

INDEX

- ★ **INTRODUCTION**
- ★ **MOTIVATION**
- ★ **OBJECTIVE**
- ★ **SYSTEM DESIGN**
- ★ **IMPLEMENT**
- ★ **RESULT**
- ★ **LIMITATIONS**
- ★ **FUTURE WORK**
- ★ **CONCLUSION**

Introduction

“Student Payment System” is the most necessary software in today's world. For Digitalization of Bangladesh this is a most important software for any Educational institute of Bangladesh. It will help to do work of a institute easy and secured. This software will help to store students data and give the easy way to use those data for taking any actions.

Motivation

Bangladesh is going digitalized day by day. But as we have seen only some of Educational institute have a digital system of payment. Although we have seen in our college there was this system but in our university which is an International University, but there have no facility like those. That why we thought we created this software.

Objective

The objective of “Student Payment System” is to make easy to work of the management of a Educational institute.

The facilities are :

- ★ Contain Student Data.
- ★ Can Add Students Information
- ★ Store Students Details.

- ❖ Making Students Payment Slip.
- ❖ Print Payment Receipts For Necessary Transaction
- ❖ Make payment system more easier
- ❖ Help to maintain accounts in a easy and digital way

System Design

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

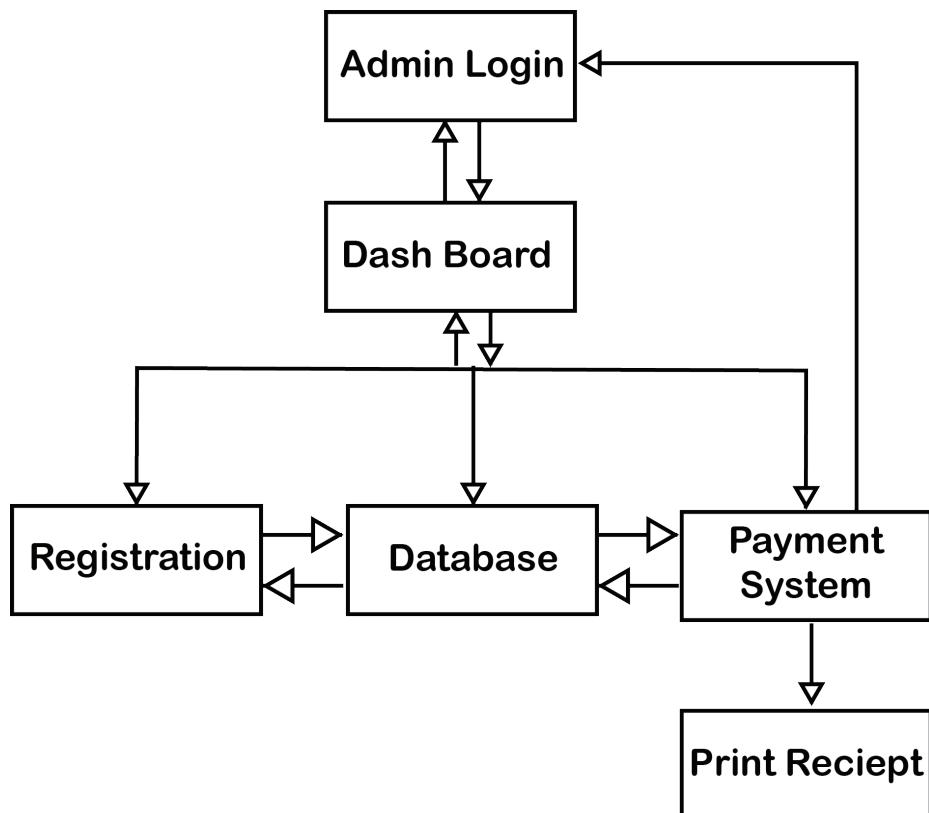


Fig : Simple UML

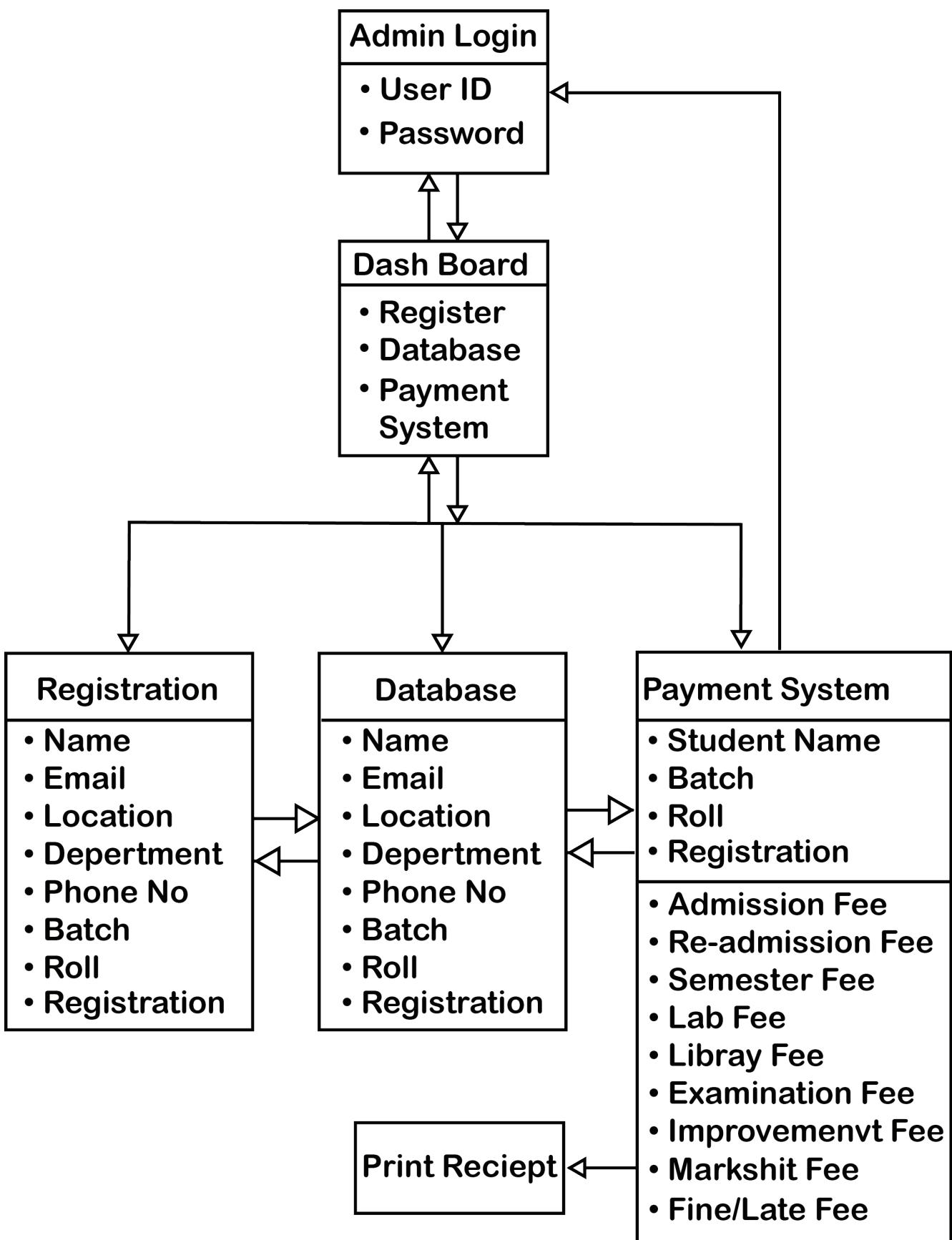


Fig : Complex UML

Implement

Here we will show a full demonstration of how our project “Student Payment System” will work. And also how we implement all kinds of works through our code will be described in this section of development.

