Institute Management System (IMS)

A PROJECT REPORT

Submitted by

Mustafiz Sharique

Under the guidance of
Mr. Mohammad Islam
Assistant Professor, Department of CS & IT

in partial fulfillment for the award of the degree

of

B.Tech

IN

COMPUTER SCIENCE

MAULANA AZAD NATIONAL URDU UNIVERSITY
Hyderabad

May 2017

MAULANA AZAD NATIONAL URDU UNIVERSITY

BONAFIDE CERTIFICATE

Certified that this project report **“Institute Management System (IMS)”** is the bonafide work of “ **Mustafiz Sharique** ” who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
DR. ABDUL WAHID	**MR. MOHAMMAD ISLAM**
	(Supervisor)
HEAD OF THE DEPARTMENT	**ASSISTANT PROFESSOR**
DEPARTMENT OF CS& IT	**DEPARTMENT OF CS&IT**

TABLE OF CONTENTS

Abstract

Institute Management System (IMS) is an application developed in Java that will be used for management of different departments in various institutes like schools, colleges and universities. IMS will have different access level for different categories of people like general users, special users, and administrators. IMS will be programmed in Java and will use SQL for database management and can also be connected to cloud database storage so as to make the data accessible from any location on the globe. IMS will be developed using an Integrated Development Environment (IDE). We are using the Netbeans IDE. It will consist of various modules to deal with different sections that can be there within an institute. There will be different tables in database to store data about students, teachers, and parents. There will also be details of fee and examination results in the database.

IMS will be developed using the Iterative model of SDLC (Software Development Life Cycle) i.e. new features and module will be added after each development cycle depending upon the need and requirements of the IMS user. We will start with a basic version of IMS that is able to store data about users like students and teachers and then keep on adding new users, new features, and new databases.

In IMS there will be user profile that will be accessed and edited by the respective users. There will be also be whole tables of data that will be accessed by the admin to maintain the database properly. Admin will be able to access and edit all the data stored with the IMS.

Students will be able to check their results, fee, attendance and time-tables. A teacher will be able to check the result of students, their attendance and time-tables. A parent will be able to do the same for their ward.

IMS has been developed keeping in view the ease for managing the data of an Institute from administrative point of view. The focus of IMS is to make administration of institutes easier. So, modules will be developed mostly for the said motive. This is also one of the reasons why we have chosen the desktop app for developing IMS.

We have used Java Swing to implement the GUI on Java platform. Java Swing provides all the necessary components like frames, buttons, labels, text fields, tables etc. that are required to make an app with good level of graphics.

List of Figures

Chapter 1: Introduction

- The idea of developing Institute Management System (IMS) was conceived after looking at the conventional way through which different institutes work and co-ordinate within their system and thus a need for a new centralized system that will help users like students, teachers and administrators to easily co-ordinate with each other and access the resources from the comfort of their personal devices.
- IMS will have separate databases for separate departments of institute to make management easy. Different level of access will be granted to teachers, students, administrators and other users will decide their accessibility to data and the data they can modify/update.
- Users will be able to see their profiles and edit them. Administrator will be able to edit an access all the data that is stored with IMS.

The users of IMS will be as follows:

- **Administrators**
- **Students**
- **Teachers**
- **Parents/Guardians**

The modules that will be developed inside IMS are:

- **User Management System.**
- **Fee Management system.**
- **Examination Management System.**
- **Student Management System.**
- **Staff Management System.**
- **Attendance/Time Table Management System.**
- **Admin Module**
- **Parent Management System**

Tools Used for Development:

- Netbeans IDE, Java, Oracle/MySql, HTML.

Software Requirements:

- Windows 7/8/10, Java Virtual Machine

Hardware Requirements:

- Processor: Minimum 1.2 GHz Intel Pentium IV or equivalent
- Memory: Minimum 2 GB
- Disk space: Minimum 1 GB of free disk space

Chapter 2:

Entity Relationship (ER) & Data Flow Diagrams (DFD)

2.1) Entity Relationship (ER)

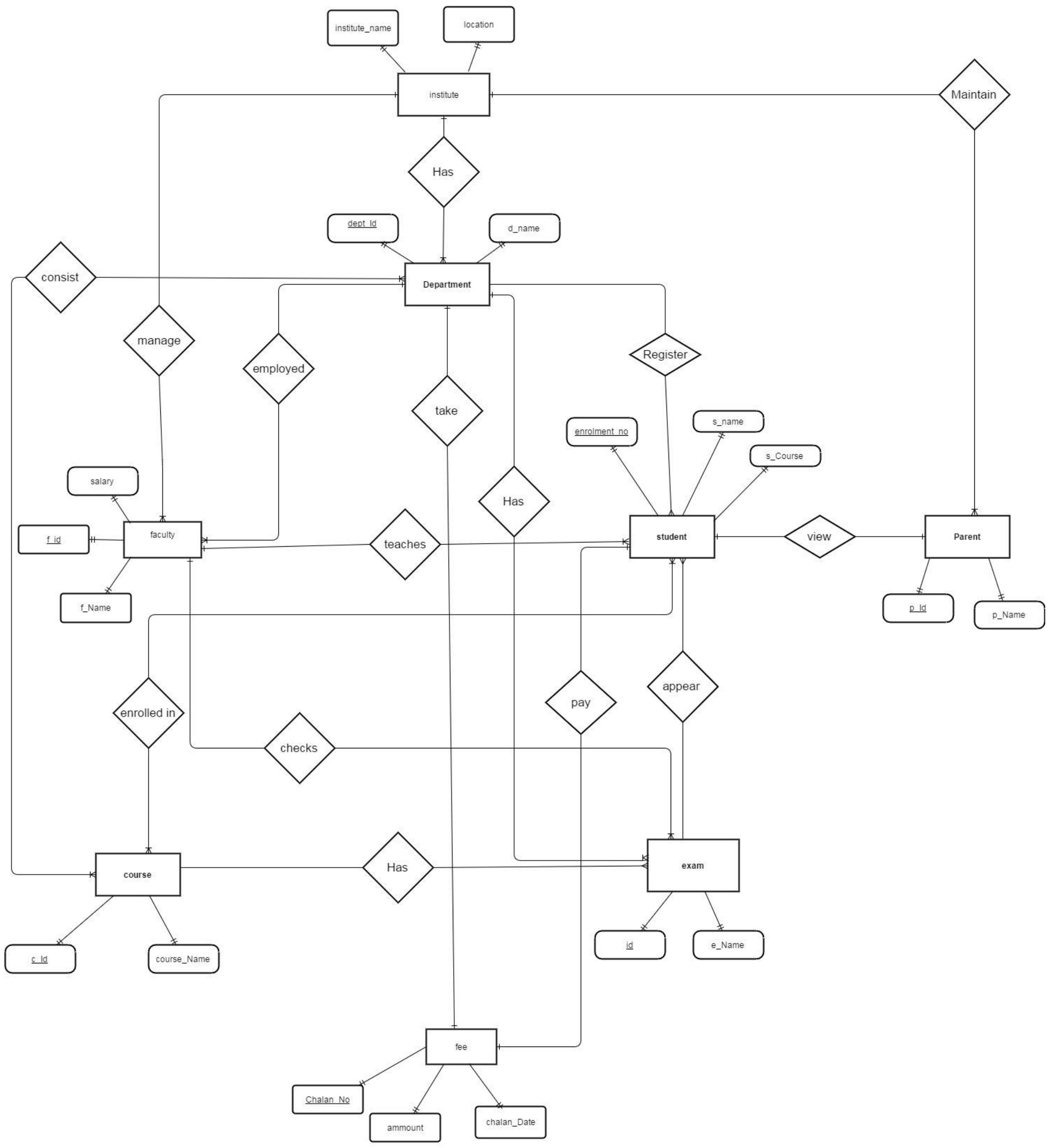


Figure 2.1

2.2) Data Flow Diagrams (DFD)

2.2.1) Level 0 DFD: Context Level

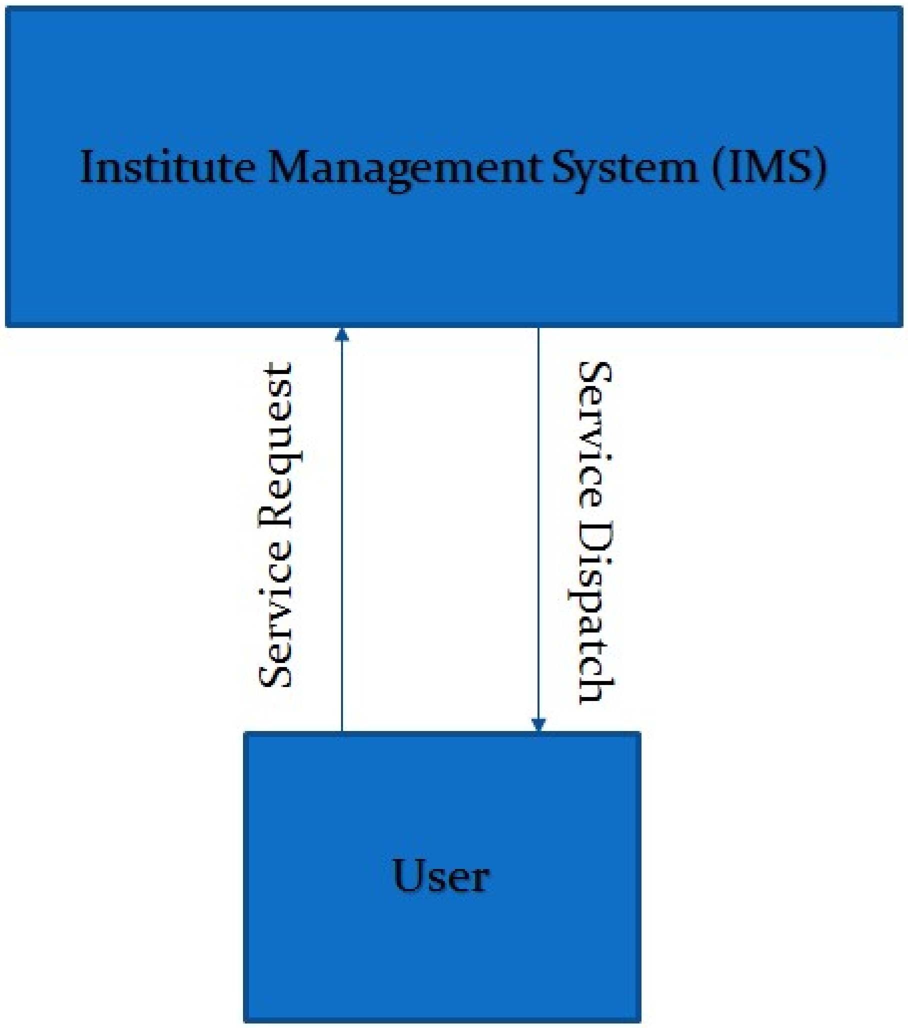


Figure 2.2.1

2.2.2) Level 1 DFD

- **DFD 1.1**

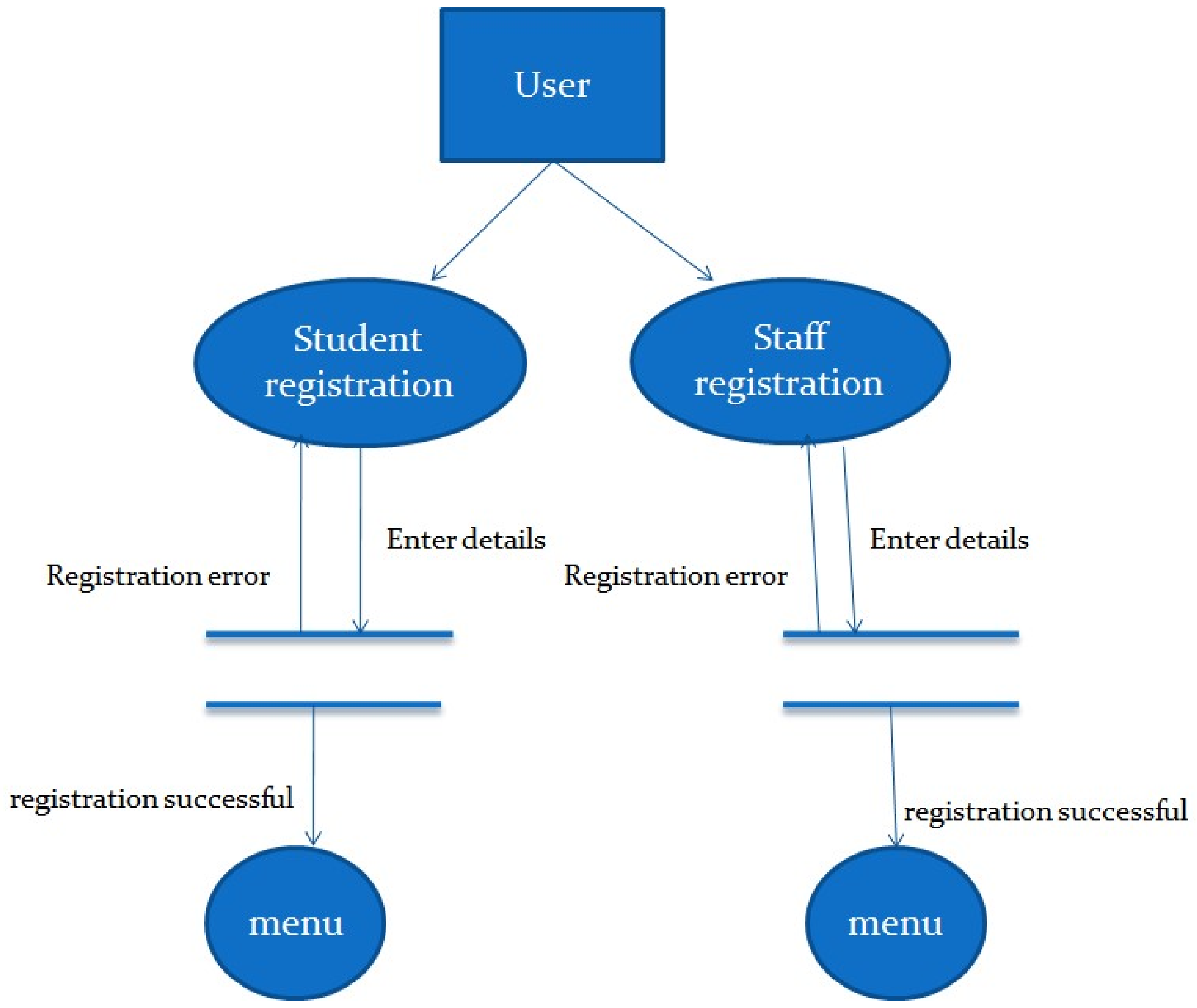

User
Student registration
Staff registration
Enter details
Registration error
Enter details
Registration error
registration successful
registration successful
menu
menu

