FYP Management System

Session: 2022-2025

Submitted by:
Burhan Ahmad 2022-CS-191

Submitted to:
Mr. Nazeef ul Haq

Deprtment of Computer Science

University of Engineering and Technoology
Lahore Pakistan

Contents

1 Introduction **3**

2 Technology Stack **3**

3 SQL Concepts **3**

3.1 JOINS 3

3.2 Aggregate Functions 3

3.3 Sub Queries 4

3.4 Group By Clause 4

4 Flow Chart **5**

5 Tables Diagram **6**

6 WireFrames **7**

7 Conclusion **9**

1 Introduction

The Final Year Project (FYP) Management System is a comprehensive platform designed to streamline and optimize the administration of university final year projects. It facilitates seamless project allocation, student grouping, and advisor assignment, ensuring efficient communication and progress tracking. With features such as evaluation management and real-time status updates, the system enhances collaboration between students, faculty, and project advisors, ultimately contributing to the successful execution of final year projects

2 Technology Stack

Leveraging Windows Form.NET for Frontend Development and SQL for Backend Operations.

3 SQL Concepts

The major concepts of SQL that have been used in this project are:

- Joins
- Sub Queries
- Aggregate Functions
- Group by Clause

3.1 JOINS

Joins in SQL are used to combine rows from two or more tables based on a related column between them. They enable you to retrieve data from multiple tables in a single query, providing a way to connect information distributed across different parts of a database. The primary reasons for using joins in SQL are

- Data Retrieval
- Normalization
- Relationship
- Efficiency
- Complex Queries

3.2 Aggregate Functions

Aggregate functions in SQL are used to perform a calculation on a set of values and return a single, summarized result. These functions are often applied to columns in a SELECT statement and are commonly used for statistical and summarization purposes. Some Aggregate Functions used in this project are:

- Count
- AVG
- MIN
- MAX

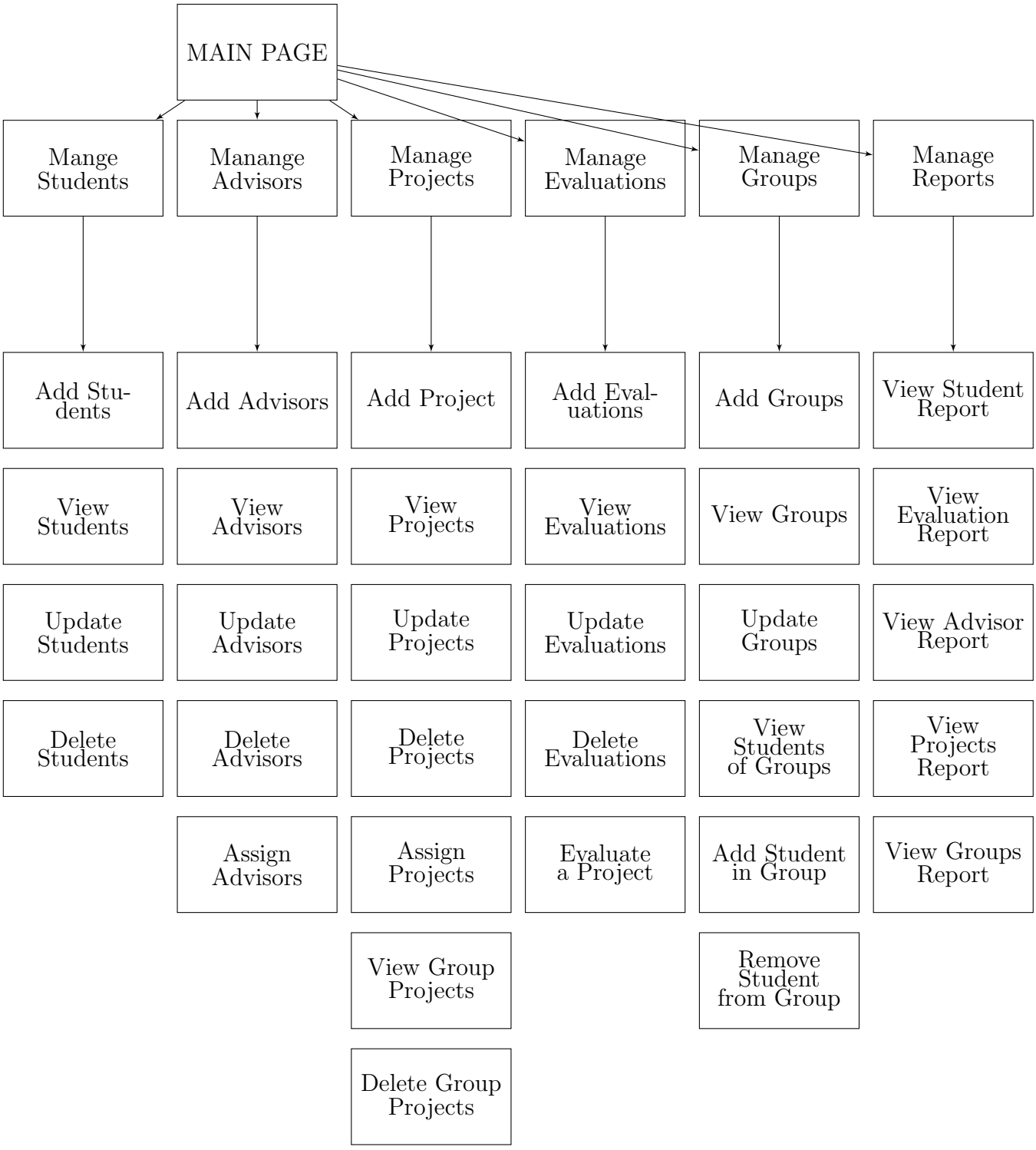
3.3 Sub Queries

A subquery, or nested query, in SQL is a query embedded within another query. The subquery is enclosed within parentheses and is used to retrieve data that will be used in the main query as a condition or value.Subqueries are powerful tools in SQL, allowing for more complex and flexible queries by breaking down problems into smaller, more manageable parts. They provide a way to filter, transform, or aggregate data within a larger query.

