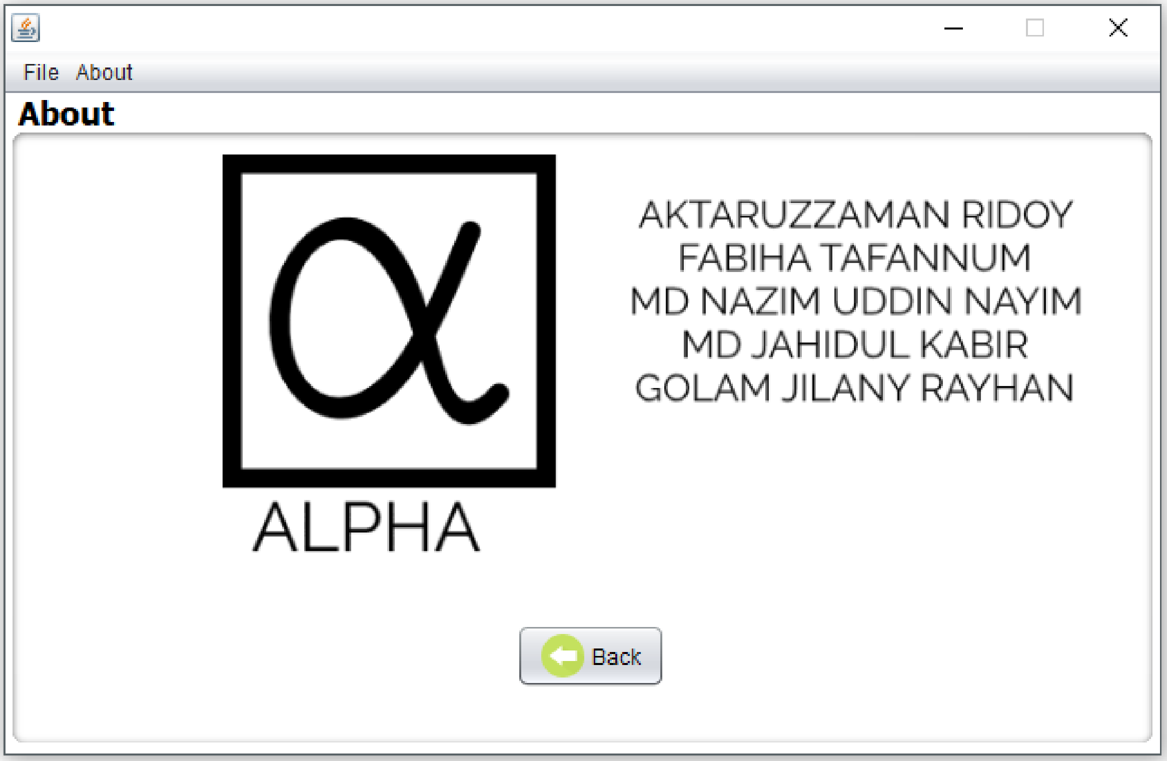


Select Course: Java


Admin
File
About
Manage Course
Search
ID
Name
Class
Select Course
Update
Delete
Back



stdId	stdName	stdClass	stdCourse
1	Ridoy	cst-16	JAVA
2	Isha	cst-16	C++
3	Nayim	cst-16	Physic
4	Jahid	cst-16	Python
5	Rayhan	cst-16	Linux

