RAMAIAH UNIVERSITY OF APPLIED SCIENCES

ASSIGNMENT

Course Code	CSC311A
Course Name	Database Systems
Programme	B. TECH
Department	CSE
Faculty	FET

Name of the Student	AISHWARYA
Reg. No	17ETCS002015
Semester/Year	6TH SEM/ 3RD YEAR
Course Leader/s	Ami Rai E.

:

RAMAIAH UNIVERSITY OF APPLIED SCIENCES

Declaration Sheet				
Student Name	AISHWARYA			
Reg. No	17ETS002015			
Programme	B. TECH		Semester/Year	6TH SEM/ 3RD YEAR
Course Code	CSC311A			
Course Title	Database Systems			
Course Date		to		
Course Leader	Ami Rai E.			

Declaration

The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Handbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly.

Signature of the Student		Date	27/04/2020
Submission date stamp (by Examination & Assessment Section)			

Signature of the Course Leader and date	Signature of the Reviewer and date

RAMAIAH UNIVERSITY OF APPLIED SCIENCES

Faculty of Engineering & Technology			
Ramaiah University of Applied Sciences			
Department	Computer Science and Engineering	**Programme**	B. Tech. Computer Science and Engineering
Semester/Batch	6th /2017		
Course Code	CSC311A	**Course Title**	Database Systems
Course Leader(s)	A. Prabhakar, Gp Capt N Rath VSM, Ami Rai E.		

Assignment			
Register No.	**17ETCS002015**	Name of Student	**AISHWARYA**

Sections		**Marking Scheme**	**Max Marks**	**First Examiner Marks**	**Second Examiner Marks**
Part A	A.1	Functional and data requirements	02		
	A.2	Implementation of database tables	03		
	A.3	Implementation of the GUI	02		
	A.4	Connection of front end with the database	02		
	A.5	Conclusion	01		
		Total Assignment Ma	**10**		

Course Marks Tabulation				
Component- 1(B) Assignment	**First Examiner**	**Remarks**	**Second Examiner**	**Remarks**
A				
Marks (out of 10)				

Signature of First Examiner

Signature of Second Examiner

A.1 List of functional and data requirements

The function requirements for the project exhibition system for RUAS are: -

FR1: The system should allow the user to login to the system as staff or student after the user input his/her credentials.

Dependent on Requirements	--
Stake Holder Owning the Requirement	End User
Example of user/system interaction for this requirement	The user whether student or staff should be able to login to the system using the login ID and password

Requirement Tag	DR1
Item Name	select
Item Description (Where/How used)	The user will select the account type for login.
Item type	Boolean
User/System interacting with the item	user
Constraints (if any)	Either of the options should be selected.

Requirement Tag	DR2
Item Name	loginID
Item Description (Where/How used)	The user will be entering his/her loginID.
Item type	Char
User/System interacting with the item	user
Constraints (if any)	The should be provided.

Requirement Tag	DR3
Item Name	password
Item Description (Where/How used)	The user will be entering his/her password
Item type	Char
User/System interacting with the item	user
Constraints (if any)	The value should match with the one stored in database with loginID.

FR2: The system should allow the user to change the password to log into the system.

Dependent on Requirements	--
Stake Holder Owning the Requirement	End User
Example of user/system interaction for this requirement	The user either student or staff should be able to change the password by using forget password option.

Requirement Tag	DR4
Item Name	Change password
Item Description (Where/How used)	The user will be able to change password
Item type	Char
User/System interacting with the item	user
Constraints (if any)	---------

FR3: the system should allow the user to register for the project exhibition using the details such as mentor name, project name, project ID, team member name and ID.

Dependent on Requirements	FR1
Stake Holder Owning the Requirement	End User
Example of user/system interaction for this requirement	If the user wants to register the project the user can do the registration.

Requirement Tag	DR5
Item Name	Project name
Item Description (Where/How used)	The user will be entering his/her project name
Item type	Char
User/System interacting with the item	user
Constraints (if any)	The value should be characters and required.

Requirement Tag	DR6
Item Name	Project ID
Item Description (Where/How used)	The user will be entering his/her project name
Item type	Char

User/System interacting with the item	user
Constraints (if any)	The value should be characters.