Figure 2.2.2

- **DFD 1.2**

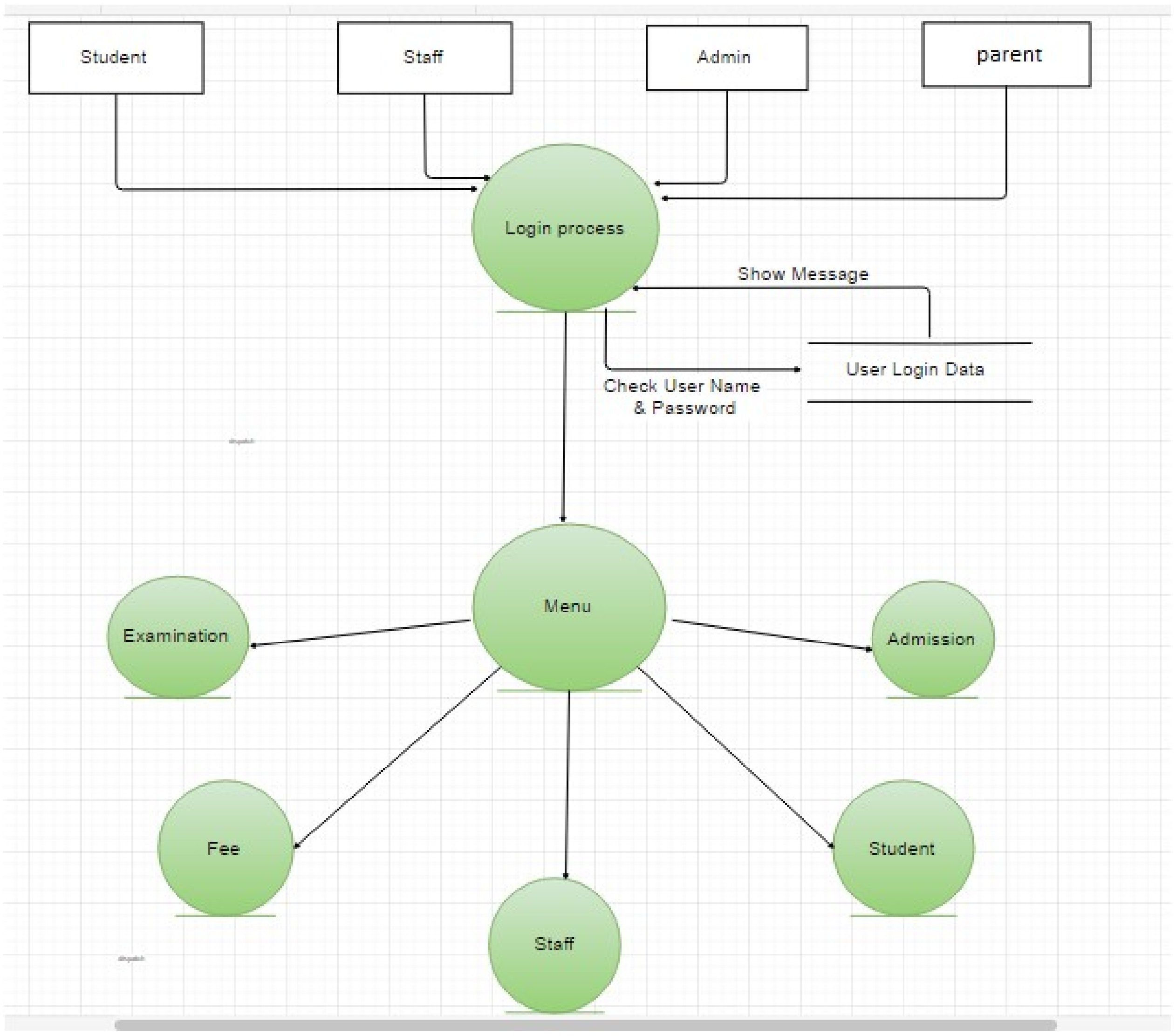


Figure 2.2.3

2.2.3) Level 2 DFD

- **DFD 2.1**

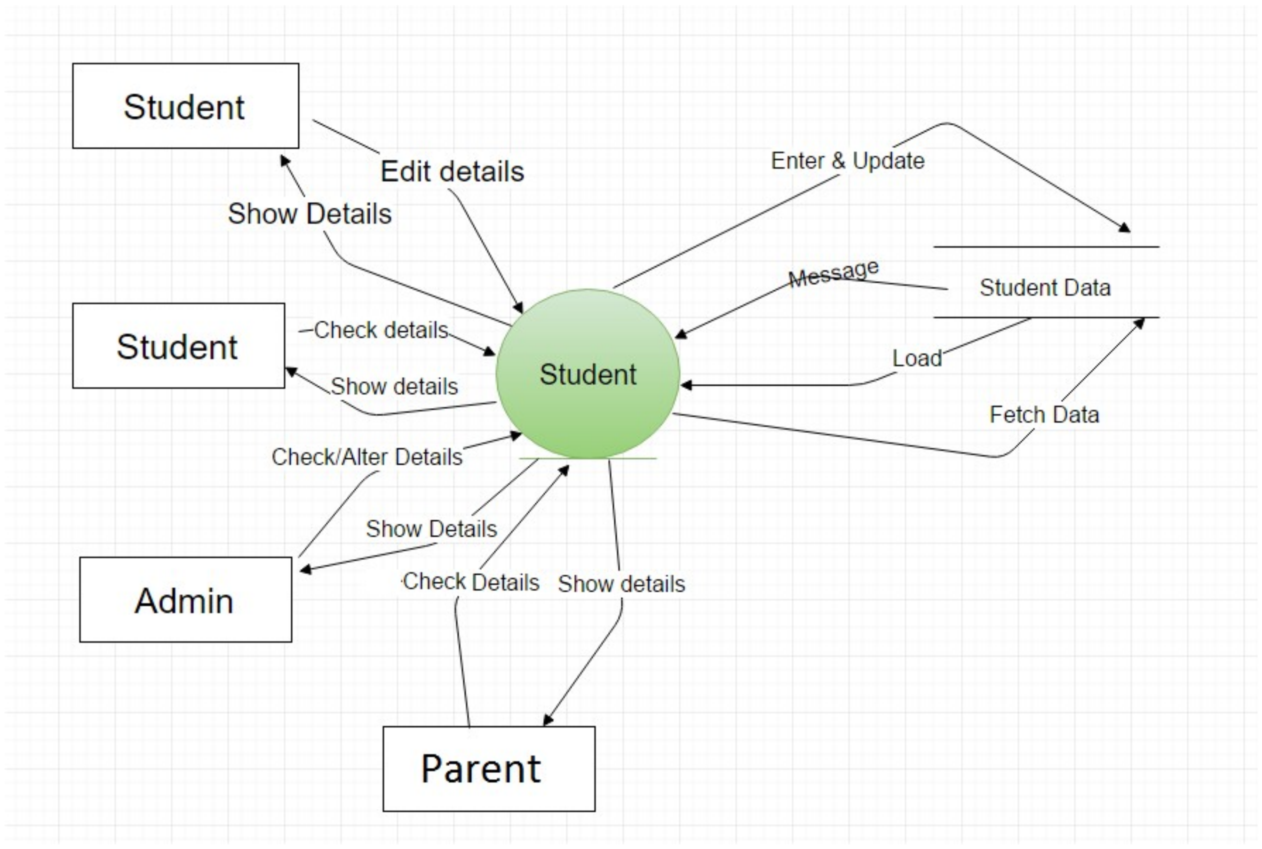


Figure 2.2.4

- **DFD 2.2**

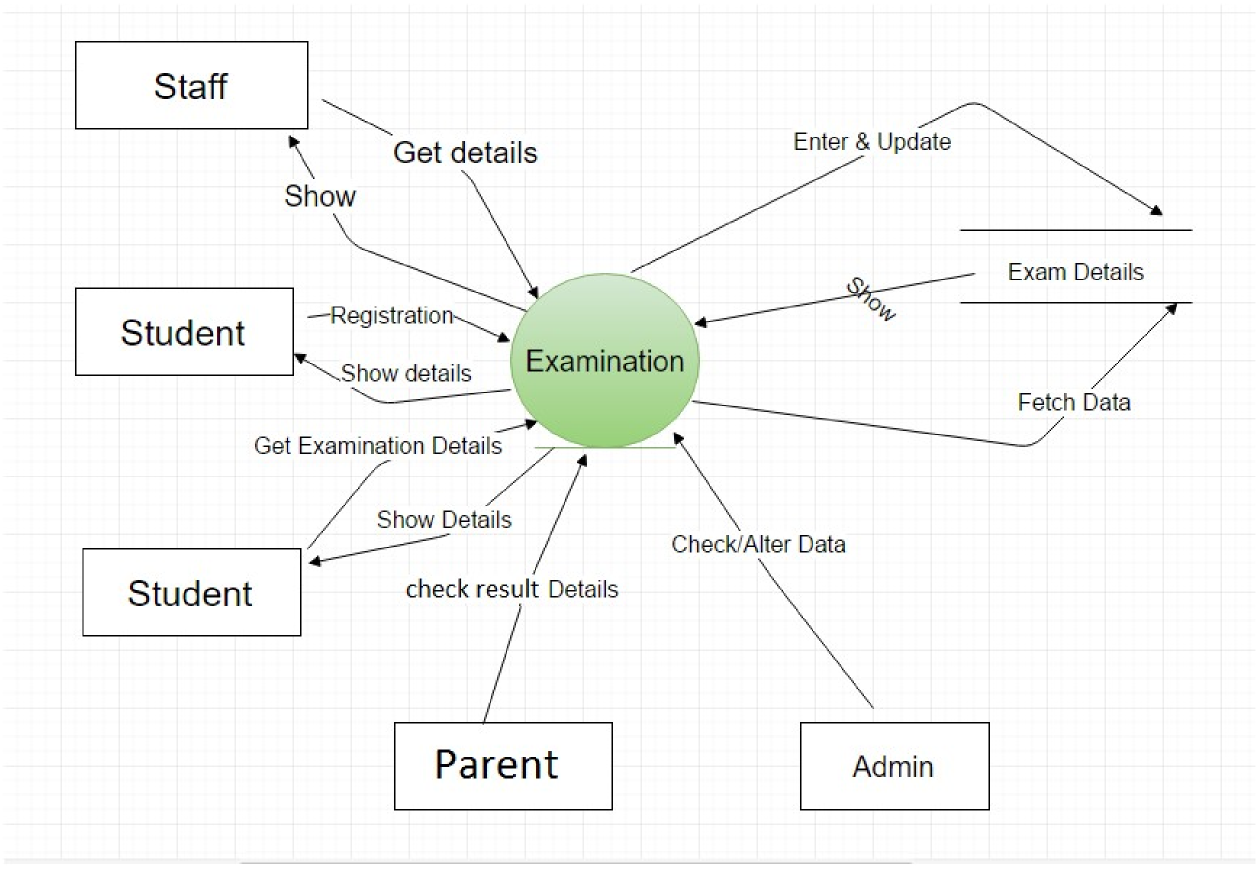


Figure 2.2.5

- **DFD 2.3**

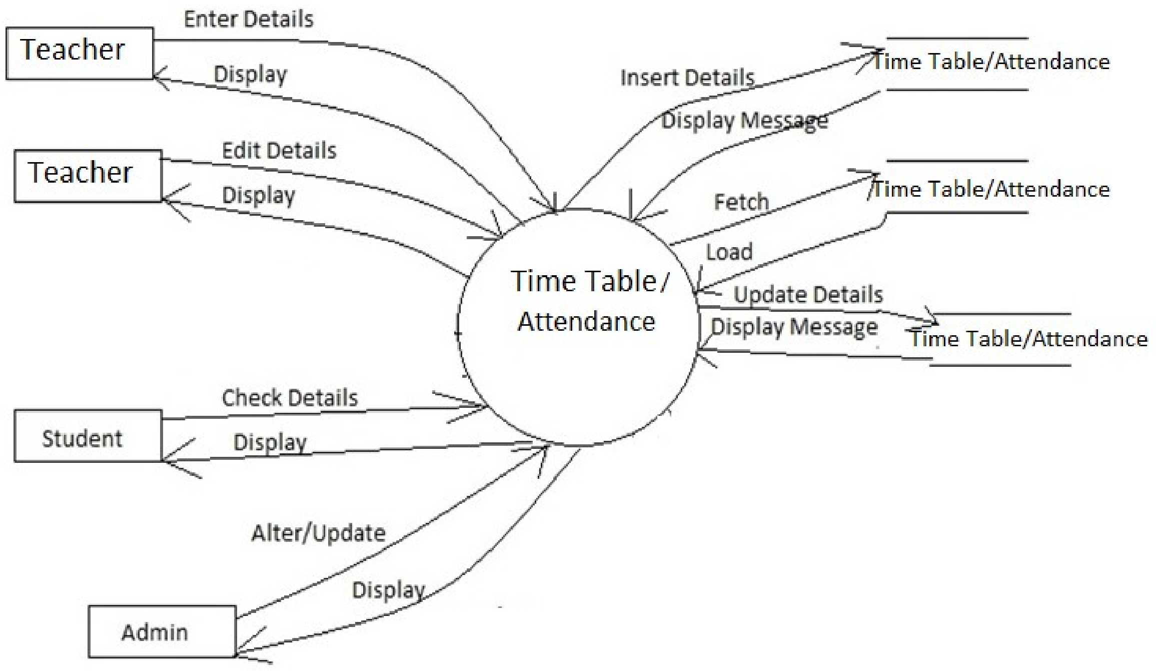


Figure 2.2.6

- **DFD 2.4**

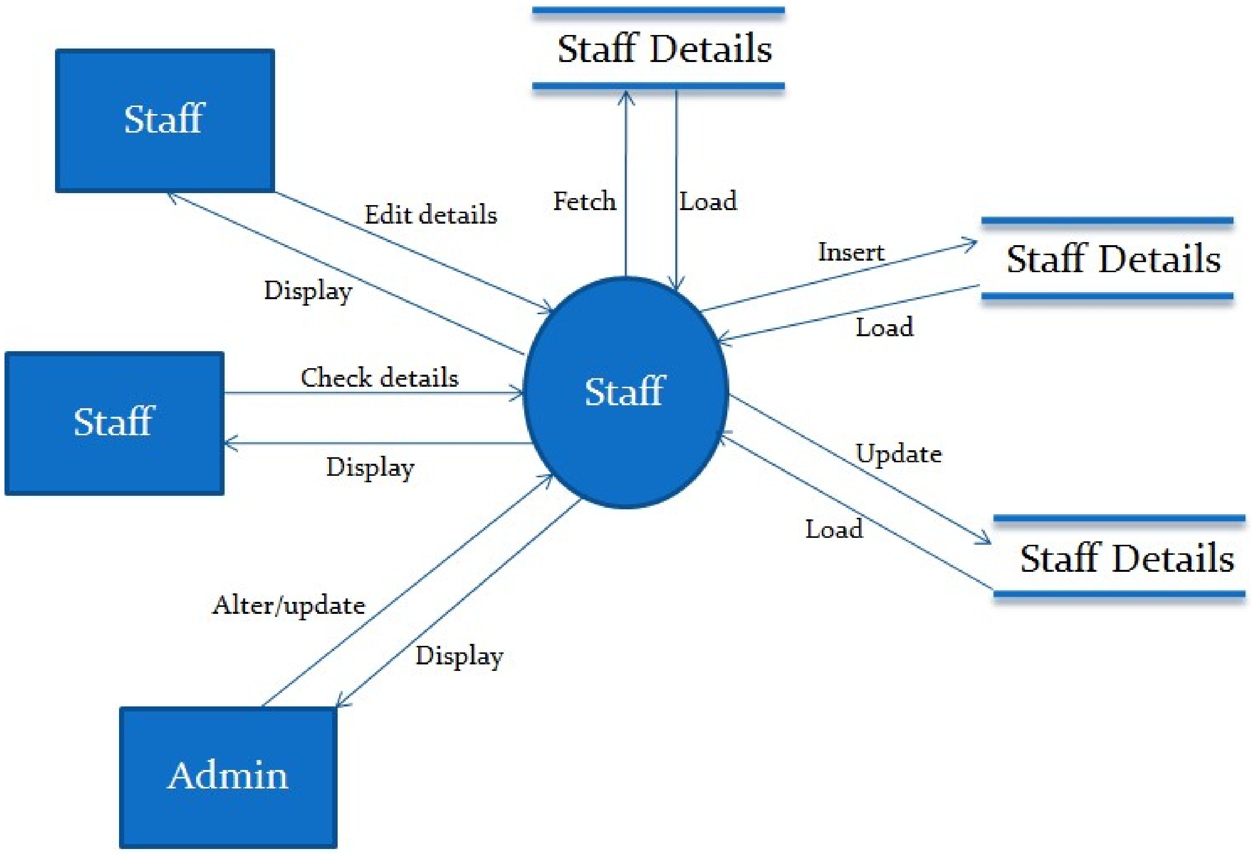


Figure 2.2.7

- **DFD 2.5**

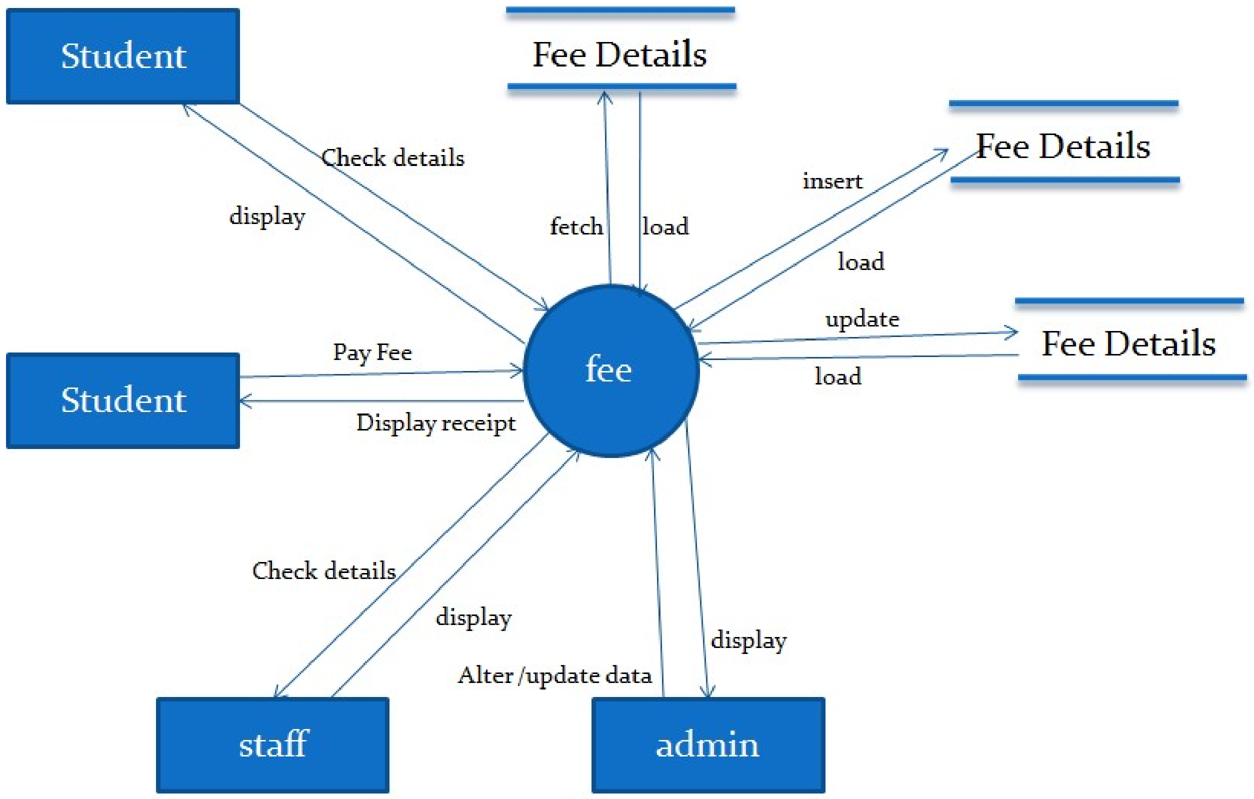


Figure 2.2.8

Chapter 3: User Management System

3.1) Login

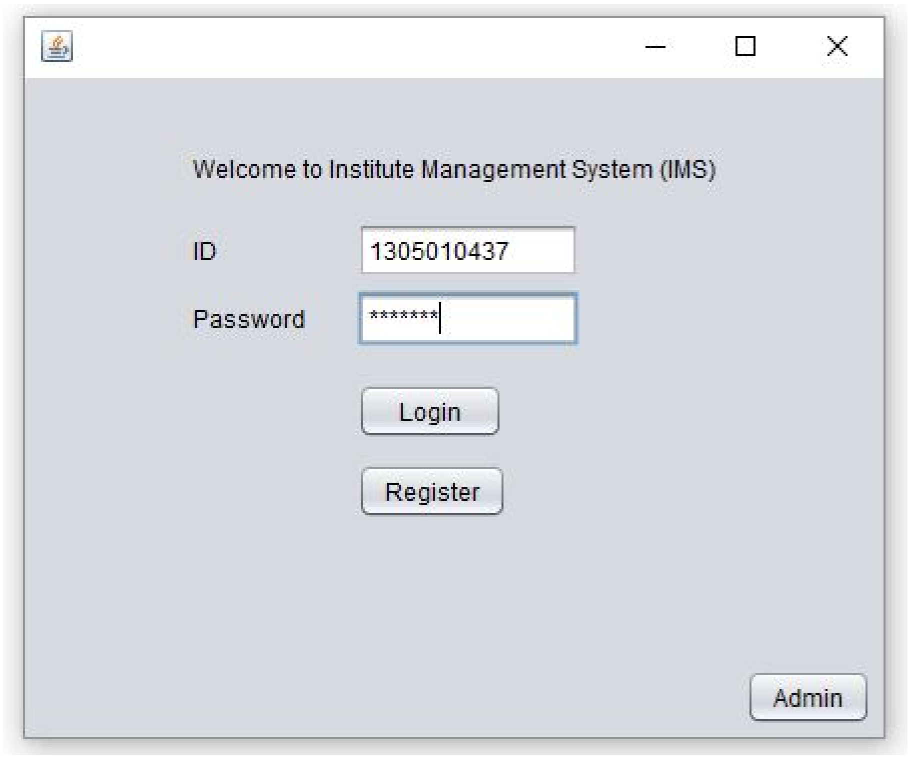


Figure 3.1

The first frame that opens up on executing the IMS will be the Login frame. Here a user will enter his login details and those details will be validated with the existing data in the database. If the data entered matched with it then the user will successfully login else he/she will be duly notified about the login failure.