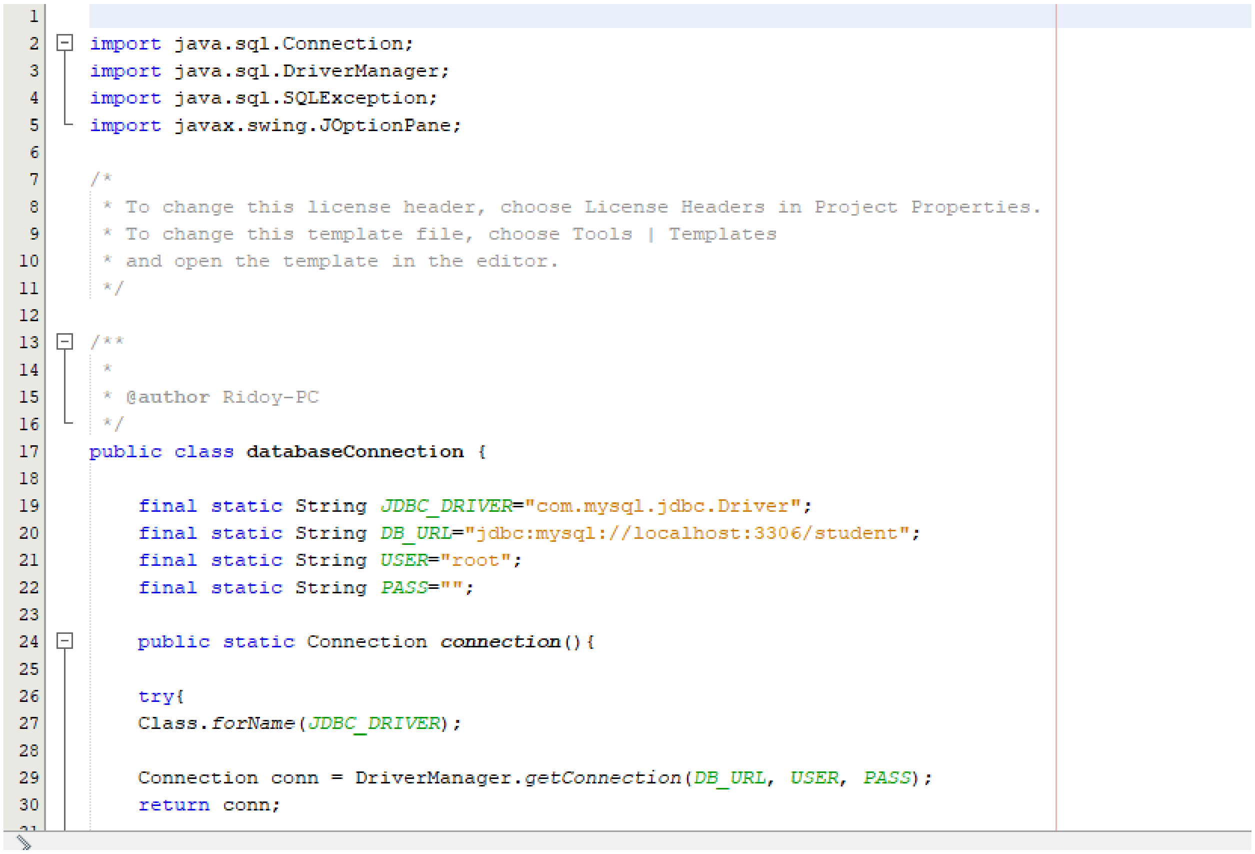


AKTARUZZAMAN RIDOY
FABIHA TAFANNUM
MD NAZIM UDDIN NAYIM
MD JAHIDUL KABIR
GOLAM JILANY RAYHAN

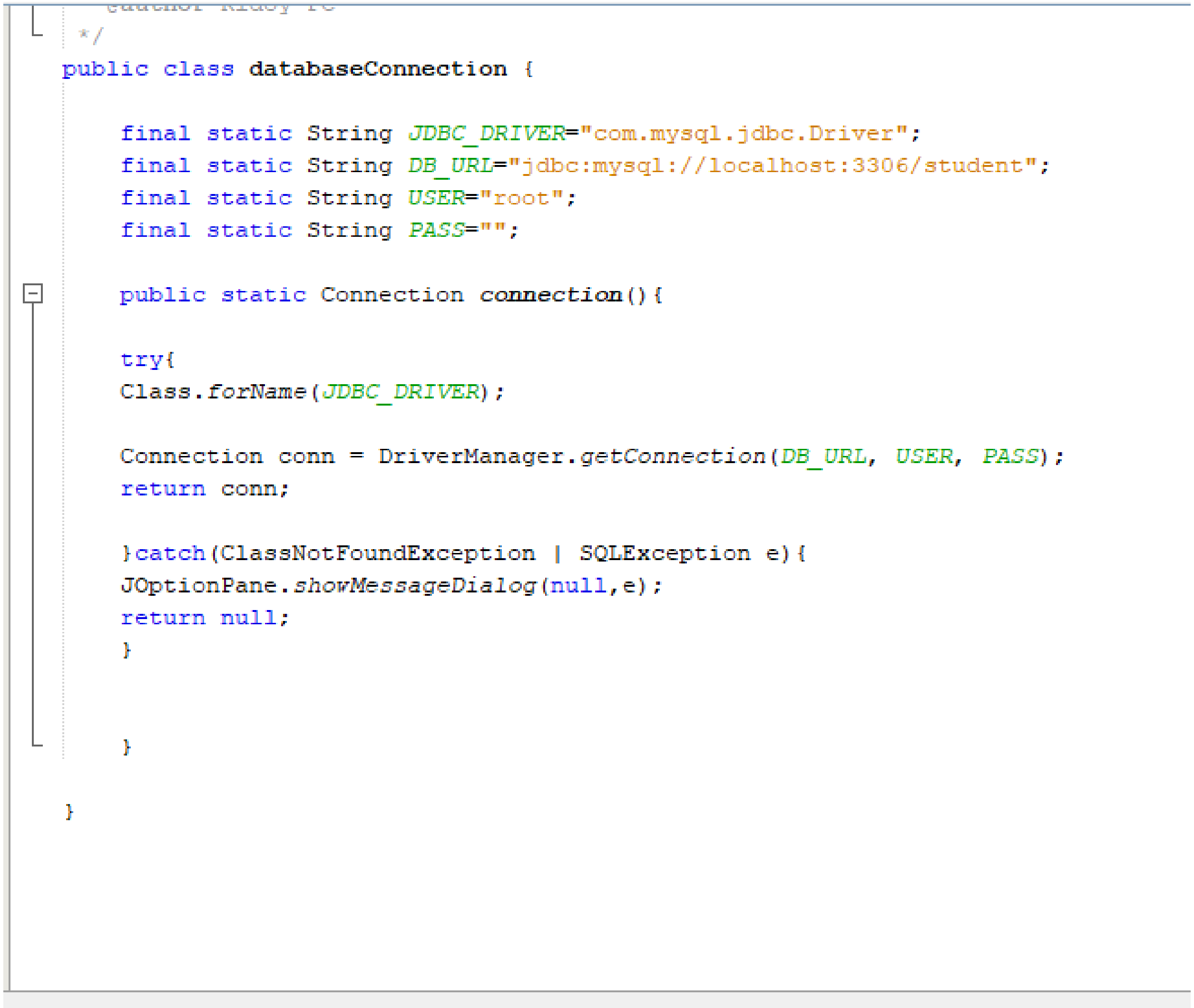
CODE

In this part we are going to show the basic source code of our project. The whole program will be uploaded in Github.

Database Connection:

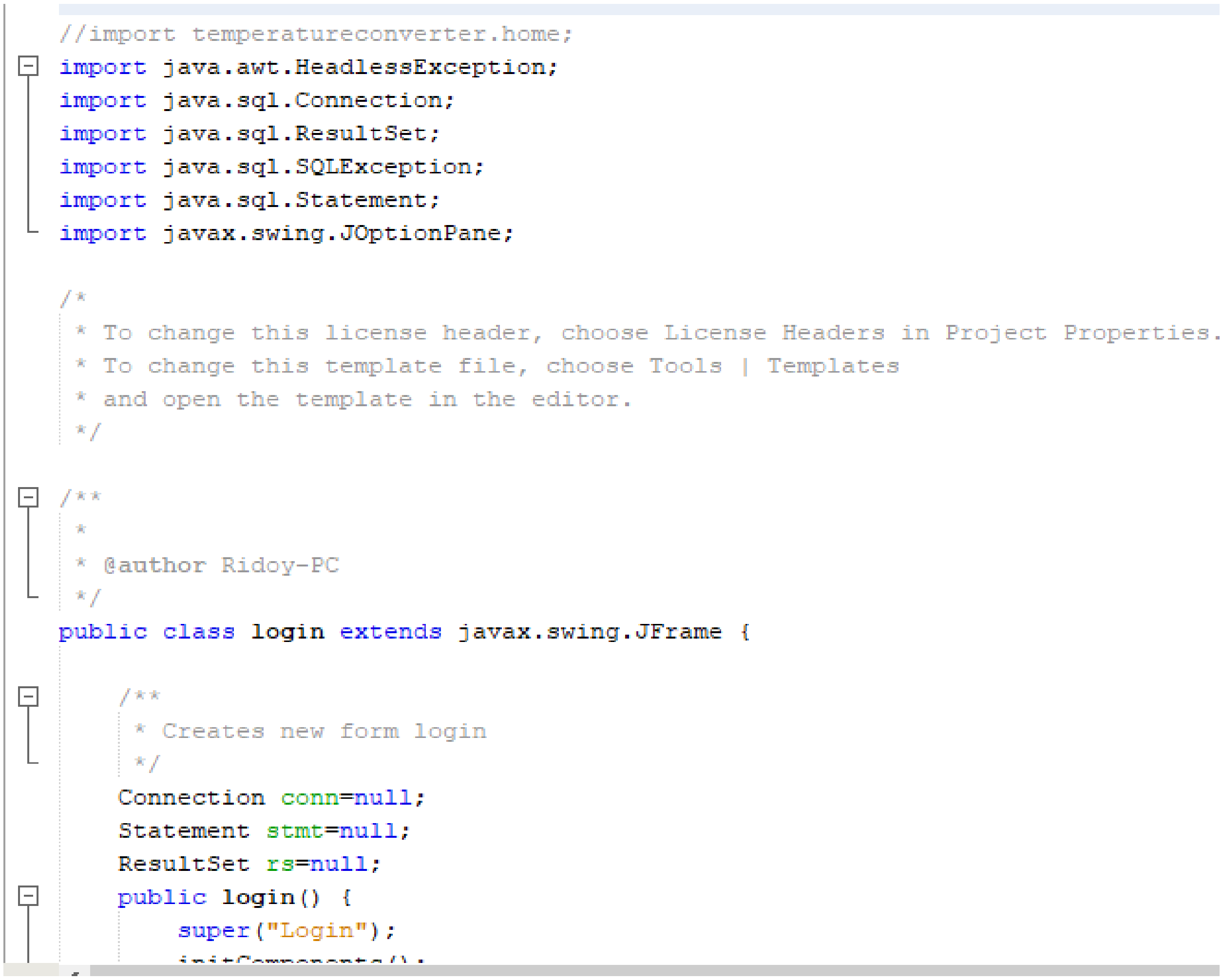


```java

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import javax.swing.JOptionPane;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author Ridoy-PC
 */
public class databaseConnection {

    final static String JDBC_DRIVER="com.mysql.jdbc.Driver";
    final static String DB_URL="jdbc:mysql://localhost:3306/student";
    final static String USER="root";
    final static String PASS="";

    public static Connection connection(){

    try{
    Class.forName(JDBC_DRIVER);

    Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
    return conn;
```

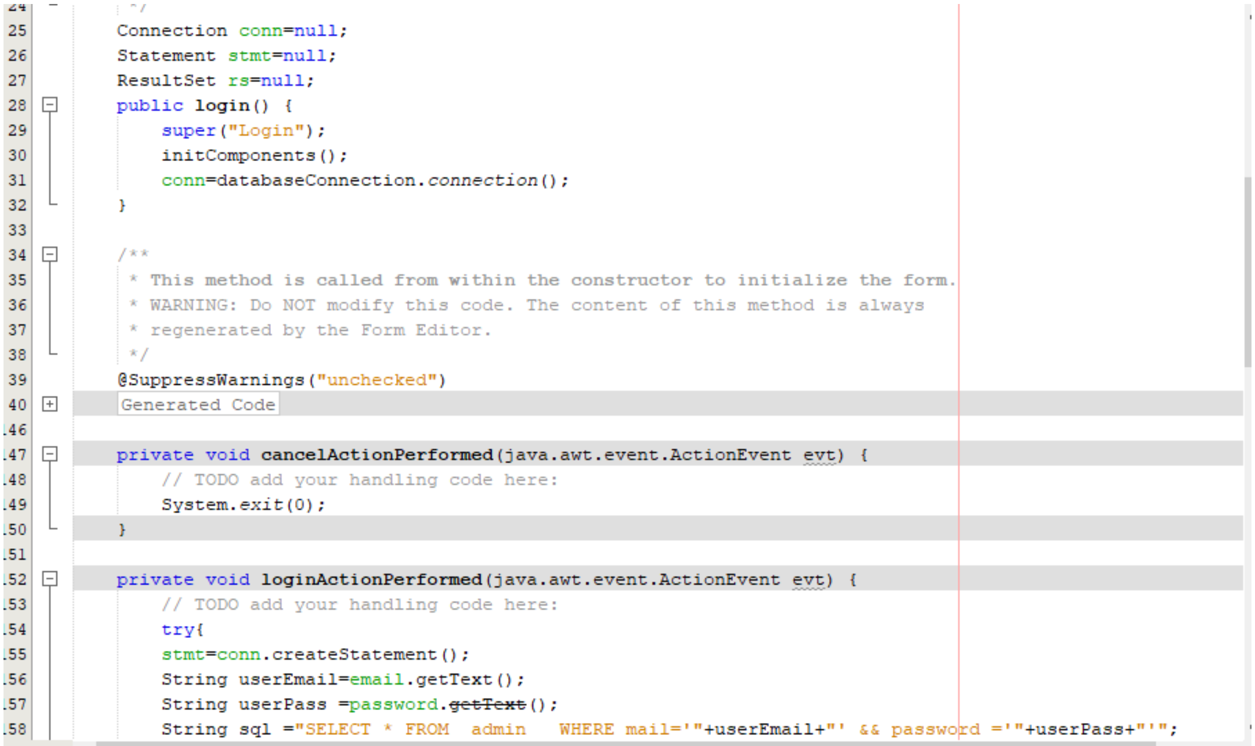


```
   @author Ridoy PC
 */
public class databaseConnection {

    final static String JDBC_DRIVER="com.mysql.jdbc.Driver";
    final static String DB_URL="jdbc:mysql://localhost:3306/student";
    final static String USER="root";
    final static String PASS="";

    public static Connection connection(){

    try{
    Class.forName(JDBC_DRIVER);

    Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
    return conn;

    }catch(ClassNotFoundException | SQLException e){
    JOptionPane.showMessageDialog(null,e);
    return null;
    }


    }

}
```