Requirement Tag	DR7
Item Name	group name
Item Description (Where/How used)	The user will be entering his/her project group name
Item type	Char
User/System interacting with the item	user
Constraints (if any)	The value should be characters and required.

Requirement Tag	DR8
Item Name	Mentor name
Item Description (Where/How used)	The user will be entering his/her project mentor name
Item type	Char
User/System interacting with the item	user
Constraints (if any)	The value should be characters and required.

Requirement Tag	DR9
Item Name	Team leader name
Item Description (Where/How used)	The user will be entering project team leader's name
Item type	Char
User/System interacting with the item	user
Constraints (if any)	The value should be characters and required.

Requirement Tag	DR10
Item Name	students name
Item Description (Where/How used)	The user will be entering project group participants name
Item type	Char
User/System interacting with the item	user
Constraints (if any)	The value should be characters and required.

Requirement Tag	DR11
Item Name	students ID

Item Description (Where/How used)	The user will be entering project group participants ID
Item type	Char
User/System interacting with the item	user
Constraints (if any)	The value should be characters and required.

FR4: the system should display the registered project details along with the room number and the table number allotted.

Dependent on Requirements	FR1
Stake Holder Owning the Requirement	End User
Example of user/system interaction for this requirement	The details of the project registered and the room number and the table number registered will be displayed to the user.

FR5: the system should be able to update the details of the project registered.

Dependent on Requirements	FR2
Stake Holder Owning the Requirement	End User
Example of user/system interaction for this requirement	The user should be able to modify details of the project registered.

FR6: the system should be able to display all the details of all the projects he/she is mentor for.

Dependent on Requirements	FR1
Stake Holder Owning the Requirement	End user
Example of user/system interaction for this requirement	User is allowed to view the projects registered under him/her.

FR7: the system should allow the user to delete the registration made for the project exhibition.

Dependent on Requirements	FR3, FR5
Stake Holder Owning the Requirement	End user
Example of user/system interaction for this requirement	The system should allow the staff/student to logout and redirect them to the login page. If a particular group cancels the registration, they should be redirected to the login page.

A.2 Implementation of relational database schema with appropriate attributes, and constraints using SQL commands

The relational database implementation of the system using SQL command are as follows:

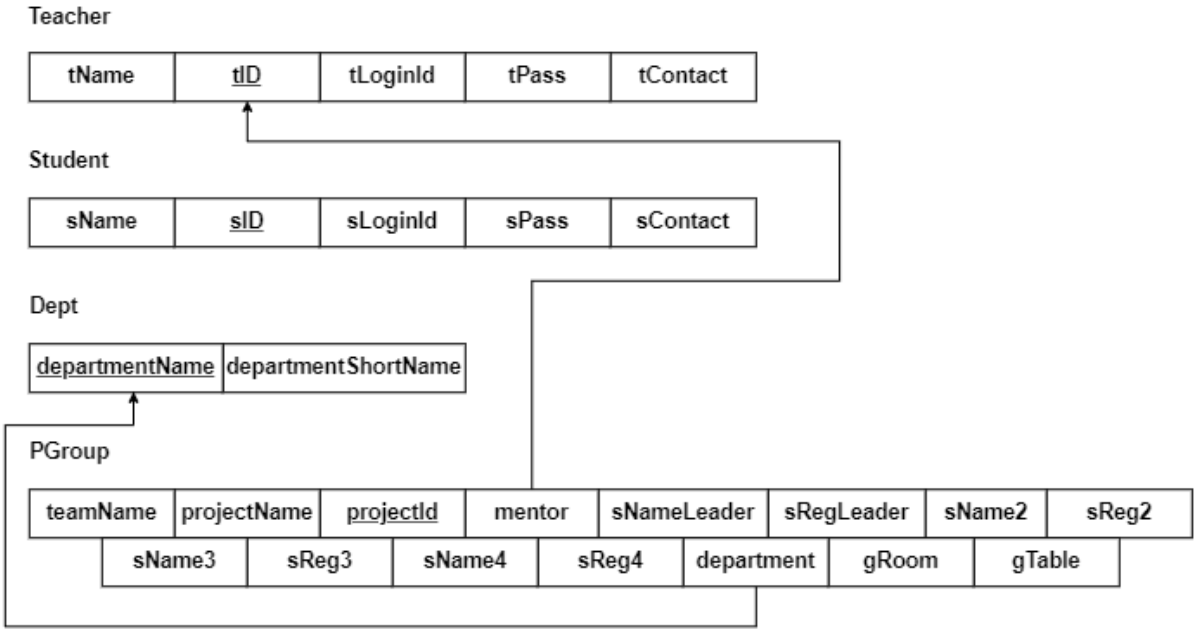
```
--DBMS Assignment
--create table teacher
create table teacher(
tName char(20) not null,
tID   char(20) not null,
tLoginId char(20) not null,
tPass char(20) not null,
tContact char(12) not null,
primary key(tID),
unique key(tLoginId),
unique key(tContact)
);
--create table student
create table student(
sName char(20) not null,
sID char(20) not null,
sLoginId char(20) not null,
sPass char(20) not null,
sContact char(12) not null,
primary key(sID),
unique key(sLoginId),
unique key(sContact)
);
--create table department
create table dept(
departmentName          char(20) primary key,
departmentShortName     char(10)
);

--create table pGroup
create table pGroup(
teamName    varchar(40) not null,
projectName varchar(40) not null,
projectId   char(20) not null,
mentor      char(20) not null,
sNameLeader char(20) not null,
sRegLeader  char(20) not null,
sName2      char(20) not null,
sReg2       char(20) not null,
sName3      char(20) not null,
sReg3       char(20) not null,
sName4      char(20) ,
sReg4       char(20) ,
department  char(30) not null,
gRoom       char(20) not null,
gTable      int not null,
primary key(projectId),
foreign key(mentor) references teacher(tID),
foreign key(department) references dept(departmentName)
);
-- tables in the project
show tables;
-- description of tables
desc pGroup;
desc teacher;
desc dept;
```

```
--inserting values to the teacher table
insert into teacher values('Rinki Sharma','T001','rinkisharma001','rinki','7894561230');
insert into teacher values('Deepak v.','T009','deepv009','deep','7204306360');
insert into teacher values('Vivek Oberoi','T007','viveko009','vivek','9304569870');
insert into teacher values('Ritesh A','T005','ritz005','ritesh','9904630220');
insert into teacher values('Ami_Rai_E','T002','ami002','rami','7014130635');

--inserting values to the student table
insert into student values('Aishwarya','17ETCS02015','aishwaryajmp1998','aishu','9304390800');
insert into student values('Amresh','17ETCS02115','amresh007','sunshine','9128657860');
insert into student values('Riya Panda','17ETCS02375','pandariya02','pandar','7209274229');
insert into student values('kartik','17ETCS02175','kartikch','karti','9907722440');
insert into student values('pratyeksha','17ETCS02116','cutiep200','golu','9304392877');

--inserting values to the department
insert into dept values("Aero Space","ASE");
insert into dept values("Civil","CE");
insert into dept values("Computer Science","CSE");
insert into dept values("Electrical","EEE");
--display of the content of all the table
select * from pGroup;
select * from teacher;
select * from dept;
```



