

Student Evaluation System

Mid Term Project



Supervisor:

Mr.Nauman Babar

Registration No.

2021-CS-55

Name:

Bilal Bishart

**University of Engineering and Technology
Lahore, Pakistan**

1 Acknowledgements

I express my heartfelt gratitude to Allah Almighty for giving me the power to complete this project within the given timeframe. I am deeply grateful to Mr. Nauman Babar for his consistent support and supervision throughout this project. Additionally, I extend my appreciation to Mr. Samyan Qayyum Wahla for his passion in teaching Database Theory and for simplifying the concepts, which were helpful for me to complete this project.

Contents

1	Acknowledgements	2
2	Introduction	5
2.1	Description	5
2.2	Motivation	5
2.3	Target Audience	5
3	Functional Requirements	5
4	Database Design	6
4.1	LookUp	6
4.2	Student	7
4.3	Student Attendance	7
4.4	Class Attendance	7
4.5	CLOs	7
4.6	Rubrics	7
4.7	Rubric Levels	7
4.8	Assessments	7
4.9	Assessment Components	7
4.10	Student Result	7
5	GUI(Graphical User Interface)	8
5.1	Welcome Screen	8
5.2	Manage Student UI	10
5.3	Add Student UI	11
5.4	Manage CLOs UI	12
5.5	Manage Rubrics UI	14
5.6	Rubric Level UI	16
5.7	Add Rubric Level UI	19
5.8	Manage Assessment UI	19
5.9	Assessment Components UI	21
5.10	Student Result UI	24
5.11	Class Attendance UI	26
5.12	Generate PDF Reports UI	28
6	Generate Report Queries	29
6.1	Assessment Wise Class Report (Detailed)	29
6.2	Assessment Wise Class Report (Summerized)	30
6.3	CLO Wise Class Report (Detailed)	30
6.4	CLO Wise Class Report (Summerized)	30
6.5	Attendance Wise Class Report	31
6.6	CLO Wise Individual Report (Detailed)	31
6.7	CLO Wise Individual Report (Summerized)	31
6.8	Assessment Wise Individual Report (Detailed)	32
6.9	Assessment Wise Individual Report (Summerized)	32
7	Testing	33
8	Limitations	33

9	Future Work	33
10	Collaboration	33
11	Conclusion	33

2 Introduction

2.1 Description

The management of CLOs (Course Learning Outcomes) is a crucial aspect for universities that implement an outcome-based education model. However, as the number of students and courses increases, manual management of CLOs can be time-consuming and prone to errors. To address this issue, an automated system has been developed, which streamlines the entire process and makes it more efficient and reliable. This system offers various functionalities, such as creating, updating, retrieving, and deleting data related to students, rubrics, assessment components, and assessments. This ensures that all information is accurate and up-to-date, enabling administrators to make better-informed decisions.

The system generates comprehensive PDF reports that provide each student's performance, making it easier for administrators to evaluate their progress. The report summarizes each student's CLOs, their evaluation, the number of CLOs, the marks they received, and their total marks. With this information, administrators can identify areas that require improvement and create targeted learning plans for each student.

The system is designed exclusively for administrators, ensuring that only authorized personnel can access and modify the data, adding an extra layer of security to the system. Moreover, the system's user-friendly interface enables administrators to navigate and perform tasks without requiring any special technical knowledge. Overall, the CLOs management system is an indispensable tool for universities that wish to streamline their outcome-based education approach. By automating the process, the system reduces errors, saves time, and provides valuable insights into each student's progress.

2.2 Motivation

CLOs are based on Rubrics which can help students do their best work by telling them exactly what is expected. They also give feedback on how they're doing, which keeps students motivated. Rubrics also have different levels of achievement, which rewards students for working hard.

2.3 Target Audience

The CLOs management system is intended for the faculty department of the university, which consists of professors, lecturers, program coordinators, and other staff members responsible for overseeing the academic performance of students. Its purpose is to assist them in managing the CLOs of all current university students in various departments and assessing their development.

3 Functional Requirements

The responsibility of assigning marks to students and the authority to modify or remove mark-related is dedicated to administrator. The administrator can enter assessment information and allocate grades to student as required. The system allows the administrator to create PDF reports. Admin can perform following operations:

1. CRUD operations on student data.
2. Mark attendance of students according to date.
3. Performs CRUD operations on CLOs.
4. Each CLO has multiple rubrics

5. Each Rubric has maximum of 4 rubric levels
6. Performs CRUD operations on assessment.
7. Each assessment has different components
8. Calculates result based on assesment, assessment component and rubric level
9. Generates pdf reports based on different sections.

4 Database Design

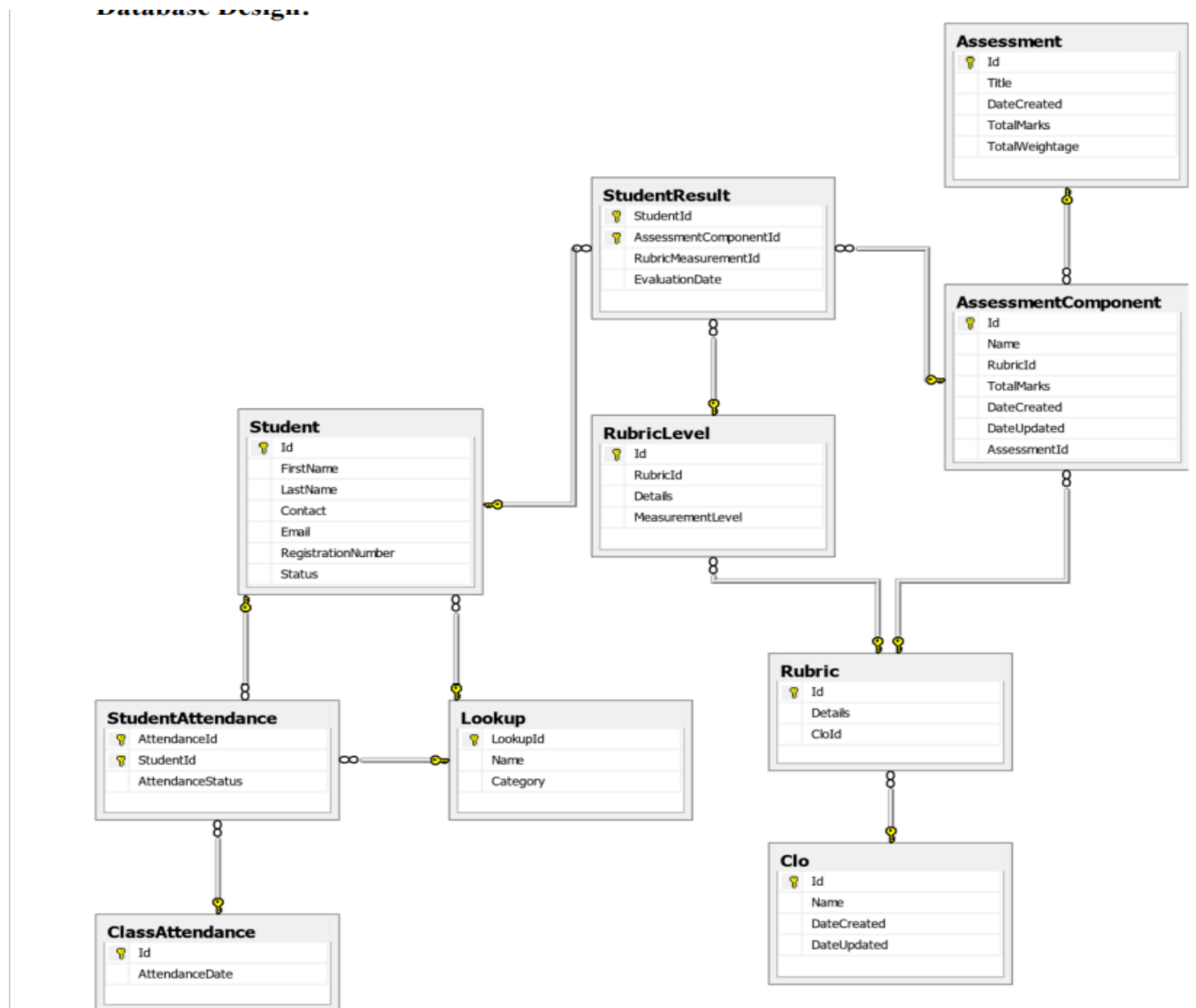


Figure:4.1 Database Design

4.1 LookUp

The Lookup table stores information about Student status and attendance status.

4.2 Student

The Student table contains information about each individual student, such as their ID, first and last name, contact information, email address, registration number, and status.

4.3 Student Attendance

The Student Attendance table stores information about individual student attendance, including an attendance ID, student ID, and attendance status.

4.4 Class Attendance

This table contains information about attendance for a particular class, including an ID and date of the attendance record.

4.5 CLOs

This table holds information about Course Learning Outcomes associated with assessments, including an ID, name, creation date, and last update date.

4.6 Rubrics

This table contains details about rubrics used in assessments, including an ID, rubric details, and the associated CLO ID.

4.7 Rubric Levels

The Rubric Level table contains information about the levels of rubrics, including an ID, associated rubric ID, details about the level, and measurement level of the rubric.

4.8 Assessments

The Assessment table holds information about assessments, such as ID, title, creation date, total marks available, and total weightage.

4.9 Assessment Components

This table contains information about components of assessments, including an ID, name, associated rubric ID, creation date, total marks, and last update date.

4.10 Student Result

This table stores information about individual student results based on assessment components, including student ID, assessment component ID, rubric measurement ID, and evaluation date.

