**ABSTRACT**

This Student Information System was developed using Java as the Frontend and MS Access Database integration. This system has all the object components in java such as buttons, text fields, radio button, combo box, list, images, checkbox that you can refer on how to use these components. It has also events for ActionEvents, ListSelection and ItemStateChanged and this system deals with education establishment to manage student data which providing capabilities for the management or administrator to manage student details by using database for storing information. This system can add, edit/update, delete, and search a particular student. It also filters wrong inputs, incorrect statements, etc.This system is very useful for small schools,colleges and universities.There are possibilities to add very good topic for students because people can add many features in this system.The main advantages of choosing this project idea is that,people can see available real time student information that will help to get and add to your student information system.The important options this system having is creating the student information,updating etc, where data can be searched using the student unique ID’s provided in the database which is accessed by the student information system.

**INTRODUCTION**

A student information system(SIS) is a management information system for education establishments to manage student data. Student information system provides capabilities for managing student-related data needs in a institution.This system vary in capability,from objective components that are implemented in relatively small organization to cover student records alone, which aim to cover most aspects of the running multi-campus organisations. From application,to career services and online education,some of may include in the products like admissions process, enrolling new students and student accounts,maintaining the student records.

In the previous system there is a step-by-step process from circulation of form(s) to student and receiving that forms(s) is consumes more time and we cannot verify identity. Thus the system of entering of user form(s) and documentation is much hectic.So when faculty wants to look at particular detail of student then faculty have to go through bunch of papers and find their particular detail.Above process consumes more time in searching of details.

This system we introduced reports so by using different reports we easily filter required data and sort through required data.In student information system we have a facility to store required detail for read it by the inputs given by the user,And here it helps multiple students in entering their details were they or other can access their data by unique ID of each student.

**Importance of Student Information System**

The competitiveness of most companies is in a large degree based on the effective use of information technologies and information systems especially. The main purpose of information systems are provide the right information to the right people at the right time. It is used to track, store, manipulate and distribute the information from gathered data to appropriate persons when necessary.

1. The first significant advantage of a student information system is that as a institution you are able to keep proper track of data related to students.
2. Can access all the data and more by using a unique identification number that you have provided to a student.
3. With the help of this system no longer would admin need to type in all the data of the students as well as others in the school on excel sheets for long hours!

It can all now be organized in a format that happens to be simple and accessible to one and all.

**DESCRIPTION**

The Student information system consists of Class with class name studentInformationSystem as public which extends JFrames implements ItemListener,ActionListener,ListSelectionListener.

JFrames which is used for inheriting the base class. Were JFrame class is a type of container which inherits the java.awt. Frames class.JFrame works like the main window where components like labels,buttons,textfields are added to create a GUI and The **Java ItemListener** is notified whenever you click on the checkbox. It is notified against ItemEvent. The **ItemListener** interface is found in **java**.awt.event were Listens for **item** events on a check box, which determines whether a lable is visible. CheckBoxDemo .

Action listeners are probably the easiest and most common event handlers to implement. One can implement an action listener to define what should be done when an user performs certain operation. An action event occurs, whenever an action is performed by the user. Examples: When the user clicks a button, chooses a menu item, presses Enter in a text field.Here It has only one method: actionPerformed(). The actionPerformed() method is invoked automatically whenever you click on the specified components.

In ListSelectionListener it Returns an array of all the ListSelectionListeners added to this JList by way of add ListSelectionListener.

RADIO BUTTON: A radio button or option button is a **graphical** control element that allows the user to choose only one of a **predefined set** of mutually exclusive options.

The Class.forName(String name, boolean initialize, ClassLoader loader) it makes java load the class sun.jdbc.odbc.jdbcOdbcDriver.In he case of JDBC driver when you open a database connection.Note that for newer JDBC drivers,this line isn’t necessary anymore-JDBC now uses another mechanism o automatically find drivers.

The DriverManager class acts as an interface between user and drivers. It keeps track of the drivers that are available and handles establishing a connection between a database and the appropriate driver. The DriverManager class maintains a list of Driver classes that have registered themselves by calling the method DriverManager.registerDriver().

**showMessageDialog(**) method is used to create a message dialog with given string and the messageType.

**showConfirmDialog()** method is used to create a dialog with the options Ok ,Cancel ;and Select an Option.

**CODE**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.util.\*;

import java.sql.\*;

import javax.swing.event.ListSelectionEvent;

import javax.swing.event.ListSelectionListener;

public class studentInformationSystem extends JFrame implements ItemListener,ActionListener, ListSelectionListener{

JPanel panel= new JPanel();

JLabel backGround = new JLabel(new ImageIcon("images3.jpg"));

JLabel labelID= new JLabel("ID No.:");

JLabel labelFName= new JLabel("First Name:");

JLabel labelMidName= new JLabel("Middle Initial:");

JLabel labelLastName = new JLabel("Last Name:");

JLabel labelAge= new JLabel("Age:");

JLabel labelFather= new JLabel("Father name:");

JLabel labelMother= new JLabel("Mother name:");

JLabel labelAadhar= new JLabel("Aadhar number:");

JLabel labelDOJ= new JLabel("Date of joining:");

JLabel labelPN= new JLabel("Phone number:");

JLabel labelEID= new JLabel("Email ID:");

JLabel labelGender= new JLabel("Gender:");

//JLabel labelStatus= new JLabel("Status:");

JLabel labelCity= new JLabel("City:");

JLabel labelRequirements = new JLabel ("Requirements:");

JLabel labelYearLevel = new JLabel ("Year Level:");

//JTextField

JTextField txtID=new JTextField ();

JTextField txtFName= new JTextField ();

JTextField txtMidName= new JTextField (0);

JTextField txtLastName= new JTextField ();

JTextField txtAge= new JTextField ();

JTextField txtFather= new JTextField ();

JTextField txtMother= new JTextField ();

JTextField txtAadhar= new JTextField ();

JTextField txtDOJ= new JTextField ();

JTextField txtPN= new JTextField ();

JTextField txtEID= new JTextField ();

JTextField txtCity= new JTextField();

//JTextField txtStatus = new JTextField();

JTextField txtRDiploma= new JTextField();

JTextField txtRBTECH = new JTextField();

JTextField txtRIT = new JTextField();

//JTextField txtRBirth = new JTextField();

JTextField txtRCSE = new JTextField();

JTextField txtRCSSE = new JTextField();

JTextField txtREEE = new JTextField();

JTextField txtFIRSTYEAR = new JTextField();

JTextField txtSECONDYEAR = new JTextField();

JTextField txtTHIRDYEAR= new JTextField();

JTextField txtFOURTHYEAR= new JTextField();

JTextField txtGender = new JTextField();

//RadioButton

JRadioButton RBMale=new JRadioButton("Male");

JRadioButton RBFemale= new JRadioButton("Female");

JRadioButton RBHidden=new JRadioButton("Hidden");

ButtonGroup bgroup= new ButtonGroup();

//checkbox for requirements

JCheckBox CBDiploma = new JCheckBox("Diploma");

JCheckBox CBBTECH = new JCheckBox("B.tech");

JCheckBox CBIT= new JCheckBox ("IT");

JCheckBox CBCSE= new JCheckBox ("CSE");

JCheckBox CBCSSE= new JCheckBox ("CSSE");

JCheckBox CBEEE= new JCheckBox ("EEE");

//checkbox for year level

JCheckBox CBFIRSTYEAR = new JCheckBox("First year");

JCheckBox CBSECONDYEAR = new JCheckBox("Second year");

JCheckBox CBTHIRDYEAR= new JCheckBox ("Third year");

JCheckBox CBFOURTHYEAR= new JCheckBox ("Fourth year");

//listbox

String [] listcity= {"Jaipur","Bengaluru","Mumbai","Kolkata","Pune","Hyderabad","Ahmedabad","Chennai","Surat","Chandigarh"};

JList citylist = new JList(listcity);

//Button

JButton New= new JButton ("New");

JButton save= new JButton ("Save");

JButton update = new JButton ("Update");

JButton delete= new JButton ("Delete");

JButton search= new JButton ("Search");

JButton cancel=new JButton ("Cancel");

Connection cn;

Statement st;

PreparedStatement ps;

public studentInformationSystem() {

super ("Student Information System");

setDefaultCloseOperation (JFrame.EXIT\_ON\_CLOSE);

panel.setLayout(null);

panel.add(backGround);

backGround.setBounds(350,300,310,250);

panel.setBackground(Color.pink);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*JLabel

panel.add(labelRequirements);

labelRequirements.setBounds (280,5,120,30);

panel.add(labelYearLevel);

labelYearLevel.setBounds (500,5,120,30);

panel.add(labelID);

labelID.setBounds (30,20,120,30);

panel.add(labelFName);

labelFName.setBounds (30,50,120,30);

panel.add(labelMidName) ;

labelMidName.setBounds (30,80,120,30);

panel.add(labelLastName);

labelLastName.setBounds (30,110,120,30);

panel.add(labelAge);

labelAge.setBounds (30,140,120,30);

panel.add(labelFather);

labelFather.setBounds (30,170,120,30);

panel.add(labelMother);

labelMother.setBounds (30,200,120,30);

panel.add(labelAadhar);

labelAadhar.setBounds (30,230,120,30);

panel.add(labelDOJ);

labelDOJ.setBounds (30,260,120,30);

panel.add(labelPN);

labelPN.setBounds (30,290,120,30);

panel.add(labelEID);

labelEID.setBounds (30,320,120,30);

panel.add (labelGender);

labelGender.setBounds (30,370,120,30);

panel.add (labelCity);

labelCity.setBounds (30,430,120,30);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*JTextField

panel.add (txtID);

txtID.setBounds (120,20,120,30);

txtID.setHorizontalAlignment (JTextField.LEFT);

