<u>Course Enrolment System - Project Document</u>

Name :- Nishadi Nuwanmini Wickramanayaka
Reg No :- 2016/CS/156
Index No :- 16001567

All the implementation details are divided into activities in the order I've followed as below.

Activity 01:

Read the problem definition and understood the system requirements.
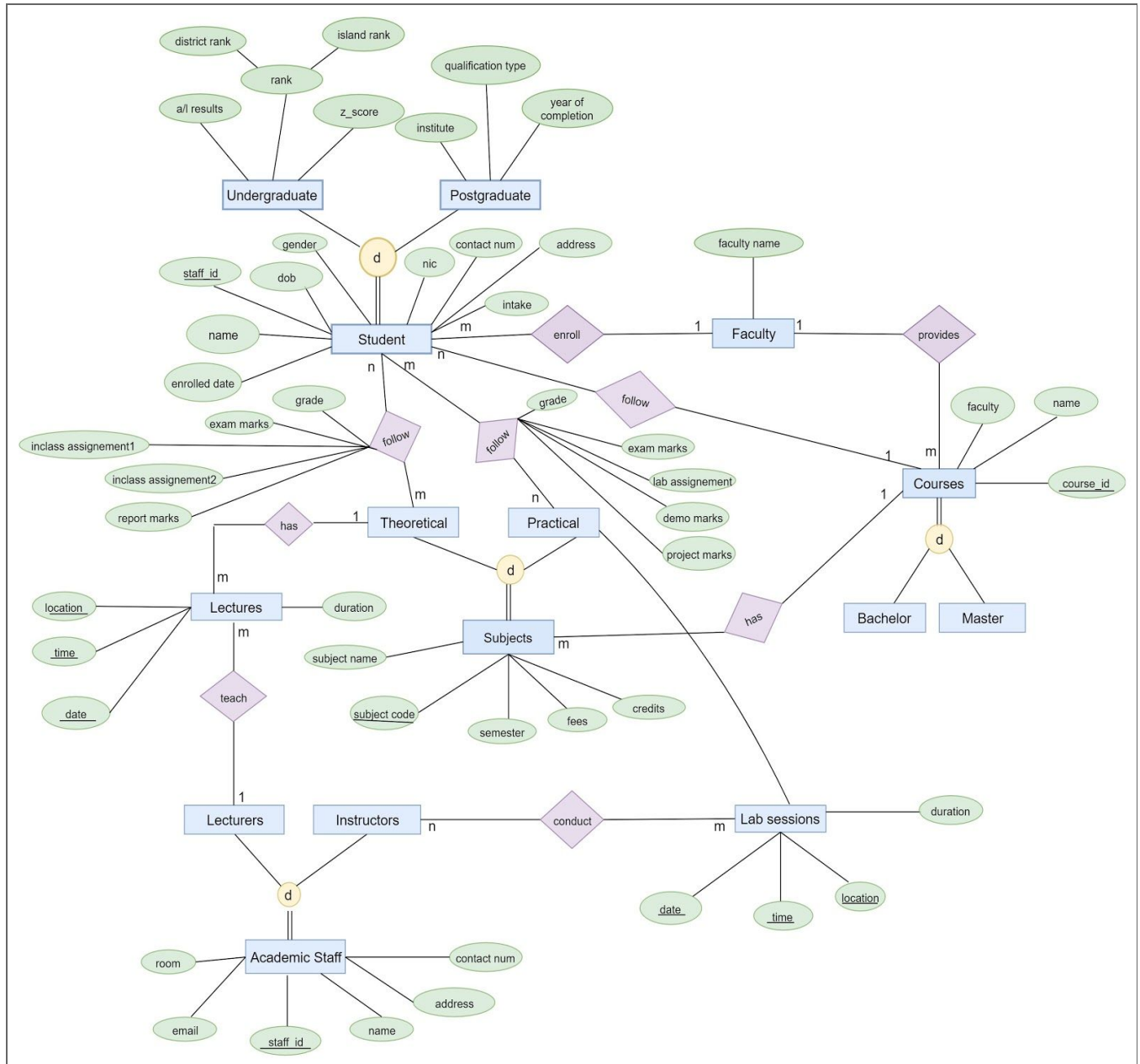
Activity 02:

Made some assumptions in order to get a clear idea of the system and to make the system boundary.(had to assume some things since they were not given in the problem definition)

Assumptions :-

1. A student can enroll in a one course at a time.(after completing the course student can enroll in another course if he/she wants)
2. Every course,provided by the university, has enrolled students.
3. A student can not follow both bachelor and master courses at the same time.
4. Every lecturer and instructor is assigned to one or more subjects.Lecturers to conduct the lecturers.Instructors to conduct the lab sessions.
5. One lecture is taken by a one lecturer and one lab is taken by two instructors.
6. A lecturer can take one or more subjects.
7. There can be subjects which don't need the lab sessions(theoretical subjects).
8. Every subject has three assignments.
9. There're no courses with only practical subjects or theoretical subjects.
10. Assigning lectures and lab sessions to relevant lecturers and instructors is upto responsible parties.It's not upto the system.After assigning them, moderators should enter those details to the system.
11. Final grade for a subject is calculated on 70% of exam marks and 30% of assignment marks.
12. Student can't obtain a final grade depending only on the assignment marks(if the student is absent for the final exam of a subject then the status of that subject for the particular student is "not completed".So he/she has to take the exam in next year)

Activity 03:

Got a clear and complete idea of the system.Identified the attributes and entities.Finally drew the ER diagram.

(if this is not clear,pdf version is available in the project folder)

Activity 04:

Mapped the ER diagram as below.

1. Academic_staff (<u>staff_id</u>, name, address, con_num, email, room, lecturer_or_instructor)

2. Courses (<u>course_id</u>, course_name, faculty, type)

3. Login (<u>username, password</u>)

4. Postgraduate (<u>std_id</u>, qualification type,institute, year_completed,course_id)

5. Undergraduate (<u>std_id</u>, district_rank, island_rank, z_score, course_id)

6. Student (<u>std_id</u>, first_name, middle_name, last_name, nic, dob, connum, address, gender, intake,enrolled date, course_id)

7. Sessions (<u>date, time, location</u>, duration, lecture_or_lab, staff_id, subject_code)

8. Subjects (<u>subject_code</u>, subject_name, fees, credits, year_sem, prac_or_theo, course_id)

9. Std_practical_subjects (<u>std_id, subject_code</u>, lab_ass_marks, practical_demo_marks, project_marks, exam_marks, grade, semester)

10. Std_theoretical_subjects (<u>std_id, subject_code</u>, inclass_ass1_marks, inclass_ass2_marks, report_marks, exam_marks, grade, semester)

11. Staff_subjects (<u>staff_id, subject_code</u>)

Activity 05:

Created the database depending on the ER diagram.

11 tables :-

1. academic_staff
2. courses
3. login
4. postgraduate
5. undergraduate
6. sessions

7. staff_subjects
8. std_practical_subjects
9. std_theoretical _subjects
10. students
11. subjects

Activity 06:

Initiated the java project using NetBeans IDE 8.2.Then created a class called "DBconnect" to get the connection to the database.(following piece of code is used to get the connection)

```
String url = "jdbc:mysql://localhost:3306/course_enrolment_system" ;
String username = "root" ;
String password = "" ;
Connection con = null;
con = DriverManager.getConnection(url, username, password);
```