5 GUI(Graphical User Interface)

5.1 Welcome Screen

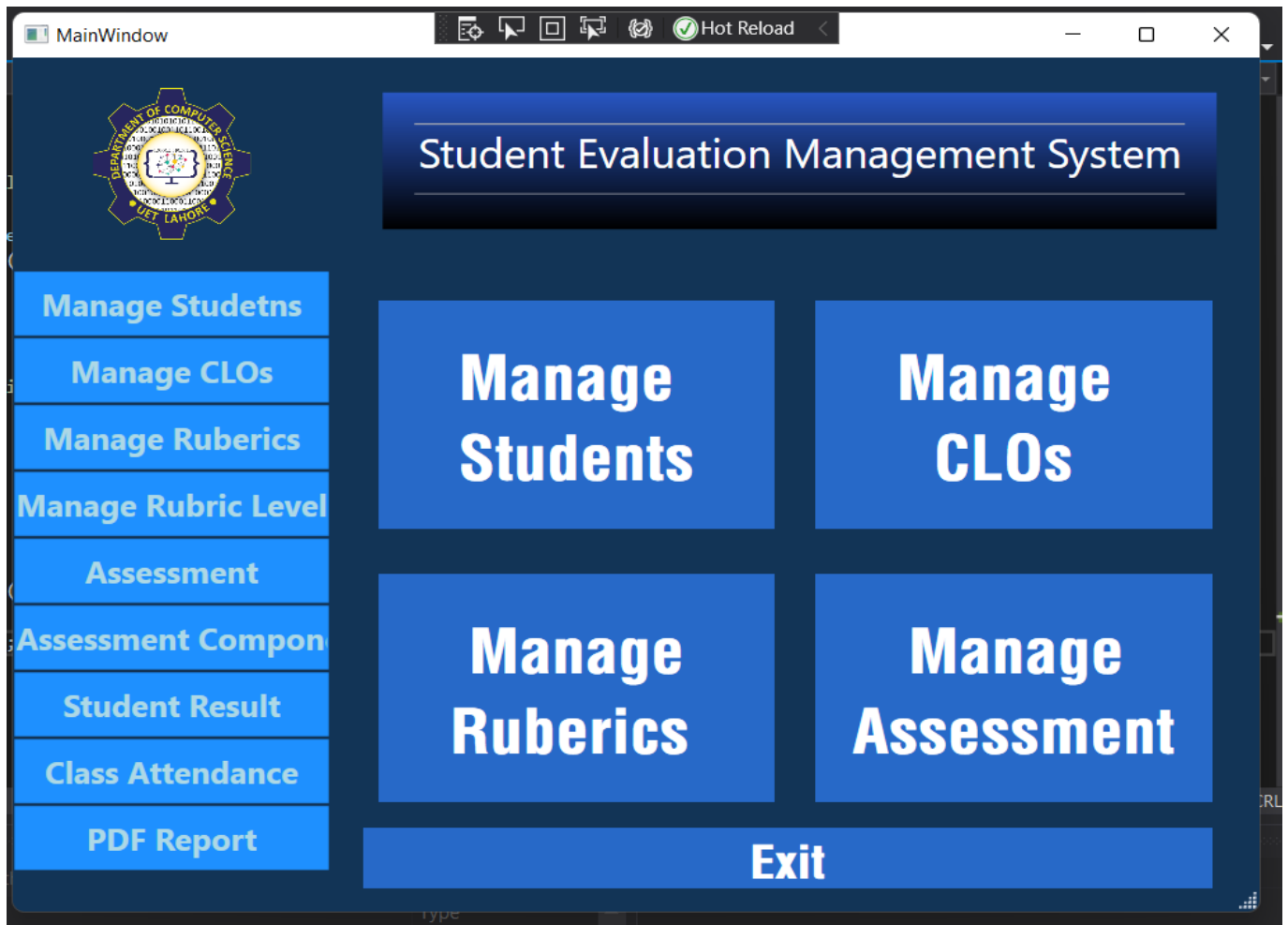


Figure 5.1.1: Welcome Page

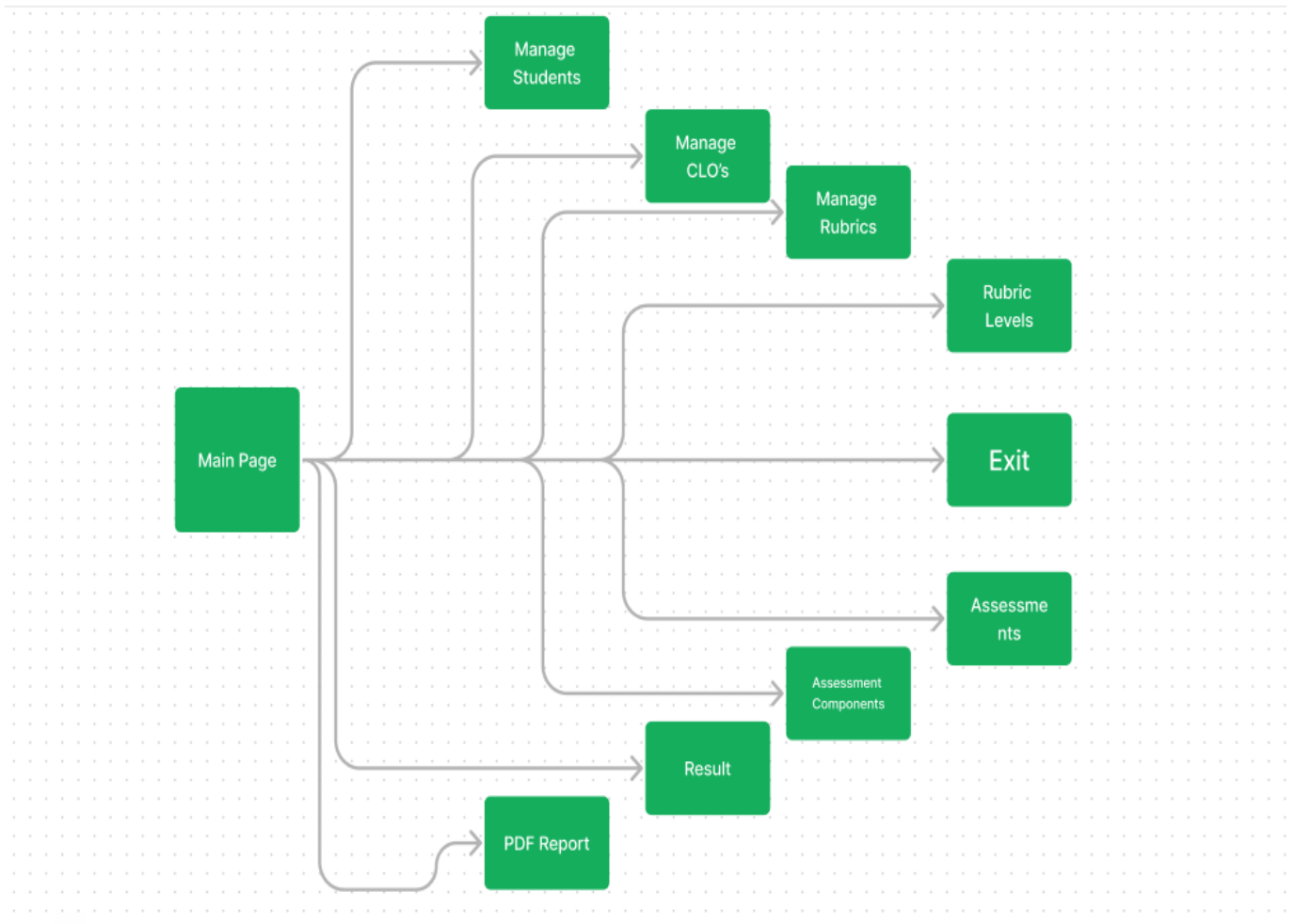


Figure 5.1.2: Welcome Page Flow Diagram

5.2 Manage Student UI

The user can add, delete, or update students by making a choice to add, update, or delete.