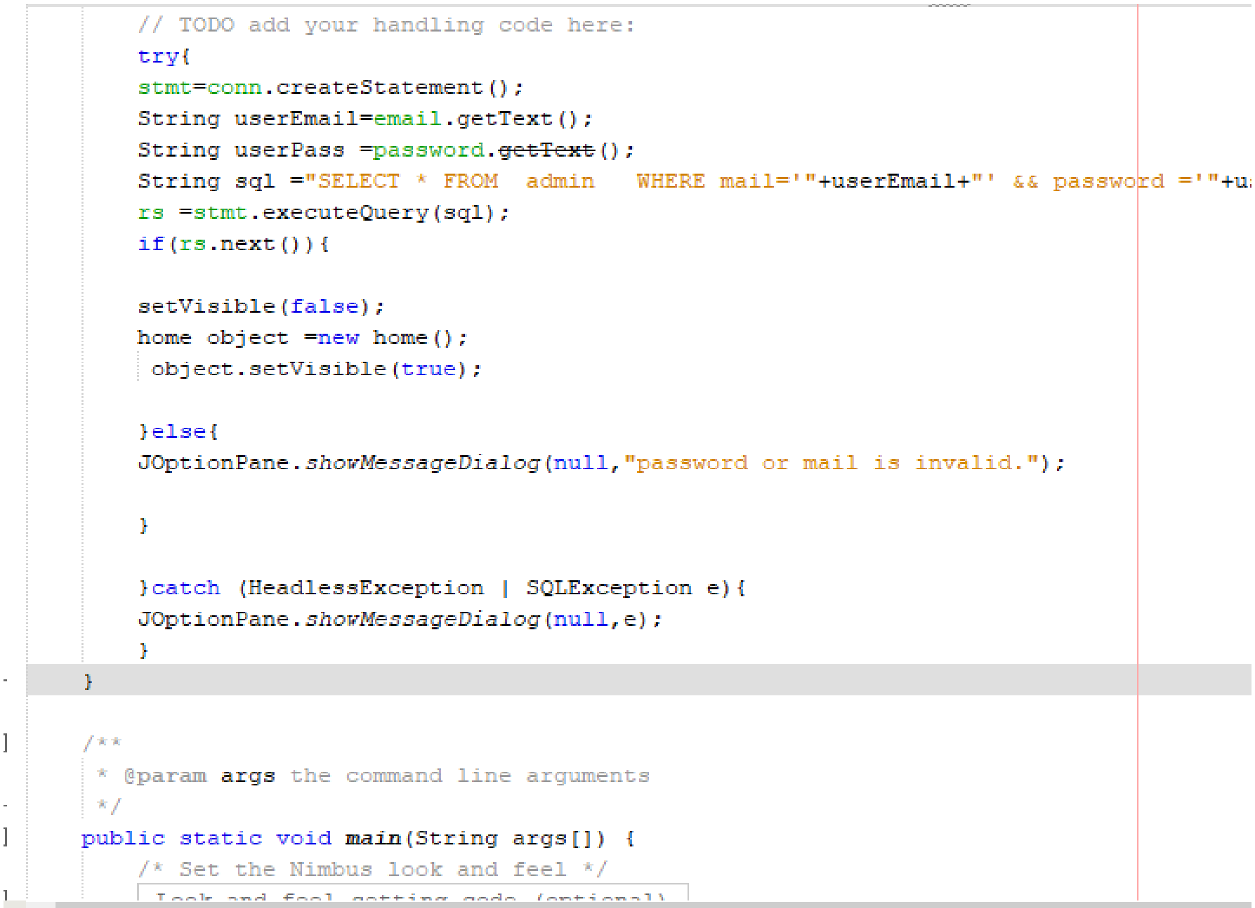
Login:



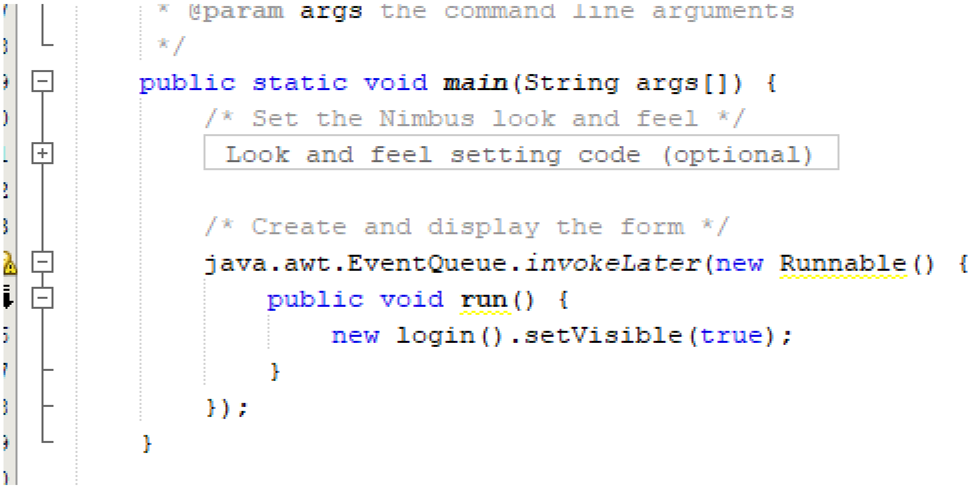
```
//import temperatureconverter.home;
import java.awt.HeadlessException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author Ridoy-PC
 */
public class login extends javax.swing.JFrame {

    /**
     * Creates new form login
     */
    Connection conn=null;
    Statement stmt=null;
    ResultSet rs=null;
    public login() {
        super("Login");
        initComponents();
```



```
     */
    Connection conn=null;
    Statement stmt=null;
    ResultSet rs=null;
    public login() {
        super("Login");
        initComponents();
        conn=databaseConnection.connection();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code

    private void cancelActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        System.exit(0);
    }

    private void loginActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try{
        stmt=conn.createStatement();
        String userEmail=email.getText();
        String userPass =password.~~getText~~();
        String sql ="SELECT * FROM  admin   WHERE mail='"+userEmail+"' && password ='"+userPass+"'";
```

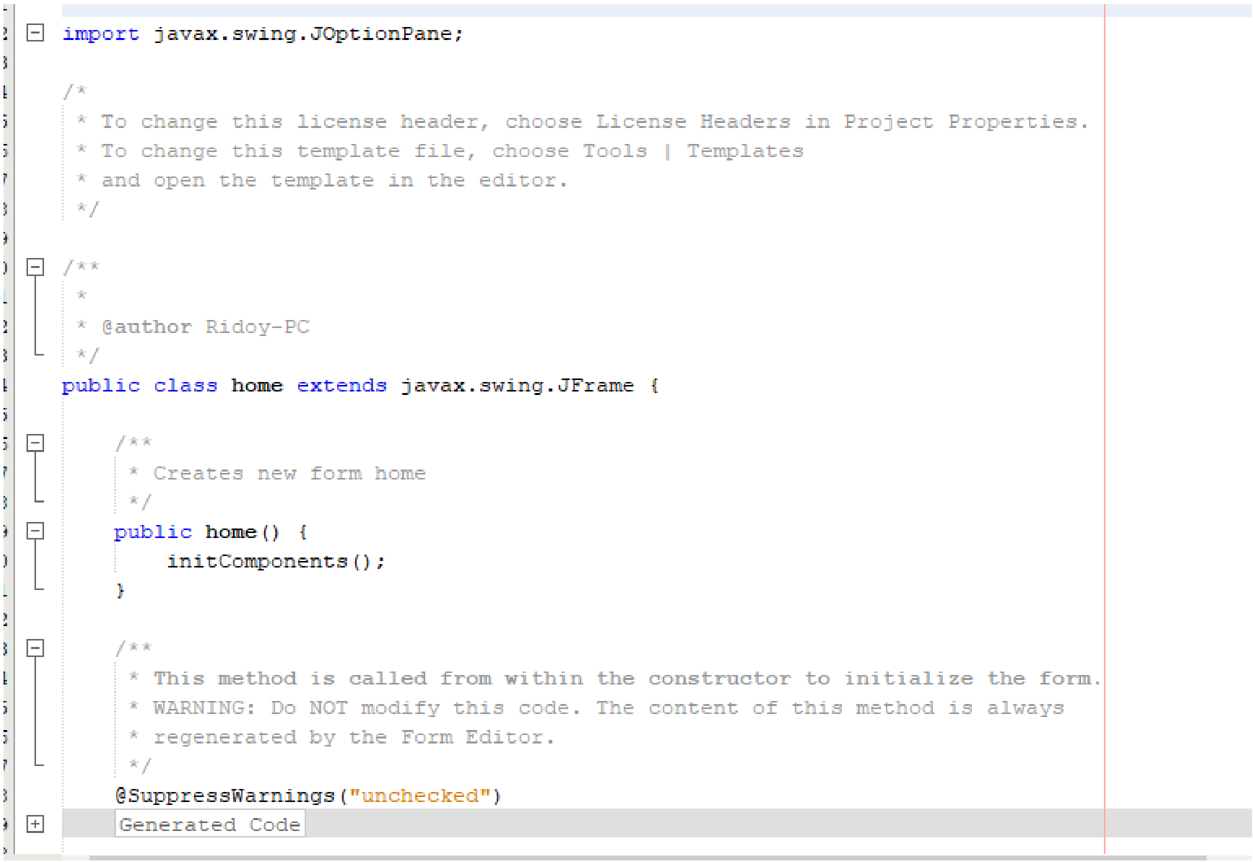


```
        // TODO add your handling code here:
        try{
        stmt=conn.createStatement();
        String userEmail=email.getText();
        String userPass =password.~~getText~~();
        String sql ="SELECT * FROM  admin   WHERE mail='"+userEmail+"' && password ='"+u
        rs =stmt.executeQuery(sql);
        if(rs.next()){

        setVisible(false);
        home object =new home();
         object.setVisible(true);

        }else{
        JOptionPane.showMessageDialog(null,"password or mail is invalid.");

        }

        }catch (HeadlessException | SQLException e){
        JOptionPane.showMessageDialog(null,e);
        }
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)
```

