Final Year Project Management System

Session: 2021 – 2025

Submitted by:

Abdullah Nasir

2021-CS-113

Supervised by:

Mr. Nazeef Ul Haq

Department of Computer Science

University of Engineering and Technology Lahore

Pakistan

Acknowledgments

I am grateful to Allah Almighty that He provided me with the strength and power to complete this project in time.

I wanted to take a moment to express my gratitude for all of the help and guidance Mr. Nazeef Ul Haq provided me throughout our project. His support was instrumental in helping me achieve my goals and I truly appreciate the time and effort he dedicated to helping me. Your feedback, suggestions, and expertise were invaluable in shaping the final outcome of the project and I am proud of what I was able to accomplish. I look forward to applying these skills in future projects.

Contents

List of Figures

Abstract

The prime objective of the final year project management system was the visualization and critical analysis regarding the use and working of databases in real-world problems. A database is an organized collection of information that can be easily accessed, managed and updated. Databases are used in a wide variety of applications, from simple storage of personal data to complex systems used by large corporations and government agencies. The main purpose of a database is to provide a structured way to efficiently store and retrieve information. By storing it in a database, users can quickly and easily access the information they need, ensuring the accuracy and consistency of the information. The scenario which was implemented to grasp the concepts of databases was the management of final year projects of the Computer Science Department of UET Lahore. The requirements of the system were not easy to fulfill but it was explained thoroughly in the case study of the project. Mainly insertion, updating, retrieval and deletion were used with some constraints to the data coming in, going out or being modified.

1 Introduction

1.1 Description

The management of final year projects is done regularly in every university. Most commonly, this process is done manually where data of students is kept in a register and insertion, modification, and deletion, are done by an attendant. Human errors are expected and commonly encountered in such a scenario. Some universities like UET, Lahore started to use file system for ease but there was a need for a better technology as there were limitations to file system as well. In a final year project management system, the system comprises all the students of the session, and groups are made for a certain project. One student can be working in one and only one group.
The evaluations of students are done by an advisory board. The advisory board is the faculty of the department given with the duty to supervise some of the projects of the students of the department. The data administrator can add the students, advisors, projects, evaluations etc. Having all the components of a system in one place helps manage the system very easily. Advisors are given different roles based on some specific skills or position in the industry. There roles are main advisor, co-advisor and industry advisor. Main advisor helps manage the whole project completely whereas co-advisor and industry advisor suggests features based on the real life problems and solutions.
There is a complete record of the dates of the events in the FYP System.

1.2 Motivation

The main motivation for this project was to help to learn querying data from the database. Simple queries along with complex queries were used to make this project work smoothly. The project was developed in C# .NET Framework using Windows Form and SQL Server Microsoft was integrated as a database. Moreover, PDF reports were generated using iTextSharp(an extension in Microsoft Visual Studio).

1.3 Target Audience

The target audience for the project are almost all the universities where final year projects are managed. This application will help the data administrator to enter data of students, advisors, projects, evaluations etc. Then this data can be used for makin important decisions.

2 Operational Details

For now the management system consists of a single user that will be acting as the only data administrator having full control over the data and the system. Following are the details of action the user can perform though the FYP Management System.

1. The admin can enter details of the student and update the entered details as well.
2. Project details can be entered and modification of records is possible.
3. Advisors can be added and updated to the system.
4. Advisors will be allocated to projects by the admin
5. Groups will be assigned with their projects by the admin. Modification and deletion of any information will be possible.
6. Evaluation details will be added by the admin and the evaluation of groups will be done by the admin too.
7. Moreover PDF reports can be generated.