Since the DB is installed in the same computer, IP address is localhost.
mysql service is running on port 3306.
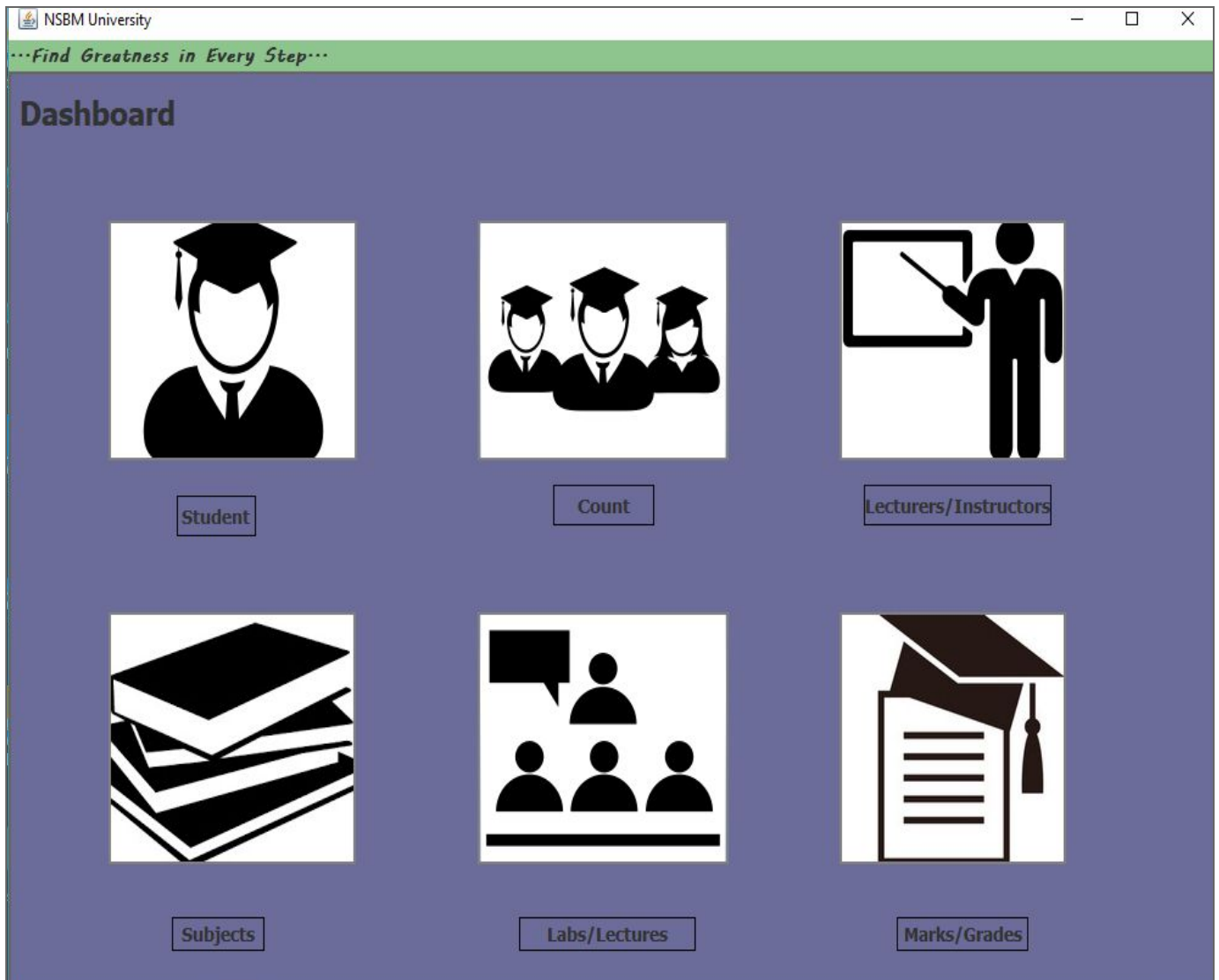Url is to find the location of the Database.

Activity 07:

Designed the user interfaces.

- *Main Page :*

Moderator should enter his/her username and the password to login to the system.

● *Dashboard :*

6 buttons. (Student,Count,Lecturers/Instructors,Subjects,Sessions,Report)
Each button will direct the user to the relevant operations.

