



STUDENT MANAGEMENT SYSTEM

Programming Project 2022



JANUARY 7, 2022

ABDELRAHMAN RASLAN – REWAN SALAH
20221460102 - 20221447143

1. Classes:

StudentMangeApp

StudentBack

Operation

operation2

2. Libraries:



java.util.Scanner;



java.sql.Connection;



java.sql.DriverManager;



java.sql.ResultSet;



java.sql.Statement;



java.sql.ResultSetMetaData;



java.sql.SQLException;

3. Method:

1) StudentMangeApp

Main

- Welcome & ask for number of Student Want to add

```
Scanner sc=new Scanner(System.in);
System.out.println("-----Welcome-----");

//Number of Student can enroll
System.out.print("Enter number of new Student to Enroll : ");
int numstd=sc.nextInt();
StudentBack[] stud=new StudentBack[numstd];
```

- Check number of Student if more than 20

```
20 // 20 student max
21 if (numstd<=20 && numstd>0)
22 {
```

```
-----Welcome-----
Enter number of new Student to Enroll : 2
```

- Repeat operations by loop until the number of students entered

```
//loop for num of students
for (int i=0;i<numstd;i++)
{
```

- Enter personal information and call constructor

```
26 // constructor : name
27 System.out.print("Enter the First Name of Student : ");
28 String fris= sc.next();
29 System.out.print("Enter the Last Name of Student : ");
30 String las= sc.next();
31 System.out.print("Enter the Year that Student join FCDS : ");
32 int lev=sc.nextInt();
33 stud[i]=new StudentBack(fris,las,lev);
```

```
-----Welcome-----
Enter number of new Student to Enroll : 2
Enter the First Name of Student : Abdelrahman
Enter the Last Name of Student : Raslan
Enter the Year that Student join FCDS : 2022
```

- Ask for many of courses & check by enroll()

```

34 //enroll
35 System.out.print("How many Course do you want to enroll : ");
36 int n = sc.nextInt();
37 stud[i].enroll(n);
38

```

```

How many Course do you want to enroll : 2
The minimum enroll is 3 courses
Try Again ,
How many Course do you want to enroll : 3
Enter Course You Want to Enroll
Course 1 : programming
Course 2 : Intro
Course 3 : data Science

```

- Enter Payment that student pay & Show new Balance

```

39 //payment
40 System.out.println("\n-----Payment-----");
41 System.out.print("Enter Your Payment :");
42 int payment=sc.nextInt();
43 stud[i].PayTuitionFees(payment);

```

```

-----Payment-----
Enter Your Payment :200
Your Payment : 200 .LE
Your Certain Balance is : 700 .LE

```

- Show Statue of Student

```

//statue
System.out.println("-----Statues----- \n"+stud[i].toString(Studid));

```

```

-----Statues-----
Student Name : Rewan Salah
Student ID : 20221001
Your Balance :700

```

- Add information to DB & Show Statue in Database

```

6 //Database
7 operation.connection(idd,fris,las,lev,Studid,bal);
8 operation2.connection();
9

```

```

done connection
done connection 2
idu      FirstName  FamilyName  StuId      StuLevl    TotalBalance  EachCourseCost
1001     Rewan         Salah       20221001   2022       700           300

```

2) StudentBack

```
1 package Project2022;//Abdelrahman - Rewan
2 import java.util.Scanner;
3 public class StudentBack {
4
5     //Variable
6     private final String FirstName;
7     private final String FamilyName;
8     public String[] CourseName;
9     private String StuId;
10    private final int StuLevl;
11    private int TotalBalance =0;
12    private static final int EachCourseCost =300;
13    private static int id=1001;
14 }
```

Variable

- private final String FirstName; → First Student name
- private final String FamilyName; → Last Student name
- public String[] CourseName; → List Contains Courses
- private String StuId; → final student id(year+unique id)
- private final int StuLevl; → Year that Student join faculty
- private int TotalBalance =0; → Balance Counter
- private static final int EachCourseCost =300; → 300 L.E for every credit hour
- private static int id=1001;→ Counter to make unique id