panel.add (txtFName);

txtFName.setBounds (120,50,120,30);

txtFName.setHorizontalAlignment (JTextField.LEFT);

panel.add (txtMidName);

txtMidName.setBounds (120,80,120,30);

txtMidName.setHorizontalAlignment (JTextField.LEFT);

panel.add (txtLastName);

txtLastName.setBounds (120,110,120,30);

txtLastName.setHorizontalAlignment (JTextField.LEFT);

panel.add (txtAge);

txtAge.setBounds (120,140,120,30);

txtAge.setHorizontalAlignment (JTextField.LEFT);

panel.add (txtFather);

txtFather.setBounds (120,170,120,30);

txtFather.setHorizontalAlignment (JTextField.LEFT);

panel.add (txtMother);

txtMother.setBounds (120,200,120,30);

txtMother.setHorizontalAlignment (JTextField.LEFT);

panel.add (txtAadhar);

txtAadhar.setBounds (120,230,120,30);

txtAadhar.setHorizontalAlignment (JTextField.LEFT);

panel.add (txtDOJ);

txtDOJ.setBounds (120,260,120,30);

txtDOJ.setHorizontalAlignment (JTextField.LEFT);

panel.add (txtPN);

txtPN.setBounds (120,290,120,30);

txtPN.setHorizontalAlignment (JTextField.LEFT);

panel.add (txtEID);

txtEID.setBounds (120,320,120,30);

txtEID.setHorizontalAlignment (JTextField.LEFT);

panel.add(txtCity);

txtCity.setBounds (130,420,100,30);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RadioButton

panel.add (RBMale);

RBMale.setBounds(100,370,90,30);

RBMale.setBackground(Color.white);

txtGender.setText("male");

panel.add (RBFemale);

RBFemale.setBounds (190,370,90,30);

RBFemale.setBackground(Color.white);

txtGender.setText("female");

bgroup.add (RBMale);

bgroup.add (RBFemale);

bgroup.add (RBHidden);

RBMale.addItemListener(this);

RBFemale.addItemListener(this);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*checkbox

panel.add (CBDiploma);

CBDiploma.setBounds (350,30,120,30);

CBDiploma.setBackground(Color.white);

CBDiploma.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent event) {

JCheckBox CBDiploma = (JCheckBox)event.getSource();

if (CBDiploma.isSelected()) {

txtRDiploma.setText("check");

} else {

txtRDiploma.setText("");

}

}

});

panel.add (CBBTECH);

CBBTECH.setBounds (350,60,120,30);

CBBTECH.setBackground(Color.white);

CBBTECH.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent event) {

JCheckBox CBBTECH = (JCheckBox)event.getSource();

if (CBBTECH.isSelected()) {

txtRBTECH.setText("check");

} else {

txtRBTECH.setText("");

}

}

});

panel.add (CBIT);

CBIT.setBounds (350,90,120,30);

CBIT.setBackground(Color.white);

CBIT.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent event) {

JCheckBox CBIT = (JCheckBox)event.getSource();

if (CBIT.isSelected()) {

txtRIT.setText("check");

} else {

txtRIT.setText("");

}

}

});

panel.add (CBCSE);

CBCSE.setBounds (350,120,120,30);

CBCSE.setBackground(Color.white);

CBCSE.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent event) {

JCheckBox CBCSE = (JCheckBox)event.getSource();

if (CBCSE.isSelected()) {

txtRCSE.setText("check");

} else {

txtRCSE.setText("");

}

}

});

panel.add (CBCSSE);

CBCSSE.setBounds (350,150,120,30);

CBCSSE.setBackground(Color.white);

CBCSSE.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent event) {

JCheckBox CBCSSE = (JCheckBox)event.getSource();

if (CBCSSE.isSelected()) {

txtRCSSE.setText("check");

} else {

txtRCSSE.setText("");

}

}

});

panel.add (CBEEE);

CBEEE.setBounds (350,180,120,30);

CBEEE.setBackground(Color.white);

CBEEE.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent event) {

JCheckBox CBEEE = (JCheckBox)event.getSource();

if (CBEEE.isSelected()) {

txtREEE.setText("check");

} else {

txtREEE.setText("");

}

}

});

panel.add (CBFIRSTYEAR);

CBFIRSTYEAR.setBounds (500,30,120,30);

CBFIRSTYEAR.setBackground(Color.white);

CBFIRSTYEAR.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent event) {

JCheckBox CBFIRSTYEAR = (JCheckBox)event.getSource();

if (CBFIRSTYEAR.isSelected()) {

txtFIRSTYEAR.setText("check");

} else {

txtFIRSTYEAR.setText("");

}

}

});

panel.add (CBSECONDYEAR);

CBSECONDYEAR.setBounds (500,60,120,30);

CBSECONDYEAR.setBackground(Color.white);

CBSECONDYEAR.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent event) {

JCheckBox CBSECONDYEAR = (JCheckBox)event.getSource();

if (CBSECONDYEAR.isSelected()) {

txtSECONDYEAR.setText("check");

} else {

txtSECONDYEAR.setText("");

}

}

});

panel.add (CBTHIRDYEAR);

CBTHIRDYEAR.setBounds (500,90,120,30);

CBTHIRDYEAR.setBackground(Color.white);

CBTHIRDYEAR.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent event) {

JCheckBox CBTHIRDYEAR= (JCheckBox)event.getSource();

if (CBTHIRDYEAR.isSelected()) {

txtTHIRDYEAR.setText("check");

} else {

txtTHIRDYEAR.setText("");

}

}

});

panel.add (CBFOURTHYEAR);

CBFOURTHYEAR.setBounds (500,120,120,30);

CBFOURTHYEAR.setBackground(Color.white);

CBFOURTHYEAR.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent event) {

JCheckBox CBFOURTHYEAR= (JCheckBox)event.getSource();

if (CBFOURTHYEAR.isSelected()) {

txtFOURTHYEAR.setText("check");

} else {

txtFOURTHYEAR.setText("");

}

}

});

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*listbox

panel.add(citylist);

citylist.setBounds (130,450,100,180);

citylist.setSelectedIndex(0);

citylist.setBackground(Color.gray);

citylist.addListSelectionListener(new ListSelectionListener(){

public void valueChanged(ListSelectionEvent cl){

txtCity.setText(listcity[citylist.getSelectedIndex()]);

}

});

//Button

panel.add (New);

New.setBounds (30,650,80,30);

panel.add (save);

save.setBounds (125,650,80,30);

panel.add (update);

update.setBounds (215,650,80,30);

panel.add (delete);

delete.setBounds (305,650,80,30);

panel.add (search);

search.setBounds (395,650,80,30);

panel.add(cancel);

cancel.setBounds (485,650,80,30);

txtID.setEnabled(false);

txtFName.setEnabled(false);

txtMidName.setEnabled(false);

txtLastName.setEnabled(false);

txtAge.setEnabled(false);

txtFather.setEnabled(false);

txtMother.setEnabled(false);

txtAadhar.setEnabled(false);

txtDOJ.setEnabled(false);

txtPN.setEnabled(false);

txtEID.setEnabled(false);

txtCity.setEnabled(false);

//txtStatus.setEnabled(false);

txtRDiploma.setEnabled(false);

txtRBTECH.setEnabled(false);

txtRIT.setEnabled(false);

txtRCSE.setEnabled(false);

txtRCSSE.setEnabled(false);

txtREEE.setEnabled(false);

//txtRBirth.setEnabled(false);

txtGender.setEnabled(false);

txtFIRSTYEAR.setEnabled(false);

txtSECONDYEAR.setEnabled(false);

txtTHIRDYEAR.setEnabled(false);

txtFOURTHYEAR.setEnabled(false);

//CBBirth.setEnabled(false);

CBDiploma.setEnabled(false);

CBBTECH.setEnabled(false);

CBIT.setEnabled(false);

CBCSE.setEnabled(false);

CBCSSE.setEnabled(false);

CBEEE.setEnabled(false);

CBFIRSTYEAR.setEnabled(false);

CBSECONDYEAR.setEnabled(false);

CBTHIRDYEAR.setEnabled(false);

CBFOURTHYEAR.setEnabled(false);

New.addActionListener (this);

save.addActionListener (this);

save.setEnabled(false);

update.addActionListener (this);

update.setEnabled(false);

delete.addActionListener (this);

delete.setEnabled(false);

search.addActionListener (this);

search.setEnabled(true);

cancel.addActionListener(this);

setContentPane(panel);

try{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

cn = DriverManager.getConnection("jdbc:odbc:ADB");

}catch(ClassNotFoundException e){

System.err.println("Failed to Load Driver");

e.printStackTrace();

}catch (SQLException e){

System.err.println("Unable to Connect");

e.printStackTrace();

}

}

public void clear(){

txtID.setText("");

txtFName.setText("");

txtMidName.setText("");

txtLastName.setText("");

txtAge.setText("");

txtFather.setText("");

txtMother.setText("");

txtAadhar.setText("");

txtDOJ.setText("");

txtPN.setText("");

txtEID.setText("");

txtCity.setText("");

txtGender.setText("");

//txtStatus.setText("");

txtRDiploma.setText("");

txtRBTECH.setText("");

txtRIT.setText("");

txtRCSE.setText("");

txtRCSSE.setText("");

txtREEE.setText("");

//txtRBirth.setText("");

txtFIRSTYEAR.setText("");

txtSECONDYEAR.setText("");

txtTHIRDYEAR.setText("");

txtFOURTHYEAR.setText("");

RBHidden.setEnabled(true);

//CBBirth.setSelected(false);

CBDiploma.setSelected(false);

CBBTECH.setSelected(false);

CBIT.setSelected(false);

CBCSE.setSelected(false);

CBCSSE.setSelected(false);

CBEEE.setSelected(false);