PGroup is shown as above because of greater number of columns

Contents of tables

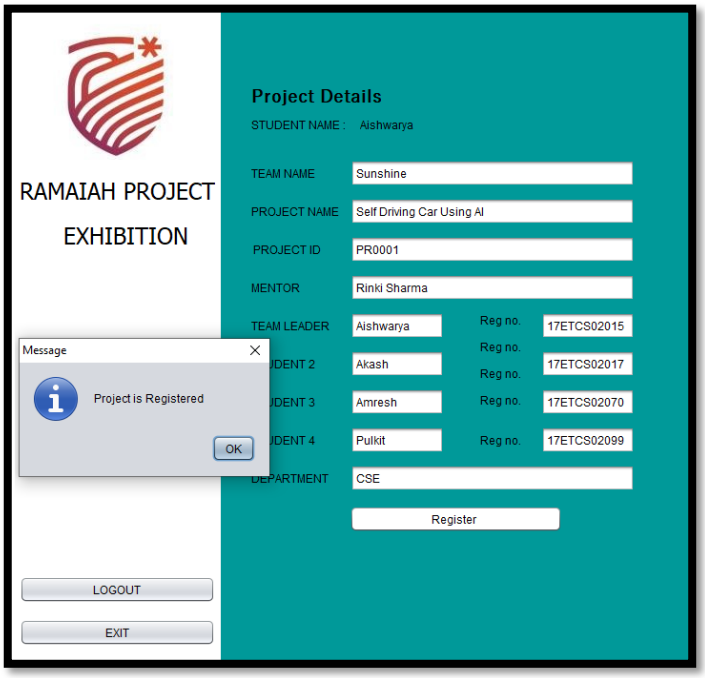
```
mysql> select * from dept;
+------------------+----------------------+
| departmentName   | departmentShortName  |
+------------------+----------------------+
| Aero Space       | ASE                  |
| Civil            | CE                   |
| Computer Science | CSE                  |
| Electrical       | EEE                  |
+------------------+----------------------+
4 rows in set (0.00 sec)
```

```
mysql> select * from student;
+------------+-------------+------------------+----------+------------+
| sName      | sID         | sLoginId         | sPass    | sContact   |
+------------+-------------+------------------+----------+------------+
| Aishwarya  | 17ETCS02015 | aishwaryajmp1998 | aishu    | 9304390800 |
| Amresh     | 17ETCS02115 | amresh007        | sunshine | 9128657860 |
| pratyeksha | 17ETCS02116 | cutiep200        | golu     | 9304392877 |
| kartik     | 17ETCS02175 | kartikch         | karti    | 9907722440 |
| Riya Panda | 17ETCS02375 | pandariya02      | pandar   | 7209274229 |
+------------+-------------+------------------+----------+------------+
5 rows in set (0.00 sec)
```

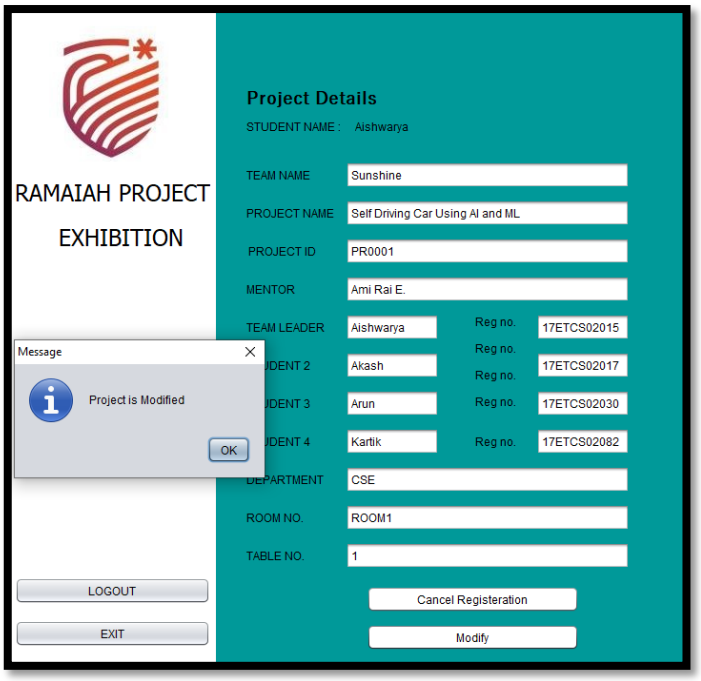
```
mysql> select * from teacher;
+--------------+------+----------------+--------+------------+
| tName        | tID  | tLoginId       | tPass  | tContact   |
+--------------+------+----------------+--------+------------+
| Rinki Sharma | T001 | rinkisharma001 | rinki  | 7894561230 |
| Ami_Rai_E    | T002 | ami002         | rami   | 7014130635 |
| Ritesh A     | T005 | ritz005        | ritesh | 9904630220 |
| Vivek Oberoi | T007 | viveko009      | vivek  | 9304569870 |
| Deepak v.    | T009 | deepv009       | deep   | 7204306360 |
+--------------+------+----------------+--------+------------+
5 rows in set (0.00 sec)
```

A.3 Implementation of GUI with options such as login, registration, updating, and cancellation

The above screenshot is of the login page of the RUAS project exhibition, where the staff or the student is able to login using the loginID and password. The user should select the account type provided must be either staff or student. The option to change the password is also provided to the user.