1. public StudentBack(String f1, String f2, int l1)

```
15      // constructor : name
16      public StudentBack(String f1, String f2, int l1)
17      {
18          this.FirstName=f1;
19          this.FamilyName=f2;
20          this.StuLevl=l1;
21          setStuId();
22          createID();
23      }
24
```

- This constructor takes First name & Last name & Year that Student join faculty then save in new variables
 - After that call setStuId(),createID() that will be explained
-

2. private void setStuId()

```
25      //student ID
26      private void setStuId()
27      {
28          this.StuId = StuLevl + "" + (id);
29          StudentMangeApp.Studid = StuId;
30      }
31
```

- This function takes Year that Student join faculty
- and add in string type to id that set as (private static int id=1001;)
- and add this string value to Studid

3. public synchronized void createID()

```
33      public synchronized void createID()
34      {
35          StuId=String.valueOf(id++);
36          StudentMangeApp.idd=StuId;
37      }
38
```

- This function create unique id by add 1 every time when function is called then set id in (idd)

4. public void enroll(int n)

```
39      //Enroll students courses
40      public void enroll(int n)
41      {
42          Scanner sc=new Scanner(System.in);
43          if (n >= 3)
44          {
45              System.out.println("Enter Course You Want to Enroll ");
46              CourseName = new String [n];
47              for(int i=0; i<CourseName.length; i++)
48              {
49                  System.out.print("Course " +(i+1)+" : ");
50                  CourseName[i]=sc.next();
51              }
52              TotalBalance=EachCourseCost*(n);
53          }
54          else
55          {
56              System.out.println("The minimum enroll is 3 courses ");
57              Scanner uu=new Scanner(System.in);
58              System.out.print("Try Again , \nHow many Course do you want to enroll : ");
59              int jj=uu.nextInt();
60              enroll(jj);
61              TotalBalance=EachCourseCost*(jj);
62          }
63      }
64
65
```

- this function in the first take times of courses that user input in main to check that courses more than 3 courses when courses is equals or more than 3 the function ask user to input the name of cursers
- then calculate total balance by multiple 300 L.E for every credit hour and number of courses that is first condition (if) ,
- second condition (else) print warning message for user and ask to try again then call enroll(); again to perform the same operations until the user enters the correct number of courses and calculate balance again

5. public void ShowBalance()

```
67 //Balance
68 public void ShowBalance()
69 {
70     System.out.println("Your Certain Balance is : "+TotalBalance+" .LE");
71 }
72
```

- This is a simple function that prints the total amount that the student will pay that was calculated before

6. public void PayTuitionFees(int pay)

```
73 //pay tuition fees
74 public void PayTuitionFees(int pay)
75 {
76     TotalBalance=TotalBalance-pay;
77     System.out.println("Your Payment : "+pay+ " .LE");
78     ShowBalance();
79     StudentMangeApp.bal=TotalBalance;
80 }
81
```

- This is a function that takes from the user in the main the amount that the student has already paid before
- and subtracts it from the total amount of the courses and prints the new amount that the user should to pay and put it in a new variable

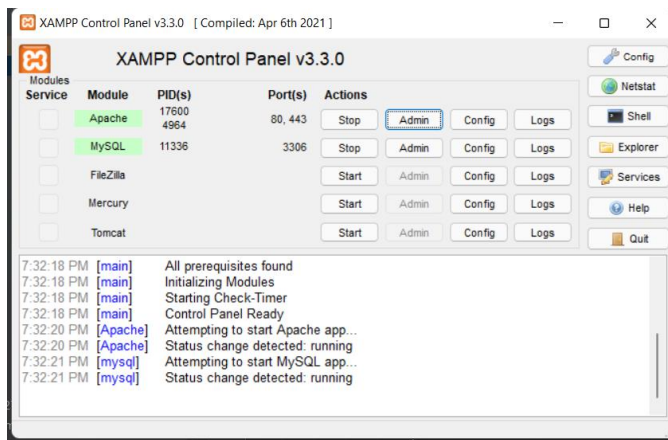
7. public String toString()

```
82 //show status.
83 public String toString()
84 {
85     return "Student Name : " + FirstName + " " + FamilyName + "\n" + "Student ID : " + StuId + "\n" + "Your Balance : " + TotalBalance;
86 }
87
88
```