Login

In this code we connected to our dashboard. The code is given below :

```
private void loinbuttonMouseClicked(java.awt.event.MouseEvent evt) {  
    //set default password  
  
    String usi = userid.getText();  
    String pas = password.getText();  
    if (usi.equals("admin") && pas.equals("1234")) {  
  
        new DashBoard().setVisible(true);  
        this.dispose();  
        userid.setText(null);  
        password.setText(null);  
  
    } else {  
        JOptionPane.showMessageDialog(null, "Login ID or Password incorrect");  
        userid.setText(null);  
        password.setText(null);  
  
    }  
}
```

Fig : Admin Login Code

Dashboard

In this code we can see our dashboard. and there are three components name “Register”, “Data”, “Payment Slip”.

The code is given below :

```
private void registerMouseClicked(java.awt.event.MouseEvent evt) {  
  
    Register reg = new Register();  
  
    reg.show();  
}  
  
private void paymentMouseClicked(java.awt.event.MouseEvent evt) {  
    home hm = new home();  
    this.dispose();  
    hm.show();  
}  
  
private void DataMouseClicked(java.awt.event.MouseEvent evt) {  
    // JOptionPane.showMessageDialog(null, "This option is underdevelopment");  
    ShowClass Show = new ShowClass();  
    this.dispose();  
    Show.show();  
}
```

Fig : Dashboard Code

Database

The most important part of a management software is Database. In this project we used a demo database. Where we store data from registration form. And we can recall it for our payment system.

The code is given below :

```
final static String driver = "org.apache.derby.jdbc.EmbeddedDriver";  
final static String DB_url = "jdbc:derby://localhost:1527/Reg";  
final static String USER = "Reg";  
final static String PASS = "1234";  
  
public static Connection getConnection(){  
    try{  
        Class.forName(driver);  
        Connection conn = DriverManager.getConnection(DB_url,USER,PASS);  
        System.out.println("Connected");  
        return conn;  
    }  
    catch(ClassNotFoundException | SQLException e){  
        //System.out.println(e);  
        JOptionPane.showMessageDialog(null, e);  
        return null;  
    }  
}
```

Fig : Database Code

Register

Here we have connected to our database with registration form, so that we can store all the student informations in our database.

The code is given below :

```
Connection conn = null;
Statement stmt = null;
ResultSet rs = null;
public Register() {
    super("Register");
    initComponents();
    conn = db.getConnection();
    this.setLocationRelativeTo(null);
```

Fig : Connection to database Code

After Connecting to Database in below code we connected text-fields with database columns.

The code is given below :

```
private void fromMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    try{
        stmt = conn.createStatement();
        String NAME = name.getText();
        String EMAIIL = email.getText();
        String LOCATION = location.getText();
        String DEPT = RegDepertment.getSelectedItem().toString();
        //RegDepertment.addItem(DEP);
        int PHONE = Integer.parseInt(phone.getText());
        String BATCH = batch.getText();
        int ROLL = Integer.parseInt(roll.getText());
        int REG_NO = Integer.parseInt(reg.getText());

        String sql = "INSERT INTO REG1 ( NAME, EMAIIL, LOCATION, DEP, PHONE, BATCH, ROLL, REG_NO)VALUES"
            + "("+NAME+", "+EMAIIL+", "+LOCATION+", "+DEPT+", "+PHONE+", "+BATCH+", "+ROLL+", "+REG_NO+")";

        stmt.executeUpdate(sql);
        JOptionPane.showMessageDialog(null, "Successfully register");
    }
    catch (NumberFormatException | SQLException e) {
        JOptionPane.showMessageDialog(null, "Please fill all the informations");
    }
}
```

Fig : Register value import to database Code

Showing Data

Here we have make our database visible as datatable where user can see every data which is store in database.