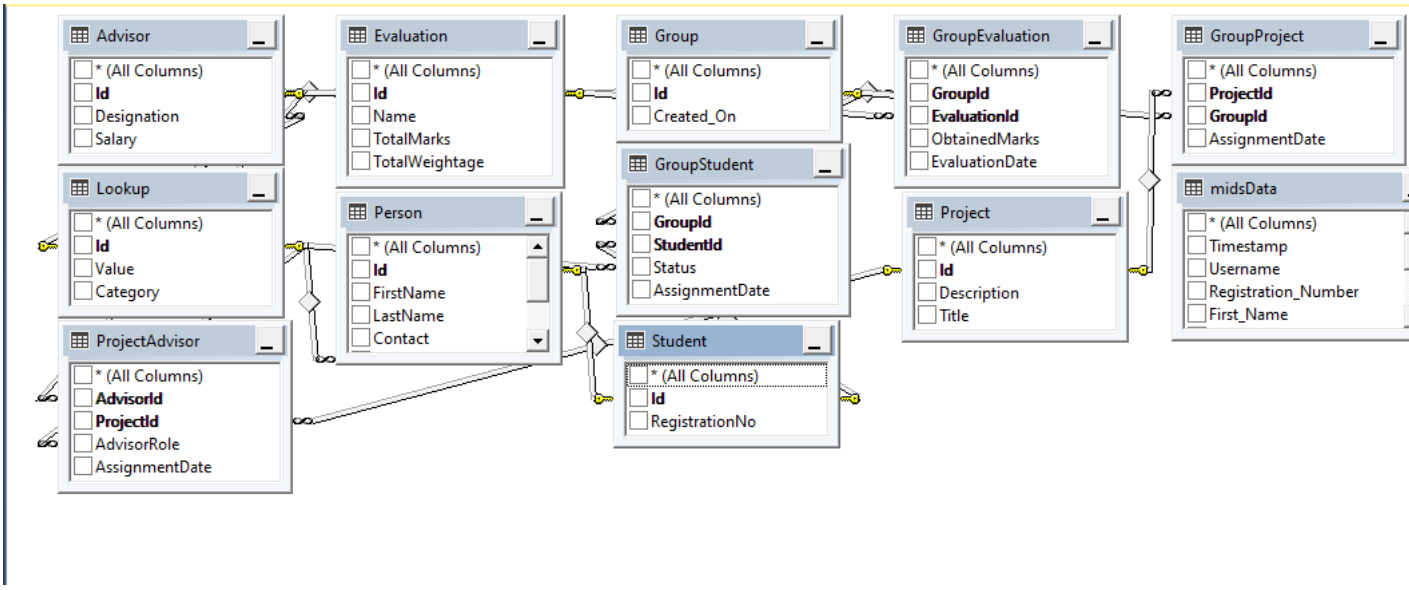
3.4 Group By Clause

The GROUP BY clause in SQL is used to group rows that have the same values in specified columns into summary rows, like "total" or "average". It is often used in conjunction with aggregate functions (such as SUM(), COUNT(), AVG(), MAX(), or MIN()) to perform operations on each group of rows.