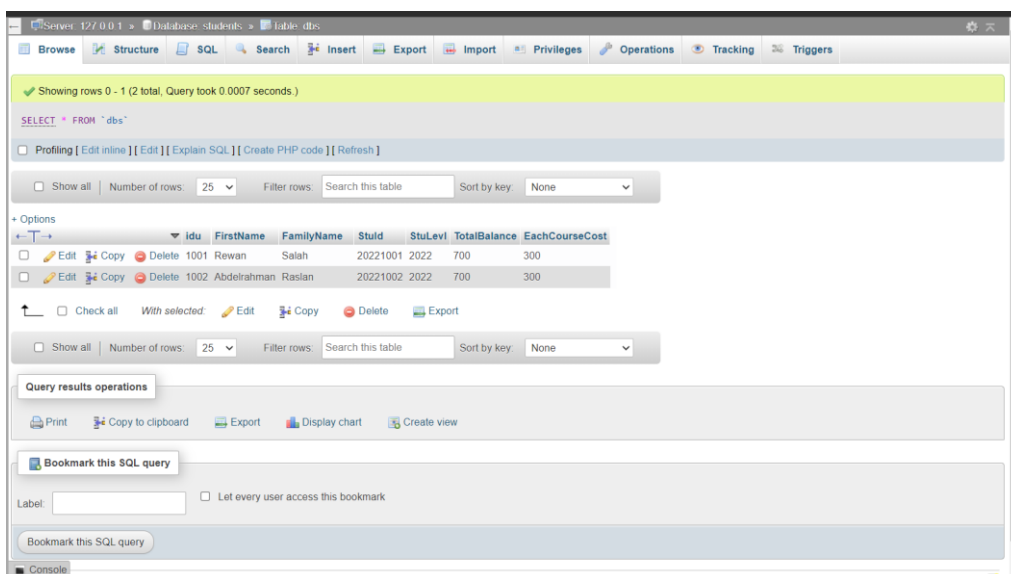
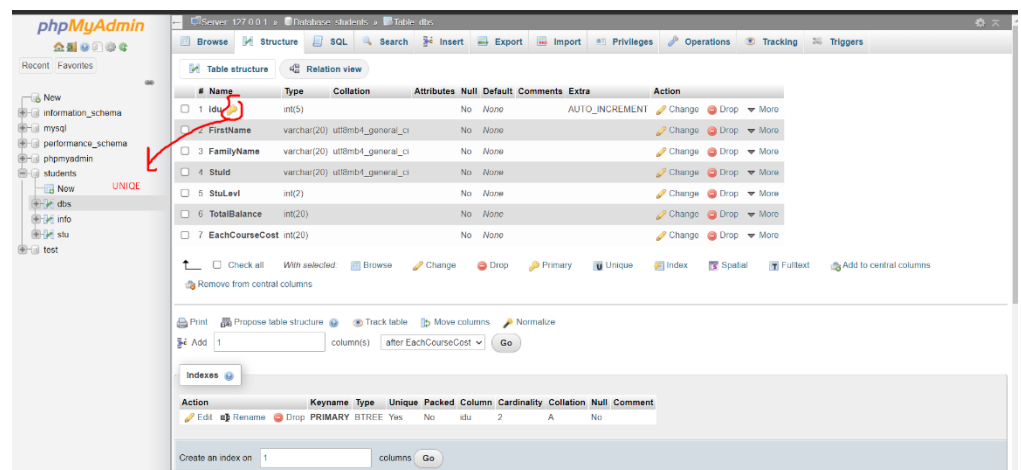
- This is a function called by this name as a kind of security because there is a built in function in java that shows all the variables in the program, this is the reason for the name,
- and as of the function, it shows all the information about the student that was collected from the user in addition to the one that was calculated