The code is given below :

```
Connection conn = null;
Statement stmt = null;
ResultSet rs = null;
public ShowClass() {
    super("REG1");
    initComponents();
    conn = db.getConnection();
    ShowRecord();
    this.setLocationRelativeTo(null);
}

public void ShowRecord(){
try{
    stmt = conn.createStatement();
    String sql = "SELECT * FROM REG.REG1";
    ResultSet res = stmt.executeQuery(sql);
    //JOptionPane.showMessageDialog(null, rs);
    jTable1.setModel(DbUtils.resultSetToTableModel(res));
}
catch (NumberFormatException | SQLException e) {
    JOptionPane.showMessageDialog(null, e);
}
```

Fig : Datatable Code

Payment Receipts

In this section we have implement to Search the data from database by using unique value of registration ID and take database value to other text fields.

The code is given below :

```
private void rnKeyPressed(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:\n    try {\n        //String sql = "SELECT NAME,BATCH,ROLL,REG_NO,DEP FROM REG1 WHERE REG_NO =?";\n        stmt=conn.createStatement();\n        int reger = Integer.parseInt(rn.getText());\n        String sql = "SELECT * FROM REG1 WHERE REG_NO = "+reger+"";\n        rs = stmt.executeQuery(sql);\n        if(rs.next()) {\n            sn.setText(rs.getString("NAME"));\n            String FN = rs.getString("BATCH");\n            bn.setText(FN);\n            String LN = rs.getString("ROLL");\n            rln.setText(LN);\n            String reg = rs.getString("REG_NO");\n            rn.setText(reg);\n            String dep = rs.getString("DEP");\n            d.setText(dep);\n        }\n    } catch (SQLException e) {\n        JOptionPane.showMessageDialog(null, e);\n    }\n}
```

Fig : Searching Registration ID and filling other data from database Code

Here we write the code to make payment receipts of a student. Where we get the amount what have to pay a student. The code is given below :

```
private void getreceiptActionPerformed(java.awt.event.ActionEvent evt) {
    // decoration
    area.setText("*****\n");
    area.setText(area.getText() + "***** Payment Receipt *****\n");
    area.setText(area.getText() + "*****\n");
    //Date
    Date obj = new Date();
    String date = obj.toString();
    area.setText(area.getText() + "" + date + "\n");
    area.setText(area.getText() + "-----\n");
    // items
    area.setText(area.getText() + " Name: " + sn.getText() + "\n");
    area.setText(area.getText() + " Department: " + d.getText() + "\n");
    area.setText(area.getText() + " Batch No: " + bn.getText() + "\n");
    area.setText(area.getText() + " Registration No: " + rn.getText() + "\n");
    area.setText(area.getText() + " Roll no: " + rln.getText() + "\n\n");
    area.setText(area.getText() + " Discription Amounts\n");
    area.setText(area.getText() + "-----\n");
    if (jCheckBox1.isSelected()) {
        area.setText(area.getText() + " Admission Fee....." + ad.getText() + "\n");
    }
    if (jCheckBox18.isSelected()) {
        area.setText(area.getText() + " Re-Admission Fee....." + rad.getText() + "\n");
    }
    if (jCheckBox19.isSelected()) {
        area.setText(area.getText() + " Semester Fee....." + sm.getText() + "\n");
    }
    if (jCheckBox20.isSelected()) {
        area.setText(area.getText() + " Lab Fee....." + labf.getText() + "\n");
    }
    if (jCheckBox21.isSelected()) {
        area.setText(area.getText() + " Library Fee....." + lb.getText() + "\n");
    }
    if (jCheckBox22.isSelected()) {
        area.setText(area.getText() + " Examination Fee....." + ef.getText() + "\n");
    }
    if (jCheckBox23.isSelected()) {
        area.setText(area.getText() + " Improvement Fee....." + im.getText() + "\n");
    }
    if (jCheckBox24.isSelected()) {
        area.setText(area.getText() + " Marks Sheet Fee....." + msf.getText() + "\n");
    }
    if (jCheckBox25.isSelected()) {
        area.setText(area.getText() + " Fine/Late fee....." + ff.getText() + "\n");
    }
    area.setText(area.getText() + "-----\n");

    //bill
    area.setText(area.getText() + " Total Cost      " + tc.getText()+"BDT" + "\n");
    area.setText(area.getText() + "-----\n\n");
}
```