3 Database Design

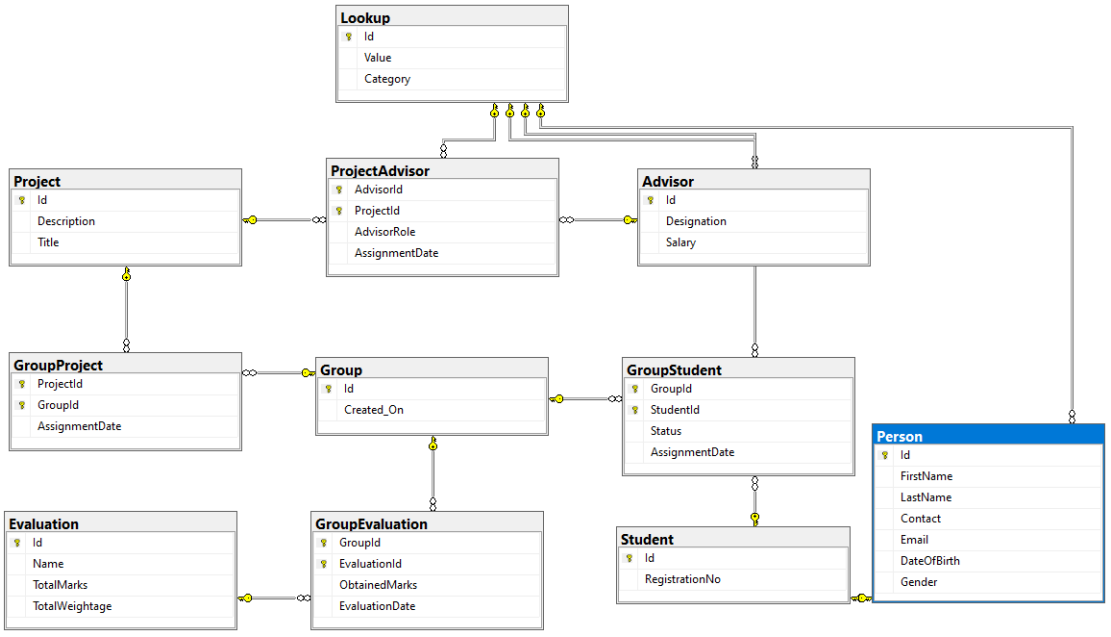

FIGURE 1: Database Diagram

3.1 Lookup

Lookup relation supports other relations in the database. It consists of advisor roles, designations and student's status and gender. Lookup.ID is its primary key which will be used as foreign key in other tables for the information.

3.2 Person

Peron consists of the data needed for a student and an advisor. Primary key is Id which is used as foreign key in student and advisor relations.

3.3 Student

Student relation consists of the student registration and Id from person relation which acts as primary key and foreign key both here.

3.4 GroupStudent

The group student consists of the GroupId that is assigned to a StudentId. Both GroupId and StudentId acts as a primary key for the relation. Assignment date and status of student in group is also entered.

3.5 Group

Group consists of the Group.Id and the date of creation. The ID acts as a primary key. It acts as foreign key in GroupProject and GroupStudent

3.6 GroupProject

The GroupProject consists of the ProjectId assigned to a GroupId and thus both these attributes act as a primary key for the relation. Assignemnet date provides date on which project was assigned.

3.7 GroupEvaluation

GroupEvaluation has EvaluationId and GroupId as its primary key. It stores the marks obtained by a group in a particular evaluation. It also has date of evaluation stored.

3.8 Evaluation

Evaluation has Id as its primary attribute. It consists of the evaluation details such as Name, Total Marks and Total Weightage.

3.9 Project

Project has its Id as its primary key. It consists the title and description of the project.

3.10 ProjectAdvisor

Project advisor consists of the advisor that is assigned to a project along with the role that is assigned to the advisor. The ProjectId and the AdvisorId acts as the primary key for the relation.

3.11 Advisor

The Advisor relation consists of the advisor's designation and the salary. Id acts as the primary key for the relation.