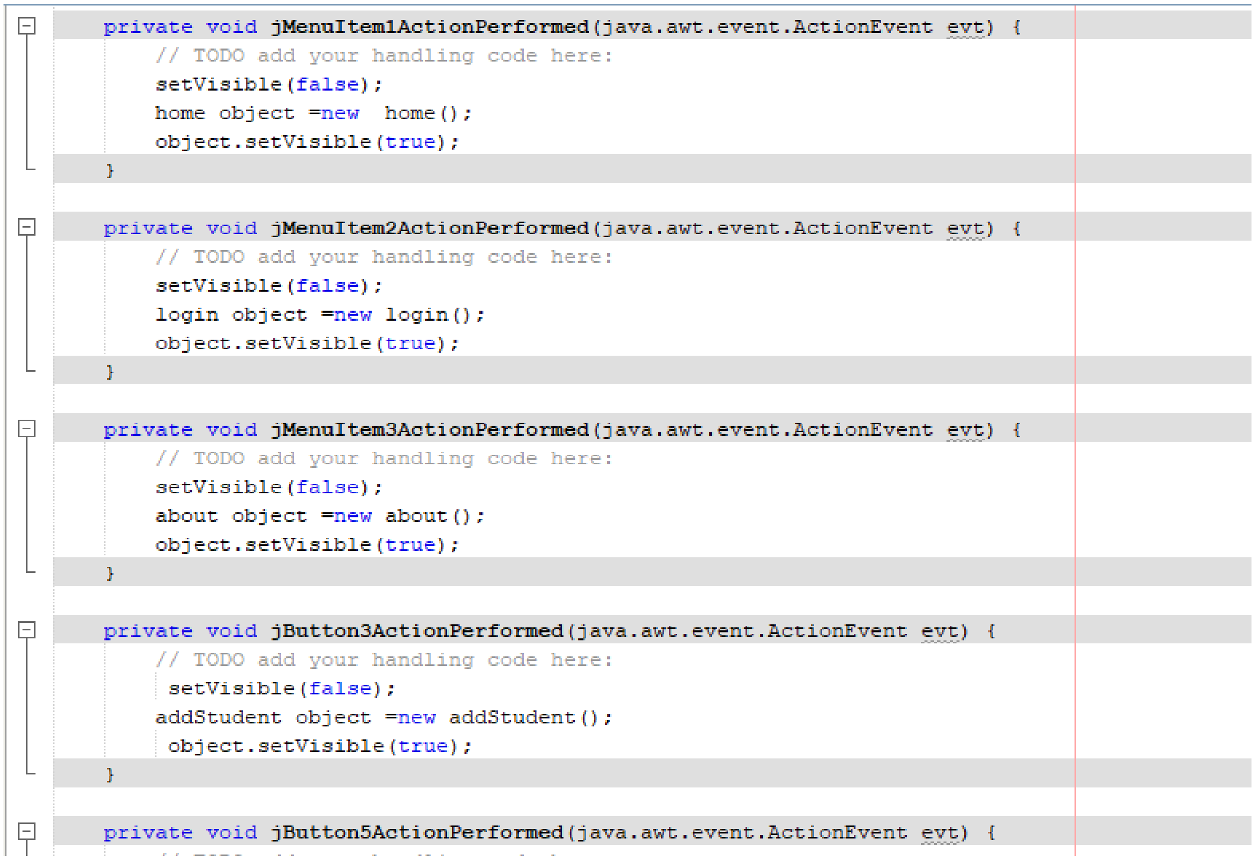


```
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new login().setVisible(true);
            }
        });
    }
```

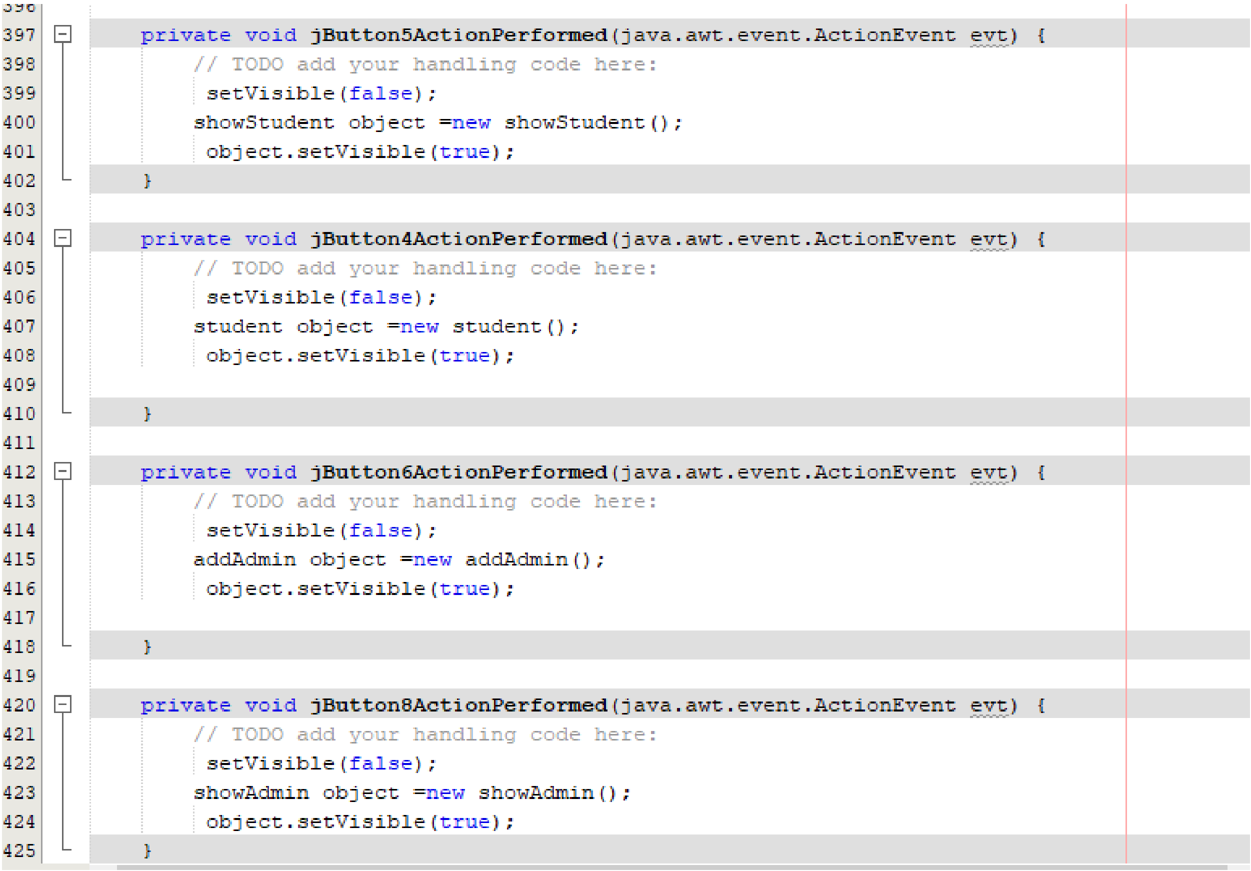
Home:



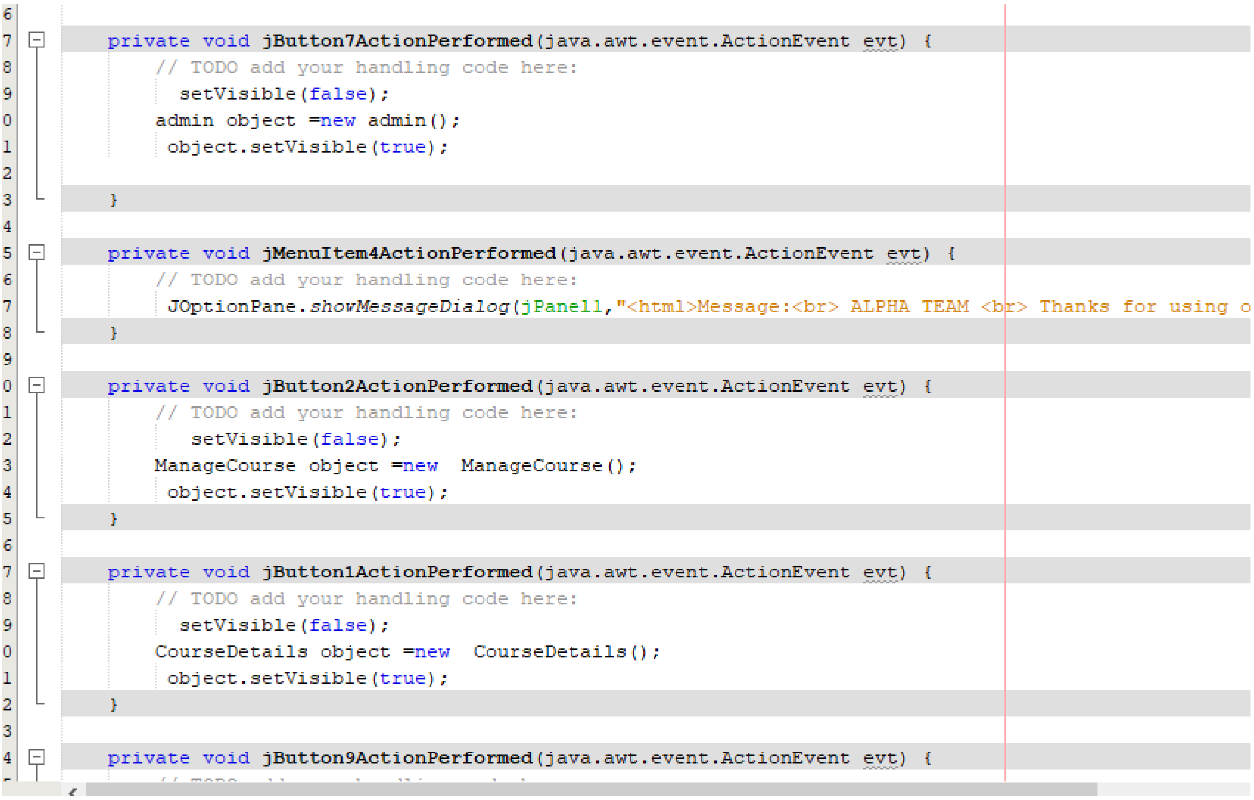
```
import javax.swing.JOptionPane;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author Ridoy-PC
 */
public class home extends javax.swing.JFrame {

    /**
     * Creates new form home
     */
    public home() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code
```