Fig : Payment Receipts Code

Some Important Codes

Here are some important codes which are used in our project.
The code is given below :

```
private void jCheckBox1ActionPerformed(java.awt.event.ActionEvent evt) {  
    if (jCheckBox1.isSelected()) {  
        ad.setEditable(true);  
        ad.setText("20000");  
        ad.requestFocus();  
    } else {  
        ad.setEditable(false);  
        ad.setText("0");  
    }  
}
```

Fig : CheckBox Click Perform Code

```
private void printreciptActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        area.print();  
    } catch (Exception e) {  
    }  
}
```

Fig : Print button Click Perform Code

```
private void totalActionPerformed(java.awt.event.ActionEvent evt) {  
    double EFR = Double.parseDouble(ad.getText());  
    double BM = Double.parseDouble(ff.getText());  
    double CC = Double.parseDouble(cf.getText());  
    double CF = Double.parseDouble(lb.getText());  
    double CFR = Double.parseDouble(rad.getText());  
    double CS = Double.parseDouble(im.getText());  
    double CSK = Double.parseDouble(msf.getText());  
    double PFR = Double.parseDouble(sm.getText());  
    double SFR = Double.parseDouble(labf.getText());  
    //.....calculation.....  
    double total =BM +CC+CF+CFR+CS+CSK+EFR+PFR+SFR ;  
    //.....show values in text field.....  
    String item_total = String.format("%3f", total);  
    tc.setText((String) item_total);  
}
```

Fig : Total Calculation button Click Perform Code

```
private void logoutMouseClicked(java.awt.event.MouseEvent evt) {  
    frame = new JFrame("Exit");  
    if ( JOptionPane.showConfirmDialog(frame, "Are you sure you want to Logout !",  
        "Logout", JOptionPane.YES_NO_OPTION) == JOptionPane.YES_NO_OPTION) {  
        login log = new login();  
        this.dispose();  
        log.show();  
    }  
}
```

Fig : Logout Click Perform Code

```
//ADD TRASSPARENT TEXTFIELD  
jPanell.setBackground(new Color(0,0,0,0));  
reg.setBackground(new Color(0,0,0,0));  
name.setBackground(new Color(0,0,0,0));  
email.setBackground(new Color(0,0,0,0));  
location.setBackground(new Color(0,0,0,0));  
RegDepertment.setBackground(new Color(0,0,0,0));  
phone.setBackground(new Color(0,0,0,0));  
batch.setBackground(new Color(0,0,0,0));  
roll.setBackground(new Color(0,0,0,0));
```

Fig : Transparent text field Code

Result

Result is the Graphical interface part of a project.
below here is our “Student Payment System” projects Results:

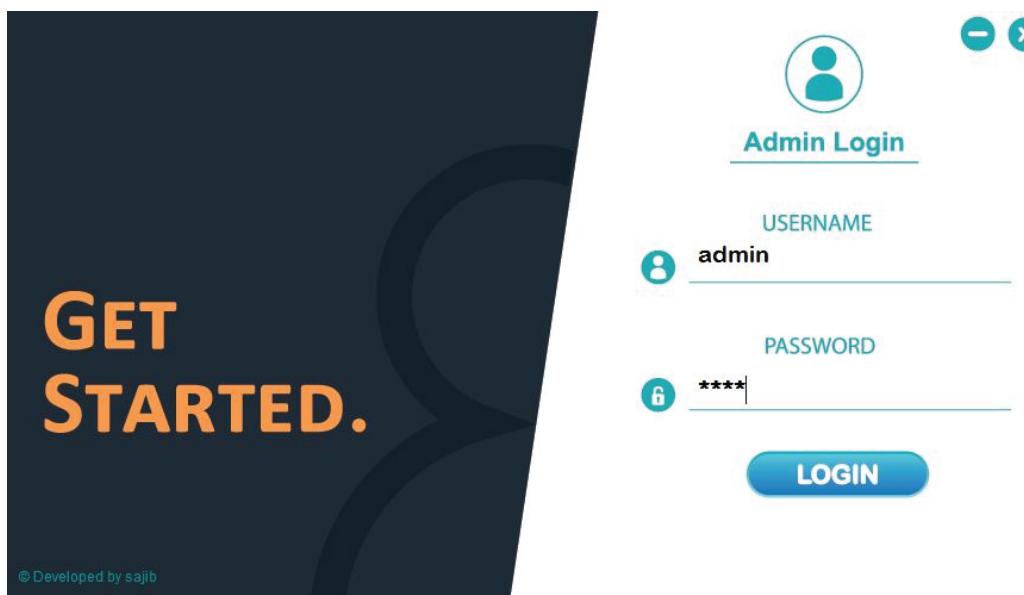


Fig : Admin Login



Fig : Dashboard

REGISTER NEW ACCOUNT

Name

Email

Location

Department

Phone

Batch No

Roll

Reg No

REGISTER

Fig : Register form

Student Payment System



Student Name :			
Registration No:			
Batch No:			
Roll No:			
Department:			
<input type="checkbox"/> Admission Fee	0		
<input type="checkbox"/> Re-Admission Fee	0		
<input type="checkbox"/> Semester Fee	0		
<input type="checkbox"/> Lab Fee	0		
<input type="checkbox"/> Library Fee	0		
<input type="checkbox"/> Examination Fee	0		
<input type="checkbox"/> Improvement Fee	0		
<input type="checkbox"/> Marks Sheet Fee	0		
<input type="checkbox"/> Fine/ Late Fee	0		
Total Cost			
Total			
Reset			

Get Receipt **Print Receipt**

Logout

© Developed by sajib

***** * Payment Receipt *

Tue Nov 27 23:09:43 BDT 2018

Name: Shamsul Arefin
Department: B.SC. IN CSE (DAY)
Batch No: D43
Registration No: 104244
Roll no: 3

Discription Amounts

Semester Fee..... 24500
Examination Fee..... 500
Marks Sheet Fee..... 500

Total Cost 25500.000000BDT

Fig : Payment form & Receipt

REG1							
NAME	EMAIL	LOCATION	PHONE	BATCH	ROLL	REG_NO	DEP
Sikder Sajib	sajib.rw99@gmail.c...	Dhaka	1521435666	43	34	102488	
Md. Tuhin	mdtuhinjoy43@gm...	Daudkandi, Cumilla	1953107214	43	10	104311	
Afsana Akther ALl_o	afsanaazziz768@gm...	Mymensingh	1960849675	43	29	104408	
Shamsul Arefin	shamsularefin500...	Dhaka, Bangladesh	1786832822	D43	3	104244	B.Sc. IN CSE (DAY)
Sikder Sajib Al Sha...	sajib.rw99@gmail.c...	Jatrabari, Dhaka	1521435666	D-43	34	104428	B.Sc. IN CSE (DAY)

Fig : Datatable of Database

Back

Limitations

There are everything have some of limitations.
here is some limitation of our software:

- Its is not fully user friendly.
- Only one admin can access.
- Database is not Online.
- Can use only one user at a time.
- Not ready for industrial purpose.
- Need to improve user interface.
- Not well protected.
- Software is not portable.

Future Work

Here is some of works what we want to implement in our software in future to work smoothly:

- Have to make it fully user friendly.
- Have to make it multiple admin accessable.
- Have to make it Online.
- Have to make it multiple admin accessable at a time.
- Have to make it ready for industrial user.
- Have to make it improve user interface.
- Have to make it well protected.
- Have to make the Software portable.

Conclusion

Overall the project will be a good implementation for payment of an educational institute. Though it have some limitations but we can improve it and make it useful for farther use.