4 Graphical User Interface (GUI)

4.1 Loading Page

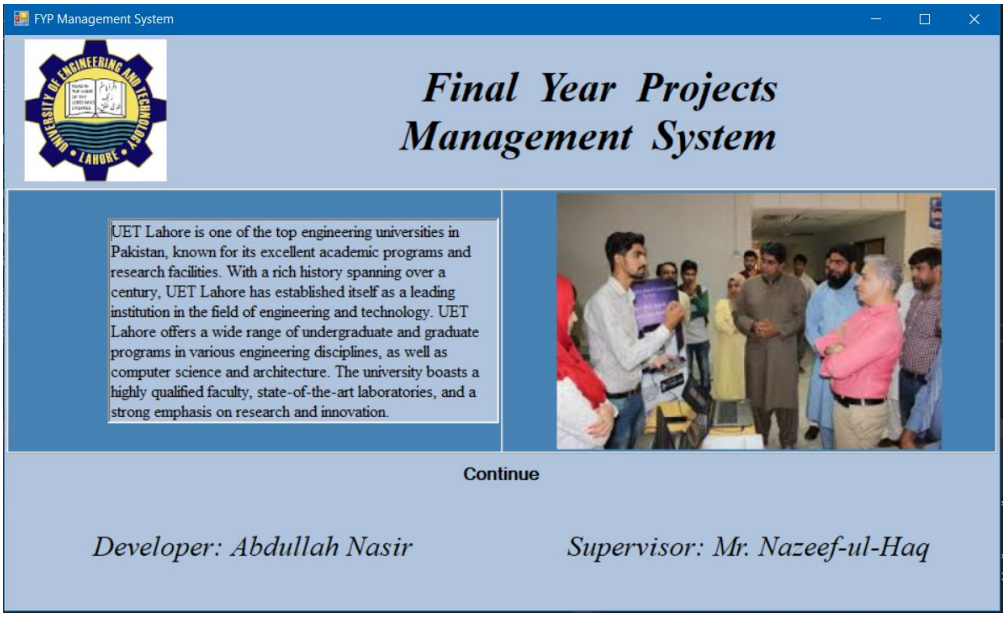

FIGURE 2: Loading Page

This page is added to enhance user experience of the application.

4.2 Home Page

This page is the considered the home page of application as all the new processes start from this page.

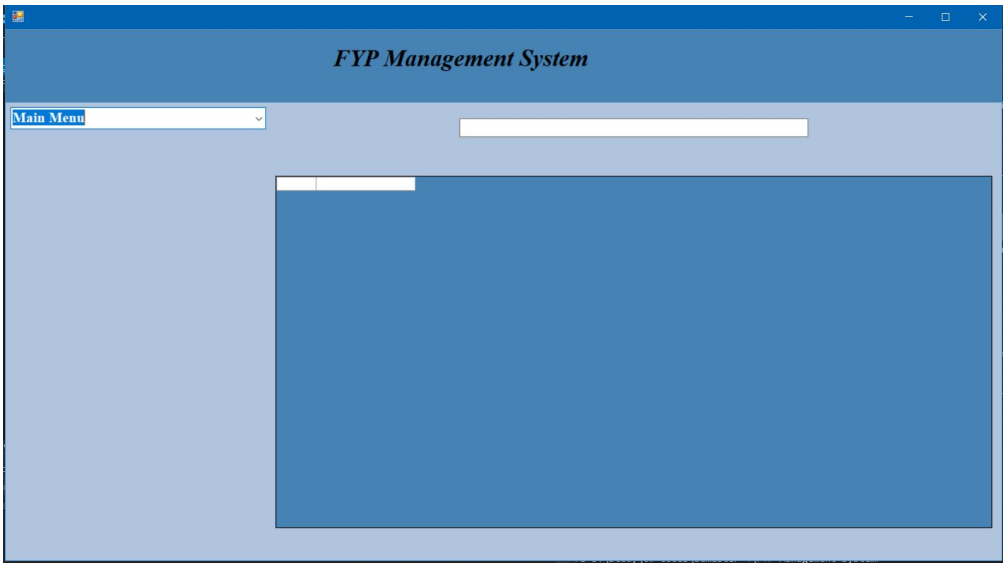

FIGURE 3: Home Page

4.3 Insert/Update Student Page

FIGURE 4: Update Students Page

This page is used to insert and update students.