CBFIRSTYEAR.setSelected(false);

CBSECONDYEAR.setSelected(false);

CBTHIRDYEAR.setSelected(false);

CBFOURTHYEAR.setSelected(false);

//statbox.setSelectedIndex(0);

}

public void actionPerformed (ActionEvent e){

Object source = e.getSource();

if (source==New){

save.setEnabled (true) ;

update.setEnabled (false);

delete.setEnabled (false);

search.setEnabled (false);

txtID.setEnabled(true);

txtFName.setEnabled(true);

txtMidName.setEnabled(true);

txtLastName.setEnabled(true);

txtAge.setEnabled(true);

txtFather.setEnabled(true);

txtMother.setEnabled(true);

txtAadhar.setEnabled(true);

txtDOJ.setEnabled(true);

txtPN.setEnabled(true);

txtEID.setEnabled(true);

txtCity.setEnabled(true);

//txtStatus.setEnabled(true);

txtRDiploma.setEnabled(true);

txtRBTECH.setEnabled(true);

txtRIT.setEnabled(true);

txtRCSE.setEnabled(true);

txtRCSSE.setEnabled(true);

txtREEE.setEnabled(true);

//txtRBirth.setEnabled(true);

txtGender.setEnabled(true);

txtFIRSTYEAR.setEnabled(true);

txtSECONDYEAR.setEnabled(true);

txtTHIRDYEAR.setEnabled(true);

txtFOURTHYEAR.setEnabled(true);

//CBBirth.setEnabled(true);

CBDiploma.setEnabled(true);

CBBTECH.setEnabled(true);

CBIT.setEnabled(true);

CBCSE.setEnabled(true);

CBCSSE.setEnabled(true);

CBEEE.setEnabled(true);

CBFIRSTYEAR.setEnabled(true);

CBSECONDYEAR.setEnabled(true);

CBTHIRDYEAR.setEnabled(true);

CBFOURTHYEAR.setEnabled(true);

txtID.setText("");

txtFName.setText("");

txtMidName.setText("");

txtLastName.setText("");

txtAge.setText("");

txtFather.setText("");

txtMother.setText("");

txtAadhar.setText("");

txtDOJ.setText("");

txtPN.setText("");

txtEID.setText("");

txtCity.setText("");

txtGender.setText("");

//txtStatus.setText("");

txtRDiploma.setText("");

txtRBTECH.setText("");

txtRIT.setText("");

txtRCSE.setText("");

txtRCSSE.setText("");

txtREEE.setText("");

//txtRBirth.setText("");

txtFIRSTYEAR.setText("");

txtSECONDYEAR.setText("");

txtTHIRDYEAR.setText("");

txtFOURTHYEAR.setText("");

}

if(source == save){

String id=txtID.getText();

String fname=txtFName.getText();

String midname=txtMidName.getText();

String lastname=txtLastName.getText();

String father = txtFather.getText();

String mother = txtMother.getText();

String aadhar = txtAadhar.getText();

String age= txtAge.getText();

String doj= txtDOJ.getText();

String pn= txtPN.getText();

String eid= txtEID.getText();

String city=txtCity.getText();

String gender=txtGender.getText();

//String status =txtStatus.getText();

String diploma =txtRDiploma.getText();

String btech=txtRBTECH.getText();

String it=txtRIT.getText();

String cse=txtRCSE.getText();

String csse=txtRCSSE.getText();

String eee=txtREEE.getText();

//String birth=txtRBirth.getText();

String firstyear=txtFIRSTYEAR.getText();

String secondyear=txtSECONDYEAR.getText();

String thirdyear=txtTHIRDYEAR.getText();

String fourthyear=txtFOURTHYEAR.getText();

if (!gender.equals("")&&!id.equals("") && !fname.equals("")&& !midname.equals("") && !lastname.equals("") && !age.equals("")&& !father.equals("")&& !mother.equals("")&& aadhar.equals("")&& doj.equals("")&& pn.equals("")&& eid.equals("")&& !city.equals("")){

New.setEnabled (true);

update.setEnabled (false);

delete.setEnabled (false);

search.setEnabled(true);

//----------------------------Check--------------------

int idt=0;

int idtChecker=0;

int aget=0;

int agetChecker=0;

int checkerfnamet=0;

int checkermidnamet=0;

int checkerlastnamet=0;

int checkerfathert=0;

int checkermothert=0;

int aadhart=0;

int aadhartChecker=0;

int dojt=0;

int dojtChecker=0;

int pnt=0;

int pntChecker=0;

int checkereidt=0;

try{

aget=Integer.parseInt(txtAge.getText());

if(aget==0) {

JOptionPane.showMessageDialog(null,"Change Age","Warning",JOptionPane.WARNING\_MESSAGE);

agetChecker=1;

}

}catch(NumberFormatException nfe){

JOptionPane.showMessageDialog(null,"Age should be Numbers only","Warning",JOptionPane.WARNING\_MESSAGE);

aget=0;

agetChecker=1;

}

try{

idt=Integer.parseInt(txtID.getText());

}catch(NumberFormatException nfe){

JOptionPane.showMessageDialog(null,"ID should be Numbers only","Warning",JOptionPane.WARNING\_MESSAGE);

idtChecker=1;

}

try{

aadhart=Integer.parseInt(txtAadhar.getText());

if(aadhart==0) {

JOptionPane.showMessageDialog(null,"Change Aadhar","Warning",JOptionPane.WARNING\_MESSAGE);

aadhartChecker=1;

}

}catch(NumberFormatException nfe){

JOptionPane.showMessageDialog(null,"Aadhar should be Numbers only","Warning",JOptionPane.WARNING\_MESSAGE);

aadhart=0;

aadhartChecker=1;

}

try{

dojt=Integer.parseInt(txtDOJ.getText());

if(dojt==0) {

JOptionPane.showMessageDialog(null,"Change date of joining","Warning",JOptionPane.WARNING\_MESSAGE);

dojtChecker=1;

}

}catch(NumberFormatException nfe){

JOptionPane.showMessageDialog(null,"date of joining should be Numbers only","Warning",JOptionPane.WARNING\_MESSAGE);

dojt=0;

dojtChecker=1;

}

try{

pnt=Integer.parseInt(txtPN.getText());

if(pnt==0) {

JOptionPane.showMessageDialog(null,"Change phone number","Warning",JOptionPane.WARNING\_MESSAGE);

pntChecker=1;

}

}catch(NumberFormatException nfe){

JOptionPane.showMessageDialog(null,"phone numner should be Numbers only","Warning",JOptionPane.WARNING\_MESSAGE);

pnt=0;

pntChecker=1;

}

String checkfname=txtFName.getText();

String checklastname=txtLastName.getText();

String checkmidname=txtMidName.getText();

String checkfather=txtFather.getText();

String checkmother=txtMother.getText();

String checkeid=txtEID.getText();

if(checkfname.matches(".\*\\d.\*")){

JOptionPane.showMessageDialog(null,"Invalid input in First Name","Warning",JOptionPane.WARNING\_MESSAGE);

checkerfnamet=1;

} else{

checkerfnamet=0;

}

if(checklastname.matches(".\*\\d.\*")){

JOptionPane.showMessageDialog(null,"Invalid input in Last Name","Warning",JOptionPane.WARNING\_MESSAGE);

checkerlastnamet=1;

} else{

checkerlastnamet=0;

}

if(checkmidname.matches(".\*\\d.\*")){

JOptionPane.showMessageDialog(null,"Invalid input in Mid Name","Warning",JOptionPane.WARNING\_MESSAGE);

checkermidnamet=1;

} else{

checkermidnamet=0;

}

if(checkfather.matches(".\*\\d.\*")){

JOptionPane.showMessageDialog(null,"Invalid input in Father Name","Warning",JOptionPane.WARNING\_MESSAGE);

checkerfathert=1;

} else{

checkerfathert=0;

}

if(checkmother.matches(".\*\\d.\*")){

JOptionPane.showMessageDialog(null,"Invalid input in Mother Name","Warning",JOptionPane.WARNING\_MESSAGE);

checkermothert=1;

} else{

checkermothert=0;

}

if(checkeid.matches(".\*\\d.\*")){

JOptionPane.showMessageDialog(null,"Invalid input in eid Name","Warning",JOptionPane.WARNING\_MESSAGE);

checkereidt=1;

} else{

checkereidt=0;

}

if(txtMidName.getText().length()>1){

JOptionPane.showMessageDialog(null,"Set Middle initial to one character only!","Warning",JOptionPane.WARNING\_MESSAGE);

checkermidnamet=1;

}

if(checkerfnamet==1||checkermidnamet==1||checkerlastnamet==1||idtChecker==1||agetChecker==1||aadhartChecker==1||checkerfathert==1||checkermothert==1||dojtChecker==1||pntChecker==1||checkereidt==1){

}

//----------------------------End of Check ;P--------------------

else{

String idx = String.valueOf(idt);

try{

if (!gender.equals("")&&!idx.equals("") && !fname.equals("")&& aadhart!=0 && !mother.equals("") && !father.equals("")&& !midname.equals("") && !lastname.equals("") && aget!=0&& !city.equals("")&& dojt!=0&& pnt!=0&& !eid.equals("")) {

st= cn.createStatement();

ps=cn.prepareStatement("INSERT INTO Login" + " (ID,FName,MidName,LastName,Age,Father,Mother,Aadhar,Gender,City,Diploma,BTECH,IT,CSE,CSSE,EEE,FIRSTYEAR,SECONDYEAR,THIRDYEAR) " + " VALUES(?,?,?,?,?,?,?,?,?,?,?,?)");

ps.setString(1,txtID.getText());

ps.setString(2,txtFName.getText());

ps.setString(3,txtMidName.getText());

ps.setString(4,txtLastName.getText());

ps.setString(5,txtAge.getText());

ps.setString(6,txtFather.getText());

ps.setString(7,txtMother.getText());

ps.setString(8,txtAadhar.getText());

ps.setString(9,txtDOJ.getText());

ps.setString(10,txtPN.getText());

ps.setString(11,txtEID.getText());

ps.setString(12,txtGender.getText());