The user will click on the register button to register his/her details if he/she is using the application for the first time.

The administrator will click on the admin button to login to the application with admin privileges.

3.1.1) The code used for the Login button is follows:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    try
    {
            String sql= "select * from register where id=? and password=?";
          PreparedStatement pst=conn.prepareStatement(sql);
          pst.setString(1,jTextField1.getText());

          pst.setString(2,jPasswordField1.getText());
           ResultSet rs1=pst.executeQuery();
          if(rs1.next())
          {
            x=rs1.getString(1);
            System.out.print("Login Successful");
            new StudentProfile().setVisible(true);
             this.dispose();
           }
          else {
        System.out.println("Invalid Username/Password");
        JOptionPane.showMessageDialog(null, "Enter Correct Values"); }
         }
        catch(Exception e){
             JOptionPane.showMessageDialog(null, e);
          }
  }
```

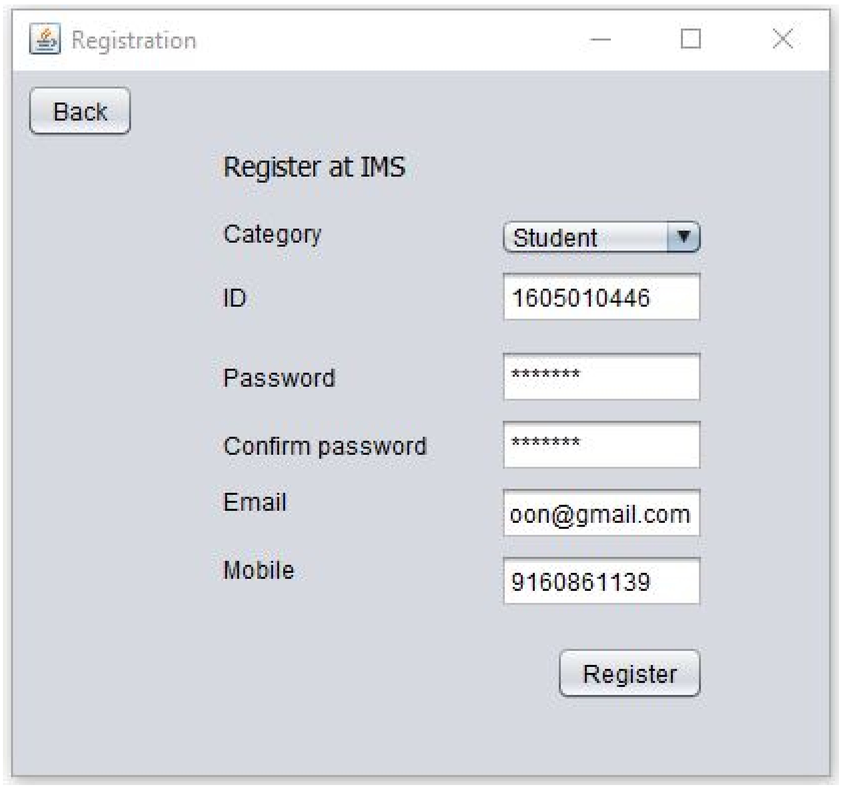
3.1.2) The code used for the Register button is follows:

```
      private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

      new Register().setVisible(true);

              this.dispose();

  }
```

3.1.3) The code used for the Admin button is follows:

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

    String adminid= JOptionPane .showInputDialog("Admin Username");
            if (adminid.equals("0000"))
            {
               JPasswordField pf = new JPasswordField();
               int adminpass = JOptionPane.showConfirmDialog(null, pf,
"Enter Password", JOptionPane.OK_CANCEL_OPTION,
JOptionPane.PLAIN_MESSAGE);
               String password = new String(pf.getPassword());
               System.out.println(pf);
            if ((adminpass == JOptionPane.OK_OPTION) &&
(password.equals("sahabji")))
            {
             new Admin().setVisible(true);
             this.dispose(); }
            else JOptionPane.showMessageDialog(null, "Incorrect
Password");
            }
            else {
                JOptionPane.showMessageDialog(null, "Incorrect Username");
      }     new Admin().setVisible(true);
             this.dispose();
            // TODO add your handling code here:
}
```

3.2) Register



Register at IMS

Category	Student
ID	1605010446
Password	*******
Confirm password	*******
Email	oon@gmail.com
Mobile	9160861139

Figure 3.2

In the Register frame a new user of any of the three categories: Student, Teacher, Parent will enter his/her details as given in the form and get registered with IMS. The details entered in the form will be stored in the register table in the database. After successful registration, the user will be will have to go to the login page and login there with details used for registration.

3.2.1) The code under the combo box used for selecting category is as follows:

```
private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {
     if (jComboBox1.getSelectedIndex()==0)
    {jLabel2.setText("Student ID");
    jTextField1.setText("");
    }

    if (jComboBox1.getSelectedIndex()==1)
    {
      jLabel2.setText("Employee ID");
      jTextField1.setText("");
    }

    if (jComboBox1.getSelectedIndex()==2)
    {
       jLabel2.setText("Student ID");
     jTextField1.setText("Your ward's ID");
       }
    }
```

3.2.2) The code under the Register button is as follows:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
conn=ConnClass.ConnectDB();
 try {
     PreparedStatement pst = null;
          if (jComboBox1.getSelectedIndex()==0)
    {pst=conn.prepareStatement("insert into register values(?,?,?,?,?)");
    }

    if (jComboBox1.getSelectedIndex()==1)
    {
      pst=conn.prepareStatement("insert into eregister values(?,?,?,?,?)");
    }

    if (jComboBox1.getSelectedIndex()==2)
    {
     pst=conn.prepareStatement("insert into pregister values(?,?,?,?,?,?)");
       }

         String s;
         {
            if (jComboBox1.getSelectedIndex()==2)
              { s="P"+jTextField1.getText();  }
            else
               {s=jTextField1.getText();}
                  }
         pst.setString(1,s);
         pst.setString(2,jComboBox1.getItemAt(jComboBox1.getSelectedIndex()));
         pst.setString(3,jPasswordField1.getText());
         pst.setString(4,jTextField4.getText());
         pst.setString(5,jTextField5.getText());
         if (jComboBox1.getSelectedIndex()==2)
         {pst.setString(6,jTextField1.getText());}
         if(!jPasswordField1.getText().equals(jPasswordField2.getText()))
         {
            JOptionPane.showMessageDialog(null, "Password Fields doesn't match!");
              }
            else
         {
         int rs1=pst.executeUpdate();
         System.out.println(rs1+" records inserted");
         if(rs1==1)
          JOptionPane.showMessageDialog(null, "Registration Successful!");
         }
       }
      catch(Exception e){
          JOptionPane.showMessageDialog(null, e);
      }
          }
```

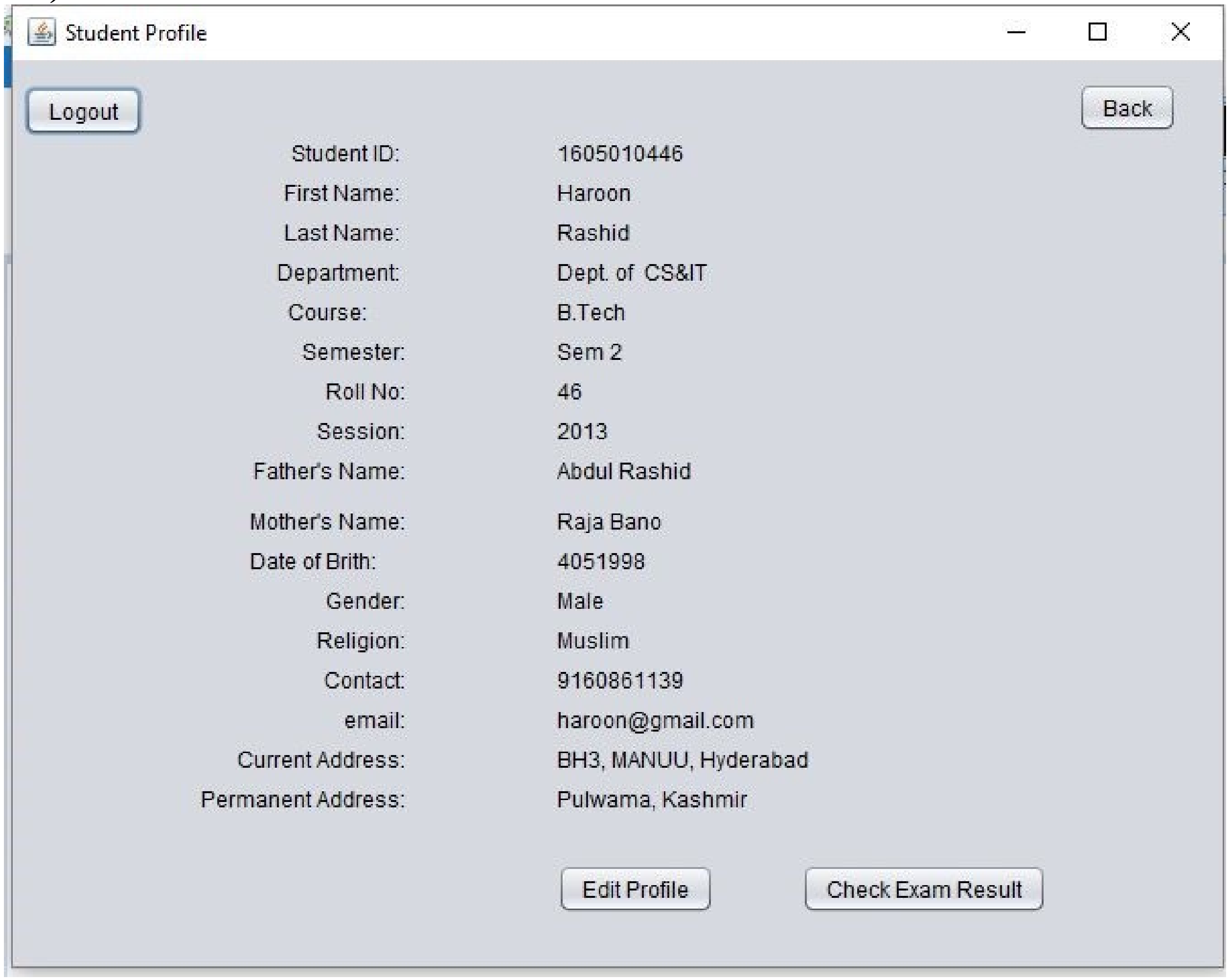
2.2.3) The code under the back button is as follows:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
       new Login().setVisible(true);
           // TODO add your handling code here:
   }
```

Chapter 4:

Student Management System

4.1) Student Profile.



Student ID:	1605010446
First Name:	Haroon
Last Name:	Rashid
Department:	Dept. of CS&IT
Course:	B.Tech
Semester:	Sem 2
Roll No:	46
Session:	2013
Father's Name:	Abdul Rashid
Mother's Name:	Raja Bano
Date of Brith:	4051998
Gender:	Male
Religion:	Muslim
Contact:	9160861139
email:	haroon@gmail.com
Current Address:	BH3, MANUU, Hyderabad
Permanent Address:	Pulwama, Kashmir

Figure 4.1

Student Profile will contain all the necessary details about the student that will be entered by the student after logging into the IMS for the first time.
After Entering the details the student will be also be able to Edit Profile.

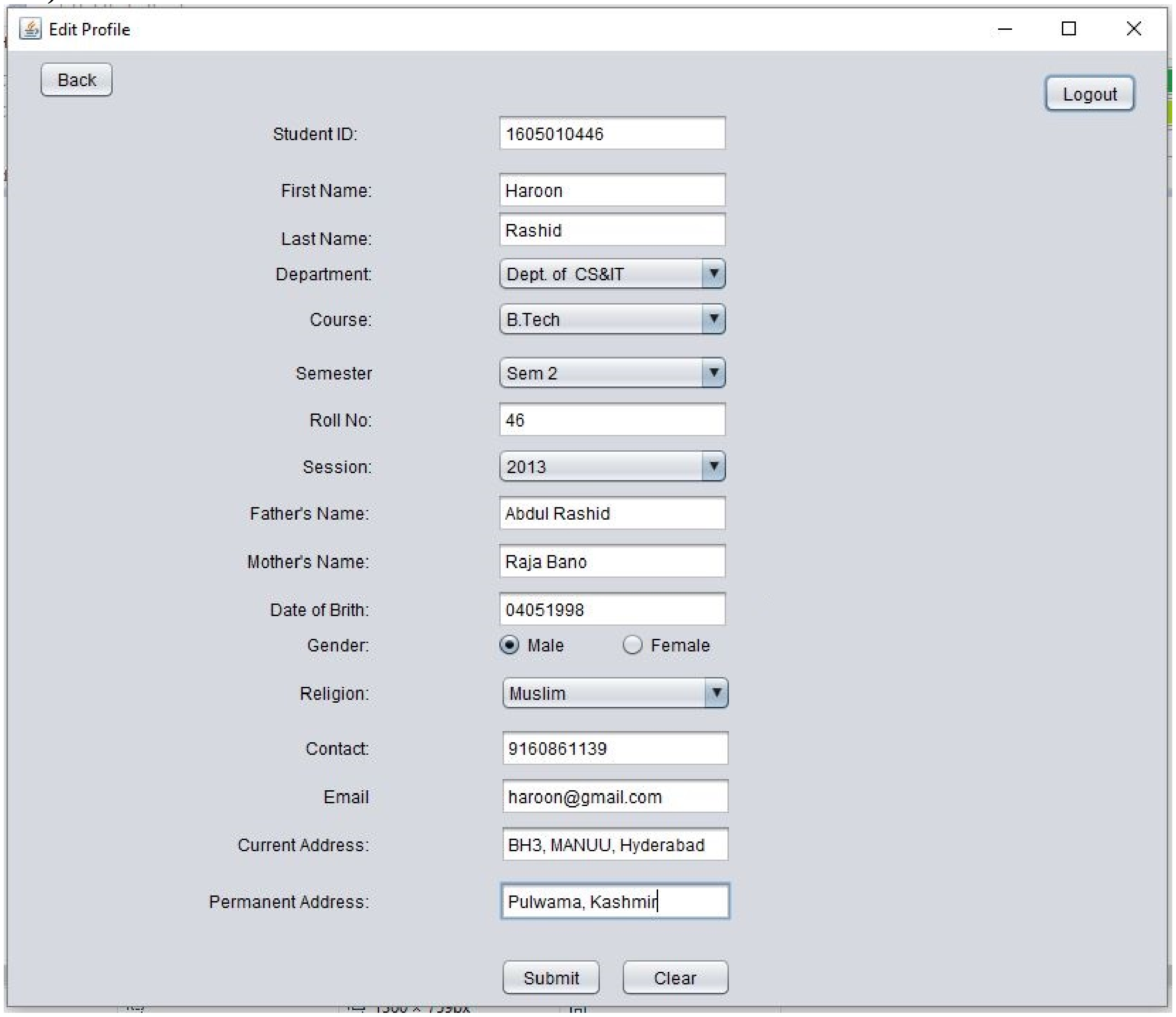
4.1.1) The code under Edit Profile button is as follows:

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
this.dispose();
new StudentEdit().setVisible(true);// TODO add your handling code here:
   }
```

4.1.2) The code under Logout button is as follows:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
this.dispose();
new Login().setVisible(true);
// TODO add your handling code here:
    }
```

4.2) Edit Profile



Student ID:	1605010446
First Name:	Haroon
Last Name:	Rashid
Department:	Dept. of CS&IT
Course:	B.Tech
Semester	Sem 2
Roll No:	46
Session:	2013
Father's Name:	Abdul Rashid
Mother's Name:	Raja Bano
Date of Brith:	04051998
Gender:	[✓ Male] Female
Religion:	Muslim
Contact:	9160861139
Email	haroon@gmail.com
Current Address:	BH3, MANUU, Hyderabad
Permanent Address:	Pulwama, Kashmir

Figure 4.2

In the Edit Profile frame a student will be able to Edit all the details that are in his/her profile and update them by clicking on the submit button.