4.4 Insert/Update Project Page

This page is used to insert and update projects.

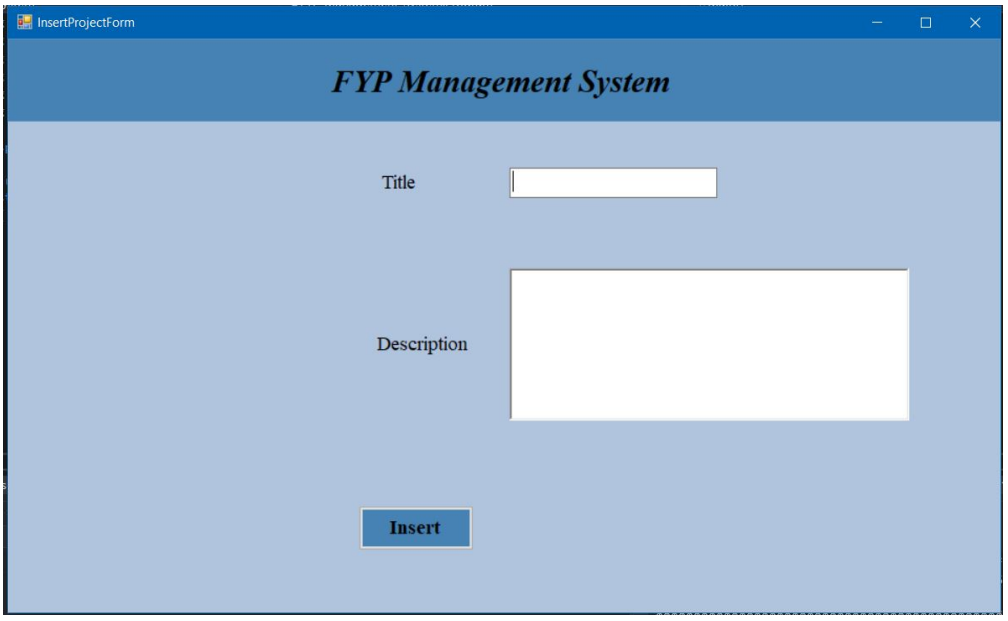

FIGURE 5: Insert new projects in system

4.5 Insert/Update Evaluation Page

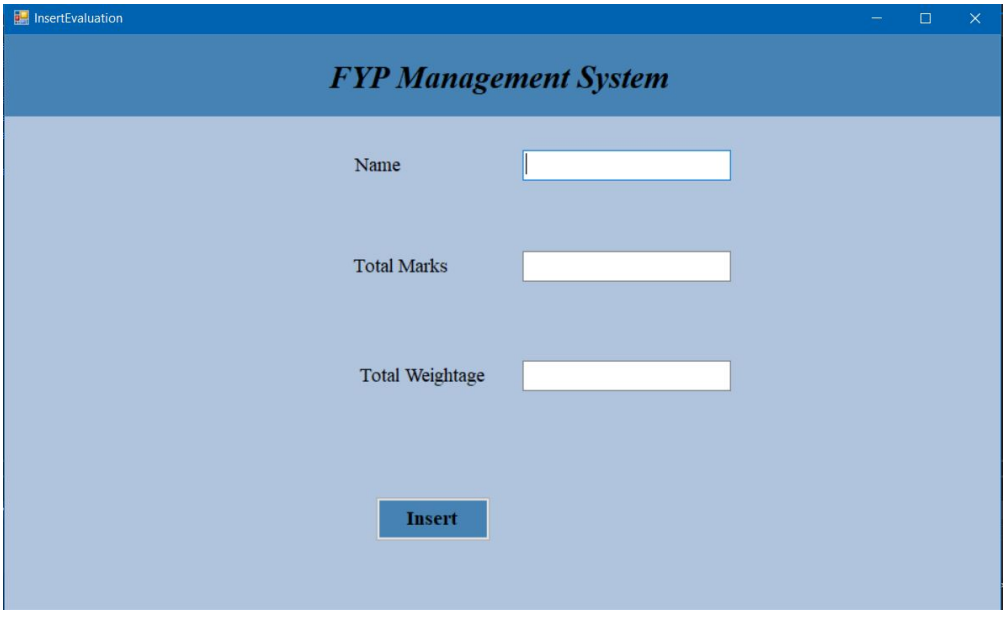

FIGURE 6: Create projects and allocate projects to groups page