● *Student Operations :*

3 buttons.Each will direct the user to the relevant operations.
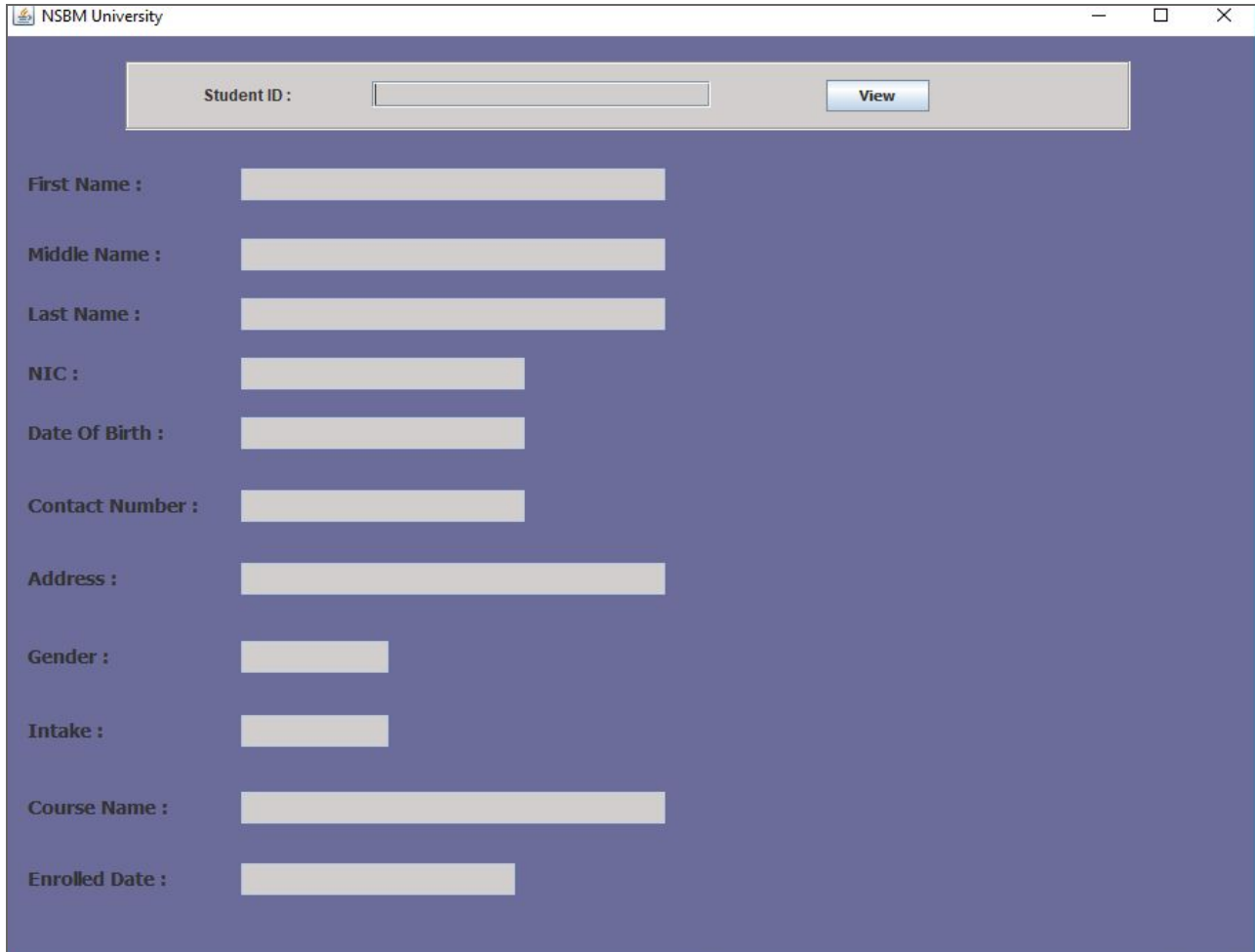
● *New Student :*

Moderator should enter the required details of the students.Tabbed pane is used to divide specific requirements for the undergraduate and the postgraduate students.