//ps.setString(13,txtStatus.getText());

ps.setString(13,txtCity.getText());

if(diploma.equals("check")){

ps.setString(14,"check");

}

else{

ps.setString(14,"null");

}

if(btech.equals("check")){

ps.setString(15,"check");

}

else{

ps.setString(15,"null");

}

if(it.equals("check")){

ps.setString(16,"check");

}

else{

ps.setString(16,"null");

}

if(cse.equals("check")){

ps.setString(17,"check");

}

else{

ps.setString(17,"null");

}

if(csse.equals("check")){

ps.setString(18,"check");

}

else{

ps.setString(18,"null");

}

if(eee.equals("check")){

ps.setString(19,"check");

}

else{

ps.setString(19,"null");

}

/\*if(birth.equals("check")){

ps.setString(21,"check");

}

else{

ps.setString(21,"null");

}\*/

if(firstyear.equals("check")){

ps.setString(21,"check");

}

else{

ps.setString(21,"null");

}

if(secondyear.equals("check")){

ps.setString(22,"check");

}

else{

ps.setString(22,"null");

}

if(thirdyear.equals("check")){

ps.setString(23,"check");

}

else{

ps.setString(23,"null");

}

if(fourthyear.equals("check")){

ps.setString(24,"check");

}

else{

ps.setString(24,"null");

}

ps.executeUpdate();

JOptionPane.showMessageDialog(null,"New account has been successfully added.","Student Information System",JOptionPane.INFORMATION\_MESSAGE);

txtID.setEnabled(false);

txtFName.setEnabled(false);

txtMidName.setEnabled(false);

txtLastName.setEnabled(false);

txtAge.setEnabled(false);

txtFather.setEnabled(false);

txtMother.setEnabled(false);

txtAadhar.setEnabled(false);

txtDOJ.setEnabled(false);

txtPN.setEnabled(false);

txtEID.setEnabled(false);

txtCity.setEnabled(false);

//txtStatus.setEnabled(false);

txtRDiploma.setEnabled(false);

txtRBTECH.setEnabled(false);

txtRIT.setEnabled(false);

txtRCSE.setEnabled(false);

txtRCSSE.setEnabled(false);

txtREEE.setEnabled(false);

//txtRBirth.setEnabled(false);

txtFIRSTYEAR.setEnabled(false);

txtSECONDYEAR.setEnabled(false);

txtTHIRDYEAR.setEnabled(false);

txtFOURTHYEAR.setEnabled(false);

txtGender.setEnabled(false);

//CBBirth.setEnabled(false);

CBDiploma.setEnabled(false);

CBBTECH.setEnabled(false);

CBIT.setEnabled(false);

CBCSE.setEnabled(false);

CBCSSE.setEnabled(false);

CBEEE.setEnabled(false);

CBFIRSTYEAR.setEnabled(false);

CBSECONDYEAR.setEnabled(false);

CBTHIRDYEAR.setEnabled(false);

CBFOURTHYEAR.setEnabled(false);

st.close();

clear();

}

else{

JOptionPane.showMessageDialog(null,"Fill Up the Empty Fields!.","Student Information System",JOptionPane.ERROR\_MESSAGE);

}

}catch(SQLException sqlEx){

sqlEx.printStackTrace();

JOptionPane.showMessageDialog(null,"Unable to save! ID number already taken.","Student Information System",JOptionPane.ERROR\_MESSAGE);}

}

}

else{

JOptionPane.showMessageDialog(null,"Fill Up the Empty Fields!.","warning",JOptionPane.ERROR\_MESSAGE);

}

}

if (source==delete){

New.setEnabled (true);

save.setEnabled (true);

delete.setEnabled (true);

search.setEnabled (true);

}

if (source==search){

delete.setEnabled(true);

update.setEnabled(true);

save.setEnabled(false);

//CBBirth.setSelected(false);

CBBTECH.setSelected(false);

CBDiploma.setSelected(false);

CBIT.setSelected(false);

CBCSE.setSelected(false);

CBCSSE.setSelected(false);

CBEEE.setSelected(false);

CBFIRSTYEAR.setSelected(false);

CBSECONDYEAR.setSelected(false);

CBTHIRDYEAR.setSelected(false);

CBFOURTHYEAR.setSelected(false);

txtID.setEnabled(true);

txtFName.setEnabled(true);

txtMidName.setEnabled(true);

txtLastName.setEnabled(true);

txtAge.setEnabled(true);

txtFather.setEnabled(true);

txtMother.setEnabled(true);

txtAadhar.setEnabled(true);

txtDOJ.setEnabled(true);

txtPN.setEnabled(true);

txtEID.setEnabled(true);

txtCity.setEnabled(true);

//txtStatus.setEnabled(true);

txtRDiploma.setEnabled(true);

txtRBTECH.setEnabled(true);

txtRIT.setEnabled(true);

txtRCSE.setEnabled(true);

txtRCSSE.setEnabled(true);

txtREEE.setEnabled(true);

//txtRBirth.setEnabled(true);

txtFIRSTYEAR.setEnabled(true);

txtSECONDYEAR.setEnabled(true);

txtTHIRDYEAR.setEnabled(true);

txtFOURTHYEAR.setEnabled(true);

txtGender.setEnabled(true);

//CBBirth.setEnabled(true);

CBDiploma.setEnabled(true);

CBBTECH.setEnabled(true);

CBIT.setEnabled(true);

CBCSE.setEnabled(true);

CBCSSE.setEnabled(true);

CBEEE.setEnabled(true);

CBFIRSTYEAR.setEnabled(true);

CBSECONDYEAR.setEnabled(true);

CBTHIRDYEAR.setEnabled(true);

CBFOURTHYEAR.setEnabled(true);

String sUser ="";

int tmp= 0;

clear();

sUser = JOptionPane.showInputDialog(null,"Enter Student ID to search.","Student Information System",JOptionPane.QUESTION\_MESSAGE);

if(!sUser.equals("")){

try{

st=cn.createStatement();

ResultSet rs=st.executeQuery("SELECT \* FROM Login WHERE ID = '" + sUser + "'");

while(rs.next()){

txtID.setText(rs.getString(1));

txtFName.setText(rs.getString(2));

txtMidName.setText(rs.getString(3));

txtLastName.setText(rs.getString(4));

txtAge.setText(rs.getString(5));

txtFather.setText(rs.getString(6));

txtMother.setText(rs.getString(7));

txtAadhar.setText(rs.getString(8));

txtDOJ.setText(rs.getString(9));

txtPN.setText(rs.getString(10));

txtEID.setText(rs.getString(11));

txtGender.setText(rs.getString(12));

if(txtGender.equals("female")){

RBFemale.setSelected(true);

}

else{

RBMale.setSelected(true);

}

txtCity.setText(rs.getString(13));

if(rs.getString(14).equals("check")){

CBDiploma.setSelected(true);

txtRDiploma.setText(("check"));

}

else{

}

if(rs.getString(15).equals("check")){

CBBTECH.setSelected(true);

txtRBTECH.setText("check");

}

else{

}

if(rs.getString(16).equals("check")){

CBIT.setSelected(true);

txtRIT.setText("check");

}

else{

}

if(rs.getString(17).equals("check")){

CBCSE.setSelected(true);

txtRCSE.setText("check");

}

else{

}

if(rs.getString(18).equals("check")){

CBCSSE.setSelected(true);

txtRCSSE.setText("check");

}

else{

}

if(rs.getString(19).equals("check")){

CBEEE.setSelected(true);

txtREEE.setText("check");

}

else{

}

/\*if(rs.getString(21).equals("check")){

CBBirth.setSelected(true);

txtRBirth.setText("check");

}

else{

}\*/

if(rs.getString(20).equals("check")){

CBFIRSTYEAR.setSelected(true);

txtFIRSTYEAR.setText("check");

}

else{

}

if(rs.getString(21).equals("check")){

CBSECONDYEAR.setSelected(true);

txtSECONDYEAR.setText("check");

}

else{

}

if(rs.getString(22).equals("check")){

CBTHIRDYEAR.setSelected(true);

txtTHIRDYEAR.setText("check");

}

else{

}

if(rs.getString(23).equals("check")){

CBFOURTHYEAR.setSelected(true);

txtFOURTHYEAR.setText("check");

}

else{

}

tmp=1;

}

st.close();

if (tmp==0){

update.setEnabled(false);

delete.setEnabled(false);

JOptionPane.showMessageDialog(null,"No record found!!.","Student Information System",JOptionPane.INFORMATION\_MESSAGE);

}

}catch(SQLException s){

JOptionPane.showMessageDialog(null,"Unable to search!.","Student Information System",JOptionPane.ERROR\_MESSAGE);

System.out.println("SQL Error" + s.toString() + " " + s.getErrorCode() + " " + s.getSQLState());

}

}

}

if(source==update){

txtID.setEnabled(true);

txtFName.setEnabled(true);

txtMidName.setEnabled(true);

txtLastName.setEnabled(true);

txtAge.setEnabled(true);

txtFather.setEnabled(true);

txtMother.setEnabled(true);

txtAadhar.setEnabled(true);

txtDOJ.setEnabled(true);

txtPN.setEnabled(true);

txtEID.setEnabled(true);

txtCity.setEnabled(true);

//txtStatus.setEnabled(true);

txtGender.setEnabled(true);

txtRDiploma.setEnabled(true);

txtRBTECH.setEnabled(true);

txtRIT.setEnabled(true);

txtRCSE.setEnabled(true);

txtRCSSE.setEnabled(true);

txtREEE.setEnabled(true);