This page prompts the user to enter details for the projects of students. Updating of these project details is done here as well.

4.6 Insert/Update Advisor Page

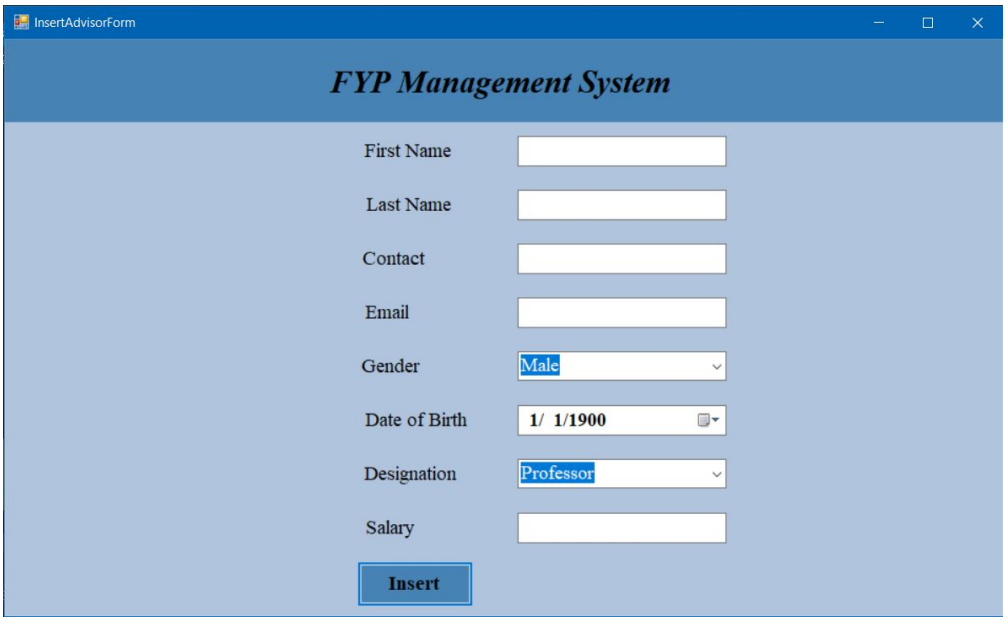

FIGURE 7: Add students in groups page

The page is used to add advisors in the database. Furthermore, modifications in data related to advisors is also done here.