● *View Student :*
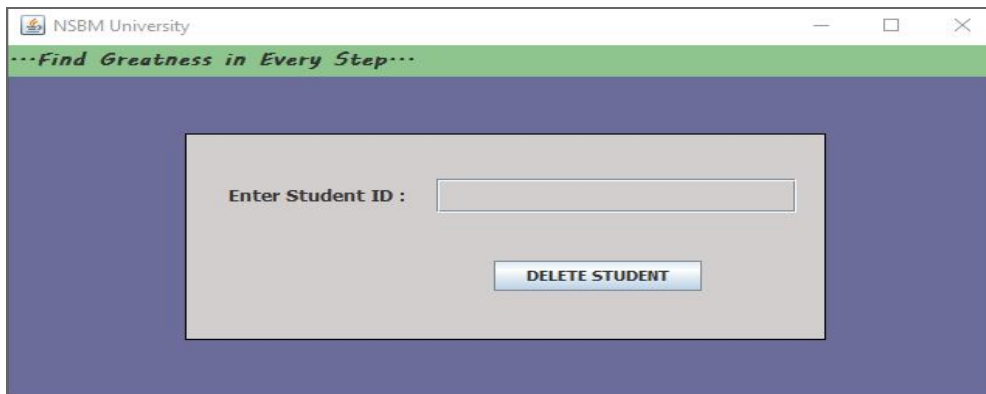

System will show the above details which is related to the entered student ID.
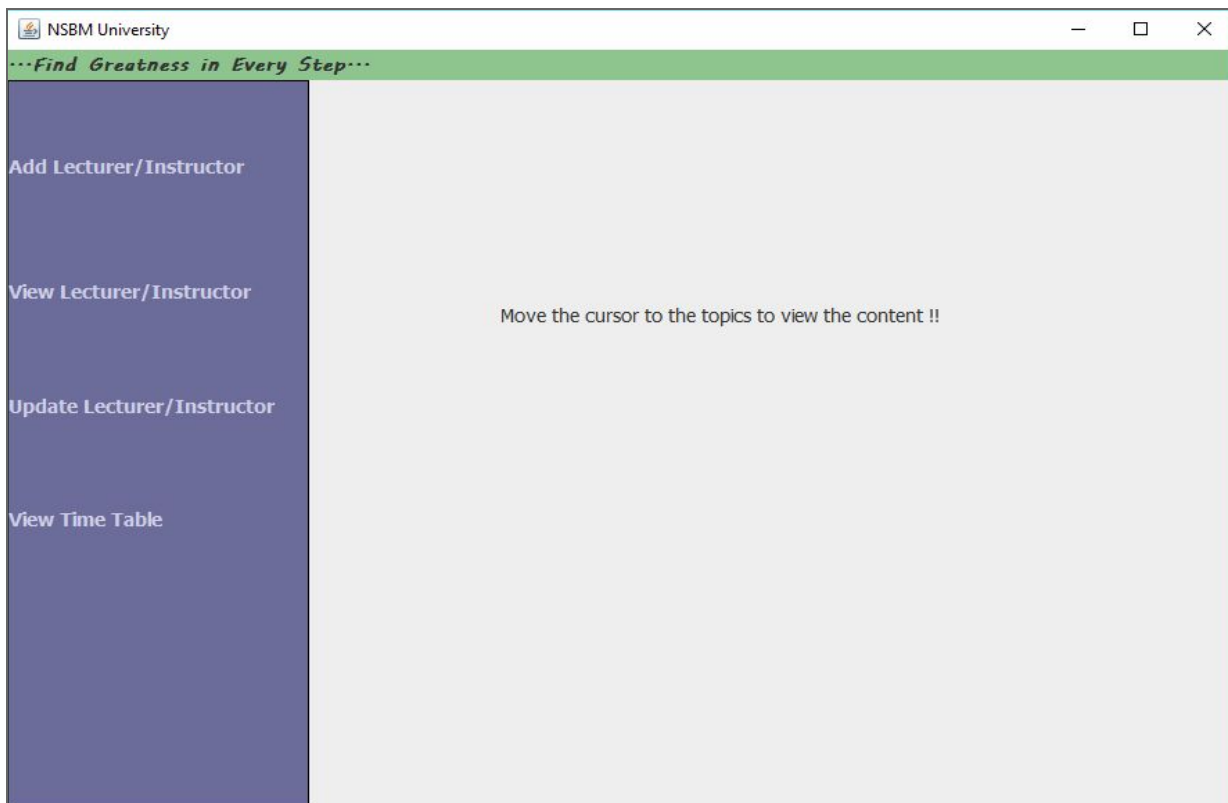Text fields are uneditable since this window doesn't provide any updating function.

● *Delete Student :*

Delete all the information related to the entered student ID.

● *Lecturers/Instructors Operations :*

This window is implemented using mouse motion(mouse moved) event.So to view the content cursor should be moved to the topic.Then it will show the related jpanel in the blank space.

● *Add Lecturer/Instructor :*

Enter the details of the lecturers/instructors and submit.Subjects combo box is to select the subjects that they are going to teach.Select a subject from the drop down and add it to the list.