//txtRBirth.setEnabled(true);

txtFIRSTYEAR.setEnabled(true);

txtSECONDYEAR.setEnabled(true);

txtTHIRDYEAR.setEnabled(true);

txtFOURTHYEAR.setEnabled(true);

//CBBirth.setEnabled(true);

CBDiploma.setEnabled(true);

CBBTECH.setEnabled(true);

CBIT.setEnabled(true);

CBCSE.setEnabled(true);

CBCSSE.setEnabled(true);

CBEEE.setEnabled(true);

CBFIRSTYEAR.setEnabled(true);

CBSECONDYEAR.setEnabled(true);

CBTHIRDYEAR.setEnabled(true);

CBFOURTHYEAR.setEnabled(true);

//----------------------------Check--------------------

int idt=0;

int idtChecker=0;

int aget=0;

int agetChecker=0;

int checkerfnamet=0;

int checkermidnamet=0;

int checkerlastnamet=0;

int checkerfathert=0;

int checkermothert=0;

int aadhart=0;

int aadhartChecker=0;

int dojt=0;

int dojtChecker=0;

int pnt=0;

int pntChecker=0;

int checkereidt=0;

try{

aget=Integer.parseInt(txtAge.getText());

if(aget==0){ agetChecker=1;

JOptionPane.showMessageDialog(null,"Change Age","Warning",JOptionPane.WARNING\_MESSAGE);

}

}catch(NumberFormatException nfe){

JOptionPane.showMessageDialog(null,"Age should be Numbers only","Warning",JOptionPane.WARNING\_MESSAGE);

agetChecker=1;

aget=0;

}

try{

aadhart=Integer.parseInt(txtAadhar.getText());

if(aadhart==0){ aadhartChecker=1;

JOptionPane.showMessageDialog(null,"Change Aadhar","Warning",JOptionPane.WARNING\_MESSAGE);

}

}catch(NumberFormatException nfe){

JOptionPane.showMessageDialog(null,"Aadhar should be Numbers only","Warning",JOptionPane.WARNING\_MESSAGE);

aadhartChecker=1;

aadhart=0;

}

try{

idt=Integer.parseInt(txtID.getText());

}catch(NumberFormatException nfe){

JOptionPane.showMessageDialog(null,"ID should be Numbers only","Warning",JOptionPane.WARNING\_MESSAGE);

idtChecker=1;

}

try{

dojt=Integer.parseInt(txtDOJ.getText());

if(dojt==0) {dojtChecker=1;

JOptionPane.showMessageDialog(null,"Change date of joining","Warning",JOptionPane.WARNING\_MESSAGE);

}

}catch(NumberFormatException nfe){

JOptionPane.showMessageDialog(null,"date of joining should be Numbers only","Warning",JOptionPane.WARNING\_MESSAGE);

dojtChecker=1;

dojt=0;

}

try{

pnt=Integer.parseInt(txtPN.getText());

if(pnt==0) {pntChecker=1;

JOptionPane.showMessageDialog(null,"Change phone number","Warning",JOptionPane.WARNING\_MESSAGE);

}

}catch(NumberFormatException nfe){

JOptionPane.showMessageDialog(null,"phone numner should be Numbers only","Warning",JOptionPane.WARNING\_MESSAGE);

pntChecker=1;

pnt=0;

}

String checkfname=txtFName.getText();

String checklastname=txtLastName.getText();

String checkmidname=txtMidName.getText();

String checkfather=txtFather.getText();

String checkmother=txtMother.getText();

String checkeid=txtEID.getText();

if(checkfname.matches(".\*\\d.\*")){

JOptionPane.showMessageDialog(null,"Invalid input in First Name","Warning",JOptionPane.WARNING\_MESSAGE);

checkerfnamet=1;

} else{

checkerfnamet=0;

}

if(checklastname.matches(".\*\\d.\*")){

JOptionPane.showMessageDialog(null,"Invalid input in Last Name","Warning",JOptionPane.WARNING\_MESSAGE);

checkerlastnamet=1;

} else{

checkerlastnamet=0;

}

if(checkmidname.matches(".\*\\d.\*")){

JOptionPane.showMessageDialog(null,"Invalid input in Mid Name","Warning",JOptionPane.WARNING\_MESSAGE);

checkermidnamet=1;

} else{

checkermidnamet=0;

}

if(checkmother.matches(".\*\\d.\*")){

JOptionPane.showMessageDialog(null,"Invalid input in Mother Name","Warning",JOptionPane.WARNING\_MESSAGE);

checkermothert=1;

} else{

checkermothert=0;

}

if(checkfather.matches(".\*\\d.\*")){

JOptionPane.showMessageDialog(null,"Invalid input in Father Name","Warning",JOptionPane.WARNING\_MESSAGE);

checkerfathert=1;

} else{

checkerfathert=0;

}

if(checkeid.matches(".\*\\d.\*")){

JOptionPane.showMessageDialog(null,"Invalid input in eid Name","Warning",JOptionPane.WARNING\_MESSAGE);

checkereidt=1;

} else{

checkereidt=0;

}

if(txtMidName.getText().length()>1){

JOptionPane.showMessageDialog(null,"Set Middle initial to one character only!","Warning",JOptionPane.WARNING\_MESSAGE);

checkermidnamet=1;

}

if(checkerfnamet==1||checkermidnamet==1||checkerlastnamet==1||idtChecker==1||agetChecker==1||aadhartChecker==1||checkerfathert==1||checkermothert==1||dojtChecker==1||pntChecker==1||checkereidt==1)

{

}

//----------------------------End of Check--------------------

else{

String idx = String.valueOf(idt);

String fname=txtFName.getText();

String midname=txtMidName.getText();

String lastname=txtLastName.getText();

String father=txtFather.getText();

String mother=txtMother.getText();

String eid= txtEID.getText();

String city=txtCity.getText();

String gender=txtGender.getText();

//String status=txtStatus.getText();

String diploma=txtRDiploma.getText();

String btech=txtRBTECH.getText();

String it=txtRIT.getText();

String cse=txtRCSE.getText();

String csse=txtRCSSE.getText();

String eee=txtREEE.getText();

//String birth=txtRBirth.getText();

String firstyear=txtFIRSTYEAR.getText();

String secondyear=txtSECONDYEAR.getText();

String thirdyear=txtTHIRDYEAR.getText();

String fourthyear=txtFOURTHYEAR.getText();

try{

if (!gender.equals("")&&!idx.equals("") && !fname.equals("")&& aadhart!=0 && !mother.equals("") && !father.equals("")&& !midname.equals("") && !lastname.equals("") && aget!=0&& !city.equals("")&& dojt!=0&& pnt!=0&& !eid.equals("")){

st=cn.createStatement();

PreparedStatement ps = cn.prepareStatement("UPDATE Login SET First = '" + fname + "',father='" + father+ "',mother='"+ mother+"',aadhar='" + aadhart + "',MI = '" + midname + "',Last= '" + lastname+"',Age = '"+aget+"',Gender='"+gender+"',City='"+city+"',Diploma='"+diploma+"',btech='"+btech+"',IT='"+it+"',WHERE ID = '" + idt + "',cse='"+cse+"',csse='"+csse+"',eee='"+eee+"',firstyear='"+firstyear+"',secondyear='"+secondyear+"',thirdyear='"+thirdyear+"',fourthyear='"+fourthyear+"',doj='"+dojt+"',eid='"+eid+"',pn='"+pnt+"'");

ps.executeUpdate();

JOptionPane.showMessageDialog(null,"Account has been successfully updated.","Student Information System",JOptionPane.INFORMATION\_MESSAGE);

txtID.setEnabled(false);

txtFName.setEnabled(false);

txtMidName.setEnabled(false);

txtLastName.setEnabled(false);

txtAge.setEnabled(false);

txtFather.setEnabled(false);

txtMother.setEnabled(false);

txtAadhar.setEnabled(false);

txtCity.setEnabled(false);

//txtStatus.setEnabled(false);

txtGender.setEnabled(false);

txtRDiploma.setEnabled(false);

txtRBTECH.setEnabled(false);

txtRIT.setEnabled(false);

txtRCSE.setEnabled(false);

txtRCSSE.setEnabled(false);

txtREEE.setEnabled(false);

//txtRBirth.setEnabled(false);

txtFIRSTYEAR.setEnabled(false);

txtSECONDYEAR.setEnabled(false);

txtTHIRDYEAR.setEnabled(false);

txtFOURTHYEAR.setEnabled(false);

//CBBirth.setEnabled(false);

CBDiploma.setEnabled(false);

CBBTECH.setEnabled(false);

CBIT.setEnabled(false);

CBCSE.setEnabled(false);

CBCSSE.setEnabled(false);

CBEEE.setEnabled(false);

CBFIRSTYEAR.setEnabled(false);

CBSECONDYEAR.setEnabled(false);

CBTHIRDYEAR.setEnabled(false);

CBFOURTHYEAR.setEnabled(false);

txtID.requestFocus(true);

clear();

st.close();

}

else{

JOptionPane.showMessageDialog(null,"Please Fill Up The Empty Fields","Warning",JOptionPane.WARNING\_MESSAGE);//please Fill Up the Empty Fields!