3) Data Base

- Use XAMPP for start server for database :



- MySQL DB (IN PHP ADMIN)



1. public static void connection(String oo,String f , String la, int o, String ii,int bb)

```
1 package Project2022;
2 import java.sql.*;
3 public class Operation {
4     public static void connection(String oo,String f , String la, int o, String ii,int bb){
5         try {
6             Connection c = DriverManager.getConnection(
7                 url: "jdbc:mysql://localhost/students?serverTimezone=UTC",
8                 user: "Rewan",
9                 password: "Rewan2002"
10            );
11            System.out.println("done connection");
12            Statement s = c.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
13            ResultSet r = s.executeQuery( sql: "SELECT `idu`, `FirstName`, `FamilyName`, `StuId`, `StuLevl`, `TotalBalance`, `EachC"
14            // ResultSetMetaData rm = r.getMetaData();
15            r.previous();
16            r.moveToInsertRow();
17            //Scanner sc =new Scanner(System.in);
18            r.updateString( columnLabel: "idu",oo);
19            r.updateString( columnLabel: "FirstName",f);
20            r.updateString( columnLabel: "FamilyName",la);
21            r.updateString( columnLabel: "StuId",ii);
22            r.updateInt( columnLabel: "StuLevl",o);
23            r.updateInt( columnLabel: "TotalBalance",bb);
24            r.updateInt( columnLabel: "EachCourseCost", x: 300);
25            r.insertRow();
26        }
27        catch (SQLException e) {
28            e.printStackTrace();
29        }
30    }
31 }
```

- This function is about linking the java code to the database
- Connection c → contain (url of mysql server & user & password)
- ResultSet r → write sql on it to select columns of table in DB
- r.previous() → To make the cursor enter from the bottom of the table for easy access to it in the class operation2
- use Parameters to update in every row and column
- Add each element to the data base in the specified row and column

2. public static void connection()

```
9 public static void connection(){
10     try {
11         Connection c = DriverManager.getConnection(
12             url: "jdbc:mysql://localhost/students?serverTimezone=UTC",
13             user: "abdel",
14             password: "12345aaaaa"
15         );
16         System.out.println("done connection 2");
17         Statement s = c.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
18         ResultSet r = s.executeQuery("SELECT `idu`, `FirstName`, `FamilyName`, `StuId`, `StuLevl`, `TotalBalance`, `EachCourseCost`");
19         ResultSetMetaData rm = r.getMetaData();
20         r.next();
21         for (int i=1; i<=rm.getColumnCount(); i++){
22             System.out.printf(
23                 "%-15s ", rm.getColumnName(i)
24             );
25         }
26         System.out.println("\n");
27         r.last();
28         System.out.printf(
29             "%-15s %-15s %-15s %-15s %-15s %-15s %-15s\n",
30             r.getString( columnLabel: "idu"),
31             r.getString( columnLabel: "FirstName"),
32             r.getString( columnLabel: "FamilyName"),
33             r.getString( columnLabel: "StuId"),
34             r.getInt( columnLabel: "StuLevl"),
35             r.getInt( columnLabel: "TotalBalance"),
36             r.getInt( columnLabel: "EachCourseCost")
37         );
38     } catch (SQLException e) {
39         e.printStackTrace();
40     }
41 }
```

- This function is in Class operation2 The main task is to link back to the Database and show the student's status more clearly.
- ResultSetMetaData rm → for show type and name for columns in this project
- Loop in rm.getColumnName(i) to show all columns name
- r.last() → To make sure the cursor is in the last row
- Print each item in the specified row

We hope that you will like the program

Source Code :

<https://drive.google.com/drive/folders/1jhp0SmUdgkg-JYN1-gjaBoQvhjqxNVI?usp=sharing>

Thank You