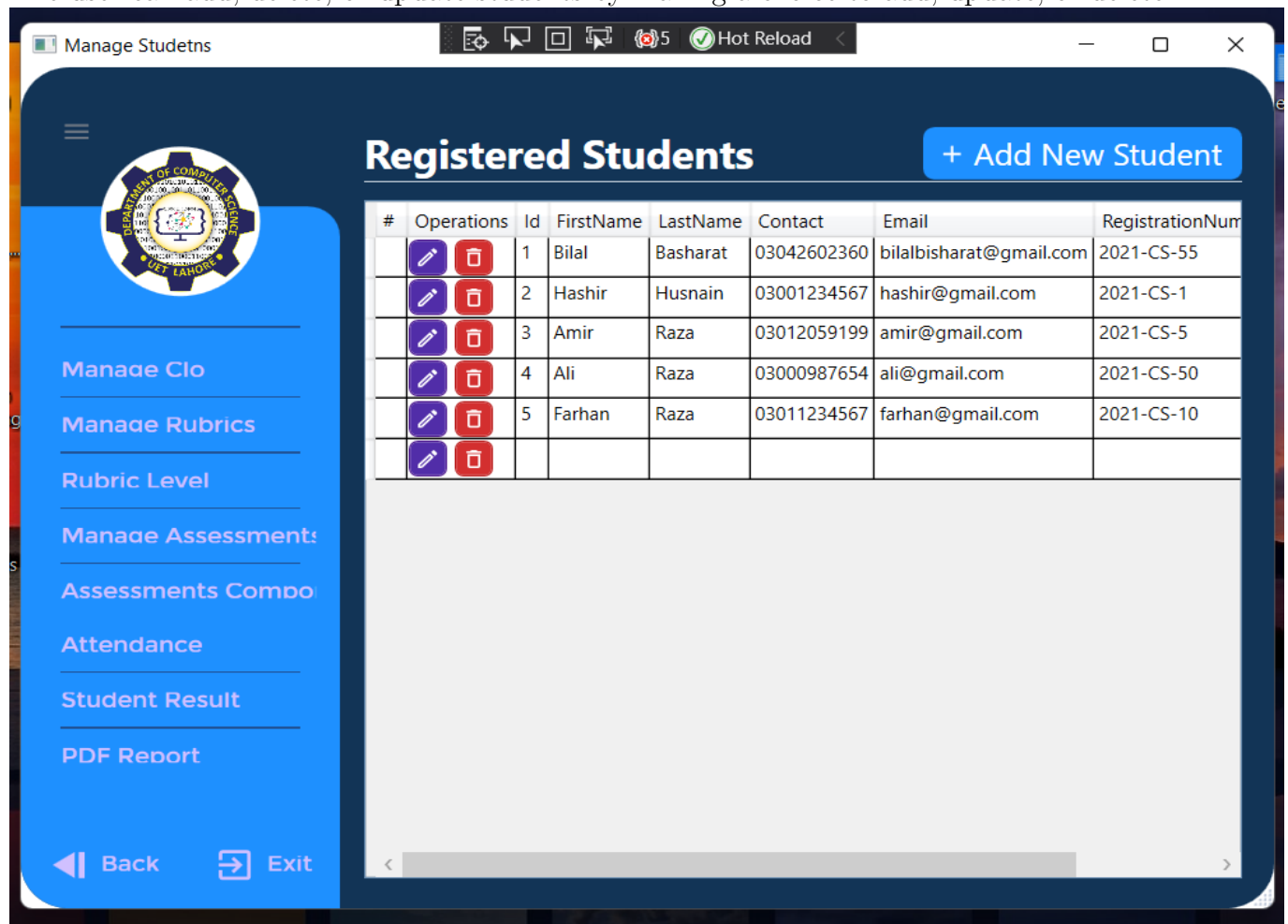


Figure 5.2: Student Page

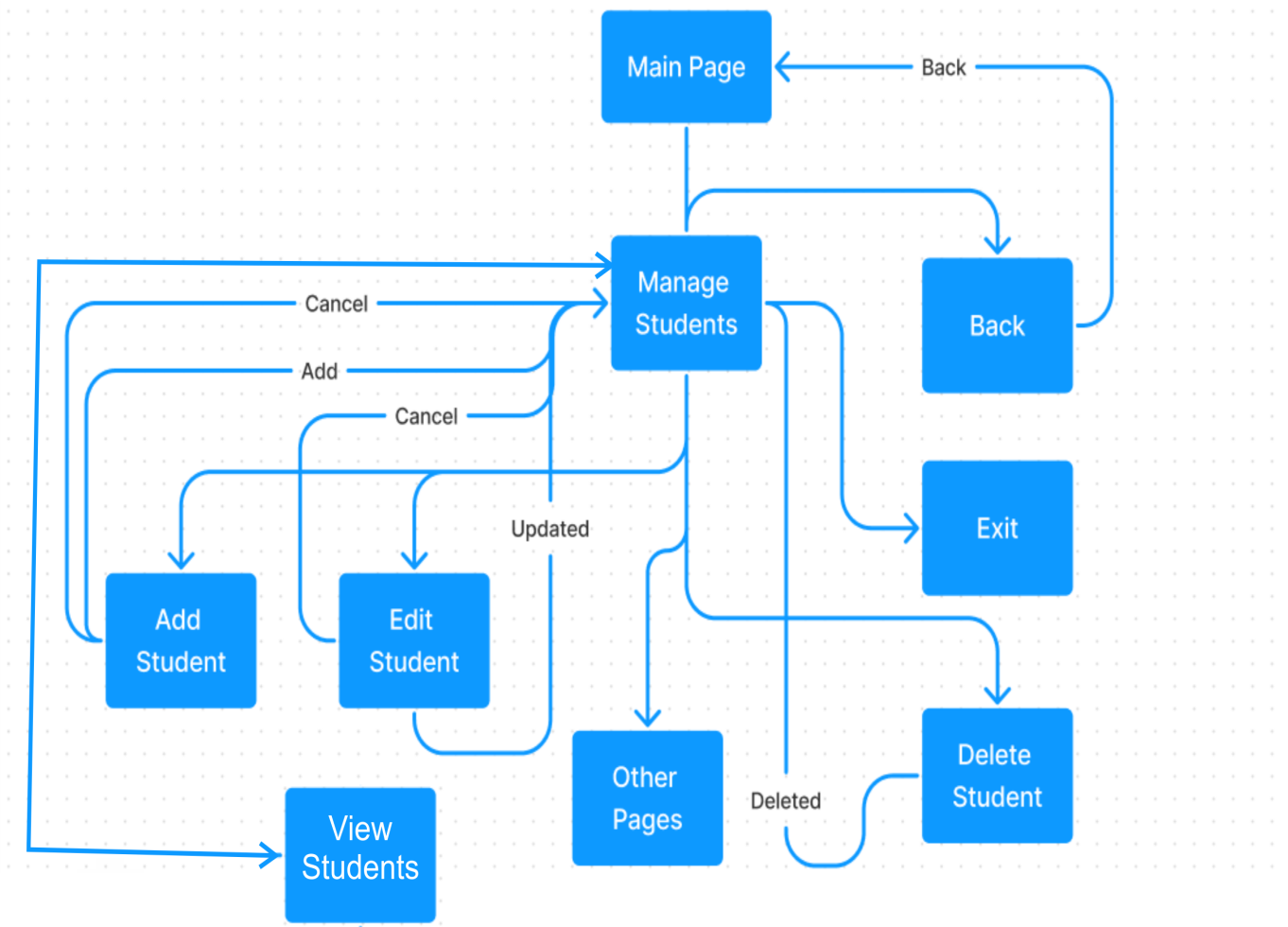
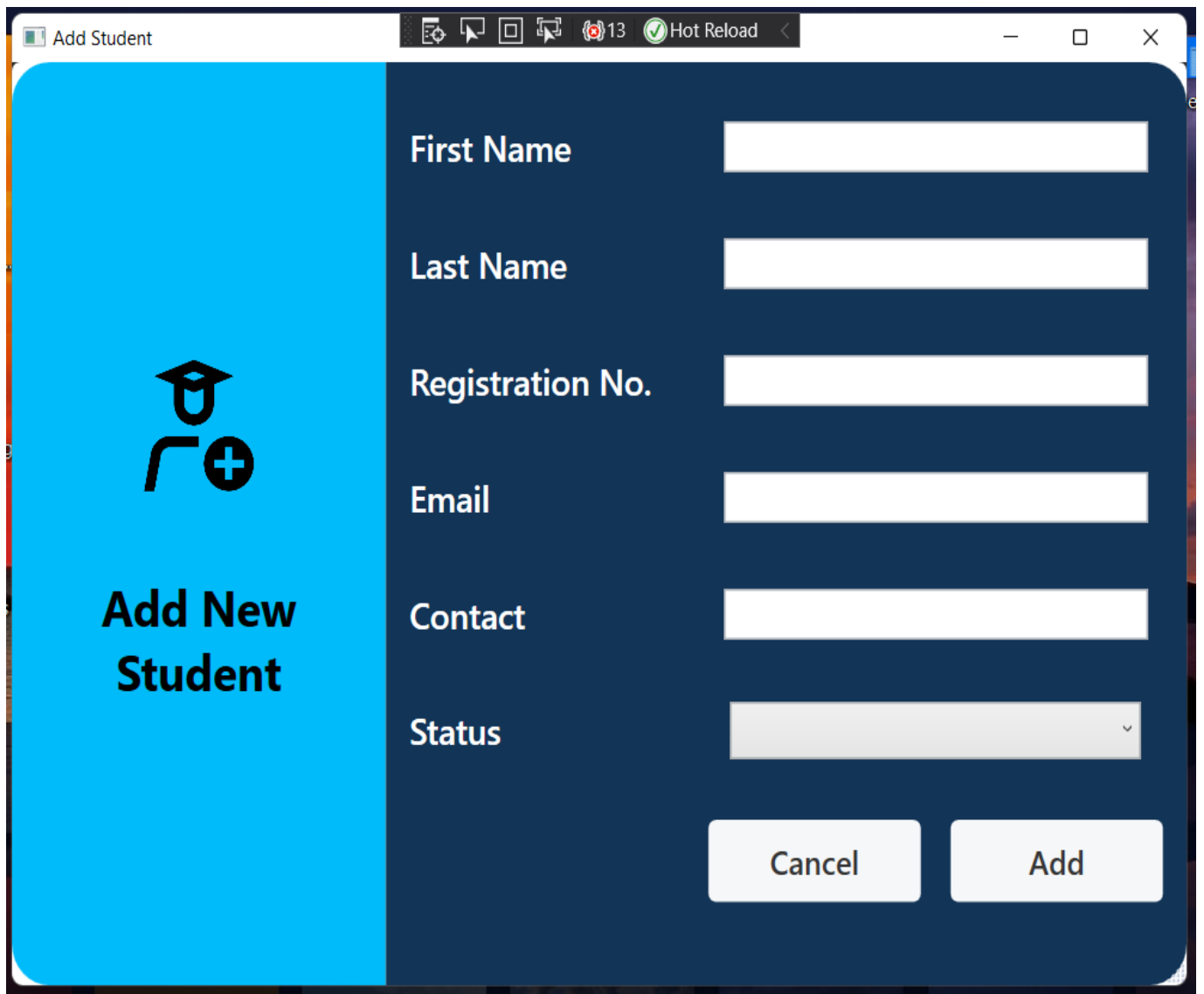


Figure 5.2.1: Activity Diagram

5.3 Add Student UI

In this page you can add a student by providing relevant details.



The screenshot shows a web browser window titled "Add Student". The browser's address bar and developer tools are visible at the top. The page has a dark blue background. On the left side, there is a light blue vertical panel containing a black icon of a graduation cap with a plus sign inside a circle, and the text "Add New Student" in bold black font. On the right side, there are six input fields with labels: "First Name", "Last Name", "Registration No.", "Email", "Contact", and "Status". The "Status" field is a dropdown menu. At the bottom right, there are two buttons: "Cancel" and "Add".

Figure 5.3: Add Student Page

5.4 Manage CLOs UI

This page information about CLOs. You can add new CLO, edit or delte CLOs.



Manage Students

Manage Rubrics

Rubric Level

Manage Assessment

Assessments Comparison

Attendance

Student Result

PDF Report

Back

Exit

ManageCLOs

CLO Name

+ Add

#	Operations	Id	Name	DateCreated	DateUpdated
	<div>Edit</div> <div>Delete</div>	1	Clo2	3/9/2023 11:18:25 AM	3/9/2023 11:18:47 AM
	<div>Edit</div> <div>Delete</div>	2	Clo2	3/9/2023 1:28:50 PM	3/9/2023 1:28:50 PM
	<div>Edit</div> <div>Delete</div>	3	Clo3	3/9/2023 1:28:54 PM	3/9/2023 1:28:54 PM
	<div>Edit</div> <div>Delete</div>				

Figure 5.4: Manage CLOs Page

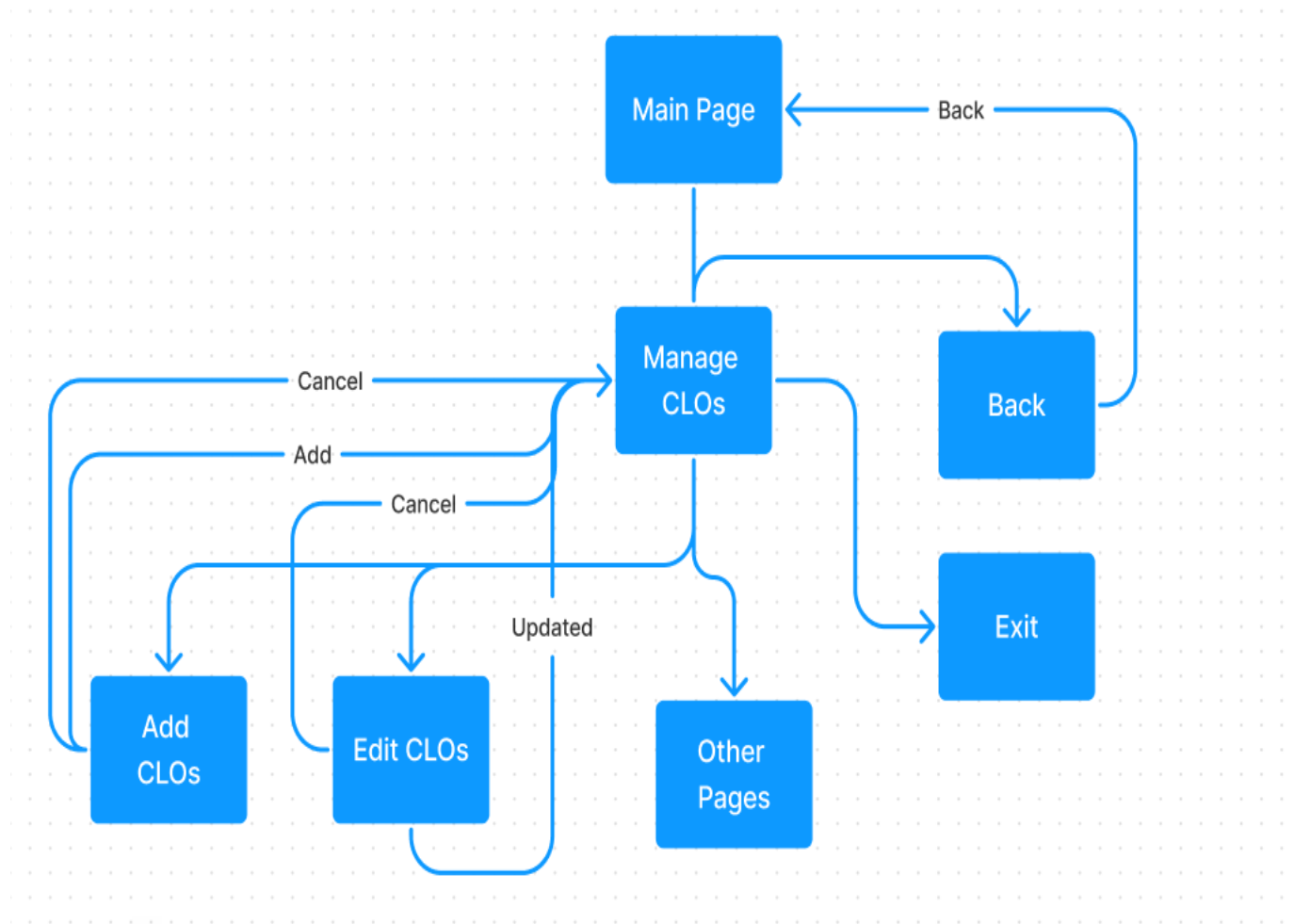


Figure 5.4.2: CLOs FLOW Diagram

5.5 Manage Rubrics UI

This provides information about rubrics i.e. you can add, edit or delete rubrics.