}

}catch (SQLException y){

JOptionPane.showMessageDialog(null,"Unable to update!.","Student Information System",JOptionPane.ERROR\_MESSAGE);

System.out.println("SQL Error" + y.toString() + " " +y.getErrorCode() + " " + y.getSQLState());

if (!gender.equals("")&&!idx.equals("") && !fname.equals("")&& aadhart!=0 && !mother.equals("") && !father.equals("")&& !midname.equals("") && !lastname.equals("") && aget!=0&& !city.equals("")&& dojt!=0&& pnt!=0&& !eid.equals("")) {

}

else{

delete.setEnabled(false);

update.setEnabled(false);

save.setEnabled(false);

}

}

}

}

if(source==delete){

if(!txtID.getText().equalsIgnoreCase("")){

try{

PreparedStatement ps = cn.prepareStatement("DELETE FROM Login WHERE ID ='"+ txtID.getText() + "'");

ps.executeUpdate();

JOptionPane.showMessageDialog(null,"Account has been successfully deleted.","Payroll System: User settings ",JOptionPane.INFORMATION\_MESSAGE);

txtID.setEnabled(false);

txtFName.setEnabled(false);

txtMidName.setEnabled(false);

txtLastName.setEnabled(false);

txtAge.setEnabled(false);

txtFather.setEnabled(false);

txtMother.setEnabled(false);

txtAadhar.setEnabled(false);

txtDOJ.setEnabled(false);

txtEID.setEnabled(false);

txtPN.setEnabled(false);

txtCity.setEnabled(false);

//txtStatus.setEnabled(false);

txtGender.setEnabled(false);

txtRDiploma.setEnabled(false);

txtRBTECH.setEnabled(false);

txtRIT.setEnabled(false);

txtRCSE.setEnabled(false);

txtRCSSE.setEnabled(false);

txtREEE.setEnabled(false);

//txtRBirth.setEnabled(false);

txtFIRSTYEAR.setEnabled(false);

txtSECONDYEAR.setEnabled(false);

txtTHIRDYEAR.setEnabled(false);

txtFOURTHYEAR.setEnabled(false);

//CBBirth.setEnabled(false);

CBDiploma.setEnabled(false);

CBBTECH.setEnabled(false);

CBIT.setEnabled(false);

CBCSE.setEnabled(false);

CBCSSE.setEnabled(false);

CBEEE.setEnabled(false);

CBFIRSTYEAR.setEnabled(false);

CBSECONDYEAR.setEnabled(false);

CBTHIRDYEAR.setEnabled(false);

CBFOURTHYEAR.setEnabled(false);

txtID.requestFocus(true);

clear();

st.close();

}catch(SQLException s){

JOptionPane.showMessageDialog(null,"Unable to delete!.","Student Information System",JOptionPane.ERROR\_MESSAGE);

}

}

delete.setEnabled(false);

update.setEnabled(false);

save.setEnabled(false);

}

if(source==cancel){

save.setEnabled(false);

search.setEnabled(true);

New.setEnabled(true);

clear();

}

}

public void itemStateChanged (ItemEvent j) {

if(j.getSource().equals(RBFemale)){

txtGender.setText("female");

}

if(j.getSource().equals(RBMale)){

txtGender.setText("male");

}

}

public void valueChanged(ListSelectionEvent e){

}

public static void main (String args []){

JFrame frame= new studentInformationSystem();

frame.setSize(800,800);

frame.setVisible(true);

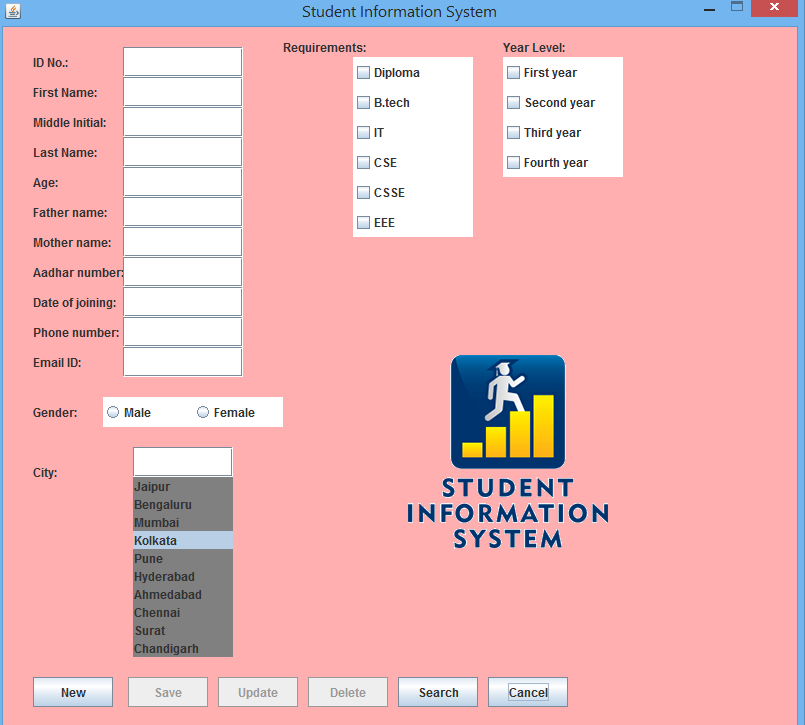
frame.setResizable(false);

}

}

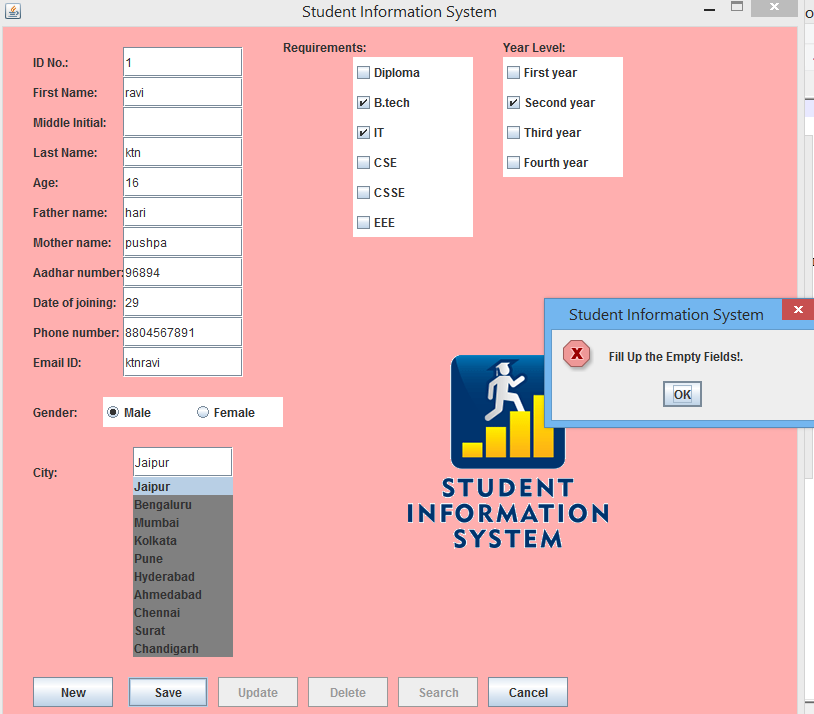
**RESULTS**

**Home Page:**



**Fig:1.Sreenshot on home page to create new student information**

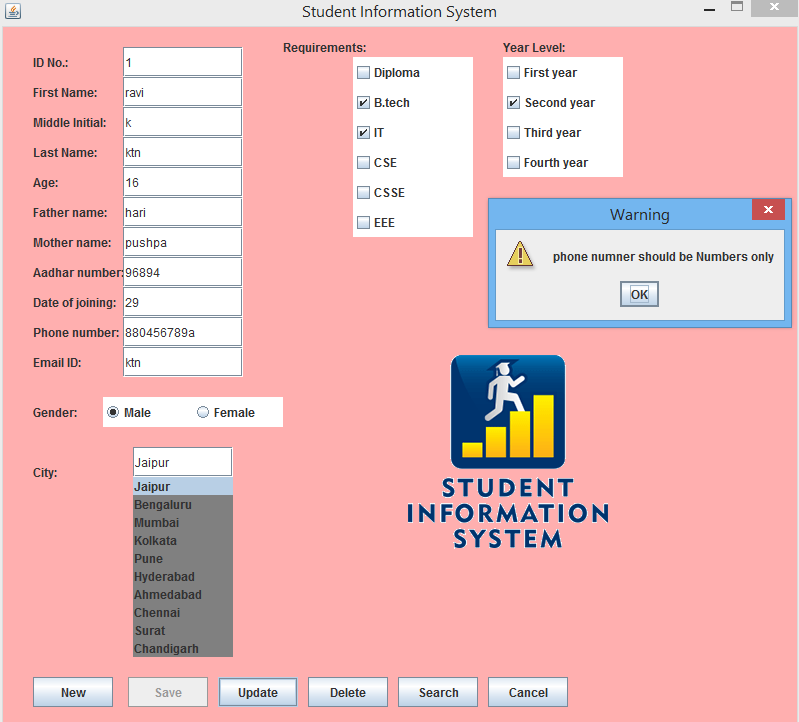
**Inserting Data:**



**Fig:2.Screenshot to show Error message on empty field.**

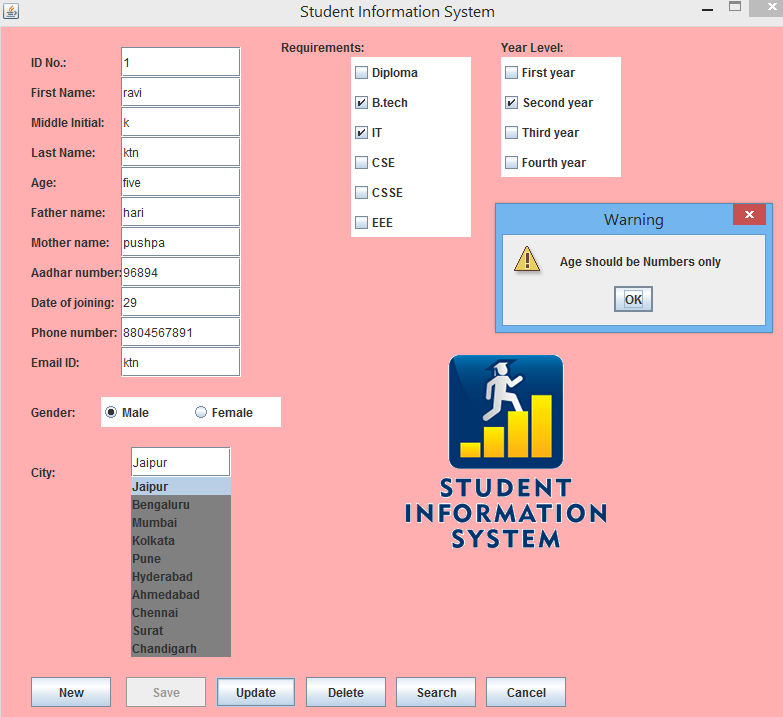
In above screenshot fig:2. according to the condition if given empty field ,

Ask you to fill the field.