4.2.1) The code for the submit button is as follows:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
PreparedStatement pst=null;
```

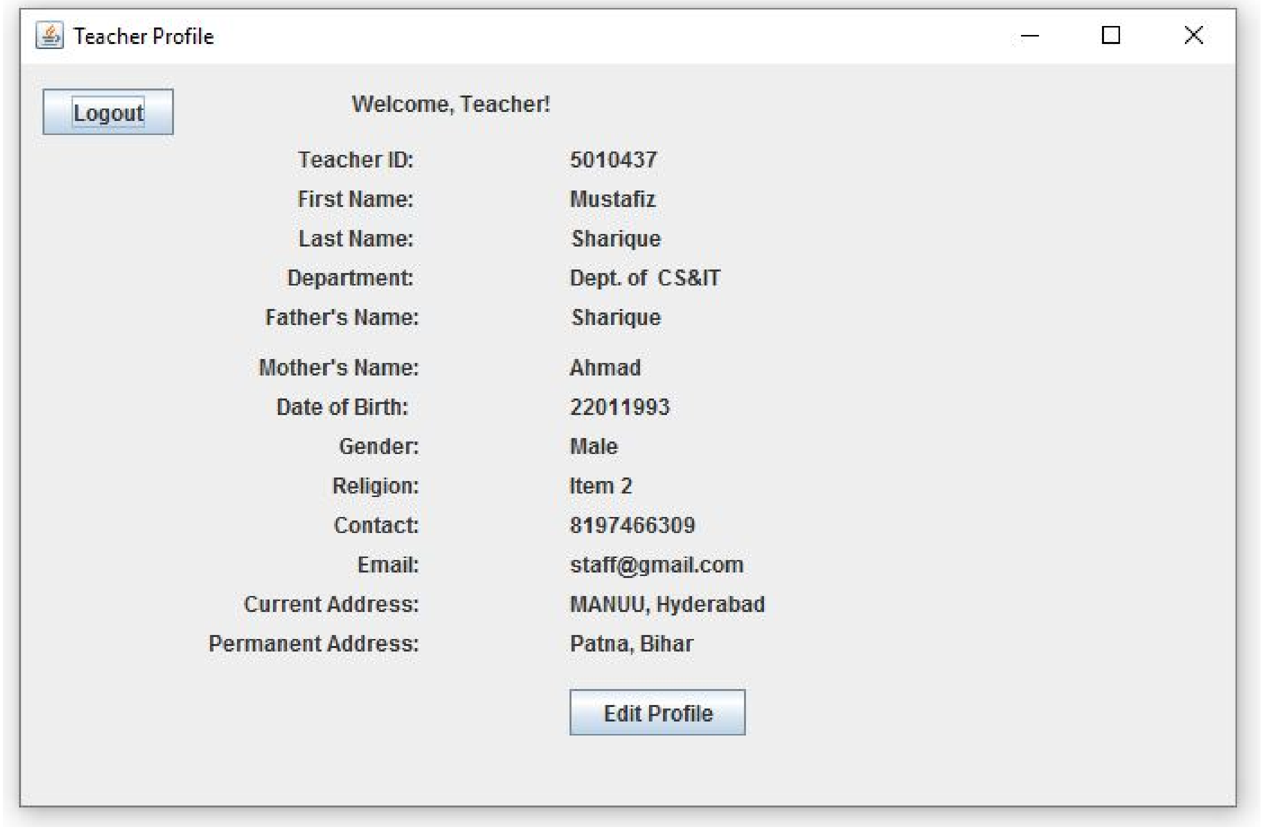
```
 try {
       PreparedStatement pst1=con.prepareStatement("select * from student where
SID=?");
        pst1.setString(1,sid.getText());
        ResultSet res=pst1.executeQuery();

        if (res.next())
        { pst=con.prepareStatement("UPDATE student SET
sfname=?,slname=?,sdept=?,scourse=?,ssemester=?,srollno=?,csession=?,fathname
=?,mothname=?,sdob=?,sgender=?,sreligion=?,scontact=?,semail=?,caddress=?,pad
dress=? WHERE sid=?");

        pst.setString(1,fname.getText());
        pst.setString(2,lname.getText());
        pst.setString(3,dept.getItemAt(dept.getSelectedIndex()));
        pst.setString(4,course.getItemAt(course.getSelectedIndex()));
        pst.setString(5,semester.getItemAt(semester.getSelectedIndex()));

        pst.setString(6,rollno.getText());
        pst.setString(7,session.getItemAt(session.getSelectedIndex()));
        pst.setString(8,fathname.getText());
        pst.setString(9,mothname.getText());
        pst.setString(10,dob.getText());
        if(male.isSelected())
                { pst.setString(11,"Male"); }
        else if(female.isSelected())
                { pst.setString(11,"Female"); }
        pst.setString(12,religion.getItemAt(religion.getSelectedIndex()));
        pst.setString(13,contact.getText());
        pst.setString(14,email.getText());
        pst.setString(15,caddress.getText());
        pst.setString(16,paddress.getText());
        pst.setString(17,sid.getText());

     }   else
        {  pst=con.prepareStatement("insert into student
values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
        pst.setString(1,sid.getText());
        pst.setString(2,fname.getText());
        pst.setString(3,lname.getText());
        pst.setString(4,dept.getItemAt(dept.getSelectedIndex()));
        pst.setString(5,course.getItemAt(course.getSelectedIndex()));
        pst.setString(6,semester.getItemAt(semester.getSelectedIndex()));

        pst.setString(7,rollno.getText());
        pst.setString(8,session.getItemAt(session.getSelectedIndex()));
        pst.setString(9,fathname.getText());
        pst.setString(10,mothname.getText());
        pst.setString(11,dob.getText());
        if(male.isSelected())
                { pst.setString(12,"Male"); }
        else if(female.isSelected())
                { pst.setString(12,"Female"); }
        pst.setString(13,religion.getItemAt(religion.getSelectedIndex()));
        pst.setString(14,contact.getText());
        pst.setString(15,email.getText());
        pst.setString(16,caddress.getText());
        pst.setString(17,paddress.getText());
        }
        int rs1=pst.executeUpdate();
        System.out.println(rs1+" records inserted");
        if(rs1==1)
        { JOptionPane.showMessageDialog(null, "Successfully Submitted!");

         }
      }
      catch(Exception e){
         JOptionPane.showMessageDialog(null, e);
      }
  }
```

4.2.2) The code for the Clear button is as follows:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:
fname.setText("");
lname.setText("");
fname.setText("");
dept.setSelectedIndex(0);
course.setSelectedIndex(0);
semester.setSelectedIndex(0);
lname.setText("");
rollno.setText("");
fathname.setText("");
mname.setText("");
dob.setText("");
contact.setText("");
email.setText("");
caddress.setText("");
paddress.setText("");


   }
```

Chapter 5: Teacher Management System

5.1) Teacher Profile



Welcome, Teacher!

Teacher ID:	5010437
First Name:	Mustafiz
Last Name:	Sharique
Department:	Dept. of CS&IT
Father's Name:	Sharique
Mother's Name:	Ahmad
Date of Birth:	22011993
Gender:	Male
Religion:	Item 2
Contact:	8197466309
Email:	staff@gmail.com
Current Address:	MANUU, Hyderabad
Permanent Address:	Patna, Bihar

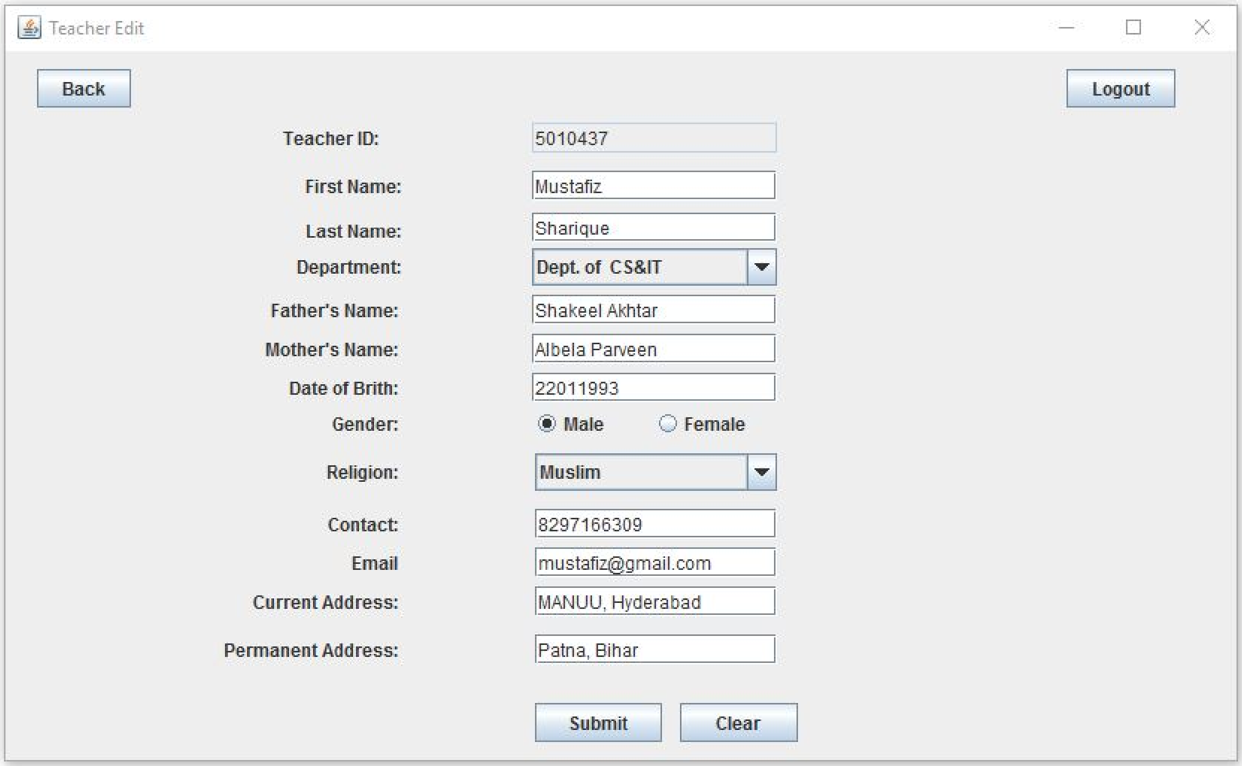
Figure 5.1

After logging in to the IMS, a teacher will be redirected to teacher profile, where he will see his/her details and will also be able to Edit the profile.

5.1.1) The code under Edit Profile button is:

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
this.dispose();
new TeacherEdit().setVisible(true);// TODO add your handling code here:
    }
```

5.2) Teacher Edit



Teacher ID:	5010437
First Name:	Mustafiz
Last Name:	Sharique
Department:	Dept. of CS&IT
Father's Name:	Shakeel Akhtar
Mother's Name:	Albela Parveen
Date of Brith:	22011993
Gender:	[✓ Male] Female
Religion:	Muslim
Contact:	8297166309
Email	mustafiz@gmail.com
Current Address:	MANUU, Hyderabad
Permanent Address:	Patna, Bihar

Figure 5.2

After clicking on the Edit Profile button, a teacher will get to edit his/her details and update them.

5.2.1) The code that will be executed when the above window gains focus is as f follows:

```
private void formWindowGainedFocus(java.awt.event.WindowEvent evt) {
    System.out.println("Hello ADMIN Teacher EDIT");
    tid.setText(obj1.x1);

    try{ System.out.println("Hello To Prepare");
      PreparedStatement pst1=con.prepareStatement("select * from teacher where
tid=?");
      System.out.println("Prepared");
      tid.setText(obj1.x1);
      pst1.setString(1, tid.getText());
      System.out.println("Try statemnet");
      ResultSet rs=pst1.executeQuery();

      if(rs.next())
```

```
        {

        fname.setText(rs.getString(2));
        lname.setText(rs.getString(3));

        fathname.setText(rs.getString(5));
        mothname.setText(rs.getString(6));
        dob.setText(rs.getString(7));

        contact.setText(rs.getString(10));
        email.setText(rs.getString(11));
        caddress.setText(rs.getString(12));
        paddress.setText(rs.getString(13));



        }
}
    catch (Exception e)
     { JOptionPane.showMessageDialog(null, e);
       }          // TODO add your handling code here:
    }
```

5.2.2) The code under Submit button is as follows:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

        con=ConnClass.ConnectDB();
        PreparedStatement pst=null;
 try {

          PreparedStatement pst1=con.prepareStatement("select * from teacher where
TID=?");
          pst1.setString(1,tid.getText());
          ResultSet res=pst1.executeQuery();

           if (res.next())
           {  pst=con.prepareStatement("UPDATE teacher SET
tfname=?,tlname=?,tdept=?,fathname=?,mothname=?,tdob=?,tgender=?,treligion=?
,tcontact=?,temail=?,caddress=?,paddress=? WHERE tid=?");
```

```
        pst.setString(1,fname.getText());
        pst.setString(2,lname.getText());
        pst.setString(3,dept.getItemAt(dept.getSelectedIndex()));
        pst.setString(4,fathname.getText());
        pst.setString(5,mothname.getText());
        pst.setString(6,dob.getText());
        if(male.isSelected())
                { pst.setString(7,"Male"); }
        else if(female.isSelected())
                { pst.setString(7,"Female"); }
        pst.setString(8,religion.getItemAt(religion.getSelectedIndex()));
        pst.setString(9,contact.getText());
        pst.setString(10,email.getText());
        pst.setString(11,caddress.getText());
        pst.setString(12,paddress.getText());
        pst.setString(13,tid.getText());

         }
         else
         {
        pst=con.prepareStatement("insert into teacher
values(?,?,?,?,?,?,?,?,?,?,?,?,?)");
        pst.setString(1,tid.getText());
        pst.setString(2,fname.getText());
        pst.setString(3,lname.getText());
        pst.setString(4,dept.getItemAt(dept.getSelectedIndex()));
        pst.setString(5,fathname.getText());
        pst.setString(6,mothname.getText());
        pst.setString(7,dob.getText());
        if(male.isSelected())
                { pst.setString(8,"Male"); }
        else if(female.isSelected())
                { pst.setString(8,"Female"); }
        pst.setString(9,religion.getItemAt(religion.getSelectedIndex()));
        pst.setString(10,contact.getText());
        pst.setString(11,email.getText());
        pst.setString(12,caddress.getText());
        pst.setString(13,paddress.getText());

         }
        int rs1=pst.executeUpdate();
        System.out.println(rs1+" records inserted");
        if(rs1==1)
        { JOptionPane.showMessageDialog(null, "Successfully Submitted!");

        }
    }
    catch(Exception e){
         JOptionPane.showMessageDialog(null, e);
    }       // TODO add your handling code here:
}
```

Chapter 6:
Admin Module

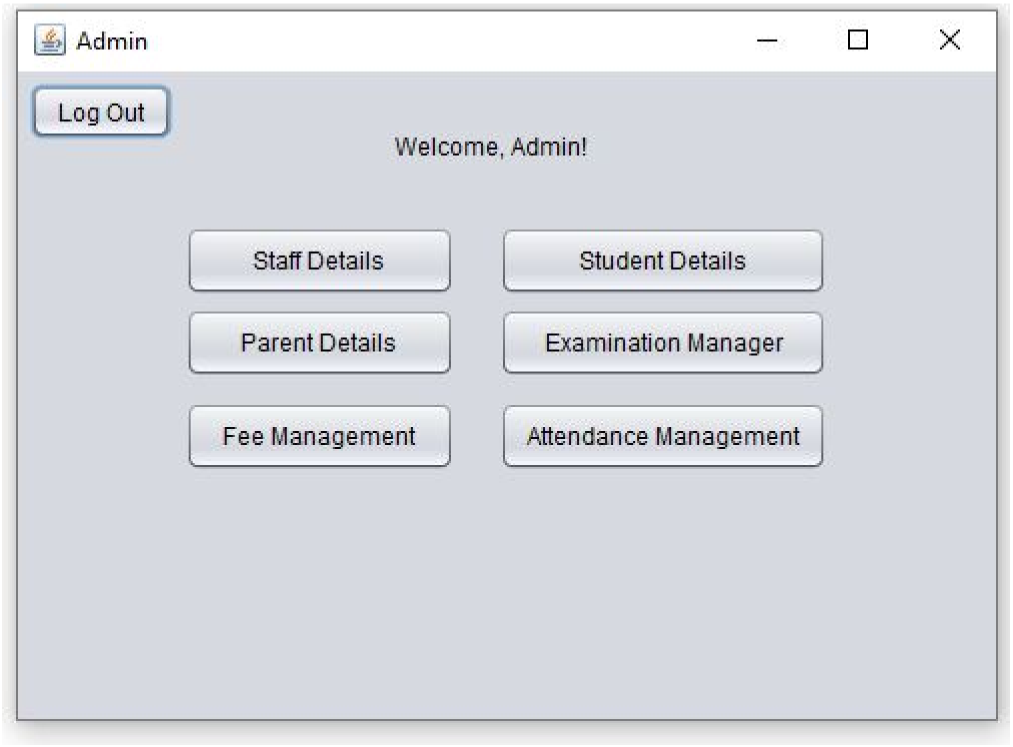
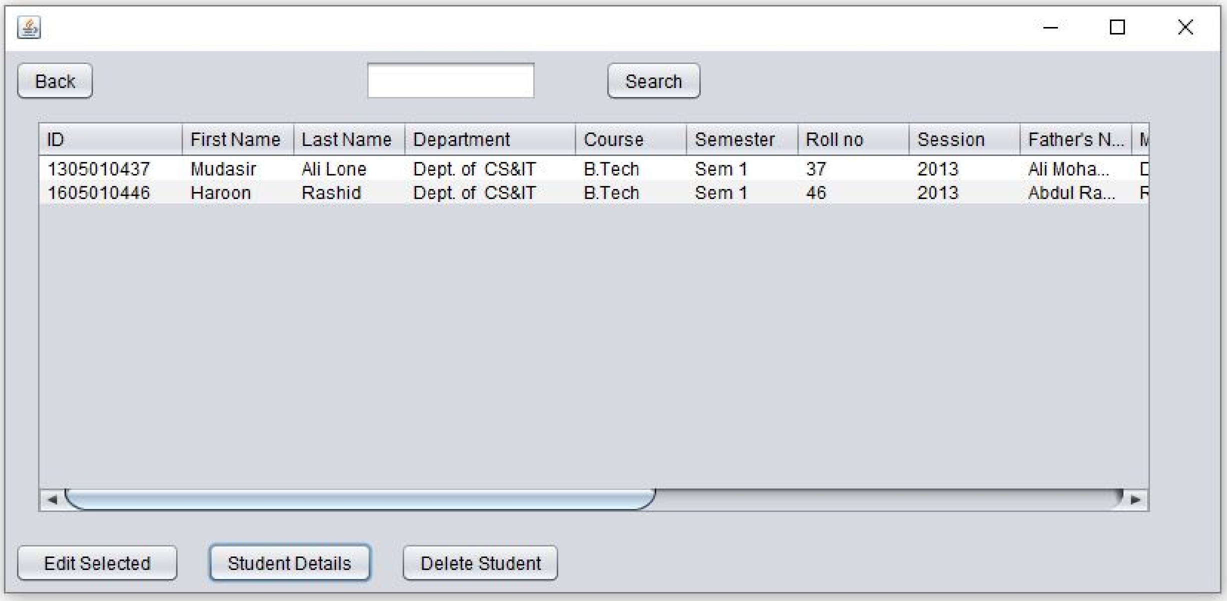