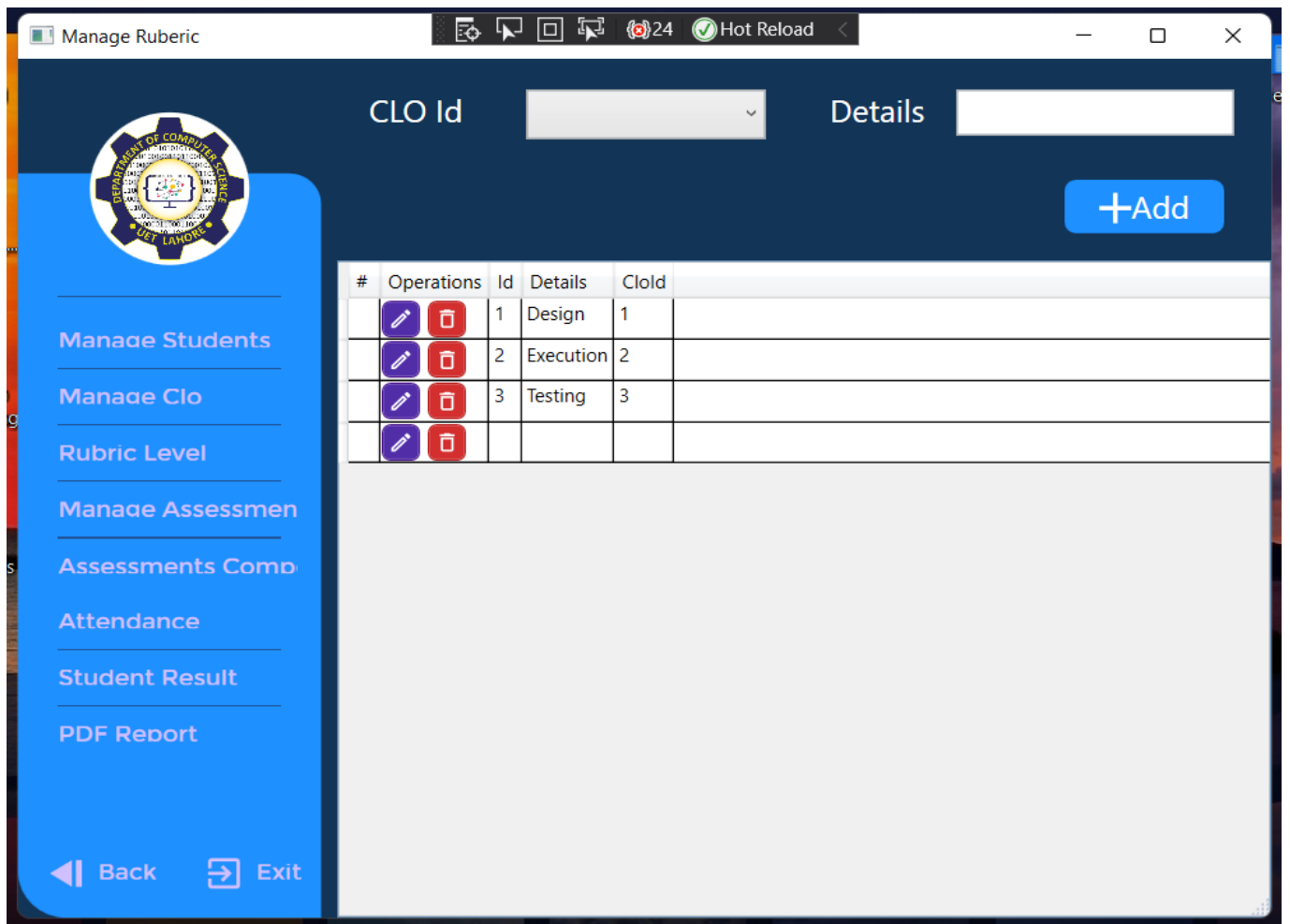


Figure 5.5.1: Manage Rubrics UI

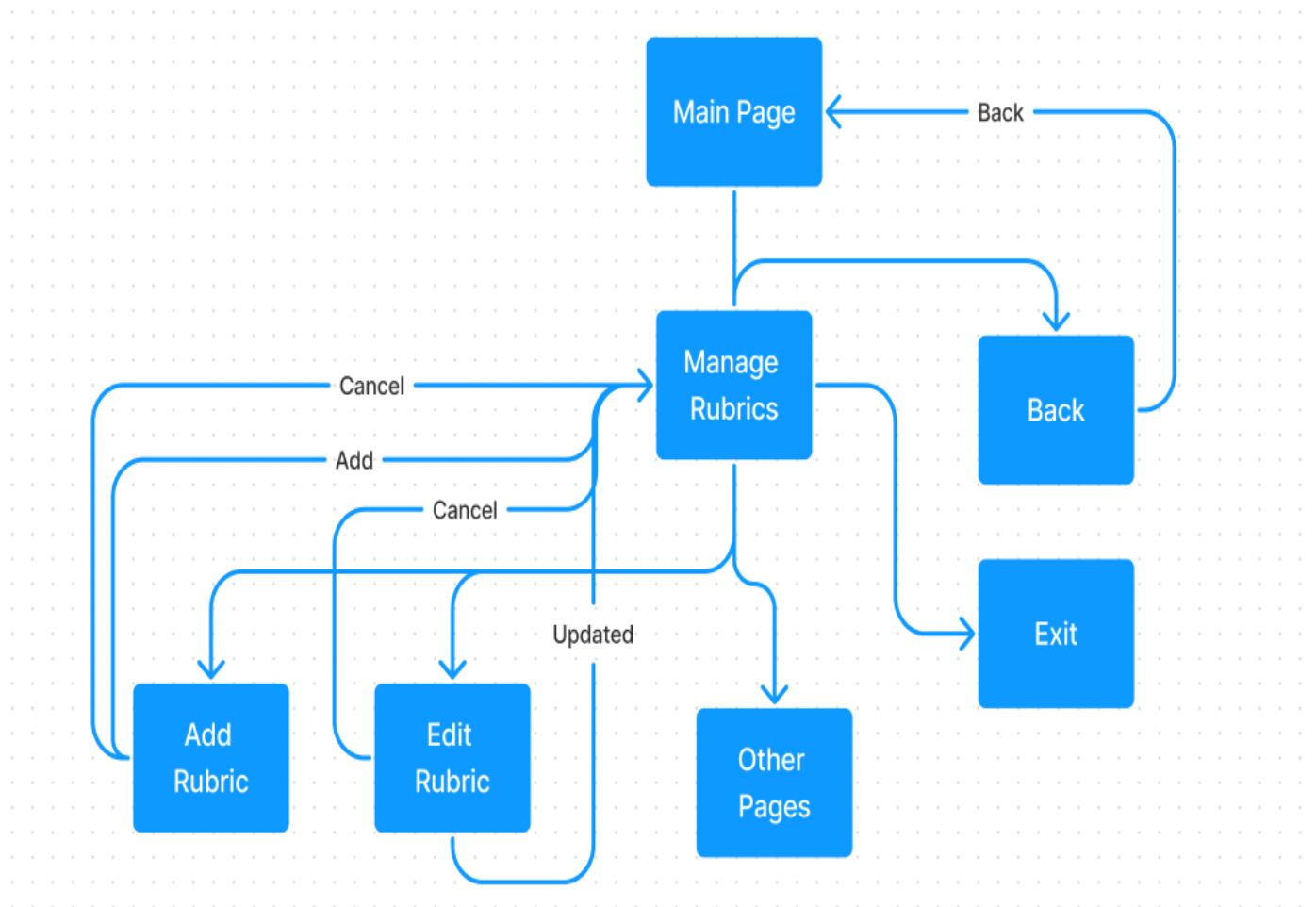



Figure 5.5.2: Rubrics Flow Diagram

5.6 Rubric Level UI

This provides information about Rubric levels. you can add new level by simply clicking on add new level button and provide the relevant information for level addition. You can also edit or delete rubric levels.



Manage Students

Manage CLO

Manage Rubrics

Manage Assessment

Assessments Comparison

Attendance

PDF Report

Back

Exit

Rubric Level

Rubric Id

Measurement Level

Details

+ Add New Levels



























#	Operations	Id	RubricId	Details	MeasurementLevel	
	 	1	3	Unsatisfactory	1	
	 	2	3	Fair	2	
	 	3	3	Good	3	
	 	4	3	Exception	4	
	 	5	1	Lack of Syntax knowledge	1	
	 	6	1	minor errors	2	
	 	7	1	could be more efficient	3	
	 	8	1	No Errors	4	
	 	9	2	Irresponsible	1	
	 	10	2	reusability and cohesion	2	
	 	11	2	redundant	3	
	 	12	2	coded properly	4	
	 					

Figure 5.6: Rubric Levels UI

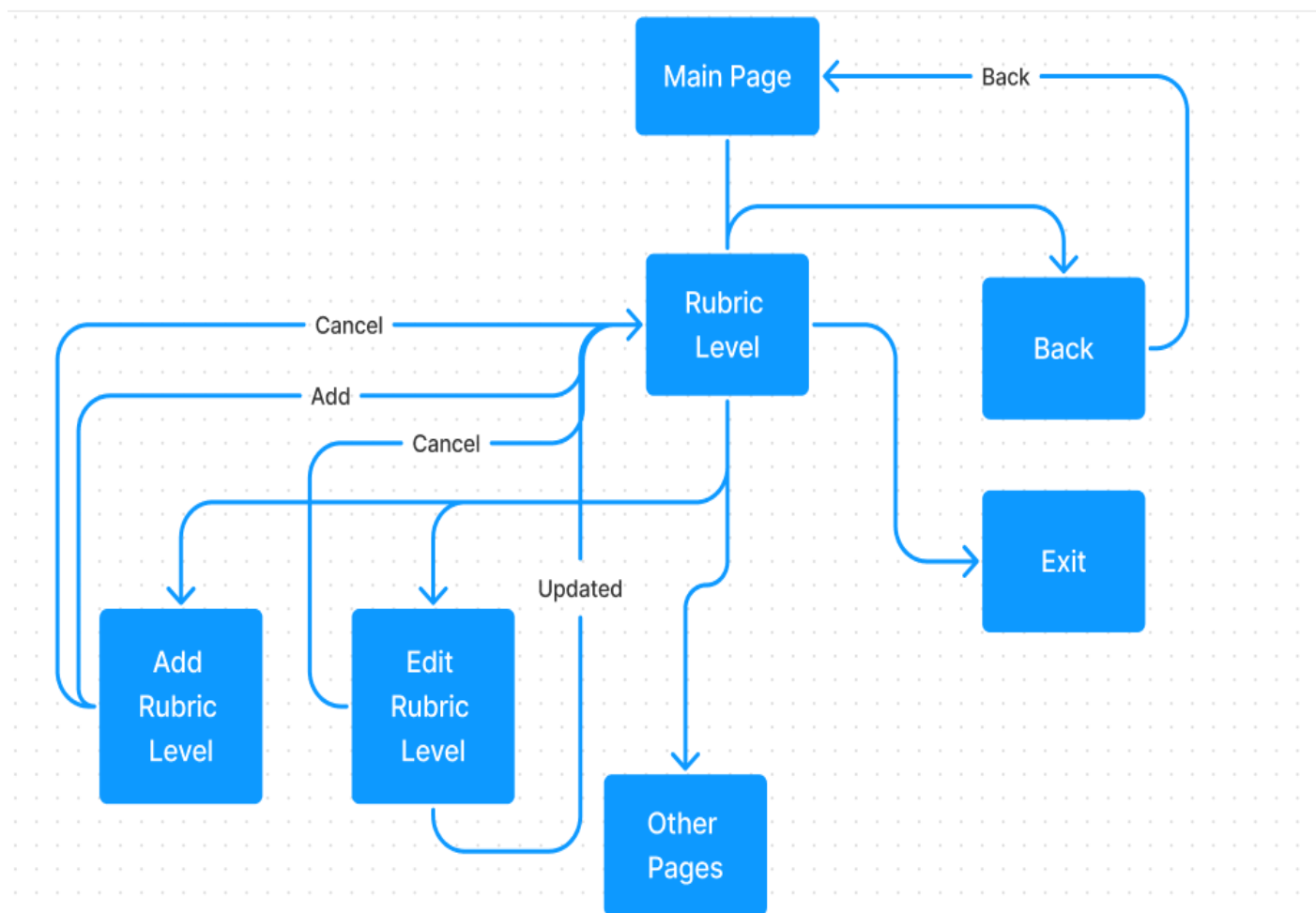


Figure 5.6.2: Rubric Level Flow Diagram

5.7 Add Rubric Level UI

In this page you have to provide relevant information of the rubric level to add it into the system.

The screenshot shows a web application window titled "Add Rubric Level". The browser's address bar shows "Add Rubric Level" and a "Hot Reload" button. The application has a dark blue theme. On the left, a light blue sidebar contains a 3x3 grid icon with checkmarks and the text "Add Rubric Level". The main area has a form with the following fields:

- Rberic Id**: A dropdown menu.
- Exceptional**: A text input field.
- Good**: A text input field.
- Fair**: A text input field.
- Unsatisfactory**: A text input field.

At the bottom right, there are two buttons: "Cancel" and "Add".

Figure 5.6: Add Rubric Level

5.8 Manage Assessment UI

This page gives information about different assessments based on academic activities. You can add new assessment, update or delete an assessment.