4.7 Create/Update Group Page

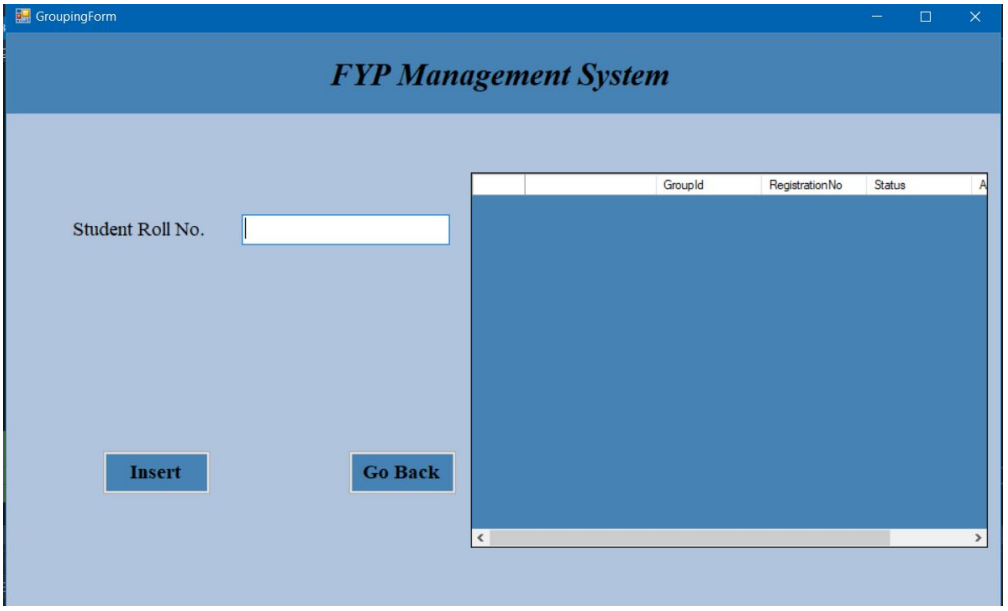

FIGURE 8: Edit projects page

The page is used to create a new group and add students to it. Moreover it can also be used to enter or remove students from an existing group.

4.8 Group Evaluation Page

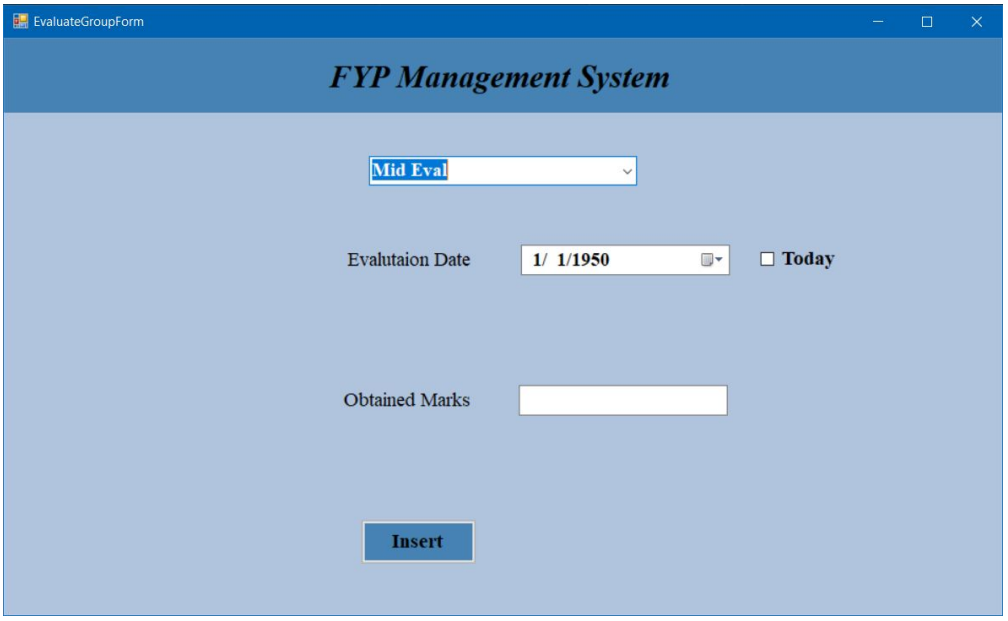

FIGURE 9: Add advisors to system page

The page is used to add information regarding the evaluation of the groups. Updating of such information takes place at this page too.

4.9 Assign Project Page

This page is used to assign project to the group and assign advisors to the projects.