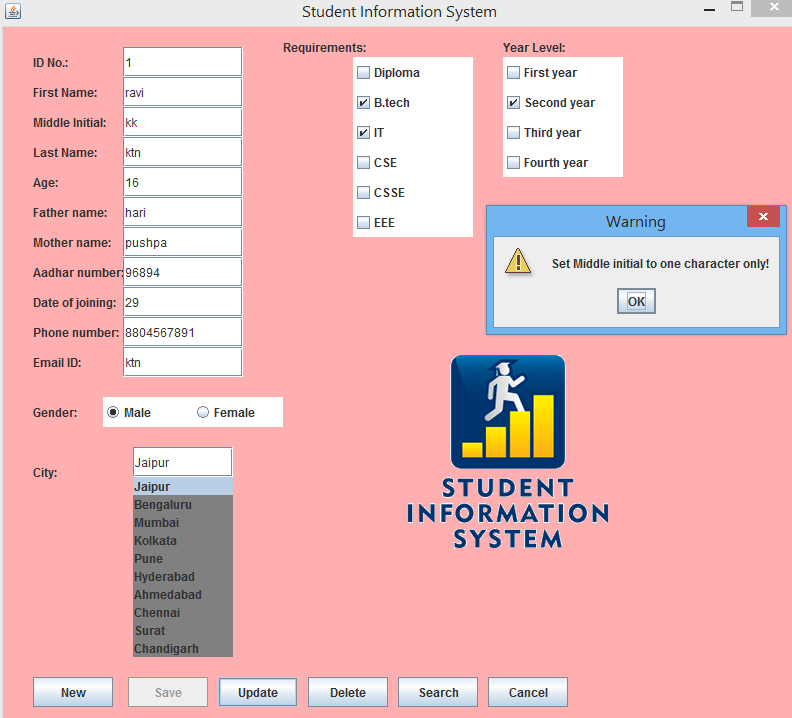
**Fig:3.Screenshot to show the warning message regarding phone number**

In above screenshot fig:3. Insertion of information is done according to the conditions and were phone number should only contains numeric values.



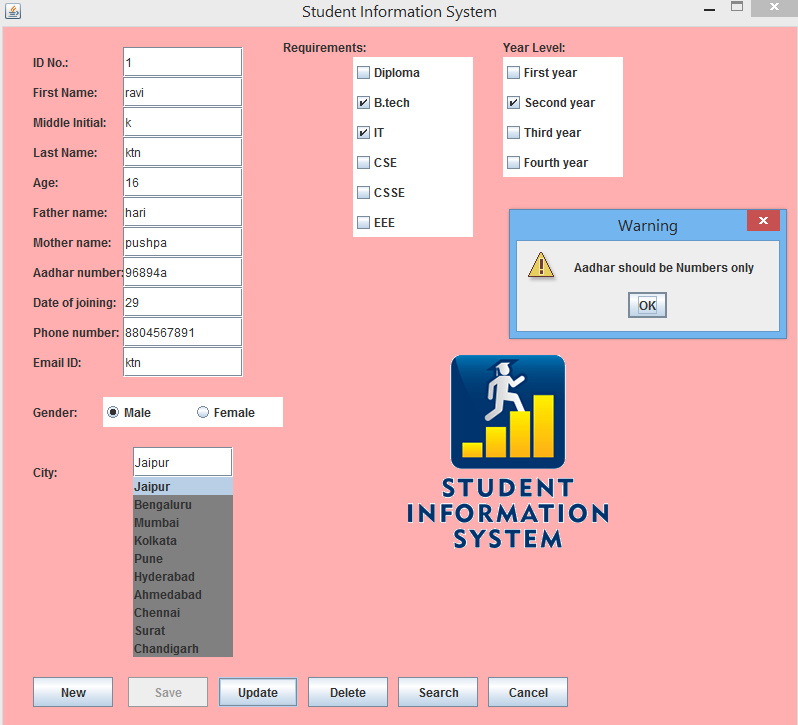
**Fig:4.Screenshot to show the warning message regarding Age**

In above screenshot fig:4. Insertion of information is done according to the conditions and were age should be entered in integers format only.



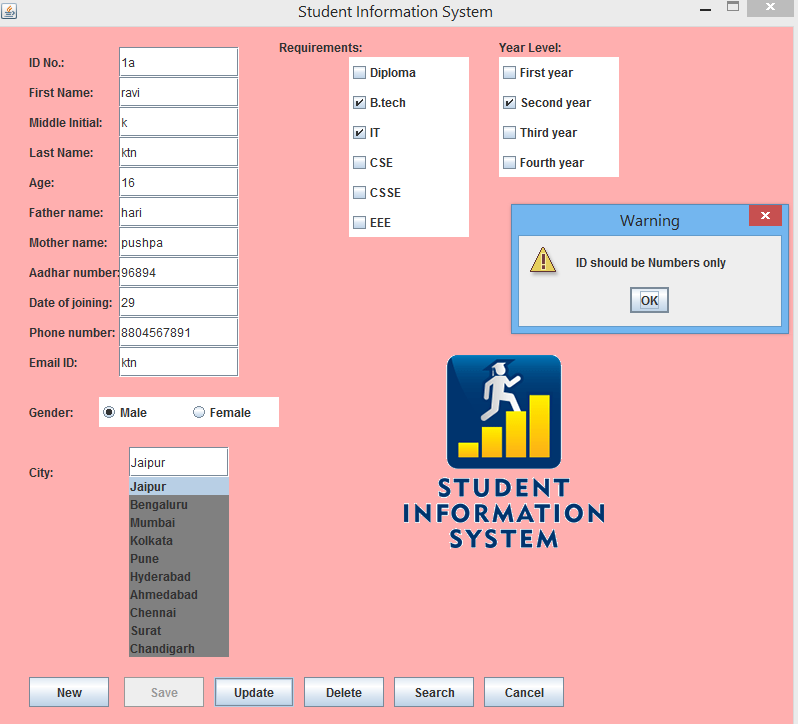
**Fig:5.Screenshot to show the Warning message regarding character length**

In above screenshot fig:5. According to the given condition the middle name should not exceed single character.



**Fig:6.Screenshot to show Warning message on Aadhar number entry**

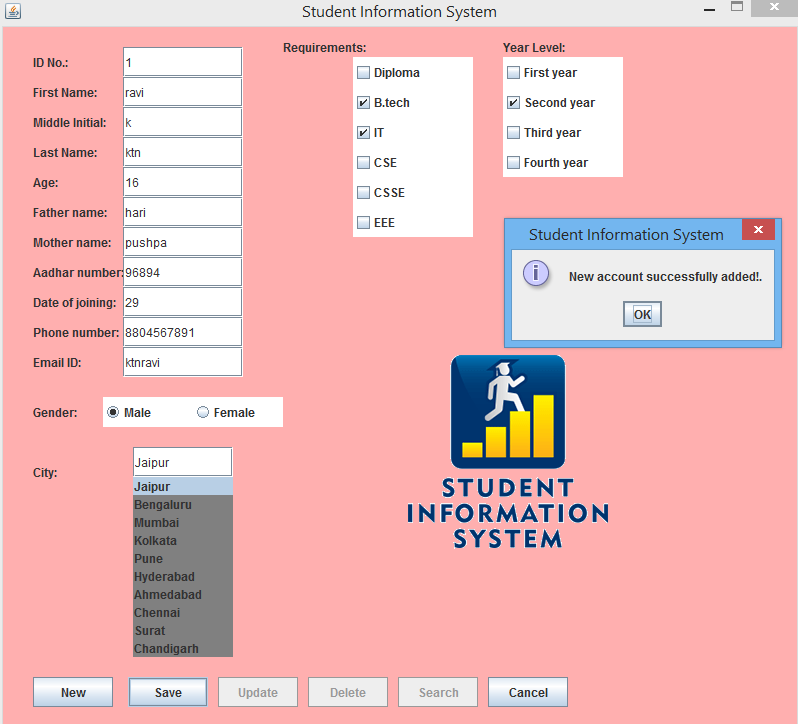
In above screenshot fig:6.according to the given condition Aadhar number should only contain integer values.