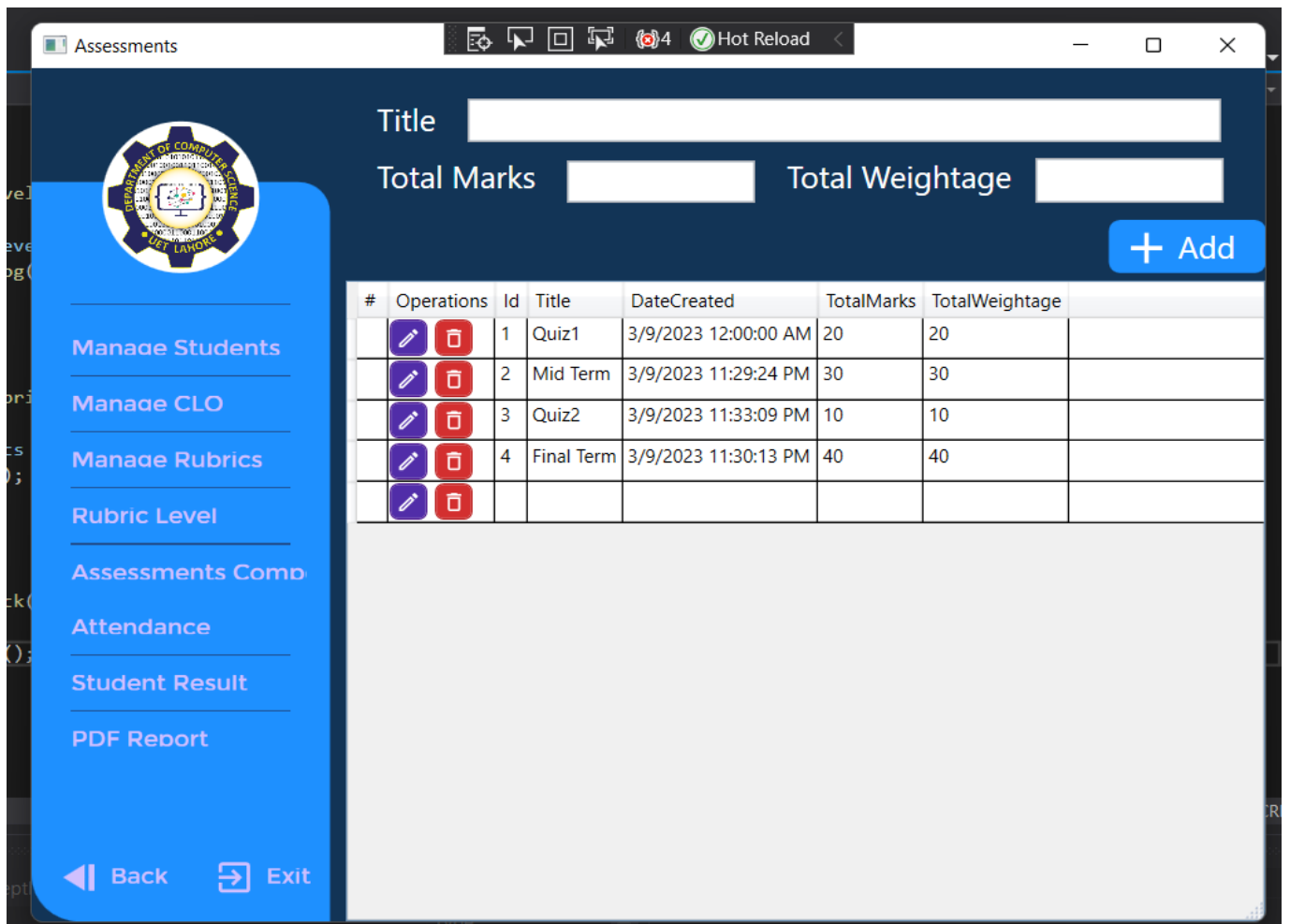


Figure 5.8.1: Manage Assessment UI

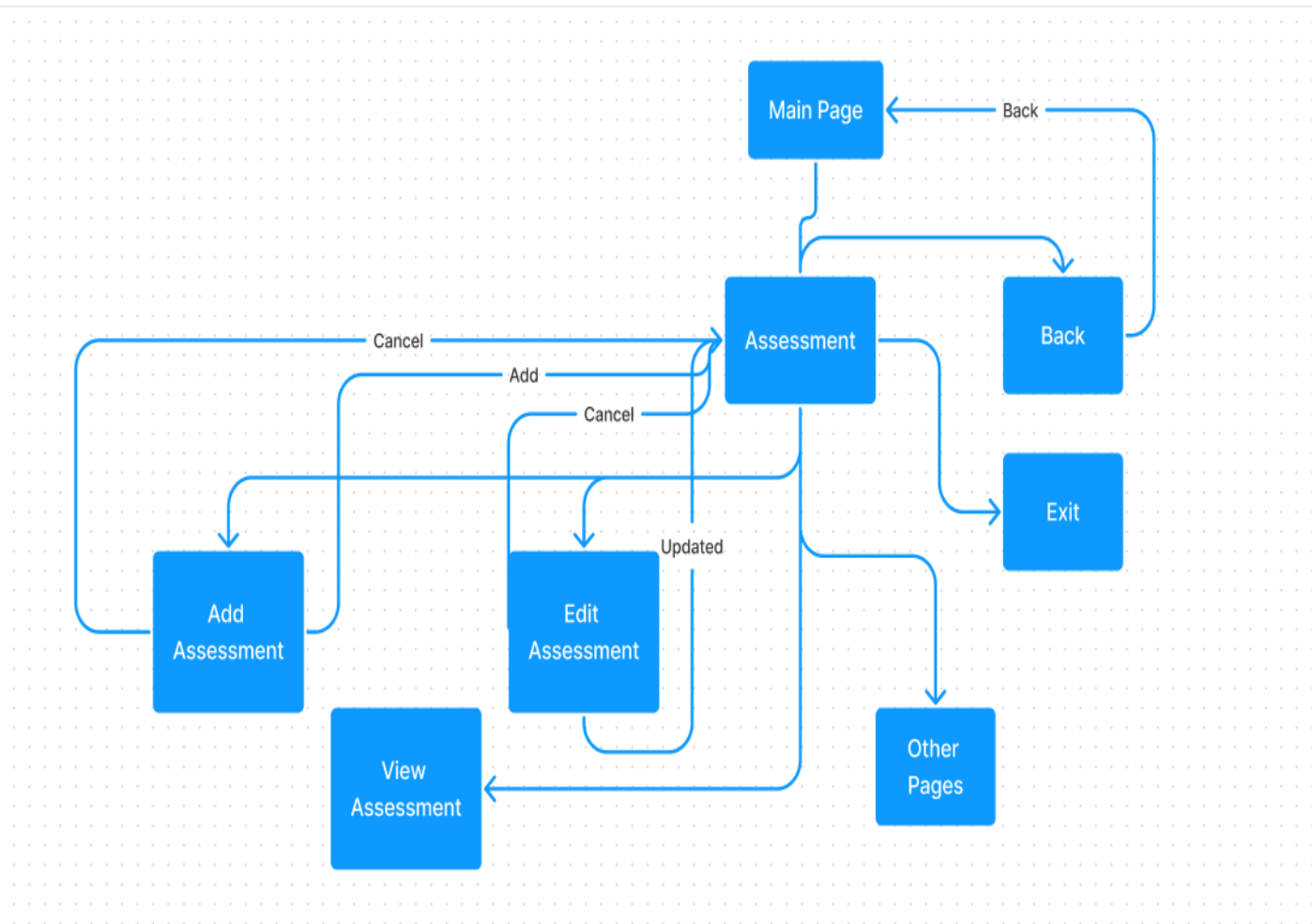


Figure 5.8.2: Manage Assessment Flow Diagram

5.9 Assessment Components UI

This page gives information about different components of an assessments. You can add new component, update or delete a component.