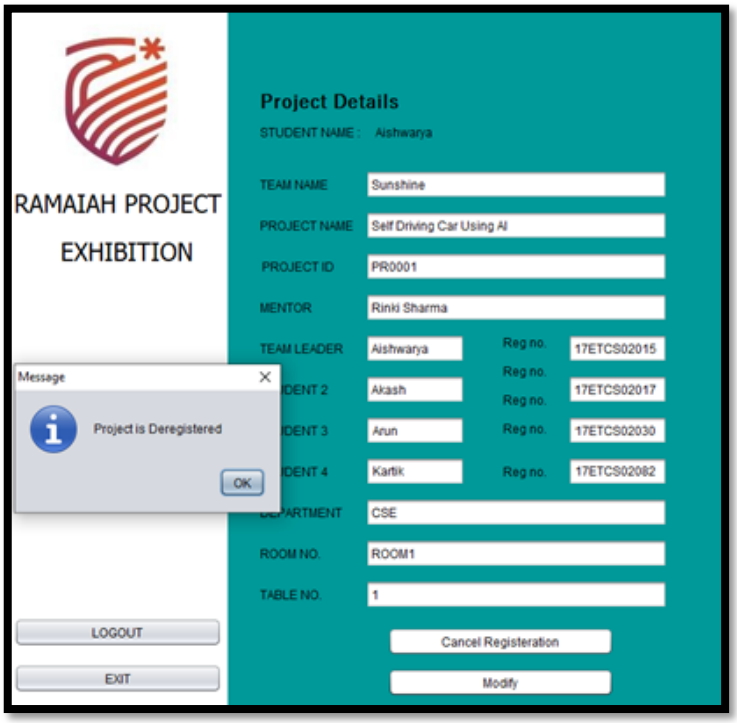
The above screenshot represents the welcome and registration page for the user, where the user is able to do registration for the project, by entering the credentials for project such as project name, ID, Mentor, team leader, participating students in group and their respective registration numbers, and the department.



The above screenshot represents the page for the updating the details of the project to modify the registration made. The user is able to update the details of the registration. The user is able to logout from the system at any moment the user wants. The user is also able to exit from the system.

The above screenshot represents the login page selected by the staff for the login.

The above screenshot displays the result obtained after the staff logs into the system. It displays the total registered project for which staff is mentor for and overall projects registered.



The above screenshot displays the feature to cancel the registration made for the project exhibition.

The above screenshot displays after the change password option is choosen. The user is allowed to change the password of the login account, by providing the login ID and contact number. The user is able to verify the password changed. The systems ask the user to enter the new password to be entered twice.

A.4 Connection of front end with the database

1. Login

```
private void ButtonLoginActionPerformed(java.awt.event.ActionEvent evt) {
    projectRegistered = false;
    ButtonLogout.setVisible(true);
    if (accountType.getSelectedIndex() == 0) {
        subPanel.removeAll();
        subPanel.add(studentPanel);
        subPanel.repaint();
        subPanel.revalidate();
        try {
            checkPreviousProjects(loginIdtext.getText());
        } catch (Exception e) {
            System.out.println(e);
        }
        if (projectRegistered == false) {
            allowToRegister();
        }
    } else {
        staffCalculation(loginIdtext.getText());
        subPanel.removeAll();
        subPanel.add(staffPanel);
        subPanel.repaint();
        subPanel.revalidate();
    }
}
```