Figure 6

Admin will have complete access to all the details of all the users and modules within IMS. An admin will be able to edit, modify and delete the data. Admin module will have further sub-modules as follows:

- Staff Details
- Student Details
- Parent Details
- Examination Manager
- Fee Management
- Attendance Management

6.1) Admin-Student



ID	First Name	Last Name	Department	Course	Semester	Roll no	Session	Father's N...	M
1305010437	Mudasir	Ali Lone	Dept. of CS&IT	B.Tech	Sem 1	37	2013	Ali Moha...	D
1605010446	Haroon	Rashid	Dept. of CS&IT	B.Tech	Sem 1	46	2013	Abdul Ra...	R

Figure 6.1

Admin after logging in and clicking on Student Details button will be able to see all the details of students that they have registered with the IMS by clicking on Student Details and modify them using Edit Selected. Admin can also search for details of a particular student by entering his ID and then clicking Search button. To modify the details of any student he will click on his data row in the table and then click Edit Student.

6.1.1) The code for Student Details button is as follows:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
   DefaultTableModel model = (DefaultTableModel) jTable1.getModel();


      try {

         PreparedStatement pst = conn.prepareStatement("select * from student");

         ResultSet rs = pst.executeQuery();
         int i = 0;
         model.setRowCount(0);

            while (rs.next()) {
              model.addRow(new Object[]{rs.getInt(1), rs.getString(2), rs.getString(3),
rs.getString(4), rs.getString(5), rs.getString(6), rs.getString(7), rs.getString(8),
rs.getString(9), rs.getString(10), rs.getString(11), rs.getString(12), rs.getString(13),
rs.getString(14), rs.getString(15), rs.getString(16), rs.getString(17)});
                i++;

            }

            if (i < 1) {

                JOptionPane.showMessageDialog(null, "No Records Found");

            }
            else

            if (i > 1) {

                System.out.println(i + " Record Found");

            }

        } catch (Exception e) {

            JOptionPane.showMessageDialog(null, e);

        }
    }
```

6.1.2) The code for Edit Selected is as follows:

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
int row = jTable1.getSelectedRow();
int column = jTable1.getSelectedColumn();
x1 = Integer.toString((int) jTable1.getValueAt(row,0));

this.dispose();
new AdminStudentEdit().setVisible(true);
// TODO add your handling code here:
    }
```

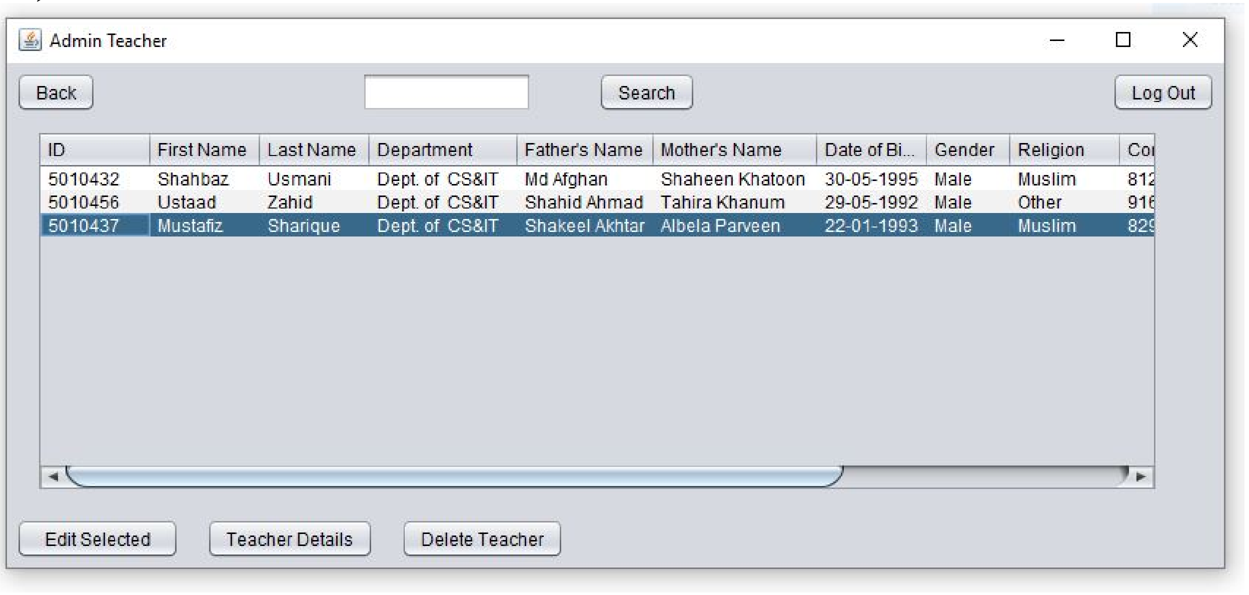
6.1.3) The code for Search button is as follows:

```
   private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();


        try {

             PreparedStatement pst = conn.prepareStatement("select * from student
where sid=?");
            pst.setString(1, search.getText());


            ResultSet rs = pst.executeQuery();
            int i = 0;
            model.setRowCount(0);

                while (rs.next()) {

                    model.addRow(new Object[]{rs.getInt(1), rs.getString(2),
rs.getString(3), rs.getString(4), rs.getString(5), rs.getString(6), rs.getString(7),
rs.getString(8), rs.getString(9), rs.getString(10), rs.getString(11), rs.getString(12),
rs.getString(13), rs.getString(14), rs.getString(15), rs.getString(16),
rs.getString(16)});
                i++;

            }

            if (i < 1) {

                JOptionPane.showMessageDialog(null, "No Records Found");

            }
            else

            if (i > 1) {

                System.out.println(i + " Record Found");

            }
        } catch (Exception e) {

            JOptionPane.showMessageDialog(null, e);

        }

    }
```

6.1.4) The code for Delete button is as follows:

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
int row = jTable1.getSelectedRow();
x1 = Integer.toString((int) jTable1.getValueAt(row,0));
try
{
PreparedStatement pst = conn.prepareStatement("DELETE FROM student
WHERE SID=?");
pst.setString(1,x1);
 int rs = pst.executeUpdate();

  if(rs==1)
        { JOptionPane.showMessageDialog(null, "Successfully Deleted!"); }
}
catch (Exception e) {

            JOptionPane.showMessageDialog(null, e);

        }
    }
```

6.2) Admin-Teacher



ID	First Name	Last Name	Department	Father's Name	Mother's Name	Date of Bi...	Gender	Religion	Co
5010432	Shahbaz	Usmani	Dept. of CS&IT	Md Afghan	Shaheen Khatoon	30-05-1995	Male	Muslim	812
5010456	Ustaad	Zahid	Dept. of CS&IT	Shahid Ahmad	Tahira Khanum	29-05-1992	Male	Other	916
5010437	Mustafiz	Sharique	Dept. of CS&IT	Shakeel Akhtar	Albela Parveen	22-01-1993	Male	Muslim	829

Figure 6.2

The admin will be able to see all the details of teachers that they have registered with the IMS by clicking on Teacher Details and modify them using Edit Selected. Admin can also search for details of a particular teacher by entering his employee ID and then clicking Search button. To modify the details of any teacher he will click on his data row in the table and then click Edit Student. Delete Teacher will delete the selected teacher from the database.

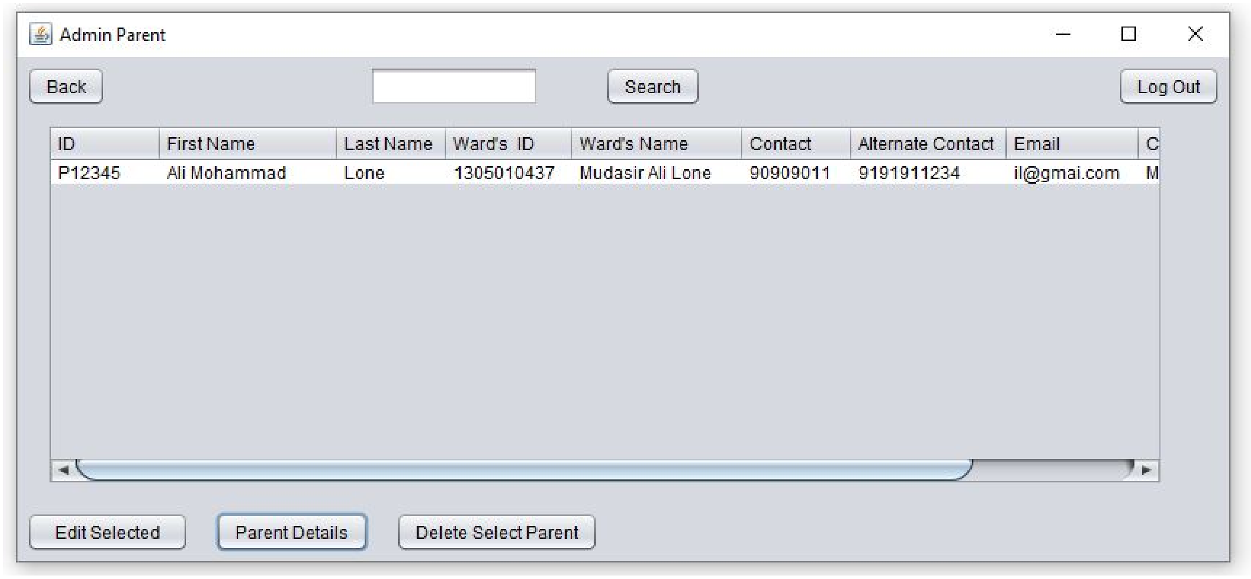
6.2.1) The code for Teacher Details button is as follows:

```
  private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
     // conn=ConnClass.ConnectDB();
    DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

          //model.addRow(new Object[]{"Column 1", "Column 2", "Column
3","Column 4","Column 5"});

       try {

            PreparedStatement pst = conn.prepareStatement("select * from teacher");

            ResultSet rs = pst.executeQuery();
            int i = 0;
            model.setRowCount(0);

               while (rs.next()) {

              model.addRow(new Object[]{rs.getInt(1), rs.getString(2), rs.getString(3),
rs.getString(4), rs.getString(5), rs.getString(6), rs.getString(7), rs.getString(8),
rs.getString(9), rs.getString(10), rs.getString(11), rs.getString(12),
rs.getString(13)});
               i++;

            }

            if (i < 1) {

               JOptionPane.showMessageDialog(null, "No Records Found");

            }
            else

            if (i > 1) {

               System.out.println(i + " Record Found");

            }

       } catch (Exception e) {

            JOptionPane.showMessageDialog(null, e);

       }     // TODO add your handling code here:
   }
```

6.2.2) The code for Edit Selected is as follows:

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
int row = jTable1.getSelectedRow();
int column = jTable1.getSelectedColumn();
x1 = Integer.toString((int) jTable1.getValueAt(row,0));
this.dispose();
new AdminTeacherEdit().setVisible(true);
// TODO add your handling code here:
    }
```

6.3) Admin-Parent



ID	First Name	Last Name	Ward's ID	Ward's Name	Contact	Alternate Contact	Email	C
P12345	Ali Mohammad	Lone	1305010437	Mudasir Ali Lone	90909011	9191911234	il@gmai.com	M

Figure 6.3

Admin-Parent module will allow the admin to see the details of all the Parents registered with IMS and edit their details and delete them.

6.3.1) The code under Parent Details Button is as follows:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
      DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
              try {
             PreparedStatement pst = conn.prepareStatement("select * from parent");
            ResultSet rs = pst.executeQuery();
            int i = 0;
            model.setRowCount(0);
                while (rs.next()) {
```

```
            model.addRow(new Object[]{rs.getString(1), rs.getString(2),
rs.getString(3), rs.getString(4), rs.getString(5), rs.getString(6), rs.getString(7),
rs.getString(8), rs.getString(9), rs.getString(10)});
                i++;
            }
            if (i < 1) {
                JOptionPane.showMessageDialog(null, "No Records Found");
            }
            else
            if (i > 1) {
                System.out.println(i + " Record Found");
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }
```

6.3.2) The code under Edit Selected is as follows:

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
int row = jTable1.getSelectedRow();
int column = jTable1.getSelectedColumn();
x1 =  (String)jTable1.getValueAt(row,0);


this.dispose();
new AdminParentEdit().setVisible(true);
// TODO add your handling code here:
    }
```

Chapter 7:
Examination Management System

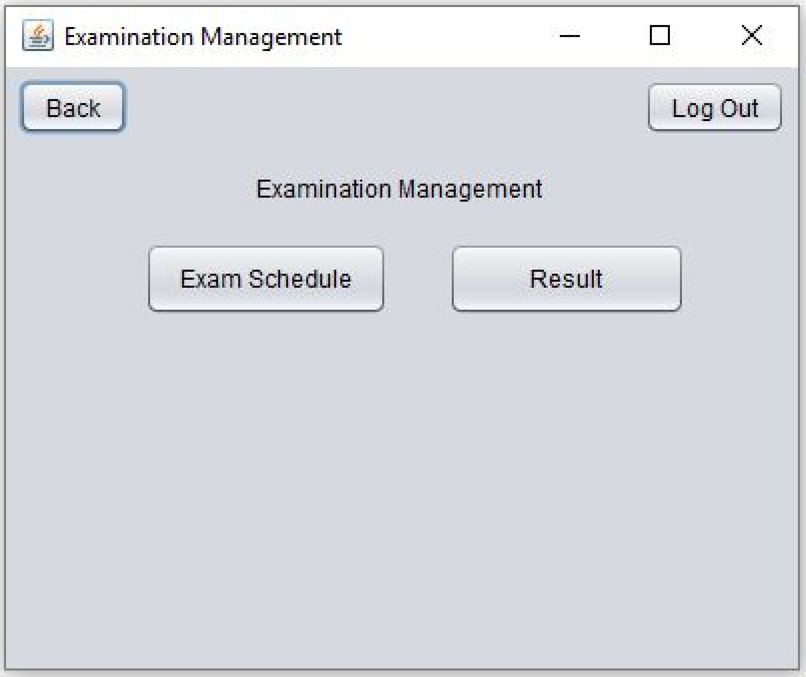


Figure 7

Examination Management System will consist of Exam Schedule and Result Management. An admin will click on Exam Schedule Button to get schedule details and Result button to access Result Management.