● *View Lecturer/Instructor :*

Enter the staff id of the lecturer or instructor to view his/her details.Text fields are uneditable to ensure that viewing the details won't make any changes of the data.

● *Update Lecturer/Instructor :*

First you have to view the details of the particular staff member that you need to update the information.Then you can do the changes(text fields are editable) and click on the update button. It'll update the database layer too.

● *View Timetable :*

Before viewing the timetable.

After viewing the timetable of a staff member.This table shows the details of the lab sessions / lectures of the particular lecturer or instructor.

- *Enter new subject :*

To enter a new subject to a course.There is a radio button group to determine whether it is a theoretical subject or a practical subject.

- *Select/Update subjects :*

To select the subjects or to change the selected subjects.

Subjects available shows the subjects that the student can select.Selected subjects shows the subjects that the student have selected.Maximum number of subjects that a student can select is four.Add button is to add new subject to the list and remove button is to remove a selected subject from the subject list.

● *Subject List :*

Shows the selected subjects for the semester.

Total = semester fees for the subjects.

- *Lectures and lab sessions :*

- *Add lecture :*

To add a lecture.Location is the hall number.

● *Add lab sessions :*

To add a lab session.Location is the lab number and the duration is set to 2 hours.(It is given in the problem definition)

● *Find ID :*

Initial look of the table(before clicking on buttons)

Lecturers IDs will give you all the names and staff IDs of the lecturers.Instructors IDs will give you all the names and staff IDs of the instructors.

- *Marks / Grades :*

'Show subject progress' shows the current state of the assignments.Grade button will calculate the grade.If examination marks field is "AB" or "not completed" grade will not be calculated.

- *Report :*

Final report of the semester.This will email to the student.

Activity 08:

Identified the classes to implement.

1. Course_enrolment_system = initiating the program.

2. DBconnect = methods related to connect the database

3. StudentDetails = getters and setters related to the variables.

4. PostgratDetails = getters and setters related to the variables.

5. UndergratDetails = getters and setters related to the variables.

6. StudentsDBOperations = methods communicating with the database(querying).

7. CountDBOperations = methods communicating with the database(querying).

8. SubjectDetails = getters and setters related to the variables.

9. SelectedSubjects = table model to get selected subject list.

10. SubjectsDBOperations = methods communicating with the database(querying).

11. LecturerInstructorDetails = getters and setters related to the variables.

12. LecInsDBOperations = methods communicating with the database(querying).

13. SMemberTimeTable = table model to get the time table of a staff member.

14. FindIDTableModel = table model to get the staff IDs.

15. ReportDetails = getters and setters related to the variables.

16. ReportTableModel = table model to get the report of a student.

17. ReportsDBOperations = methods communicating with the database(querying).

18. SessionsDetails = getters and setters related to the variables.

19. SessionsDBOperations = methods communicating with the database(querying).

Activity 09:

Implementing the system.
Important implementation details :-

- If the password or username is incorrect, system will show an error message.If not user(moderator) can login to the system successfully.

    Ex:- username - nsbm001
         Password - pwnsbm001

- Course name combo box in new student form and subject combo box in add lecturer/instructor form will be loaded from the database(from the course table)
- When viewing student and deleting student, if the student id is not found in the database table system will give a message indicating that the student id is not found.

- When deleting a student, system will ask from the user again, whether to delete the particular student or not.
- Some operations are implemented using the mouse motion (mouse move) event (not mouse click event).Those operations will be initiated when the mouse is moved to the topic.
- Viewing text boxes are set as uneditable while updating text boxes are set as editable.
- Tables are loaded using separate classes(table models).
- Since a student can select maximum four subjects, if he/she selects more than 4 subjects, system will pop up a message saying that constraint.
- Find id is to find staff ids of lecturers and instructors.
- Subject grade is available only if the student take the final exam.Otherwise he/she has to take that exam in next year in order to complete the subject.

Activity 10:

Entering data to the database.(ex:- course names,course IDs,subject names,subject IDs,hall numbers,lab numbers)

Entering dummy data to the database(ex:- student details,lecturer / instructor details)

Testing the system.

END..