Here, the function ButtonLoginActionPerformed is an event driven function.

```
public void checkPreviousProjects(String loginID) throws SQLException {
    rs = st.executeQuery("select sId,sName from student where sLoginId='"
            + loginID + "'");
    ResultSet temprs = null;
    while (!rs.next()) {
        studentNameLabel.setText(rs.getString("sName"));
        temprs = st.executeQuery("select * from pgroup where sRegLeader='"
                + rs.getString("sId") + "'");
    }
    if (!temprs.next()) {
        projectRegistered = true;
        teamNameText.setText(temprs.getString(1));
        projectNameText.setText(temprs.getString(2));
        projectIdText.setText(temprs.getString(3));
        mentorText.setText(temprs.getString(4));
        teamLeaderNameText.setText(temprs.getString(5));
        teamLeaderRegText.setText(temprs.getString(6));
        student2NameText.setText(temprs.getString(7));
        student2RegText.setText(temprs.getString(8));
        student3NameText.setText(temprs.getString(9));
        student3RegText.setText(temprs.getString(10));
        student4NameText.setText(temprs.getString(11));
        student4RegText.setText(temprs.getString(12));
        roomNoText.setText(temprs.getString(13));
        tableNoText.setText(temprs.getString(14));
    }
}
```

```
public void allowToRegister() {
    teamNameText.setEditable(true);
    projectNameText.setEditable(true);
    projectIdText.setEditable(true);
    mentorText.setEditable(true);
    teamLeaderNameText.setEditable(true);
    teamLeaderRegText.setEditable(true);
    student2NameText.setEditable(true);
    student2RegText.setEditable(true);
    student3NameText.setEditable(true);
    student3RegText.setEditable(true);
    student4NameText.setEditable(true);
    student4RegText.setEditable(true);
    roomNoLabel.setVisible(false);
    roomNoText.setVisible(false);
    tableNoLabel.setVisible(false);
    tableNoText.setVisible(false);
}
```

#	sName	sID	sLoginId	sPass	sContact
1	Aishwarya	17ETCS02015	aishwaryajmp1998	aishu	9304390800
2	Amresh	17ETCS02115	amresh007	sunshine	9128657860
3	pratyeksha	17ETCS02116	cutiep200	golu	9304392877
4	kartik	17ETCS02175	kartikch	karti	9907722440
5	Riya Panda	17ETCS02375	pandariya02	pandar	7209274229

Figure 1 Student table to check Student Credentials

#	tName	tID	tLoginId	tPass	tContact
1	Rinki Sharma	T001	rinkisharma001	rinki	7894561230
2	Ami Rai E.	T002	ami002	rami	7014130635
3	Ritesh A	T005	ritz005	ritesh	9904630220
4	Vivek Oberoi	T007	viveko009	vivek	9304569870
5	Deepak v.	T009	deepv009	deep	7204306360

Figure 2 Staff table to check Staff Credentials

#	teamName	projectName	projectId	mentor	sNameLeader	sRegLeader	sName2	sReg2	sName3	sReg3	sName4	sReg4	department	gRoom	gTable

Figure 3 pGroup table to check if any Project is registered with the student info

The above code snippet represents the code for the login page where, the user is able to login to the system. The system checks if the user trying to login has already registered for the project exhibition or not, if the user has already registered for the project then after the login the user is directed to the page displaying the project details such as all the registration details and the allocated room and table number. If the user trying to log into the system is not registered for any project, the user is directed to the registration page where, the user is allowed to register. If 'setvisible' is empty then "registration" panel will be displayed followed by retrieving details about the department and staff from department table and staff table respectively

2. Project Registration

```
private void ButtonRegisterActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String projectId = "", roomNo = "";
        int tableNo = 0;
        rs = st.executeQuery("select count(*) from pgroup");
        while (rs.next()) {
            int numberOfProjects = rs.getInt(1);
            projectId = "PR000" + 1 + numberOfProjects;
            roomNo = "ROOM" + 1 + numberOfProjects / 4;
            tableNo = 1 + numberOfProjects % 4;
        }
        st.executeUpdate("insert into pgroup values('" + teamNameText.getText() + "','"
                + projectNameText.getText() + "','" + projectId + "','"
                + mentorText.getText() + "','" + teamLeaderNameText.getText() + "','"
                + teamLeaderRegText.getText() + "','" + student2NameText.getText() + "','"
                + student2RegText.getText() + "','" + student3NameText.getText() + "','"
                + student3RegText.getText() + "','" + student4NameText.getText() + "','"
                + student4RegText.getText() + "','" + departmentText.getText() + "','"
                + roomNo + "'," + tableNo + ")");
    } catch (SQLException ex) {
        System.out.println(ex);
    }
    JOptionPane.showMessageDialog(null,"Project is Registered");
}
```