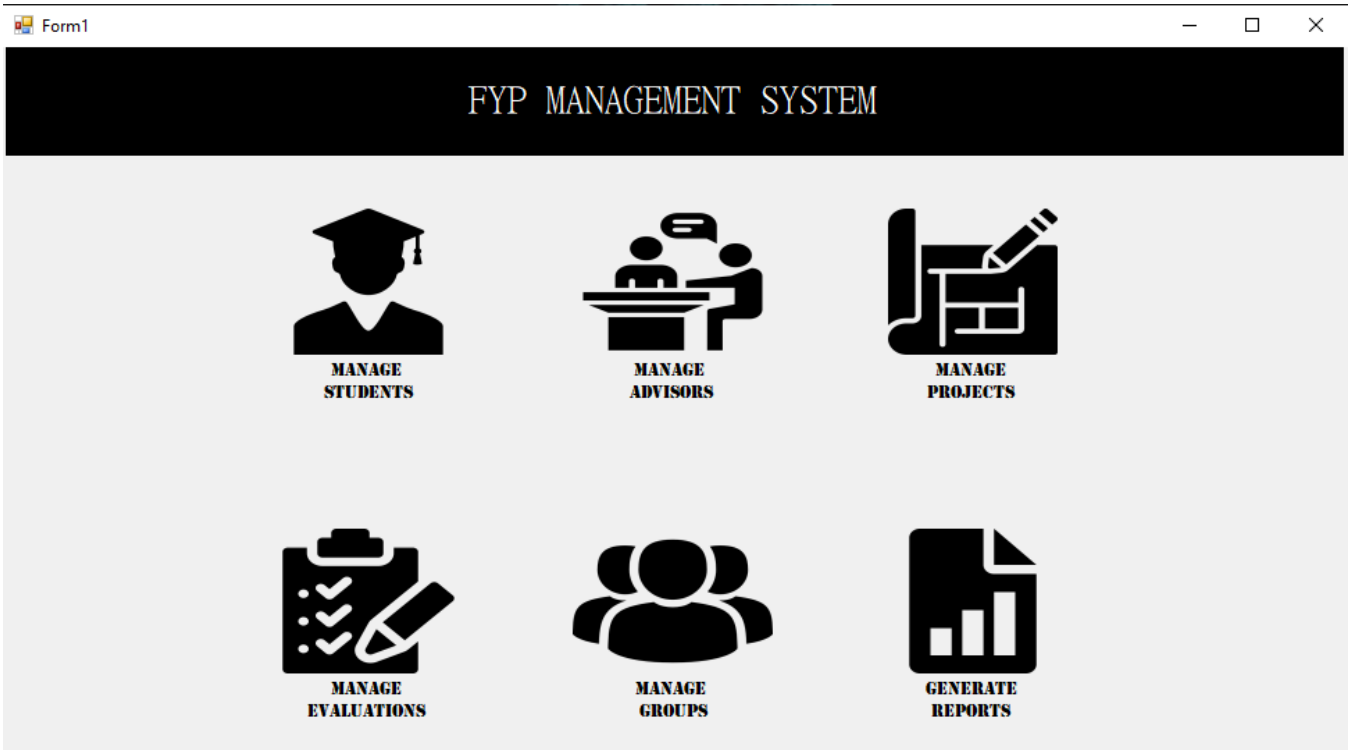
4 Flow Chart

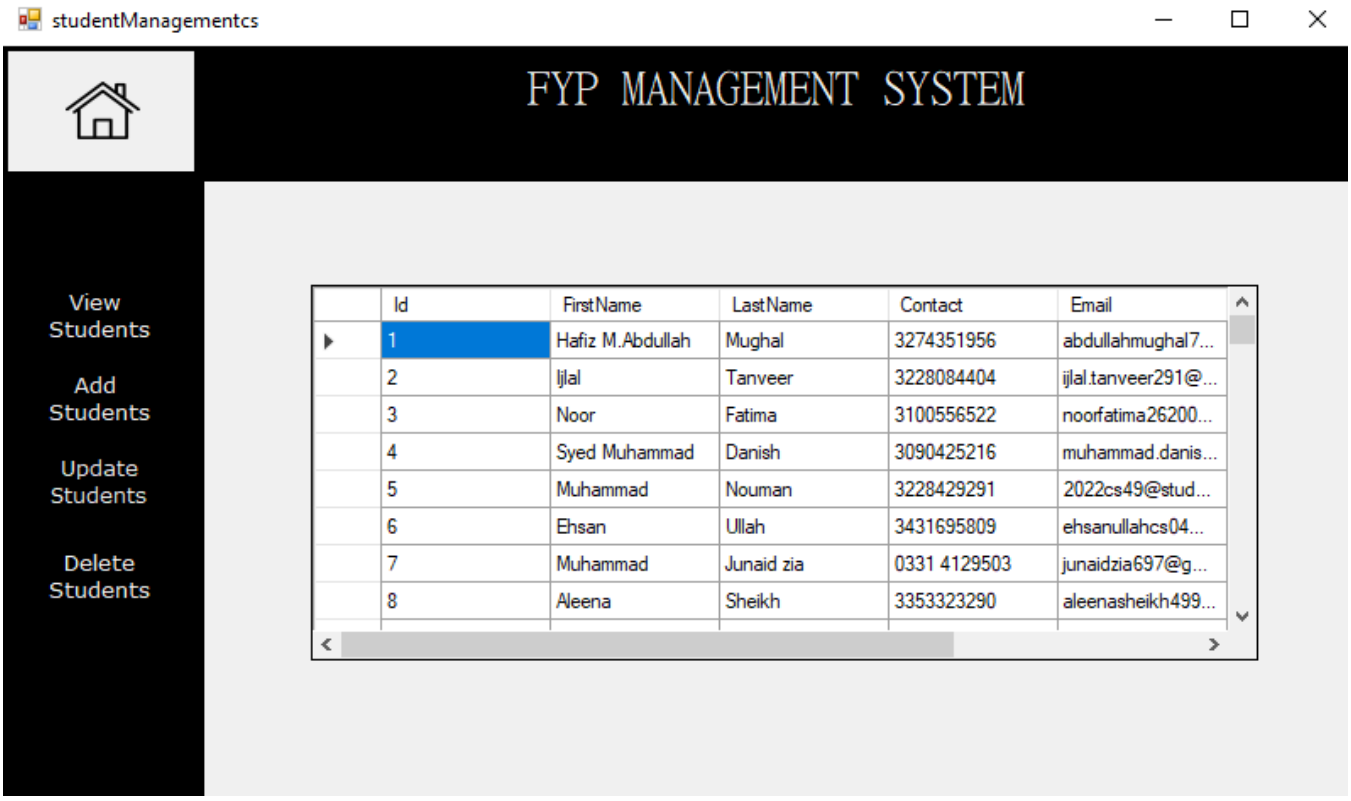

MAIN PAGE
Mange Students
Manange Advisors
Manage Projects
Manage Evaluations
Manage Groups
Manage Reports
Add Students
Add Advisors
Add Project
Add Evaluations
Add Groups
View Student Report
View Students
View Advisors
View Projects
View Evaluations
View Groups
View Evaluation Report
Update Students
Update Advisors
Update Projects
Update Evaluations
Update Groups
View Advisor Report
Delete Students
Delete Advisors
Delete Projects
Delete Evaluations
View Students of Groups
View Projects Report
Assign Advisors
Assign Projects
Evaluate a Project
Add Student in Group
View Groups Report
View Group Projects
Remove Student from Group
Delete Group Projects