**Fig:7.Screenshot to show Warning message on ID**

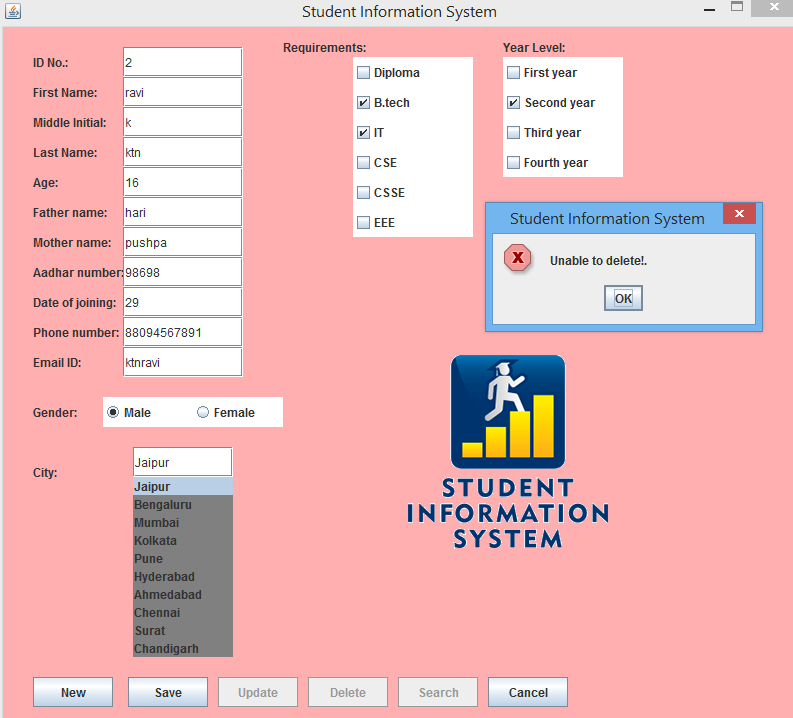
In above screenshot fig7: As per the given condition the Unique ID of each student should contain number format only.

**Saving new account:**



**Fig:8.Screenshot to show Successful creation of new account**

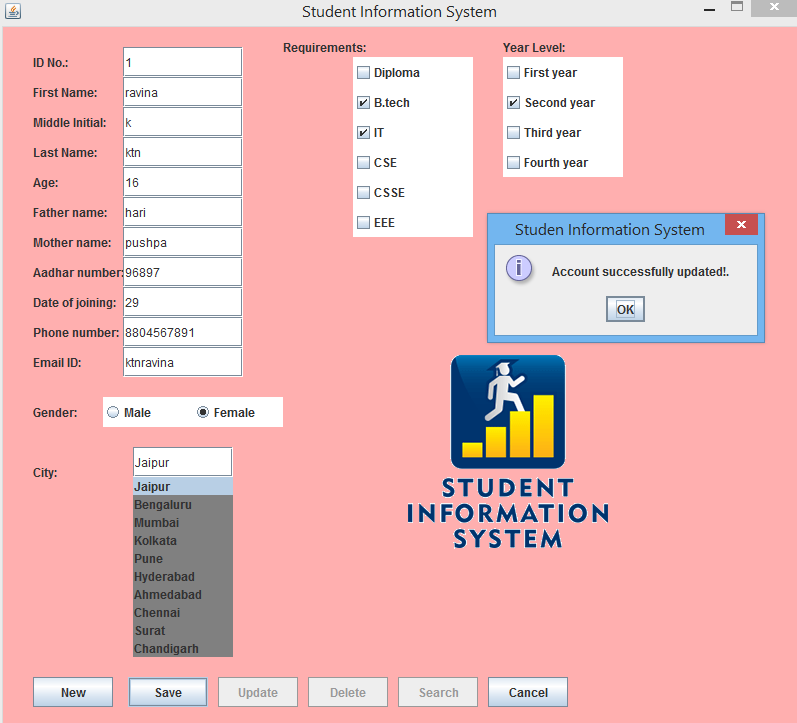
In above screenshot fig:8 After giving all valid inputs based on conditions the new student information is saved in database.



**Fig9:Screenshot to show Error message on deletion of account**

In above screenshot fig9: The data is unable to delete because of invalid ID number of student in database,as it should be given ID=1.

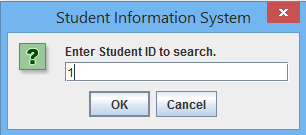
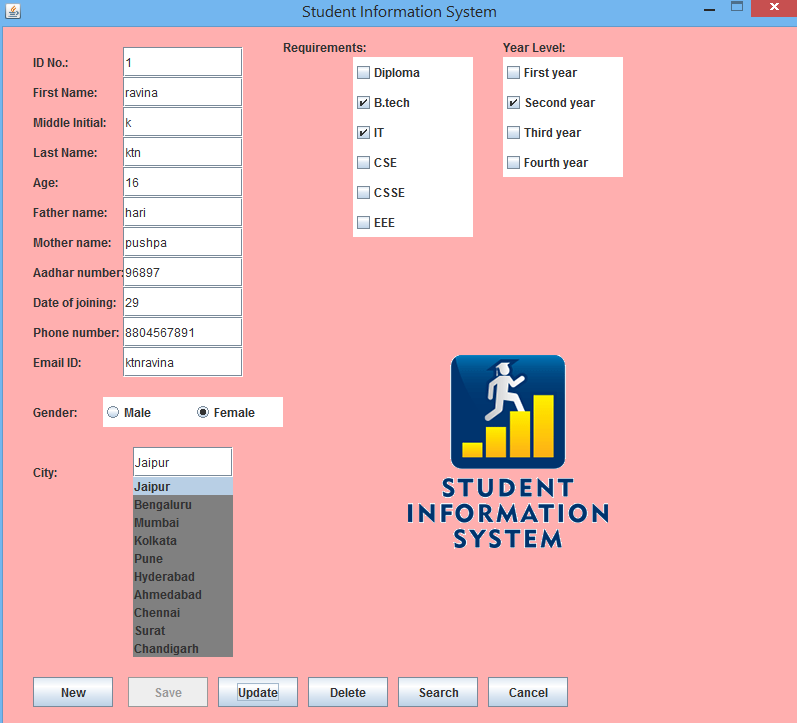
**Update:**



**Fig:10.Screenshot to show successful updation of previous account**

In above Sreenshot fig10: The information related to student of ID-1,get updated with the new information,which shows the successful updation.

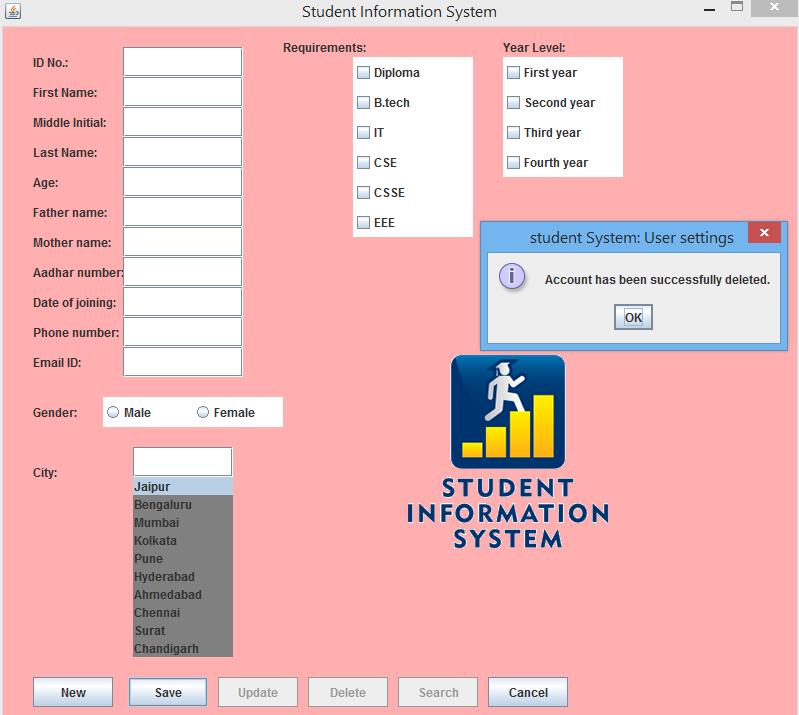
**Search:**



**Fig11:Screenshot to show the searched student information**

In above Screenshot fig11:One can get the student information , searching through unique ID of each student.

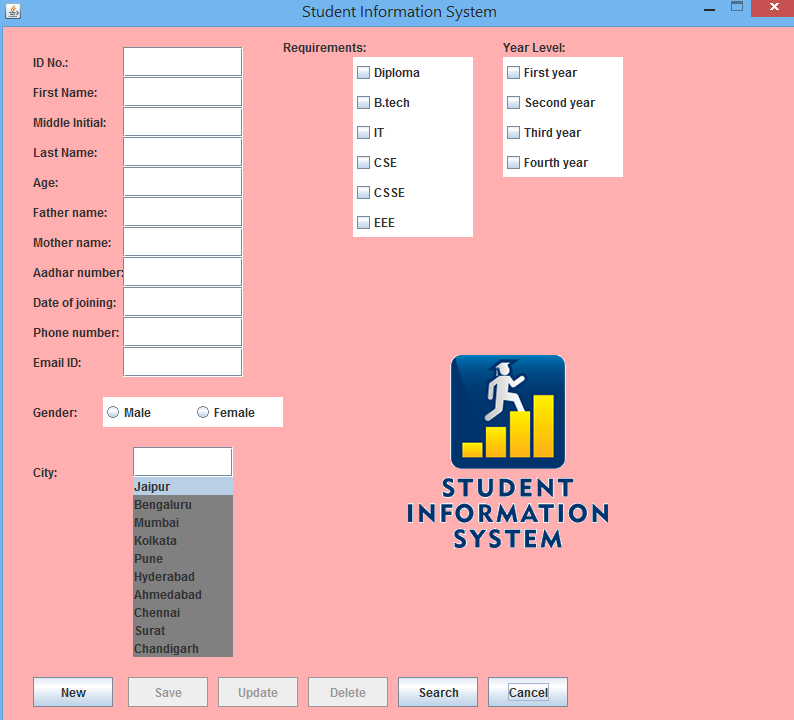
**Delete:**



**Fig:12.Screenshot to show the deletion of previous account**

In above screenshot fig:12 one can delete the student information from database by delete option using Unique ID of student.

**Cancel:**



**Fig13:Screenshot to show cancel option to start Reinsertion**