#	teamName	projectName	projectId	mentor	sNameLeader	sRegLeader	sName2	sReg2	sName3	sReg3	sName4	sReg4	department	gRoom	gTable
1	Sunshine	Self Driving C...	PR0001	Rinki Sharma	Aishwarya	17ETCS02015	Akash	17ETCS02017	Amresh	17ETCS02070	Pulkit	17ETCS02099	CSE	ROOM1	1

Figure 4 pGroup table when project is registered by the student

In the above screenshot, function ButtonRegisterActionPerformed,is for the registration for the project which appears after the login page. The user inserts the value of the required fields for the project in the fields such as team name, project name, mentor name, and student participants. After registration the room number and the table number are generated randomly between 1 to 4. The limitation for the project ID generated is that it should start with **PR00**. The details of the registered project are stored in the registration table**.**

3. Modify Project

```
private void ButtonModifyActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        rs=st.executeQuery("select projectId from pgroup where "
                + "sNameLeader='"+tempName+"' OR"
                + "sName2='"+tempName+"' OR"
                + "sName3='"+tempName+"' OR"
                + "sName2='"+tempName+"'");
        String projectId="";
        while (rs.next()){
             projectId=rs.getString("projectId");}
        st.executeUpdate("update pgroup set "
                + "teamName='" + teamNameText.getText() + "',"
                + "projectName='"+ projectNameText.getText() + "',"
                + "mentor='"+ mentorText.getText() + "',"
                + "sNameLeader='" + teamLeaderNameText.getText() + "',"
                + "SRegLeader='"+teamLeaderRegText.getText() + "',"
                + "sName2='" + student2NameText.getText() + "',"
                + "sReg2='"+ student2RegText.getText() + "',"
                + "sName3='" + student3NameText.getText() + "',"
                + "sReg3='"+ student3RegText.getText() + "',"
                + "sName4='" + student4NameText.getText() + "',"
                + "sReg4='"+ student4RegText.getText() + "',"
                + "department='" + departmentText.getText() + "'"
                + "where projectId='"+projectId+"'");
    } catch (SQLException ex) {
        System.out.println(ex);
    }
    JOptionPane.showMessageDialog(null,"Project is Modified");
}
```

#	teamName	projectName	projectId	mentor	sNameLeader	sRegLeader	sName2	sReg2	sName3	sReg3	sName4	sReg4	department	gRoom	gTable
1	Sunshine	Self Driving C...	PR0001	Ami Rai E.	Aishwarya	17ETCS02015	Akash	17ETCS02017	Arun	17ETCS02030	Kartik	17ETCS02082	CSE	ROOM1	1

Figure 5 pGroup table after modifying the project

The above screenshot represents the function ButtonModifyActionPerformed, where the values stored in the registration table can be modified if needed. If the student has already registered for the project, he/she will be directed to the modification page directly after the login.

4. Staff Panel

```
public void staffCalculation(String loginID) {
    try {
        rs = st.executeQuery("select tName from student where sLoginId='"
                + loginID + "'");
        rs = st.executeQuery("select * from pgroup where mentor='"
                + rs.getString("tName") + "'");
        projectsUnderYourselfTable.setModel(DbUtils.resultSetToTableModel(rs));
        rs = st.executeQuery("select teamName,projectName,mentor,sNameLeader from pgroup");
        allProjectsTable.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (SQLException ex) {
        System.out.println(ex);
    }
}
```

#	teamName	projectName	projectId	mentor	sNameLeader	sRegLeader	sName2	sReg2	sName3	sReg3	sName4	sReg4	department	gRoom	gTable
1	Sunshine	Self Driving C...	PR0001	Ami Rai E.	Aishwarya	17ETCS02015	Akash	17ETCS02017	Arun	17ETCS02030	Kartik	17ETCS02082	CSE	ROOM1	1

Figure 6 pGroup table used 2 times with different requirements

The above screenshot is of the staffpanel, where staffCalculation function which is displayed just after the staff logs in to the system. The page displays all the projects registered under the staff and the details of the project as well. The staff will be also able to view all the projects registered.