7.1) Result Management

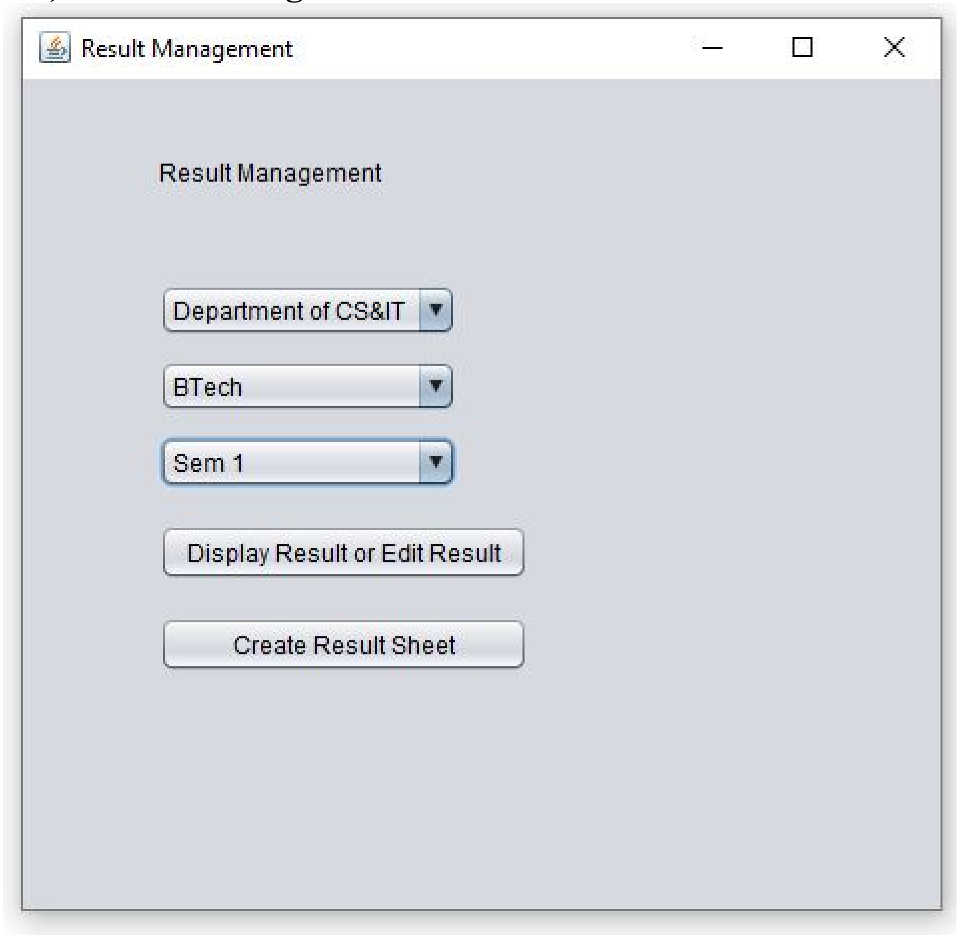


Figure 7.1

In the Result Management, the Admin will choose the Department, Course and Semester to get the Result Details or to create the result sheet. After choosing, he/she will click on the respective button.

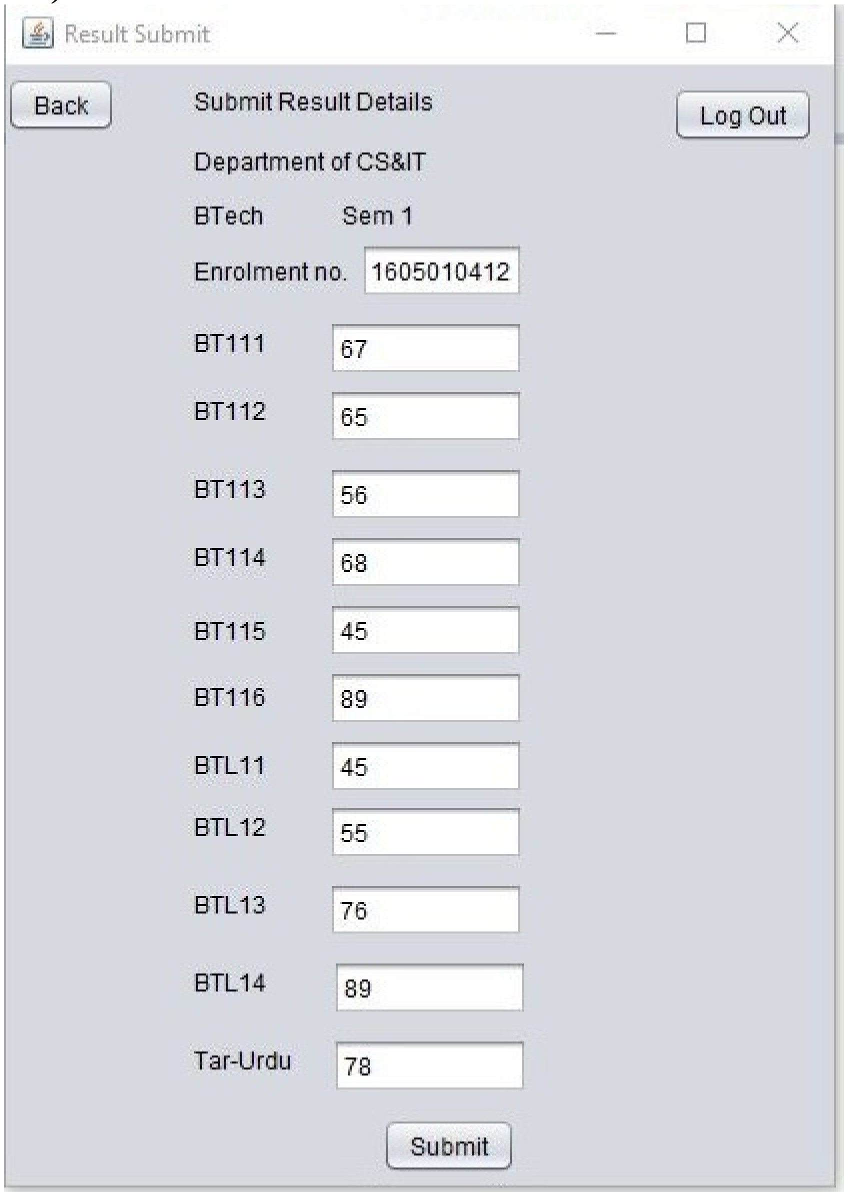
7.1.1) The code under Display Result or Edit Result button is as follows:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
if(jComboBox3.getSelectedIndex()==0)
{JOptionPane.showMessageDialog(null, "Select First");}
   else
{    this.dispose();
new AdminResultSheet().setVisible(true);
   }
```

7.1.2) The code under Create Result Sheet button is follows:

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        dept=jComboBox1.getItemAt(jComboBox1.getSelectedIndex());
        course=jComboBox2.getItemAt(jComboBox2.getSelectedIndex());
        sem=jComboBox3.getItemAt(jComboBox3.getSelectedIndex());

        if(jComboBox3.getSelectedIndex()==0)
{JOptionPane.showMessageDialog(null, "Select First");}
    else
        { this.dispose();
new AdminResultSubmit().setVisible(true); }
// TODO add your handling code here:
    }
```

7.2) Result Submit



Submit Result Details

Department of CS&IT

BTech Sem 1

Enrolment no.	1605010412
BT111	67
BT112	65
BT113	56
BT114	68
BT115	45
BT116	89
BTL11	45
BTL12	55
BTL13	76
BTL14	89
Tar-Urdu	78

Figure 7.2

After choosing the department, course, semester and then clicking on Create Result

Sheet, the Admin will enter the enrolment number of student and then his marks in respective subjects and then click on the submit button. All the marks data will be stored in the respective database of that course and semester.

7.2.1) The code under submit button is as follows:

```
  private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
con=ConnClass.ConnectDB();

 int sub1= Integer.parseInt(jTextField2.getText());
         int sub2= Integer.parseInt(jTextField3.getText());
         int sub3= Integer.parseInt(jTextField4.getText());
         int sub4= Integer.parseInt(jTextField5.getText());
         int sub5= Integer.parseInt(jTextField6.getText());
         int sub6= Integer.parseInt(jTextField7.getText());
         int sub7= Integer.parseInt(jTextField8.getText());
         int sub8= Integer.parseInt(jTextField9.getText());
         int sub9= Integer.parseInt(jTextField10.getText());
         int sub10= Integer.parseInt(jTextField11.getText());
         int sub11= Integer.parseInt(jTextField12.getText());

         int flag=1;

PreparedStatement pst=null;
 try {

         if(obj.x==1)
         {pst=con.prepareStatement("insert into BTech1
values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");

         pst.setString(1,jTextField1.getText());
         pst.setString(2,jLabel4.getText());
         pst.setString(3,jTextField2.getText());
         if (sub1>=36)
         { pst.setString(4,"P"); }
         else { pst.setString(4,"F");
           flag=0;}

         pst.setString(5,jLabel5.getText());
         pst.setString(6,jTextField3.getText());
         if (sub2>=36)
```

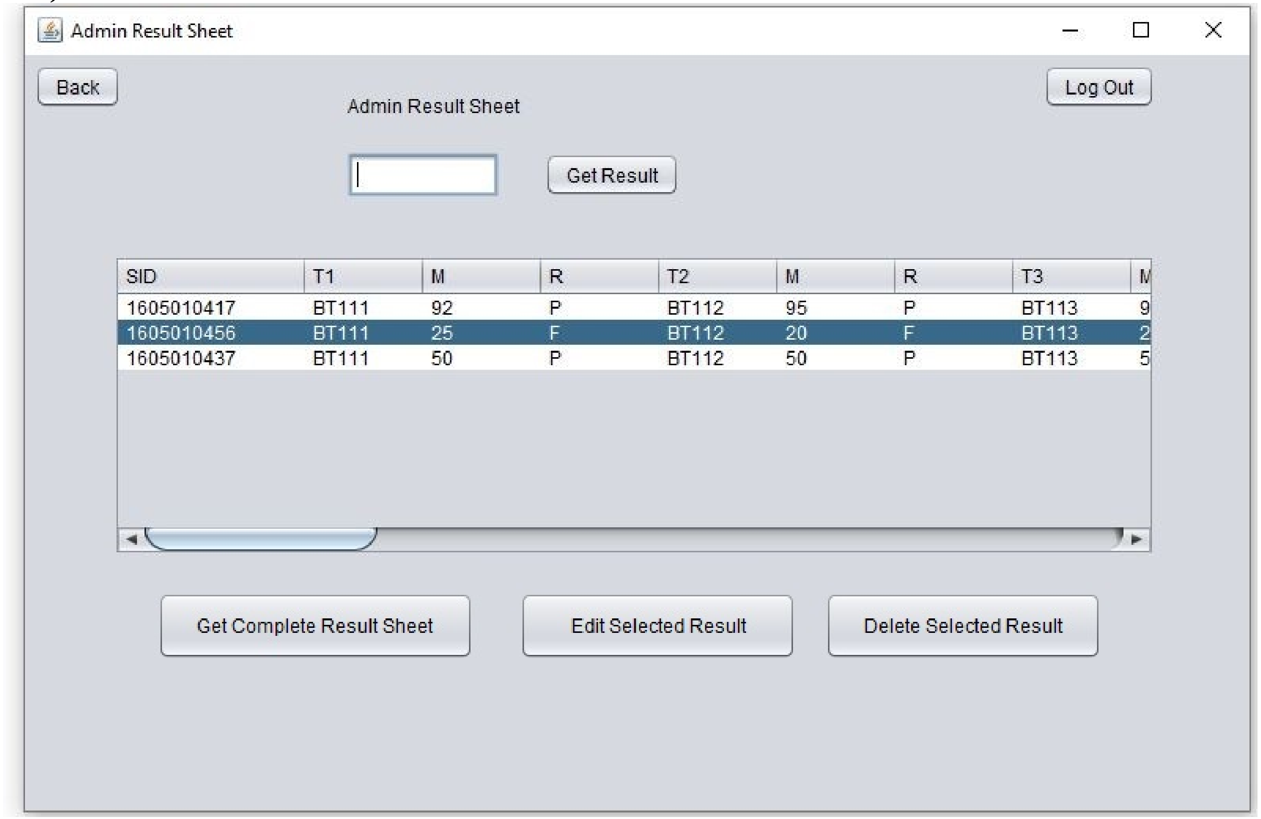
```
{ pst.setString(7,"P"); }
else { pst.setString(7,"F");
  flag=0; }

pst.setString(8,jLabel6.getText());
pst.setString(9,jTextField4.getText());
if (sub3>=36)
{ pst.setString(10,"P"); }
else { pst.setString(10,"F");
  flag=0; }

pst.setString(11,jLabel7.getText());
pst.setString(12,jTextField5.getText());
if (sub4>=36)
{ pst.setString(13,"P"); }
else { pst.setString(13,"F");
 flag=0; }

pst.setString(14,jLabel8.getText());
pst.setString(15,jTextField6.getText());
if (sub5>=36)
{ pst.setString(16,"P"); }
else { pst.setString(16,"F");
  flag=0; }

pst.setString(17,jLabel9.getText());
pst.setString(18,jTextField7.getText());
if (sub6>=36)
{ pst.setString(19,"P"); }
else { pst.setString(19,"F");
  flag=0; }

pst.setString(20,jLabel10.getText());
pst.setString(21,jTextField8.getText());
if (sub7>=36)
{ pst.setString(22,"P"); }
else { pst.setString(22,"F");
  flag=0; }

pst.setString(23,jLabel11.getText());
pst.setString(24,jTextField9.getText());
```

```
  if (sub8>=36)
  { pst.setString(25,"P"); }
  else { pst.setString(25,"F");
  flag=0; }

  pst.setString(26,jLabel12.getText());
  pst.setString(27,jTextField10.getText());
  if (sub9>=36)
  { pst.setString(28,"P"); }
  else { pst.setString(28,"F");
    flag=0; }

  pst.setString(29,jLabel13.getText());
  pst.setString(30,jTextField11.getText());
  if (sub10>=36)
  { pst.setString(31,"P"); }
  else { pst.setString(31,"F");
    flag=0; }

  pst.setString(32,jLabel14.getText());
  pst.setString(33,jTextField12.getText());
 if (sub11>=36)
  { pst.setString(34,"P"); }
  else { pst.setString(34,"F");
    flag=0; }


 int total=sub1+sub2+sub3+sub4+sub5+sub6+sub7+sub8+sub9+sub10+sub11;

  pst.setString(35,Integer.toString(total));
  if(flag==1)
  { pst.setString(36,"P");}
  else { pst.setString(36,"F");}

          }

    if(obj.x==2)
 { pst=con.prepareStatement("insert into BTech2
values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
```

```
pst.setString(1,jTextField1.getText());

  pst.setString(2,jLabel4.getText());
  pst.setString(3,jTextField2.getText());
  pst.setString(4,"P");

  pst.setString(5,jLabel5.getText());
  pst.setString(6,jTextField3.getText());
  pst.setString(7,"P");

  pst.setString(8,jLabel6.getText());
  pst.setString(9,jTextField4.getText());
  pst.setString(10,"P");

  pst.setString(11,jLabel7.getText());
  pst.setString(12,jTextField5.getText());
  pst.setString(13,"P");

  pst.setString(14,jLabel8.getText());
  pst.setString(15,jTextField6.getText());
  pst.setString(16,"P");

  pst.setString(17,jLabel9.getText());
  pst.setString(18,jTextField7.getText());
  pst.setString(19,"P");

  pst.setString(20,jLabel10.getText());
  pst.setString(21,jTextField8.getText());
  pst.setString(22,"P");

  pst.setString(23,jLabel11.getText());
  pst.setString(24,jTextField9.getText());
  pst.setString(25,"P");

  pst.setString(26,jLabel12.getText());
  pst.setString(27,jTextField10.getText());
  pst.setString(28,"P");

  pst.setString(29,"1100");
  pst.setString(30,"P");
```

```
        }
          int rs1=pst.executeUpdate();
          System.out.println(rs1+" records inserted");
          if(rs1==1)
          { JOptionPane.showMessageDialog(null, "Successfully Submitted!");
                  }
        }
        catch(Exception e){
           JOptionPane.showMessageDialog(null, e);
        }
      }
// And so on and so forth for all other semesters.
```

7.3) Result Sheet



SID	T1	M	R	T2	M	R	T3	M
1605010417	BT111	92	P	BT112	95	P	BT113	9
1605010456	BT111	25	F	BT112	20	F	BT113	2
1605010437	BT111	50	P	BT112	50	P	BT113	5

Figure 7.3

In the Examination Management the admin will choose the department, course and semester, and will then get the result details of all the students of that course and semester after clicking on Get Complete Result Sheet button.