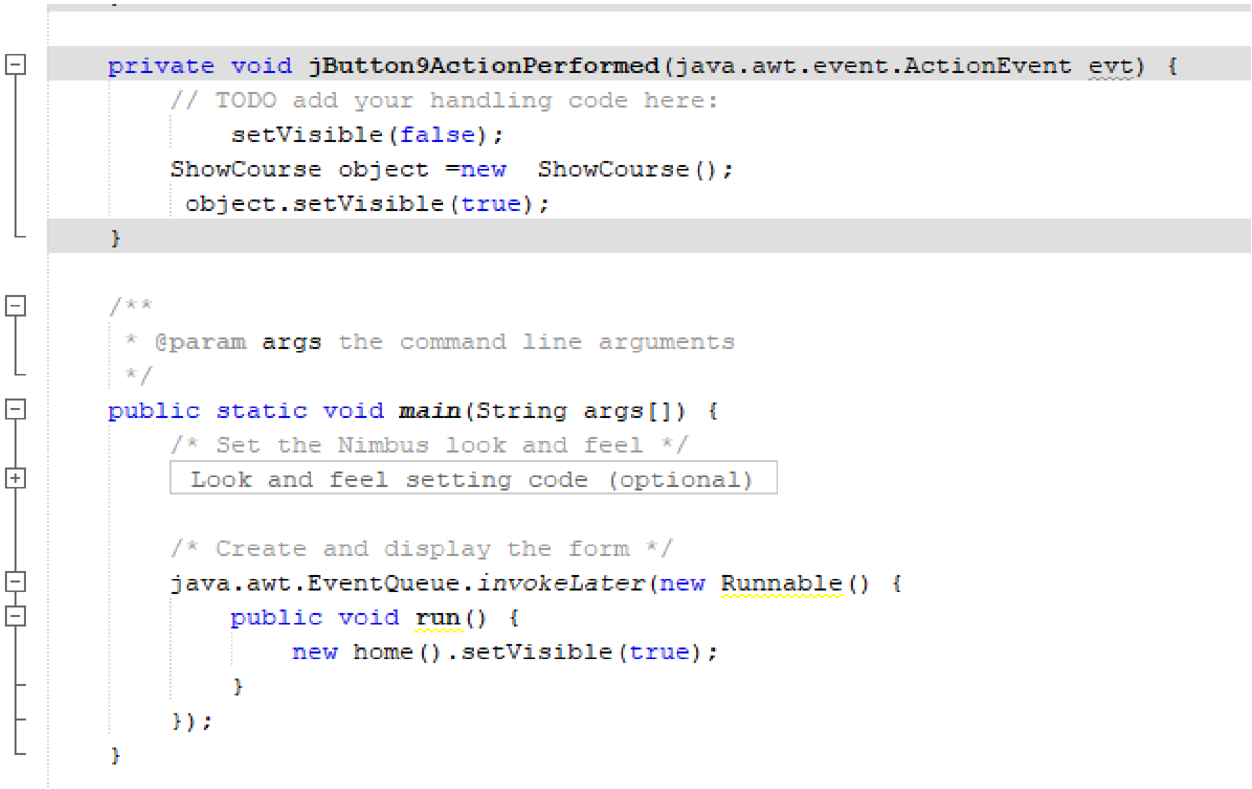
```
    private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setVisible(false);
        home object =new  home();
        object.setVisible(true);
    }

    private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setVisible(false);
        login object =new login();
        object.setVisible(true);
    }

    private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setVisible(false);
        about object =new about();
        object.setVisible(true);
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
         setVisible(false);
        addStudent object =new addStudent();
         object.setVisible(true);
    }

    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
```



```java
    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
         setVisible(false);
        showStudent object =new showStudent();
         object.setVisible(true);
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
         setVisible(false);
        student object =new student();
         object.setVisible(true);

    }

    private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
         setVisible(false);
        addAdmin object =new addAdmin();
         object.setVisible(true);

    }

    private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
         setVisible(false);
        showAdmin object =new showAdmin();
         object.setVisible(true);
    }
```



```java
    private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
          setVisible(false);
        admin object =new admin();
         object.setVisible(true);

    }

    private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
         JOptionPane.showMessageDialog(jPanel1,"<html>Message:<br> ALPHA TEAM <br> Thanks for using o
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
           setVisible(false);
        ManageCourse object =new  ManageCourse();
         object.setVisible(true);
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
          setVisible(false);
        CourseDetails object =new  CourseDetails();
         object.setVisible(true);
    }

    private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {
```



```
private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
        setVisible(false);
    ShowCourse object =new  ShowCourse();
     object.setVisible(true);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
     Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new home().setVisible(true);
        }
    });
}
```

REQUIREMENTS AND SCHEDULE POSTMORTEM

Features and Cuts: The major functionalities that we have declared in our original SRS are:

Any student's information will be saved as database and anytime the admins and even users can also see the data what will be input there. Student can Search his/her information by using his/her ID. If necessary information can be updated. In our software there is an option which is “**show students**” where all the information of students can be seen in a table.

The features above have all been done successfully and then we have decided to add more features in our software. We planned to add an option **‘Course’** for the students as if they can find it easy to choose their desired course by knowing about it properly. So again we started redesigning our software to accomplish this target and then successfully we could make it happen. Now students can visit the courses details and knowing all about the course, they can choose the proper one for themselves.

Task assignments: Team management is the most important task to finish any project. We are lucky here enough that all the members of our team contributed their **100%.** As we are Computer Science students, this kind of project was very essential for us to develop our skills and sharpen the existing ones. This part is to summarize the contribution details to the success of our team work:

Project Schedule and Team Structure:

Set specific day	We have set working day and time for each member in our group and the time is 6pm to 10pm.
Set tasks for team members	Every member got individual tasks for themselves. Such as, some worked with coding part, some had the task to find out the errors, some of us took care

	of the online side (downloading stuffs related to our project and solution of the errors) and some were to make the slide and document that we have to upload on Github.
Meeting time	We have scheduled a meeting time on every Sunday at 5pm.

Ridoy and **Fabiha** worked on the coding part. That means, designing the features and program of the featured designs was for these two.

Nayim, **Jahid** and **Rayhan** looked after the error finding and preparing the requirements part. During the program, we had to face errors sometimes for what we could not procced on. Then they helped to figure out these errors and we could continue.

We were also divide into two parts for writing the documents and presentation slides that we have to submit on Github. Here, **Ridoy** and **Fabiha** contributed to write the documents pdf and **Nayim**, **Jahid** and **Rayhan** did the slides for presentation.

We fixed a specific day and time in a week. On that day, we used to gather to share our any plans or ideas about project. We all are new in making software, so it’s normal that we have our own flaws. But all of us could overcome our lacking and finished the team work successfully by assisting our heart and soul. Last of all, we must have to say that, it was very effective method to gain knowledge and to utilize this knowledge in our future career as well.

RISK ASSESSMENT

We need a solution that makes identifying and assessing risk simple. The most effective risk identification techniques focus on root cause. It’s not enough to identify what happened. We need to understand why it happened in order to truly mitigate risk. If we want to find the risks in our project, first of all we must say about the programming side. If any code are written wrong even if a single comma/ semicolon/ dot are missing then the operation won’t work out. We have used a database software named ‘xampp’. If we don’t start it and don’t go to ‘Admin’ localhost site, the project will not run. That’s mean internet connection is also necessary to work with our project.

CONCLUSION

We have been tried our best to specify the whole design and process according to our plan of the project. We have fulfilled our expectation and completed the whole project as our designed plan. But it would be quite impossible if the teacher in charge would not help us. So before finishing this assignment, we want to cordially thank our honorable professor “**Zheng Liu”** to give us this opportunity and to show us the right way to fill up our target.