5. Change Password

```
private void ButtonChangePassActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        if (accountType.getSelectedIndex() == 0) {
            st.executeUpdate("UPDATE student SET sPass='" + changePassNewPassText.getText()
                    + "' WHERE sLoginId='" + changePassLoginIDText.getText()
                    + "' AND sContact='"+ changePassContactText.getText() + "'");
        } else {
            st.executeUpdate("UPDATE teacher SET sPass='" + changePassNewPassText.getText()
                    + "' WHERE sLoginId='" + changePassLoginIDText.getText()
                    + "' AND sContact='"+ changePassContactText.getText() + "'");
        }
    } catch (Exception e) {
        System.out.println(e);
    }
    JOptionPane.showMessageDialog(null,"Password Changed");
}
```

#	sName	sID	sLoginId	sPass	sContact
1	Aishwarya	17ETCS02015	aishwaryajmp1998	aishu	9304390800
2	Amresh	17ETCS02115	amresh007	sunshine	9128657860
3	pratyeksha	17ETCS02116	cutiep200	golu	9304392877
4	kartik	17ETCS02175	kartikch	karti	9907722440
5	Riya Panda	17ETCS02375	pandariya02	pandar	7209274229

Figure 7 Student table before changing password

#	sName	sID	sLoginId	sPass	sContact
1	Aishwarya	17ETCS02015	aishwaryajmp1998	aishwarya	9304390800
2	Amresh	17ETCS02115	amresh007	sunshine	9128657860
3	pratyeksha	17ETCS02116	cutiep200	golu	9304392877
4	kartik	17ETCS02175	kartikch	karti	9907722440
5	Riya Panda	17ETCS02375	pandariya02	pandar	7209274229

Figure 8 Student table after changing password

The above screenshot represents the ButtonChangePassActionPerformed function, which is for changing the login password. The login password using the SqL query update is changed with the help of the loginID. The password is reset and then the changed password is updated in the login table.

6. Cancel Project

```
private void ButtonCancelRegisterationActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        st.executeUpdate("delete from project where projectId='"+projectIdText.getText()+"'");
                } catch (SQLException ex) {
        System.out.println(ex);
    }
    JOptionPane.showMessageDialog(null,"Project is Deregistered");
}
```

#	teamName	projectName	projectId	mentor	sNameLeader	sRegLeader	sName2	sReg2	sName3	sReg3	sName4	sReg4	department	gRoom	gTable
1	Sunshine	Self Driving C...	PR0001	Ami Rai E.	Aishwarya	17ETCS02015	Akash	17ETCS02017	Arun	17ETCS02030	Kartik	17ETCS02082	CSE	ROOM1	1

Figure 9 pGroup table before cancelling the project

#	teamName	projectName	projectId	mentor	sNameLeader	sRegLeader	sName2	sReg2	sName3	sReg3	sName4	sReg4	department	gRoom	gTable

Figure 10 pGroup table after Cancelling the project

The above screenshot represents the function ButtonCancelRegistrationActionPerformed where, on selecting the button “cancel registration”, the project registered will be deleted. The SQL query delete from() is used to delete the data stored in the registration table.

A.5 Concluding remarks (Summary, limitations, improvements)

The system developed above to manage the group project exhibition, in the RUAS conducted by the FET works efficiently with the proper exception handling. The GUI (Graphic user interface) is developed using the jframe that is java. Swing and is linked with the MySQL database. The GUI developed for the exhibition at the RUAS is able to insert, update, delete and display a particular record of the project details. Java Swing is a part of Java Foundation Classes (JFC) that is used to create window-based applications. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java. Unlike AWT, Java Swing provides platform-independent and lightweight components. The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc. Java swing components are platform independent. The connection for the swing to the MySQL database is established, to retrieve or store data to the MySQL database. Thus, an effective code is developed for the project exhibition conducted at the RUAS.

The limitations of the result and experiment are: No data validation for input components and API is bloated. We can either use an action, a key event or a document listener to listen to button pressing even and we need to be sure they trigger the same code for same case. The SWING comes with certain disadvantages such as the components used requires certain

separate jar files to consume. And sometimes the components of the swing don't appear as it should. To develop application in swing one has to be very careful with programming. The developed system could have been improved as for the registration of the project for the exhibition there could have been a separate page displaying the details of the project registered and to make the exhibition unbiased, the details such as the marking scheme could have been included in the GUI along with the fields where the student/project obtains the marks.