7.3.1) The code under Get Complete Result Sheet button is as follows:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
         int i = 0;
        PreparedStatement pst=null;

      try {
         if (obj.x==1)
         { pst = conn.prepareStatement("select * from BTech1");

         ResultSet rs = pst.executeQuery();

         model.setRowCount(0);
            while (rs.next()) {
```

```
            model.addRow(new Object[]{rs.getString(1), rs.getString(2),
rs.getString(3), rs.getString(4), rs.getString(5),
                                rs.getString(6), rs.getString(7), rs.getString(8),
rs.getString(9), rs.getString(10),
                                rs.getString(11), rs.getString(12), rs.getString(13),
rs.getString(14), rs.getString(15),
                                rs.getString(16), rs.getString(17), rs.getString(18),
rs.getString(19), rs.getString(20),
                                rs.getString(21), rs.getString(22), rs.getString(23),
rs.getString(24), rs.getString(25),
                                rs.getString(26), rs.getString(27), rs.getString(28),
rs.getString(29), rs.getString(30),
                                rs.getString(31), rs.getString(32), rs.getString(33),
rs.getString(34), rs.getString(35),
                                rs.getString(36)});
            i++;

        }
          }

        if (obj.x==2)
        { pst = conn.prepareStatement("select * from BTech2");

        ResultSet rs = pst.executeQuery();

        model.setRowCount(0);
            while (rs.next()) {

            model.addRow(new Object[]{rs.getString(1), rs.getString(2),
rs.getString(3), rs.getString(4), rs.getString(5),
                                rs.getString(6), rs.getString(7), rs.getString(8),
rs.getString(9), rs.getString(10),
                                rs.getString(11), rs.getString(12), rs.getString(13),
rs.getString(14), rs.getString(15),
                                rs.getString(16), rs.getString(17), rs.getString(18),
rs.getString(19), rs.getString(20),
                                rs.getString(21), rs.getString(22), rs.getString(23),
rs.getString(24), rs.getString(25),
                                rs.getString(26), rs.getString(27), rs.getString(28),
rs.getString(29), rs.getString(30)});
```

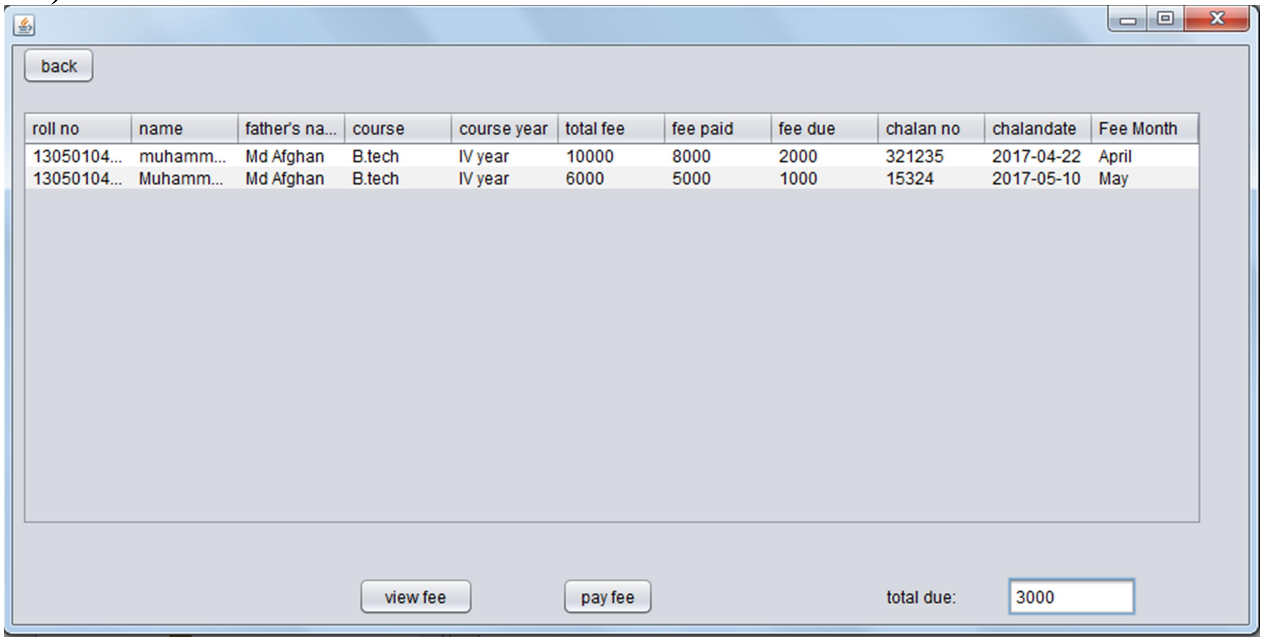
```
            i++;

        }
          }
 if (i < 1) {

            JOptionPane.showMessageDialog(null, "No Records Found");

        }
        else

        if (i > 1) {

            System.out.println(i + " Record Found");

        }

      } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);