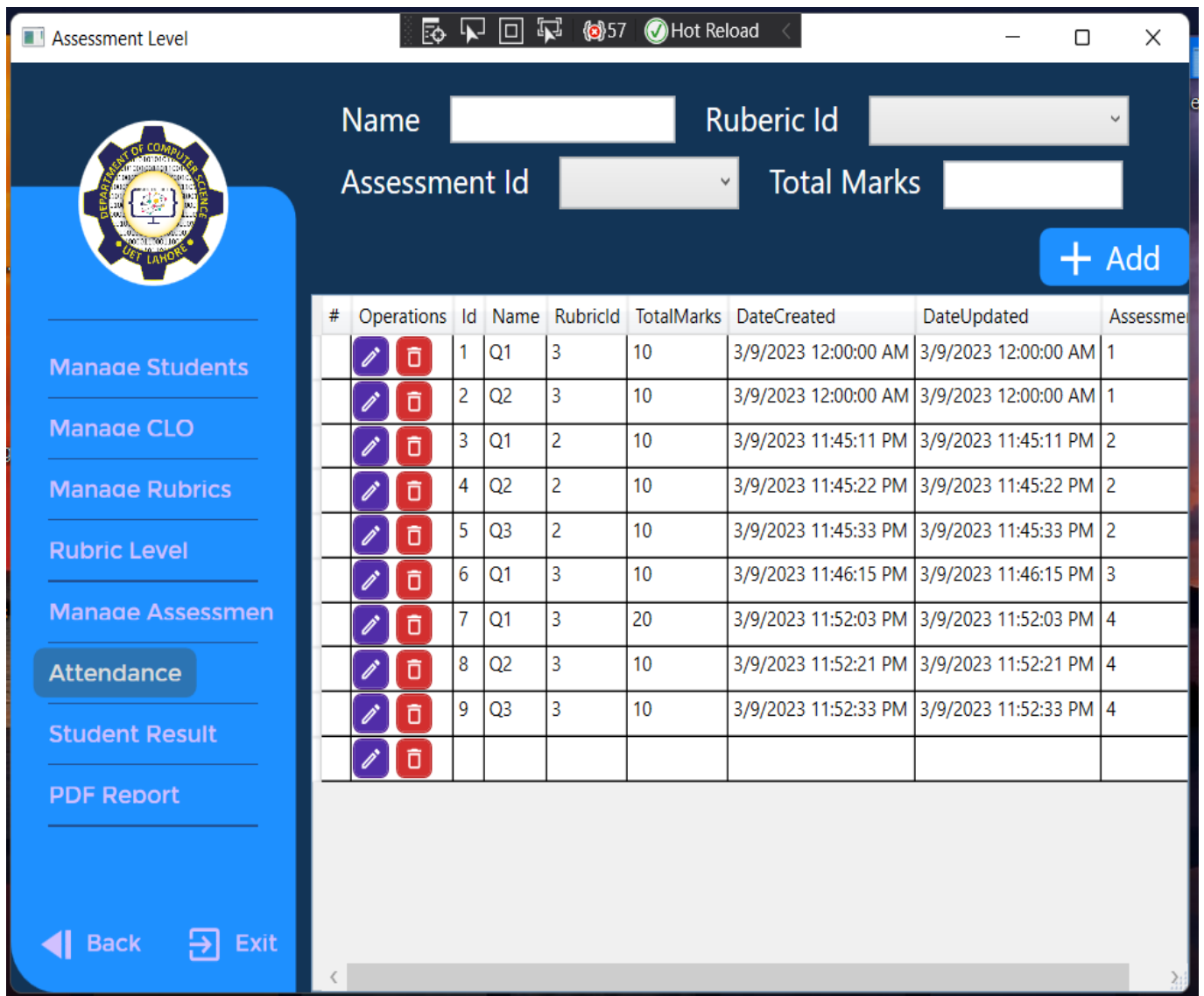


Figure 5.9.1: Assessment Components UI

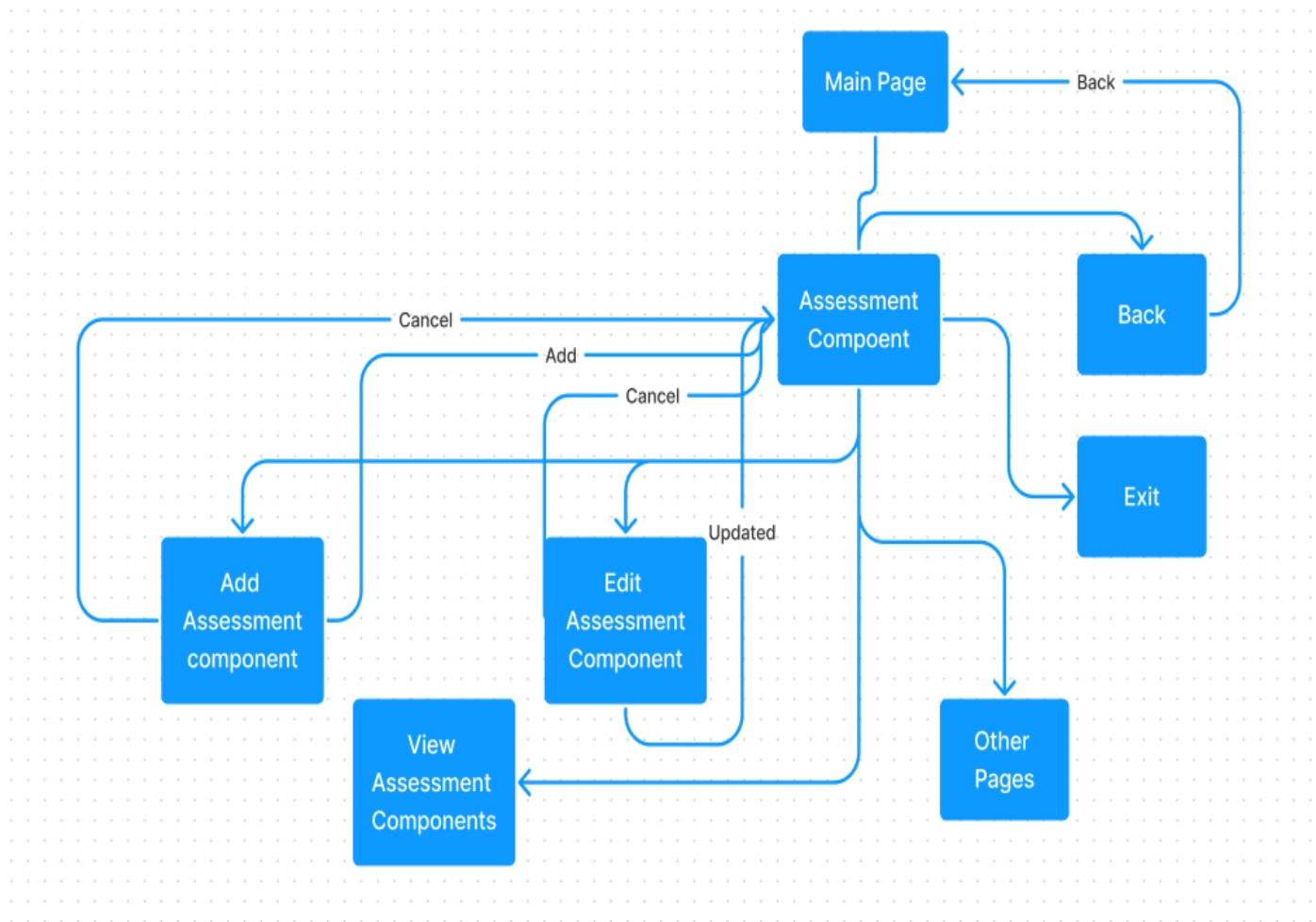


Figure 5.9.2: Assessment Components Flow Diagram

5.10 Student Result UI

This page stores information of student result based on different assessments.

The screenshot displays the 'Student Result' web application. The top header bar contains the title 'Student Result' and a 'Hot Reload' button. Below the header, the main content area is divided into a sidebar and a main panel. The sidebar, on the left, features a blue background with a circular logo at the top and a list of navigation links: 'Manage Students', 'Manage CLO', 'Manage Rubrics', 'Rubric Level', 'Manage Assessmen', 'Assessments Comp', 'Attendance', and 'PDF Report'. At the bottom of the sidebar are 'Back' and 'Exit' buttons. The main panel has a dark blue header with input fields for 'Student Id' (1), 'Assessment Id' (2), 'Student Name' (Bilal), and 'Assessment Name' (Mid Term). Below these fields are 'Load' and '+ Add' buttons. The main panel also contains a table with the following data:

Measurement Level	AssessmentComponentId	RubricId	RubricDetails	CloName	AssessmentMarks	Ass
	3	2	Execution	Clo2	30	10
	4	2	Execution	Clo2	30	10
	5	2	Execution	Clo2	30	10

Figure 5.10.1: Student Result UI

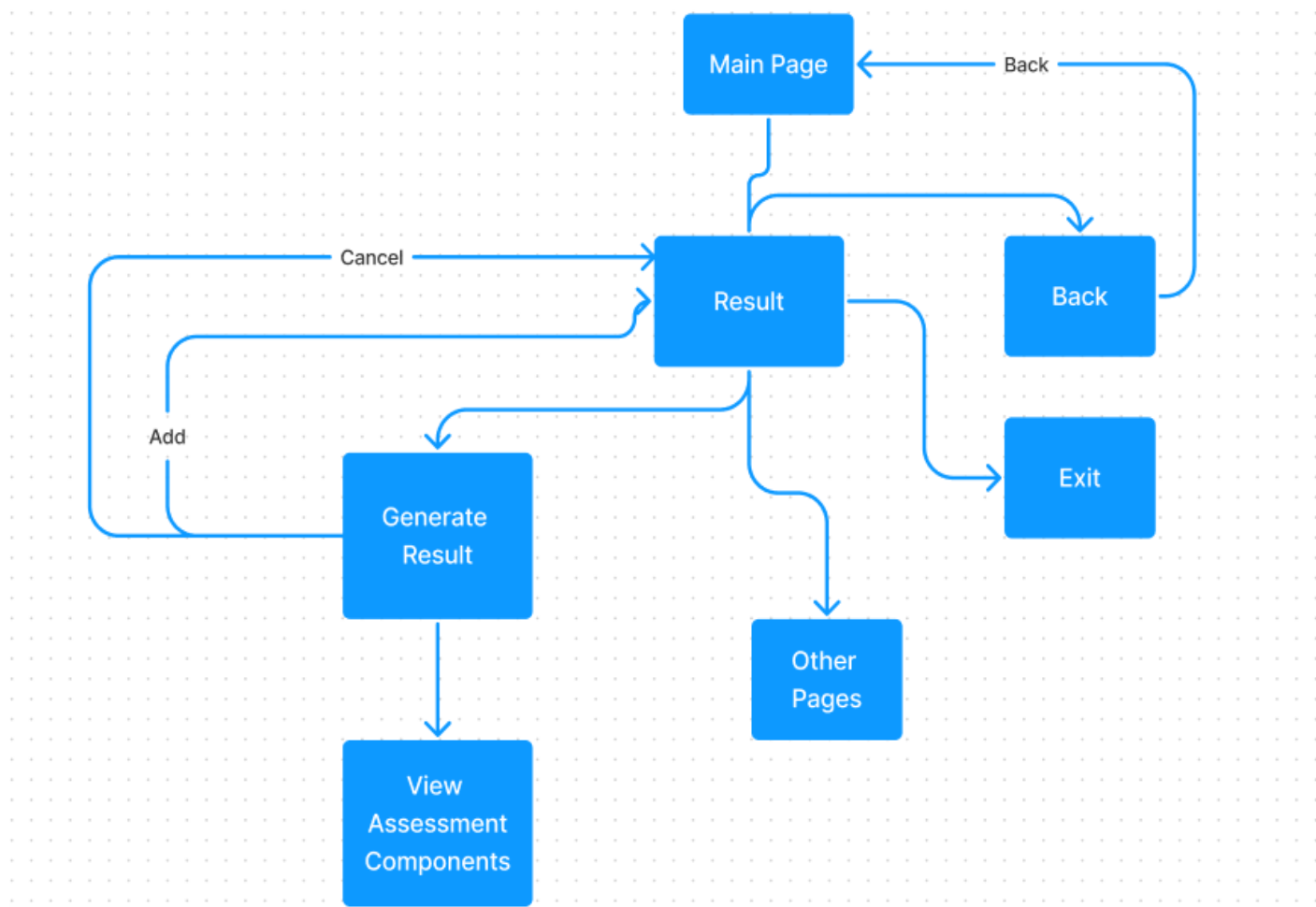


Figure 5.10.2: Student Result Flow Diagram

5.11 Class Attendance UI

This page generates pdf reports based on different .