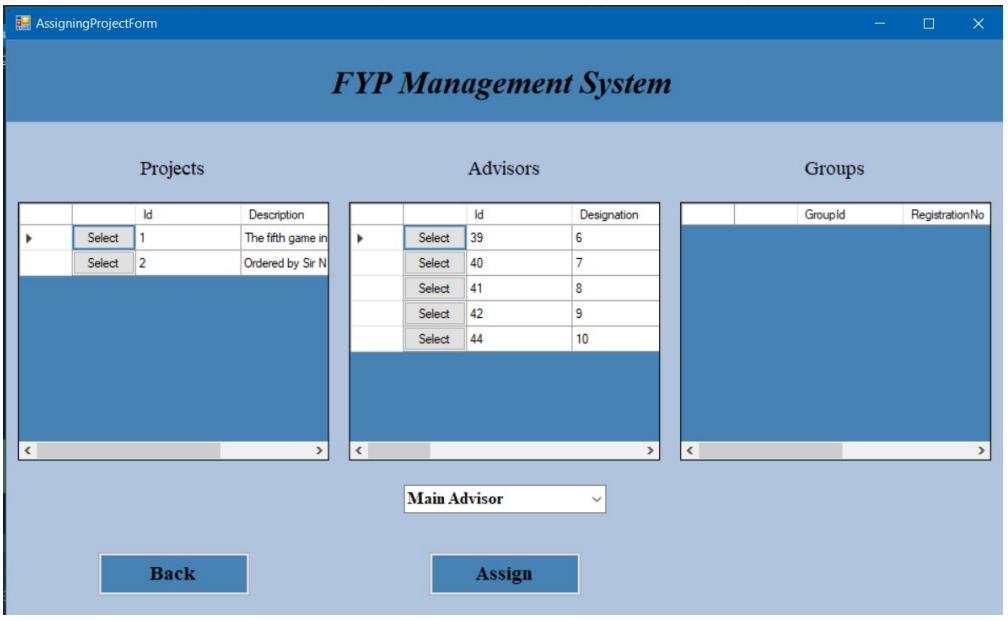

FIGURE 10: Edit advisors information page

5 Queries

5.1 Select Queries

5.1.1 Selecting all Students

```
Select * From Student Join Person On Student.Id = Person.Id
```

5.1.2 Selecting all Advisors

```
Select * From Advisor Join Person On Advisor.Id = Person.Id
```

5.1.3 Selecting all Projects

```
Select * From Project
```

5.1.4 Selecting all Evaluations

```
Select * From Evaluation
```

5.1.5 Selecting First Id for insertion using foreign key

```
Select Top(1) Id From [Group] Order By Id DESC
```

5.1.6 Selecting Groups and their Projects

```
SELECT GroupId, (Select Title From Project Where ProjectId = Id)
As Project FROM GroupProject
```

5.1.7 Selecting Projects, Groups and their advisors

```
Select g.ProjectId, p.Title, p.Description, g.GroupId,
(Select FirstName From Person Where Id = a.AdvisorId)
As Advisor, (Select Value From Lookup wHERE Id = a.AdvisorRole)
As AdvisorRole From Project As p Left Outer Join GroupProject
As g On ProjectId = Id Left Outer Join ProjectAdvisor As a On
a.ProjectId = p.Id
```

5.1.8 Selecting Students without any advisor

```
Select RegistrationNo From Student Except Select RegistrationNo
From Student as s Join GroupStudent as gs On gs.StudentId = s.Id
join GroupProject as gp On gp.GroupId = gs.GroupId Join
ProjectAdvisor as pa On pa.ProjectId = gp.ProjectId
```

5.2 Insert Queries

5.2.1 Inserting mandatory information in Person

```
Insert into Person (FirstName, Email) values (@FirstName,
@Email)
```