      }
   }
// And so on for other semesters
```

7.3.2) The code under Edit Selected Result button is as follows:

7.3.3) The code under Delete Selected Result button is as follows:

Chapter 8:
Fee Management System

8.1) Fee Details



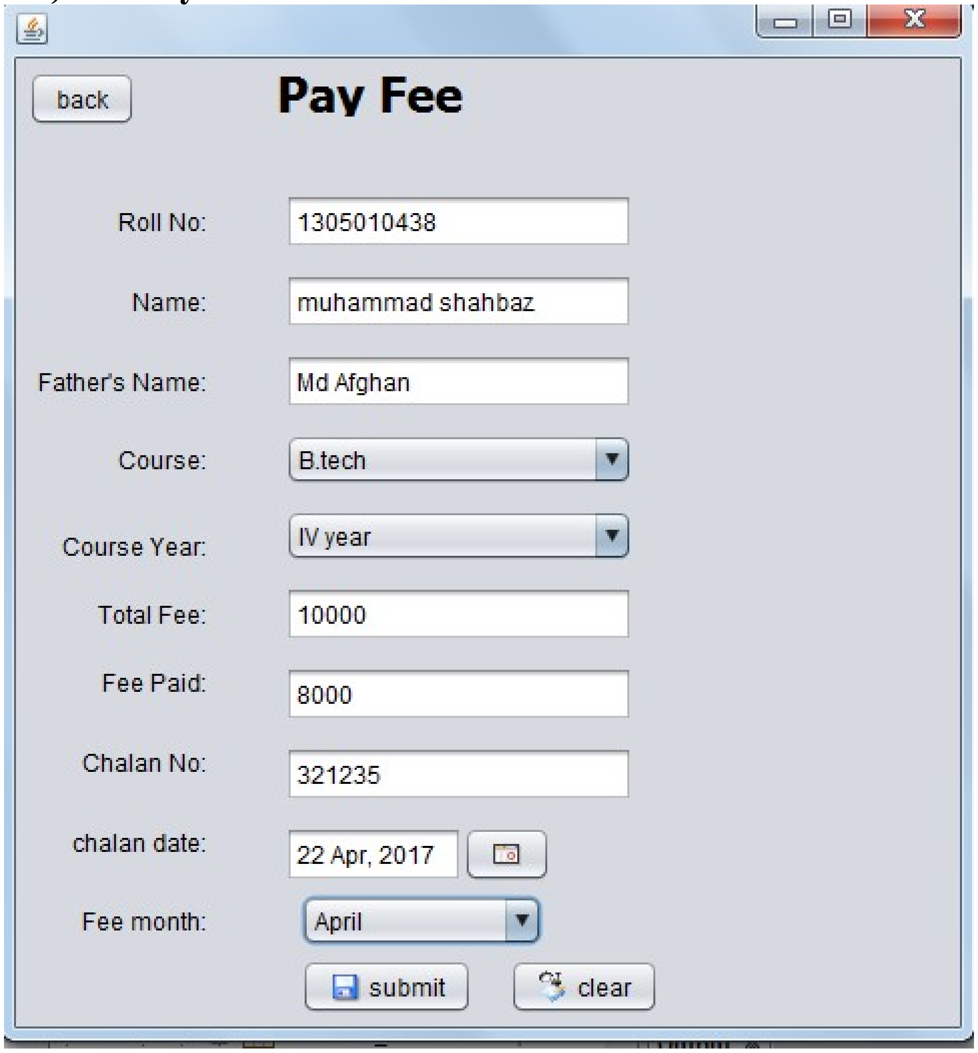
roll no	name	father's na...	course	course year	total fee	fee paid	fee due	chalan no	chalandate	Fee Month
13050104...	muhamm...	Md Afghan	B.tech	IV year	10000	8000	2000	321235	2017-04-22	April
13050104...	Muhamm...	Md Afghan	B.tech	IV year	6000	5000	1000	15324	2017-05-10	May

total due: 3000

Figure 8.1

Admin will be able to see the monthly details of all the student in the Fee Detail section and will also get to see the total amount of fee which is due from all the students.

8.2) Fee Payment



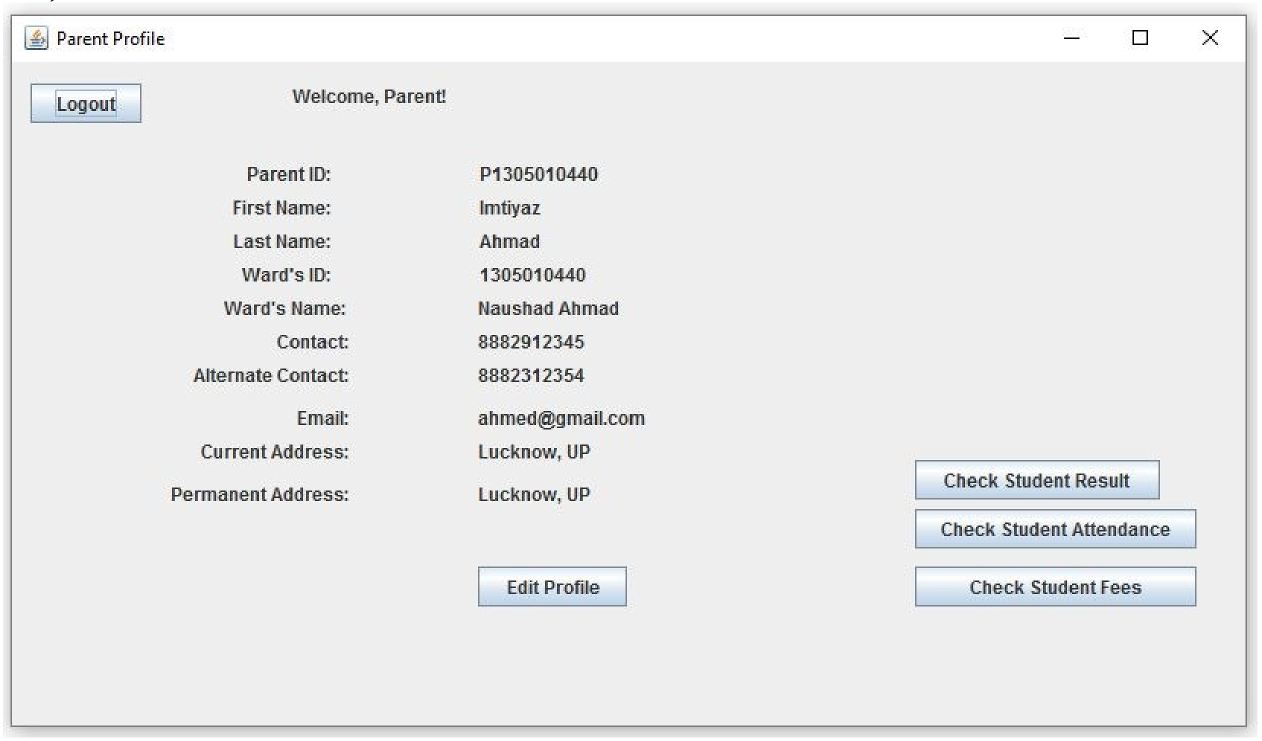
Roll No:	1305010438
Name:	muhammad shahbaz
Father's Name:	Md Afghan
Course:	B.tech
Course Year:	IV year
Total Fee:	10000
Fee Paid:	8000
Chalan No:	321235
chalan date:	22 Apr, 2017
Fee month:	April

Figure 8.2

Admin upon receiving fee from a student will update the payment details in the Pay Fee form and will then submit the details.

Chapter 9:
Parent Management System

9.1) Parent Profile



Parent Profile

Logout

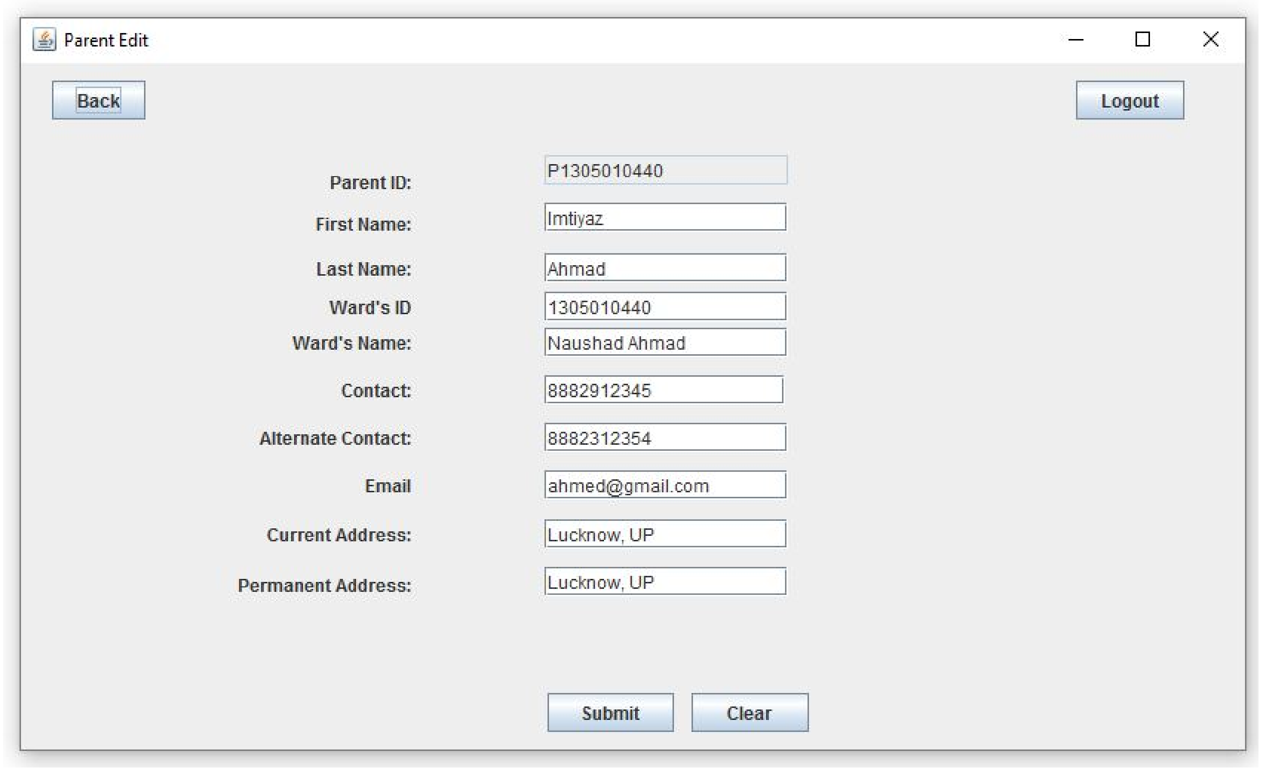
Welcome, Parent!

Parent ID:	P1305010440
First Name:	Imtiyaz
Last Name:	Ahmad
Ward's ID:	1305010440
Ward's Name:	Naushad Ahmad
Contact:	8882912345
Alternate Contact:	8882312354
Email:	ahmed@gmail.com
Current Address:	Lucknow, UP
Permanent Address:	Lucknow, UP

Figure 9.1

Parent Profile will contain all the details of Parent stored with the IMS. On clicking on Check Student Result the Parent will be able to see the result of his/her ward. On clicking on Check Student Attendance the Parent will be able to see the result of his/her ward. On clicking on Check Student Fees the Parent will be able to see the fees of his/her ward.

9.2) Parent Edit



Parent ID:	P1305010440
First Name:	Imtiyaz
Last Name:	Ahmad
Ward's ID	1305010440
Ward's Name:	Naushad Ahmad
Contact:	8882912345
Alternate Contact:	8882312354
Email	ahmed@gmail.com
Current Address:	Lucknow, UP
Permanent Address:	Lucknow, UP

Figure 9.2

Parent Edit will allow the Parent to Edit his/her details and Update them by clicking on Submit Button after entering the details in the given text fields.

Chapter 10:
Attendance Management System

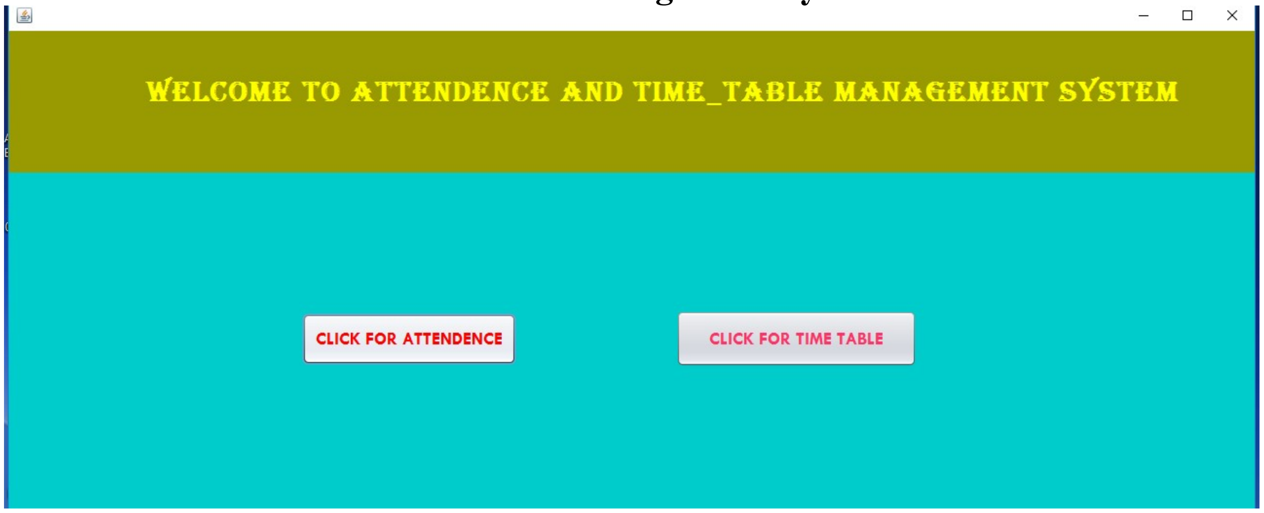
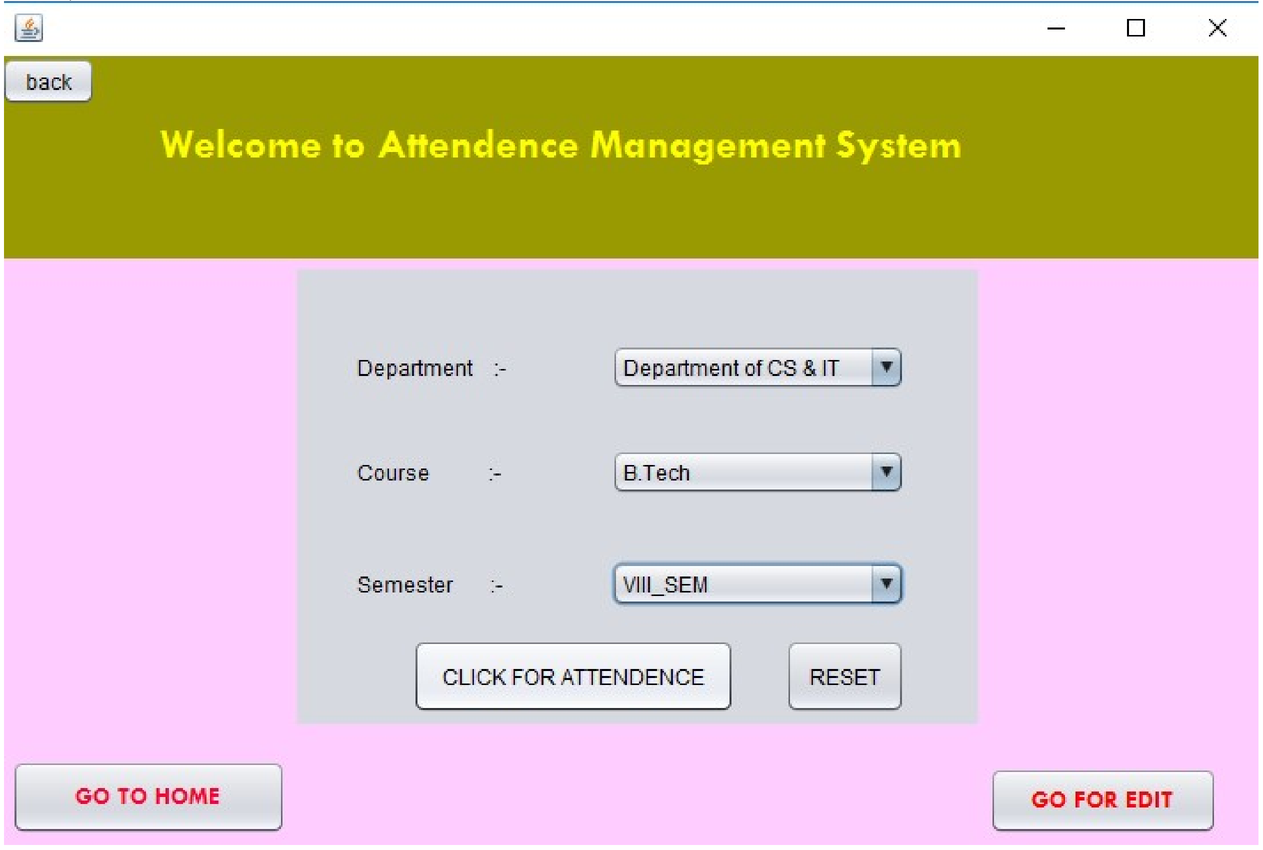


Figure 10

Attendance Management System will allow students to view their monthly attendance and course time table. Teachers will be able to view and edit the monthly attendance of students and course time table. Admin will also be able to view and edit the monthly attendance of students and course time table.

10.1) Attendance



Department :- Department of CS & IT

Course :- B.Tech

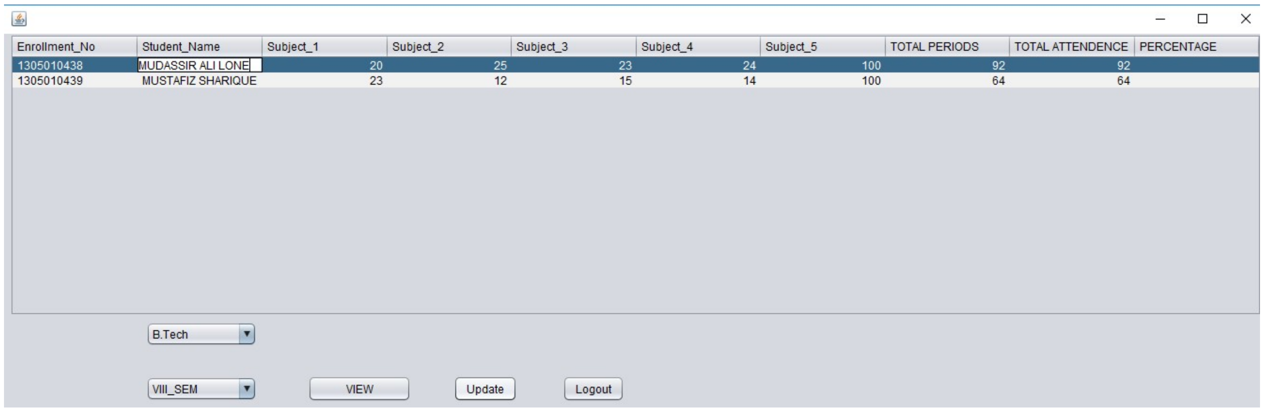
Semester :- VIII_SEM

Figure 10.1

After selecting the Department, Course, and semester. We have to click on Click for Attendance to view the attendance of all the students of the selected field. Go for Edit button will allow the teachers/admin to edit the data of attendance.

10.1.1) Attendance Details Teacher

After a teacher/admin clicks on the Click for Attendance, the following frame opens where he/she gets percentage of all the students and will also be able to edit the details.

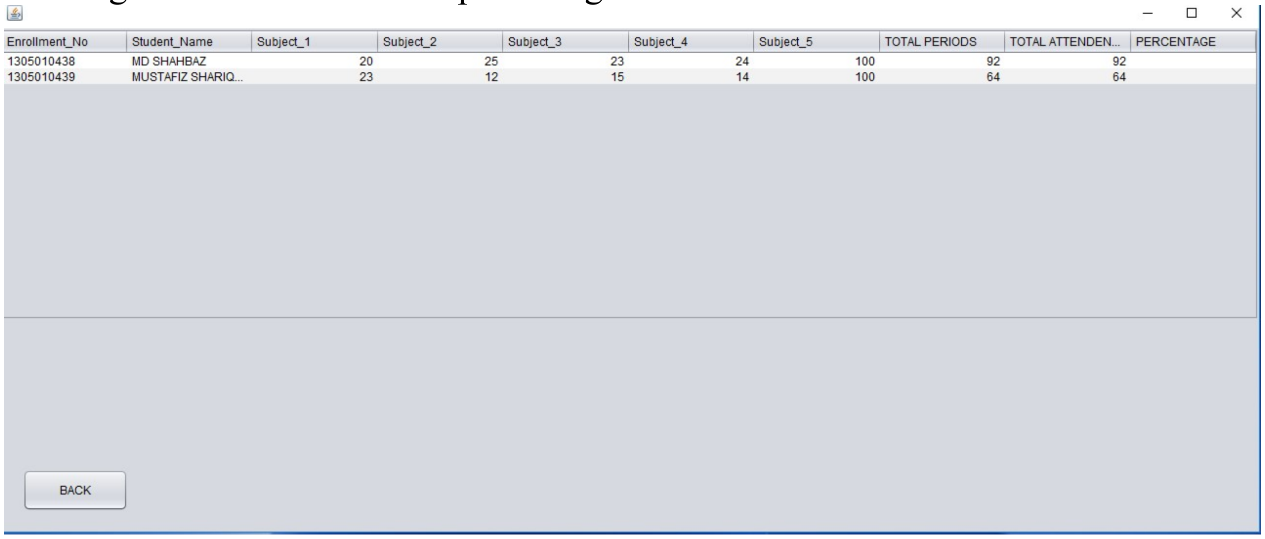


Enrollment_No	Student_Name	Subject_1	Subject_2	Subject_3	Subject_4	Subject_5	TOTAL PERIODS	TOTAL ATTENDENCE	PERCENTAGE
1305010438	MUDASSIR ALI LONE	20	25	23	24	100	92	92	
1305010439	MUSTAFIZ SHARIQUE	23	12	15	14	100	64	64	

Figure 10.1.1

10.1.2) Attendance Details Student

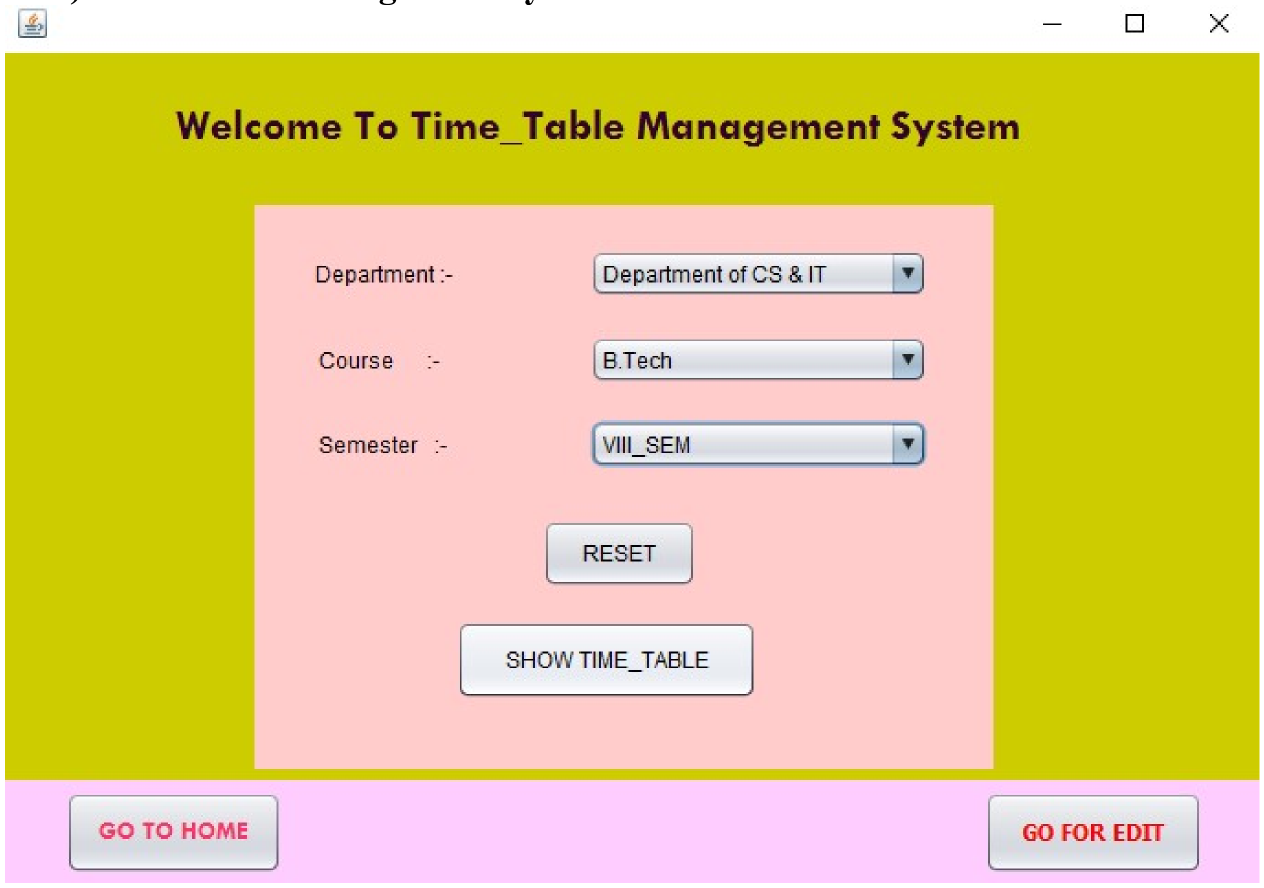
After a student clicks on the Click for Attendance, the following frame opens where he/she gets his/her attendance percentage.



Enrollment_No	Student_Name	Subject_1	Subject_2	Subject_3	Subject_4	Subject_5	TOTAL PERIODS	TOTAL ATTENDEN...	PERCENTAGE
1305010438	MD SHAHBAZ	20	25	23	24	100	92	92	
1305010439	MUSTAFIZ SHARIQ...	23	12	15	14	100	64	64	

Figure 10.1.2

10.2) Time Table Management System


Welcome To Time_Table Management System

Department :- Department of CS & IT

Course :- B.Tech

Semester :- VIII_SEM

Figure 10.2

After choosing the required department, course and semester a student will click on the Show Time_Table to see the tame table of the selected course. And Go For Edit will allow a teacher/admin to edit the Time Table.

10.2.1) Time Table View Student

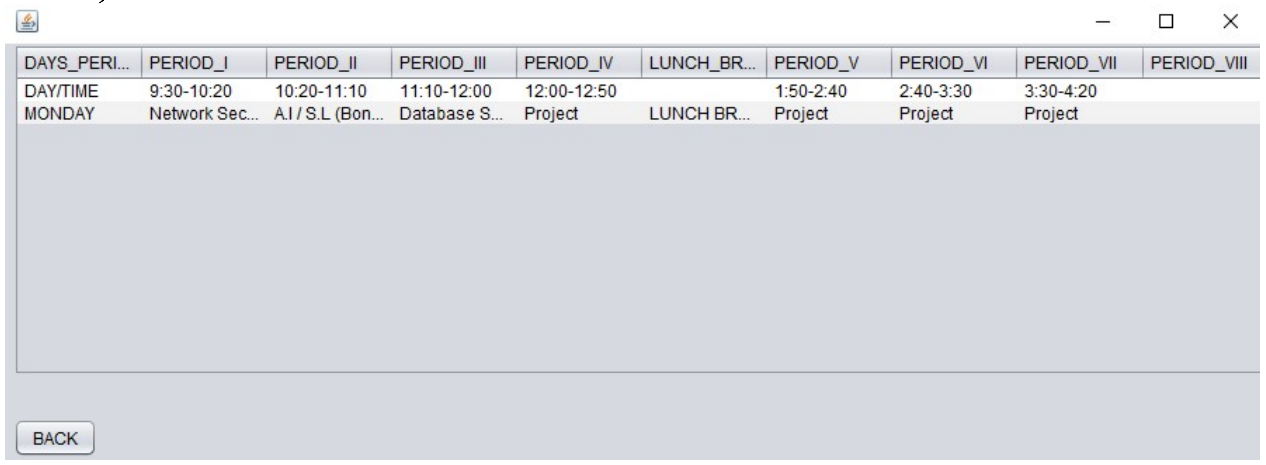
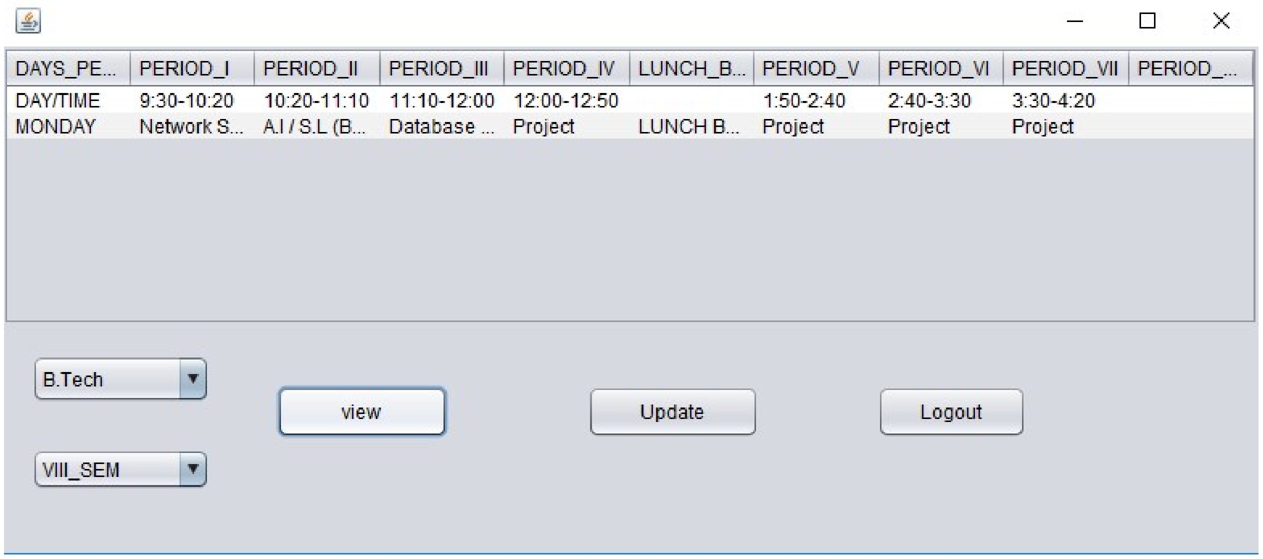

DAYS_PERI...	PERIOD_I	PERIOD_II	PERIOD_III	PERIOD_IV	LUNCH_BR...	PERIOD_V	PERIOD_VI	PERIOD_VII	PERIOD_VIII
DAY/TIME	9:30-10:20	10:20-11:10	11:10-12:00	12:00-12:50		1:50-2:40	2:40-3:30	3:30-4:20	
MONDAY	Network Sec...	A.I / S.L (Bon...	Database S...	Project	LUNCH BR...	Project	Project	Project	

Figure 10.2.1

After clicking on Show Time_Table button the respective time table details will be show to the student.

10.2.2) Time Table View for Teacher/Admin



DAYS_PE...	PERIOD_I	PERIOD_II	PERIOD_III	PERIOD_IV	LUNCH_B...	PERIOD_V	PERIOD_VI	PERIOD_VII	PERIOD_...
DAY/TIME	9:30-10:20	10:20-11:10	11:10-12:00	12:00-12:50		1:50-2:40	2:40-3:30	3:30-4:20	
MONDAY	Network S...	A.I / S.L (B...	Database ...	Project	LUNCH B...	Project	Project	Project	

Figure 10.2.2

After clicking on Show Time_Table button the respective time table details will be show to the student and he/she will also be able to edit/update the Time Table.

CHAPTER 10: REFERENCES

- http://stackoverflow.com/
- https://en.wikipedia.org/wiki/Java_(programming_language)
- https://www.oracle.com/java/index.html
- https://netbeans.org/features/java/
- https://netbeans.org/kb/docs/java/quickstart.html
- https://www.javatpoint.com/java-tutorial
- https://www.w3schools.com/sql/default.asp