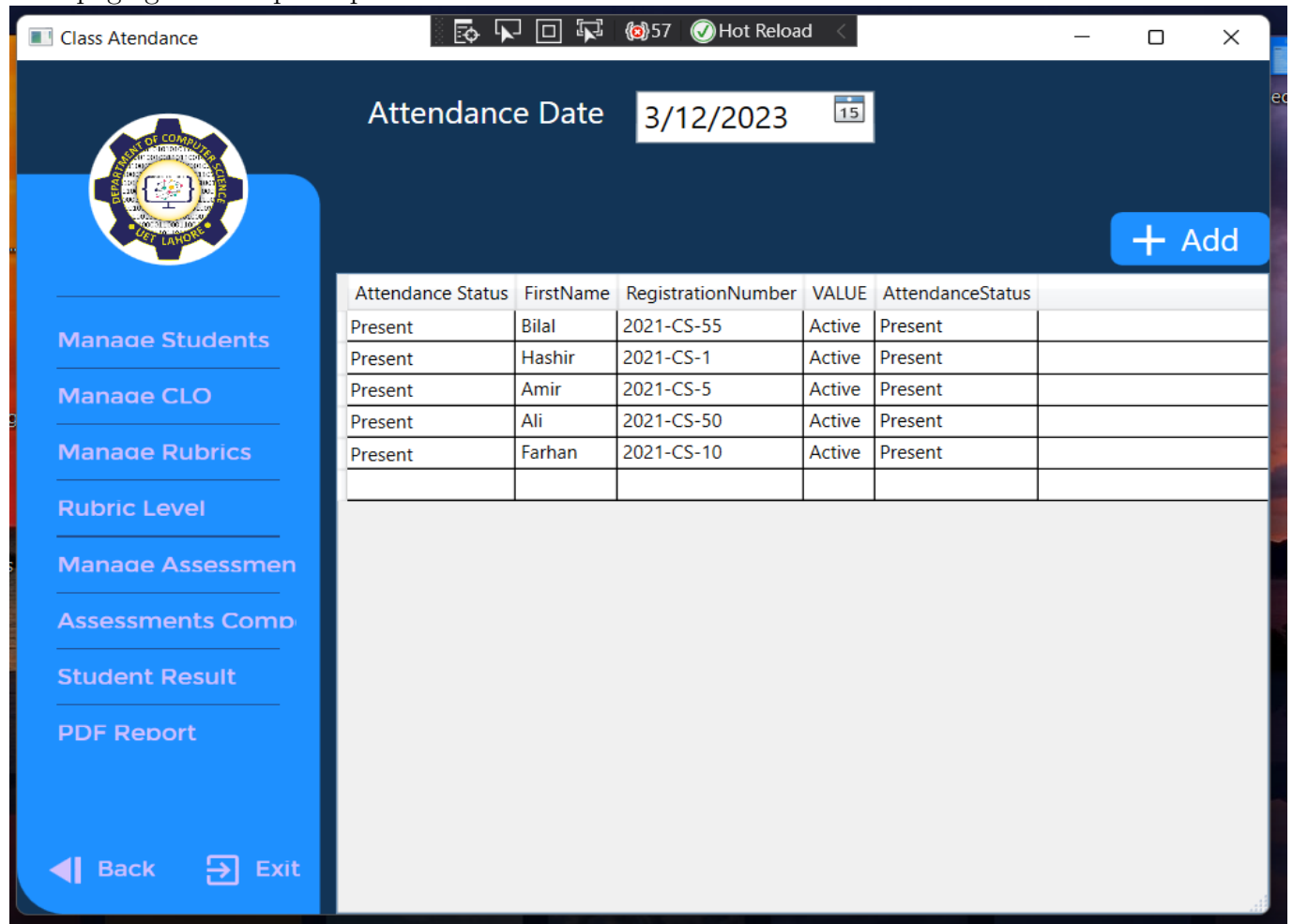


Figure 5.11.1: Class Attendance UI

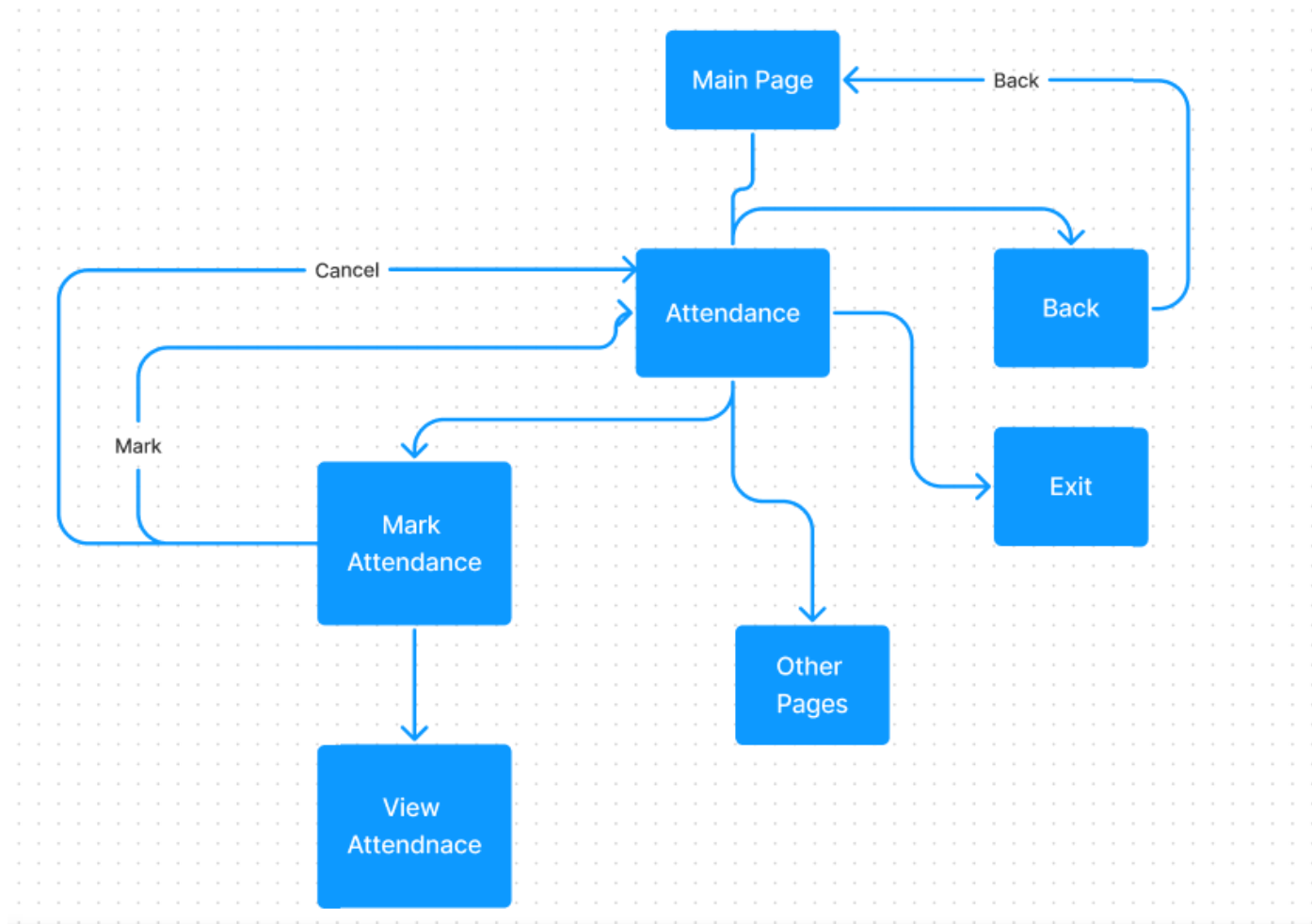
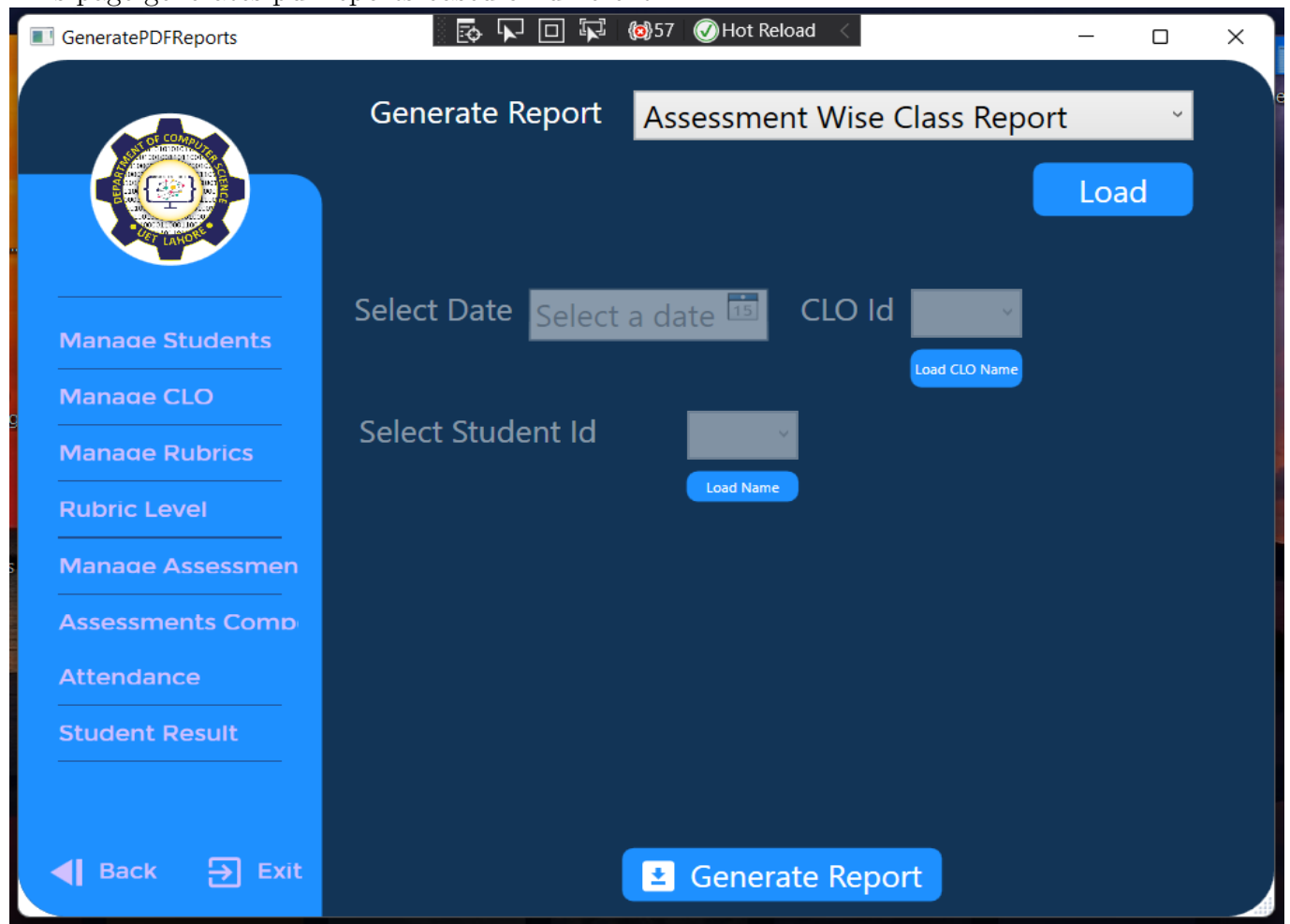


Figure 5.11.2: Class Attendance Flow Diagram

5.12 Generate PDF Reports UI

This page generates pdf reports based on different .



The screenshot displays the 'GeneratePDFReports' application window. The title bar includes standard window controls and a 'Hot Reload' button. The main interface features a dark blue background with a light blue sidebar on the left. The sidebar contains a circular logo for the 'DEPARTMENT OF COMPUTER SCIENCE UET LAHORE' and a list of menu items: 'Manage Students', 'Manage CLO', 'Manage Rubrics', 'Rubric Level', 'Manage Assessment', 'Assessments Comparison', 'Attendance', and 'Student Result'. The main content area is titled 'Generate Report' and shows a dropdown menu set to 'Assessment Wise Class Report' with a 'Load' button. Below this, there are form fields for 'Select Date' (with a calendar icon), 'CLO Id' (with a dropdown arrow), and 'Select Student Id' (with a dropdown arrow). Each of these fields has a corresponding 'Load' button ('Load CLO Name' and 'Load Name'). At the bottom of the main area is a large blue button labeled 'Generate Report'. The bottom of the sidebar contains 'Back' and 'Exit' buttons.