5.2.2 Inserting title into Projects

```
Insert into Project (Title) values (@Title)
```

5.2.3 Inserting evaluation information

```
Insert into Evaluation values (@Name, @TotalMarks,
@TotalWeightage)
```

5.2.4 Inserting into Advisor

```
Insert into Advisor (Id, Designation) values (@Id,
@Designation)
```

5.2.5 Inserting in Groups

```
Insert into GroupStudent values (@groupId, @studentId,
@status, @assignmentDate)
```

5.3 Deletion Queries

5.3.1 Deleting Group if it has no Students

```
Delete From [Group] Where Id = @gId
```

5.4 Updating Queries

5.4.1 Updating Student status in Group

```
Update GroupStudent Set Status = @status Where StudentId =
@sId And GroupId = @gId
```

5.4.2 Update Advisor Salary

```
Update Advisor Set Salary = @Salary Where Id = @id
```

5.4.3 Updating Project description

```
Update Project Set Description = @Description Where Title = @title
```

5.4.4 Updating student registration number

```
Update Student Set RegistrationNo = @rNo Where Id = @id
```

5.4.5 Updating evaluation weightage

```
Update Evaluation Set TotalWeightage = @totalWeightage
Where Id = @id
```

6 Generated Reports

6.1 Report 1

Report for Project's group and advisors.

6.1.1 Query

```
Select p.Id as Id, p.Title as Title, (Select FirstName From
Person Where pa.AdvisorId = Id) As AdvisorName, (Select
Value From Lookup Where Id = pa.AdvisorRole) As AdvisorRole,
(Select RegistrationNo From Student Where Id = gs.StudentId)
As RegistrationNumber, (Select FirstName From Person Where
Id = gs.StudentId) As StudentName From Project As p Full Outer
Join ProjectAdvisor As pa On p.Id = pa.ProjectId Full Outer
Join GroupProject as gp On gp.ProjectId = p.Id Join GroupStudent
as gs On gs.GroupId = gp.GroupId
```

6.2 Report 2

Report for Marks of students in individual evaluation.

6.2.1 Query

```
Select p.Title As [Title], s.RegistrationNo As [RegistrationNo],
e.Name as [Evaluation], ge.ObtainedMarks as [Marks] From Project
As p Join GroupProject As gp On gp.ProjectId = p.Id Join
GroupStudent As gs On gs.GroupId = gp.GroupId Join Student As s
On s.Id = gs.StudentId Join GroupEvaluation As ge On ge.GroupId =
gs.GroupId Join Evaluation As e On e.Id = ge.EvaluationId
```

6.3 Report 3

Report for Total Marks of students.

6.3.1 Query

```
Select RegistrationNo as [RegistrationNo], Sum(Marks) as
[TotalMarks] From (Select p.Title As [Title], s.RegistrationNo
As [RegistrationNo], e.Name as [Evaluation], ge.ObtainedMarks as
[Marks] From Project As p Join GroupProject As gp On gp.ProjectId
= p.Id Join GroupStudent As gs On gs.GroupId = gp.GroupId Join
Student As s On s.Id = gs.StudentId Join GroupEvaluation As
ge On ge.GroupId = gs.GroupId Join Evaluation As e On e.Id =
ge.EvaluationId) as t Group by RegistrationNo
```

6.4 Report 4

Report for students not assigned with an advisor.

6.4.1 Query

```
Select RegistrationNo From Student Except Select RegistrationNo
From Student as s Join GroupStudent as gs On gs.StudentId = s.Id
Join GroupProject as gp On gp.GroupId = gs.GroupId Join
ProjectAdvisor as pa On pa.ProjectId = gp.ProjectId
```

7 Limitations

Following are the limitations of the project:

1. The application is only oriented towards a single user

2. No login is required and thus anyone can access the application, making the data to be insecure..

3. The code needs to be changed if different reports are required. They are not generated dynamically.

8 Conclusion

The Final Year Projects Management midterm project for the fulfillment of the Database and Management System Lab was a full package to learn out the practical implementation of the queries. The full power of the project can be revealed if further time is provided. Overall it is a working project with extendability.