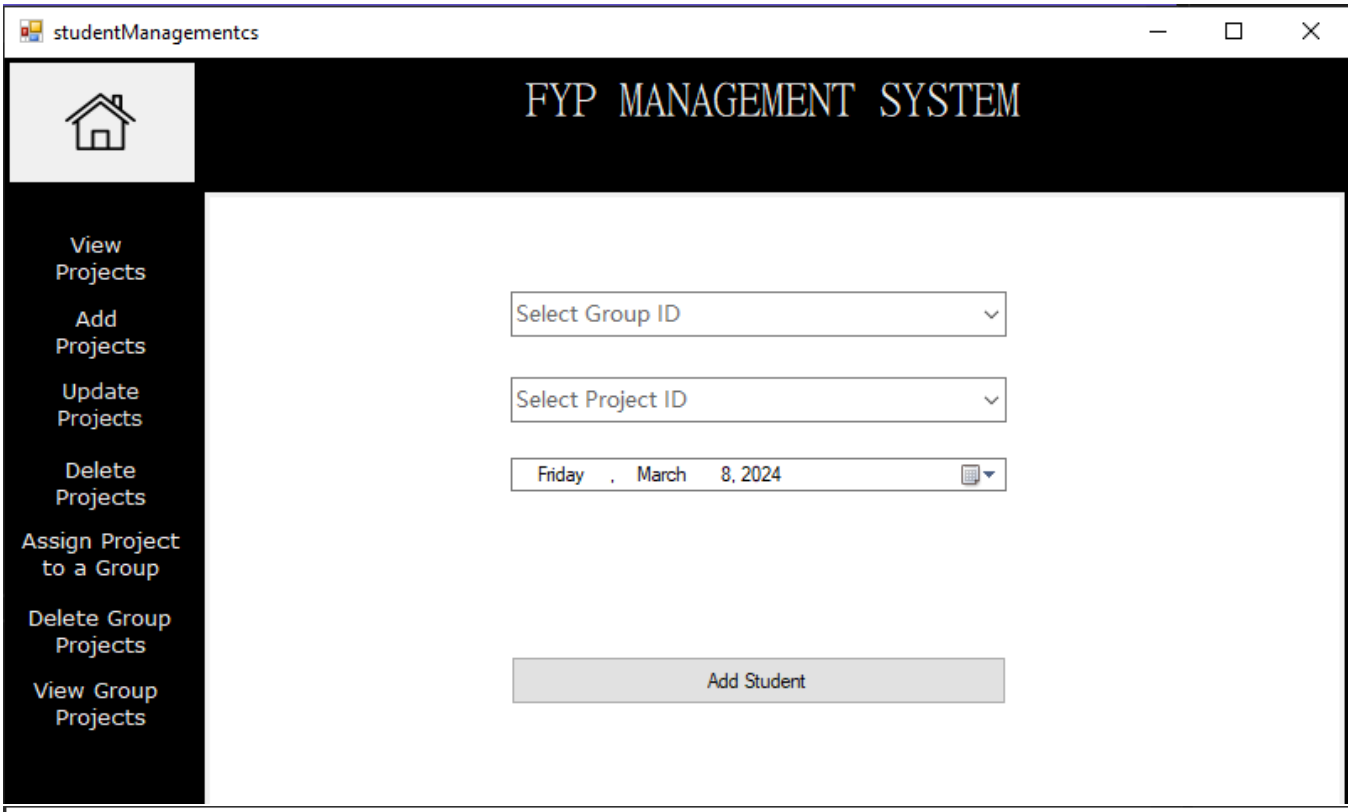
5 Tables Diagram

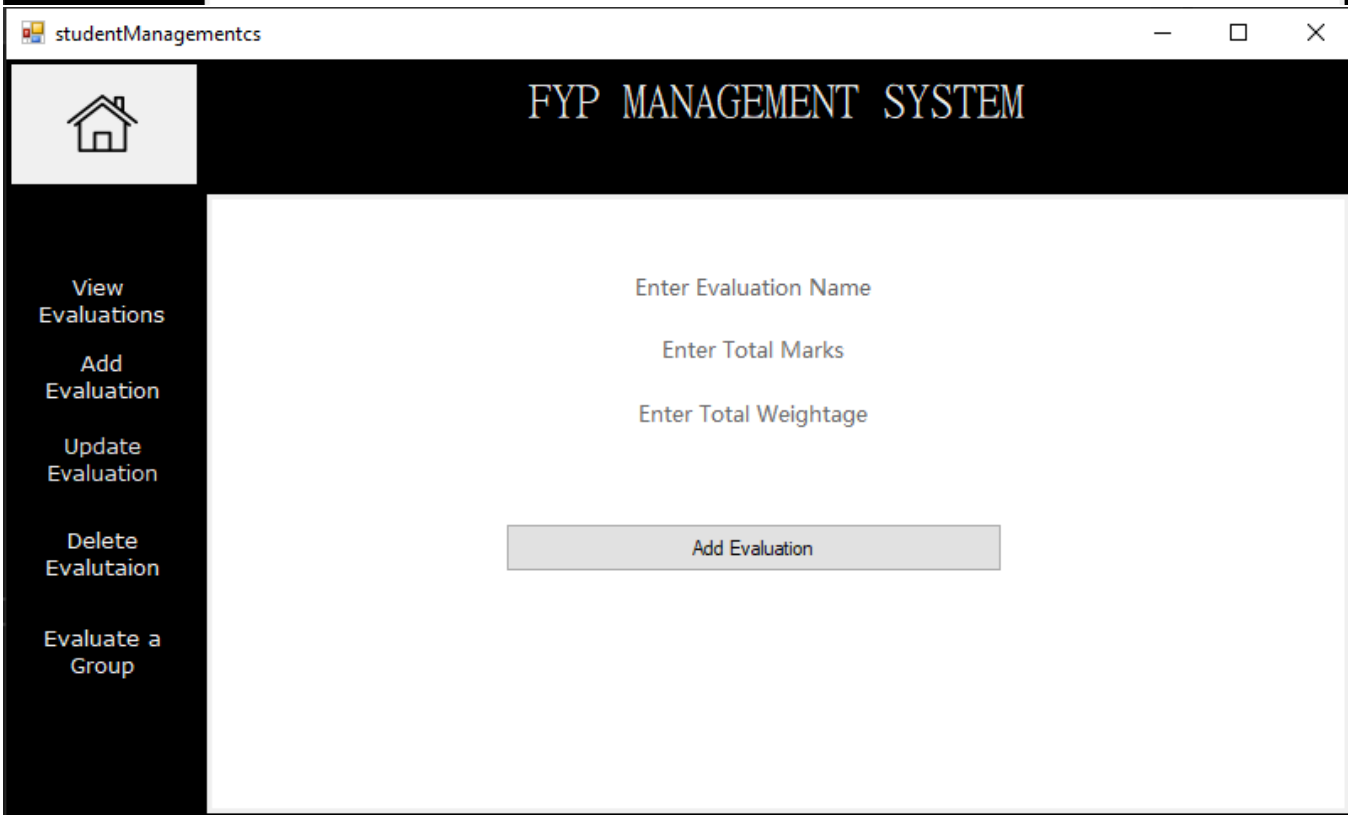


6 WireFrames






studentManagementcs
FYP MANAGEMENT SYSTEM
View Projects
Add Projects
Update Projects
Delete Projects
Assign Project to a Group
Delete Group Projects
View Group Projects
Select Group ID
Select Project ID
Friday , March 8, 2024
Add Student


studentManagementcs
FYP MANAGEMENT SYSTEM
View Evaluations
Add Evaluation
Update Evaluation
Delete Evalutaion
Evaluate a Group
Enter Evaluation Name
Enter Total Marks
Enter Total Weightage
Add Evaluation

7 Conclusion

The FYP Management System has been successfully developed and implemented to address the complex challenges associated with overseeing final year projects within the university. Through meticulous planning and execution, the system has efficiently allocated projects, grouped students, and assigned advisors, fostering seamless communication and progress tracking.he lessons learned, including the importance of data normalization and linking, have significantly contributed to the project's robust architecture.The system's impact on reducing data replication and optimizing data distribution across different layers underscores its significance. As the project concludes, appreciation is extended to mentors, teachers, and colleagues for their invaluable support, paving the way for future enhancements and continued positive contributions to the management of final year projects.