Figure 5.13.1: Generate Reports UI

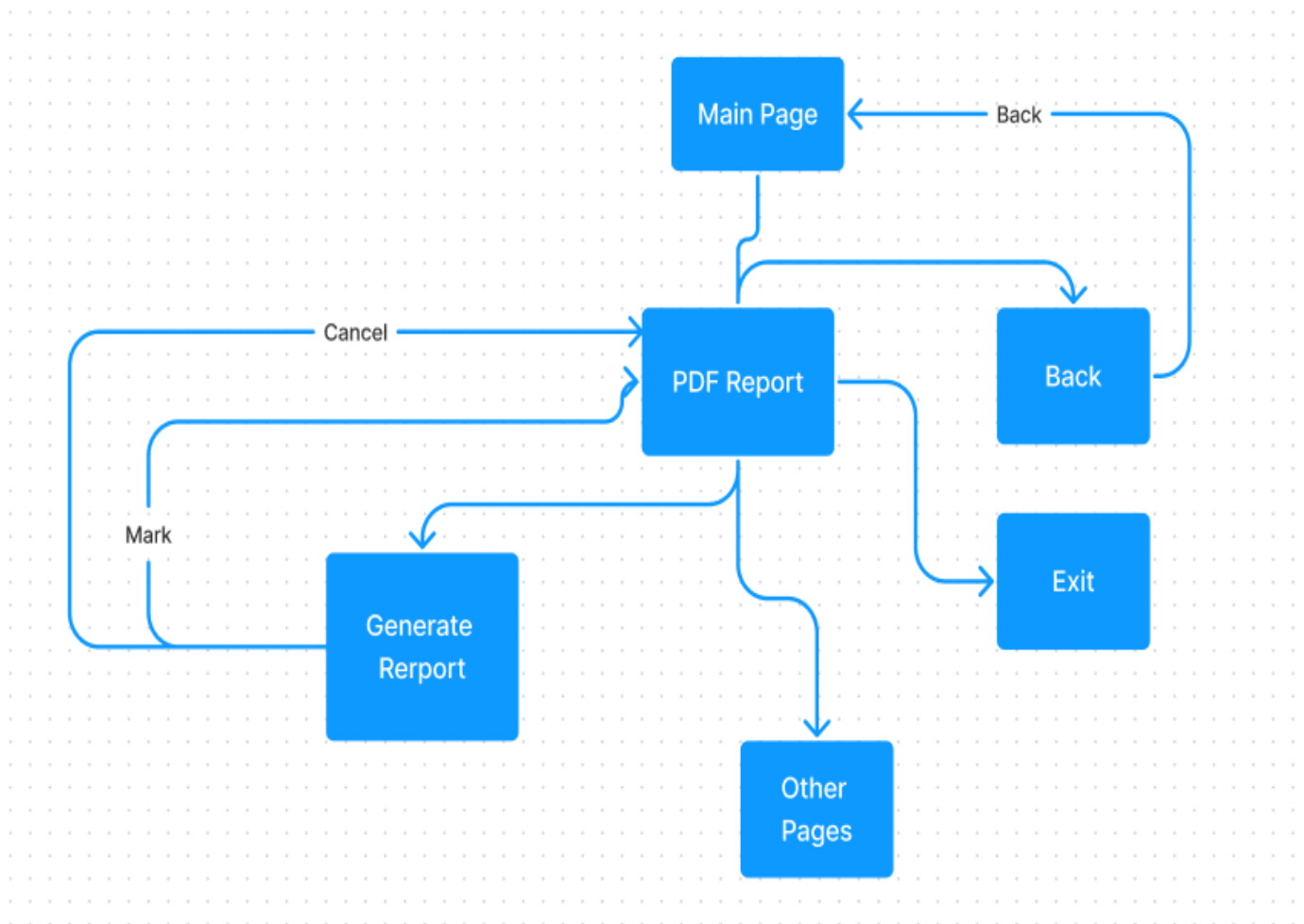


Figure 5.13.1: Generate Reports Flow Diagram

6 Generate Report Queries

6.1 Assessment Wise Class Report (Detailed)

```

SELECT S.FirstName AS StudentName, cl.Name as CloName,
R.Details as RubricDetails, Ass.Title as AssessmentTitle,
AC.Name as AssessmentComponent, AC.TotalMarks,
RL.MeasurementLevel as ObtainedMeasurementLevel,
((CAST(MeasurementLevel as float) / (SELECT
MAX(MeasurementLevel) FROM RubricLevel)) * AC.TotalMarks)
as ObtainedMarks FROM Student S
JOIN StudentResult SR ON S.Id = SR.StudentId
JOIN AssessmentComponent AC ON AC.Id = SR.AssessmentComponentId
JOIN Assessment Ass ON Ass.Id = AC.AssessmentId
JOIN RubricLevel RL ON RL.Id = SR.RubricMeasurementId
JOIN Rubric R ON R.Id = RL.RubricId
JOIN Clo cl ON cl.Id = R.CloId
GROUP BY S.FirstName, cl.Name, R.Details, Ass.Title, AC.Name,
AC.TotalMarks, RL.MeasurementLevel

```

6.2 Assessment Wise Class Report (Summerized)

```
SELECT S.FirstName AS StudentName, cl.Name as CloName,
MAX(R.Details) as RubricDetails, Ass.Title as
AssessmentTitle, SUM(AC.TotalMarks) as TotalMarks,
SUM(CAST(RL.MeasurementLevel AS float) /
maxMeasurementLevel * AC.TotalMarks) AS ObtainedMarks
FROM Student S
JOIN StudentResult SR ON S.Id = SR.StudentId
JOIN AssessmentComponent AC ON AC.Id =
SR.AssessmentComponentId
JOIN Assessment Ass ON Ass.Id = AC.AssessmentId
JOIN RubricLevel RL ON RL.Id = SR.RubricMeasurementId
JOIN Rubric R ON R.Id = RL.RubricId
JOIN Clo cl ON cl.Id = R.CloId
CROSS JOIN (SELECT MAX(MeasurementLevel) AS
maxMeasurementLevel FROM RubricLevel) maxRl
GROUP BY S.FirstName, cl.Name, Ass.Title
```

6.3 CLO Wise Class Report (Detailed)

```
SELECT cl.Name as CloDetails, S.FirstName AS StudentName,
R.Details as RubricDetails, Ass.Title as AssessmentTitle,
AC.TotalMarks, ((CAST(MeasurementLevel as float) / (SELECT
MAX(MeasurementLevel) FROM RubricLevel)) * AC.TotalMarks)
as ObtainedMarks FROM Student S
JOIN StudentResult SR ON S.Id = SR.StudentId
JOIN AssessmentComponent AC ON AC.Id = SR.AssessmentComponentId
JOIN Assessment Ass ON Ass.Id = AC.AssessmentId
JOIN RubricLevel RL ON RL.Id = SR.RubricMeasurementId
JOIN Rubric R ON R.Id = RL.RubricId
JOIN Clo cl ON cl.Id = R.CloId
GROUP BY S.FirstName, cl.Name, R.Details, Ass.Title, AC.Name,
AC.TotalMarks, RL.MeasurementLevel, Cl.Id
HAVING cl.Id = (selected Clo Id)
```

6.4 CLO Wise Class Report (Summerized)

```
SELECT
cl.Name AS CloDetails, S.FirstName AS StudentName,
MAX(R.Details) AS RubricDetails, Ass.Title AS
AssessmentTitle, SUM(AC.TotalMarks) AS TotalMarks,
SUM(CAST(RL.MeasurementLevel AS float)
maxMeasurementLevel * AC.TotalMarks) AS ObtainedMarks
FROM Student S
JOIN StudentResult SR ON S.Id = SR.StudentId
JOIN AssessmentComponent AC ON AC.Id =
SR.AssessmentComponentId
JOIN Assessment Ass ON Ass.Id = AC.AssessmentId
```

```

JOIN RubricLevel RL ON RL.Id = SR.RubricMeasurementId
JOIN Rubric R ON R.Id = RL.RubricId
JOIN Clo cl ON cl.Id = R.CloId
CROSS JOIN (SELECT MAX(MeasurementLevel) AS
maxMeasurementLevel FROM RubricLevel) maxRl
WHERE cl.Id = (SELETED CLO ID)
GROUP BY S.FirstName, cl.Name, Ass.Title

```

6.5 Attendance Wise Class Report

```

SELECT SA.StudentId as Id, S.RegistrationNumber
as RollNo, S.FirstName as StudentName, L.Name as
AttendanceStatus FROM StudentAttendance SA
JOIN ClassAttendance CA ON SA.AttendanceId = CA.Id
JOIN Student S ON S.Id = SA.StudentId
JOIN Lookup L ON L.LookupId = SA.AttendanceStatus
WHERE CA.AttendanceDate = (SELECTED DATE)

```

6.6 CLO Wise Individual Report (Detailed)

```

SELECT cl.Name as CloDetails, S.FirstName AS StudentName,
R.Details as RubricDetails, Ass.Title as AssessmentTitle,
AC.TotalMarks, ((CAST(MeasurementLevel as float) / (SELECT
MAX(MeasurementLevel) FROM RubricLevel)) * AC.TotalMarks)
as ObtainedMarks FROM Student S
JOIN StudentResult SR ON S.Id = SR.StudentId
JOIN AssessmentComponent AC ON AC.Id = SR.AssessmentComponentId
JOIN Assessment Ass ON Ass.Id = AC.AssessmentId
JOIN RubricLevel RL ON RL.Id = SR.RubricMeasurementId
JOIN Rubric R ON R.Id = RL.RubricId
JOIN Clo cl ON cl.Id = R.CloId
WHERE cl.Id = (SELECTED Id) and S.Id = (SELECTED Id)
GROUP BY S.FirstName, cl.Name, R.Details, Ass.Title,
AC.Name, AC.TotalMarks, RL.MeasurementLevel, Cl.Id

```

6.7 CLO Wise Individual Report (Summerized)

```

SELECT S.FirstName AS StudentName, clo.Name as CloName,
MAX(r.Details) as RubricDetails, Ass.Title as
AssessmentTitle, SUM(AC.TotalMarks) as TotalMarks,
SUM(CAST(RL.MeasurementLevel AS float) /
maxMeasurementLevel * AC.TotalMarks) AS ObtainedMarks
FROM Student S
JOIN StudentResult SR ON S.Id = SR.StudentId
JOIN AssessmentComponent AC
ON AC.Id = SR.AssessmentComponentId
JOIN Assessment Ass ON Ass.Id = AC.AssessmentId
JOIN RubricLevel RL ON RL.Id = SR.RubricMeasurementId

```

```

JOIN Rubric r ON r.Id = RL.RubricId
JOIN Clo clo ON clo.Id = r.CloId
CROSS JOIN (SELECT MAX(MeasurementLevel) AS
maxMeasurementLevel FROM RubricLevel) maxRl
WHERE S.Id = 1 and clo.Id GROUP BY S.FirstName,
clo.Name , Ass.Title

```

6.8 Assessment Wise Indiv dual Report (Detailed)

```

SELECT S.
FirstName AS StudentName, cl.Name as CloName,
R.Details as RubricDetails, Ass.
Title as AssessmentTitle,
AC.Name
as AssessmentComponent, AC.TotalMarks,
RL.MeasuRmentLevel
as ObtainedMeasurementLevel,
((CAST(MeasurementLevel as float) / (SELECT
MAX(MeasurementLevel)FROM RubricLevel)) * AC.TotalMarks)
as ObtainedMarks FROM Student S
JOIN StudentResult SR ON S.Id = SR.StudentId
JOIN AssessmentComponent AC ON AC.Id = SR.AssessmentComponentId
JOIN Assessment Ass ON Ass.Id = AC.AssessmentId
JOIN RubricLevel RL ON RL.Id = SR.RubricMeasurementId
JOIN Rubric R ON R.Id = RL.RubricId
JOIN Clo cl ON cl.Id = R.CloId
WHERE S.Id = (SELECTED Student Id)
GROUP BY S.FirstName, cl.Name,
R.Details, Ass.Title, AC.Name, AC.TotalMarks, RL.MeasurementLevel

```

6.9 Assessment Wise Individual Report (Summerized)

```

SELECT S.FirstName AS StudentName, cl.Name as CloName,
MAX(R.Details) as RubricDetails, Ass.Title as
AssessmentTitle, SUM(AC.TotalMarks) as TotalMarks,
SUM(CAST(RL.MeasurementLevel AS float) /
maxMeasurementLevel * AC.TotalMarks) AS ObtainedMarks
FROM Student S
JOIN StudentResult SR ON S.Id = SR.StudentId
JOIN AssessmentComponent AC
ON AC.Id = SR.AssessmentComponentId
JOIN Assessment Ass ON Ass.Id = AC.AssessmentId
JOIN RubricLevel RL ON RL.Id = SR.RubricMeasurementId
JOIN Rubric R ON R.Id = RL.RubricId
JOIN Clo cl ON cl.Id = R.CloId
CROSS JOIN (SELECT MAX(MeasurementLevel) AS
maxMeasurementLevel FROM RubricLevel) maxRl
WHERE S.Id = (SELECTED Student Id)
GROUP BY S.FirstName, cl.Name, Ass.Title

```


7 Testing

After conducting a thorough test of the project to verify the correctness of data linkage, including the testing of integrated SQL Server queries, I discovered that inactive students cannot be updated due to a constraint in the database provided to us. Also, if you want to delete an entity which gives its primary key to other entity set as foreign key. Then you have to delete that object first to delete the required onject.

8 Limitations

The project has certain limitations that need to be addressed.

1. Firstly, it is designed for a single user, which puts a lot of pressure on the admin to keep track of everything.
2. There is no login form, making the data vulnerable to unauthorized access.
3. It would be beneficial to consider consolidating multiple forms into a single form to simplify navigation for users
4. Instead of using IDs, it would be more user-friendly to use names and titles for data entry
5. there is an issue with data duplication when a student is already in the system.

9 Future Work

The project has potential for future enhancements, which include: Adding a Sign-in/Sign-up form for improved security Enabling multiple admins to access the application for ease of management Improving the data validation by preventing duplication of student registration numbers Enhancing the user interface through user control implementations.

10 Collaboration

Under the guidance of our supervisors, Mr. Nauman Babar and Mr. Samyan Qayyum Wahla, we have successfully completed the project. They have played a critical role in helping us understand the case study, as well as explaining the database schema. To ensure our comprehension, our supervisors utilized a database diagram, which provided us with a comprehensive understanding of the schema. The main objective was to familiar with the schema, enabling us to comprehend the associations between different tables and to understand the database system.

11 Conclusion

The midterm project on the database was an extensive resource for learning new concepts. There are so much small details that we often overlook while using a system. This project helped us understand the hard work required to create a system and how databases can be used to solve problems in the real world. It highlighted how the theoretical concepts covered in class can be applied effectively to address real-life issues.

The midterm project on the database was a helpful resource for learning new ideas. We usually don't think about many small details that we often overlook while using a system. This

project helped us understand how databases are used to solve real-life problems and how to write queries to generate and print data. It showed us that what we learn in class can be used to handle real-world issues effectively.