Rubrics Based Evaluation System

Session: 2020 – 2024

Submitted by:

Muhammad Nouman Butt 2020-CS-110

Supervised by:

Lecturer Nazeef-ul-Haq

Department of Computer Science and Engineering
University of Engineering and Technology
Lahore Pakistan

Declaration

I declare that the Project entitled “Rubrics Based Evaluation System” submitted by me towards to evaluation of the course “Database And Systems Lab” is a project carried by me under the supervision of Mr. Nazeef-ul-Haq, and have not been submitted somewhere else. We will solely be responsible if any kind of plagiarism is found.

Signed: Muhammad Nouman Butt
Date: 4-March-2022

Acknowledgments

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. A special gratitude I give to our report supervisor, Mr Nazeef Ul Haq, whose contribution in stimulating suggestions and encouragement, helped me to coordinate my project especially in writing this report.I have to appreciate the guidance given by other supervisor as well as the panels especially in our project presentation that has improved our presentation skills thanks to their comment and advices.

Contents

List of Figures

List of Tables

Abstract

In this project a Desktop app namely ‘Rubric Based Evaluation System’ has been developed. There are many evaluation systems which is managed by teachers and this project is also a reflection of it.

Teacher will manage student by adding their information and then will evaluate their assessments based on rubrics and rubrics level. He will also mark their attendance in this system and he will also be able to view different kinds of report to keep track on their performance.All the data will automatically be managed behind the scenes in database.

Chapter 1

Project Details

1.1 Project Description:

Rubrics Based Evaluation System is a desktop software application for the evaluation of student's assessments according to the rubrics defined against each CLO's (Course Learning Outcomes). This program is single user based application used by teacher of the course who can do multiple functionalities. He can add multiple students into his course. He can perform CRUD (Create, Read, Update and Delete) operations on Course Learning Outcomes and its Rubrics and then Rubric levels. He can also mark attendance of the students on the present day and can edit attendance of previous days as well. He can also make different assignments and can mark these assignments based on rubric levels. He can also view different reports generated which are described below. In CLO wise report of students he can view the the total number of students who passed that particular CLO or not. He can also view the rubrics that are present in each CLO. He can also see the percentage of that CLO marks. In Assessment Wise report, he can view the obtained marks of each students in each assessment along with the percentage. He can also just view result as assessment wise that in one assessment how many student has passed and what was the average score of the student. In Attendance report, he can view attendance of each student that how many times he was present until now and he can also view attendance report day wise how how many student was absent and present on particular day.

1.2 Business Need:

This Program is made to facilitate the professors or teacher of universities providing all the facilities in one platform. The Main purpose of the program is to provide a user-friendly interface and to lower the burdens of the staff to keep the manual records.

1.3 Stakeholder:

The end user of this software will be simple the professor of the university teaching a specific course. He will use it to manage all the data of the students, CLOs, Rubrics and assignments.

1.4 Project Actors:

There is only 1 users of this Program:

1.4.1 Teacher:

He is basically the main user of this program as he is the one who actually manages all data of the students and other modules such as assessments and attendance,etc.

1.5 Project Features:

Rubrics Based Evaluation System is a user friendly software and provides the following features to end user:

1.5.1 Features:

- He can perform CRUD Operations on Students.
- He can Manage CLOs, Rubrics and Rubric Levels.
- He can also mark the attendance of students.
- He can make multiple assessments using rubrics and then can mark assessments based upon rubric levels of rubrics.
- He can also view reports of multiple kinds like CLO wise report, Assessment wise report, Attendance report etc.

Chapter 2

Project Planning

Project planning involves the stages in which a project should be executed. Different people use different methodologies to build their software. I have used the Waterfall Process Model to implement the program as described below:

2.1 Waterfall Process Model:

The Waterfall process model is a linear-sequential life cycle model. In a waterfall model, one stage must complete first before moving to the next stage. It is the most basic approach used for software development because it is easy to understand and use. Waterfall process model is divided into separate phases. Following figure is the representation of all the stages of process:

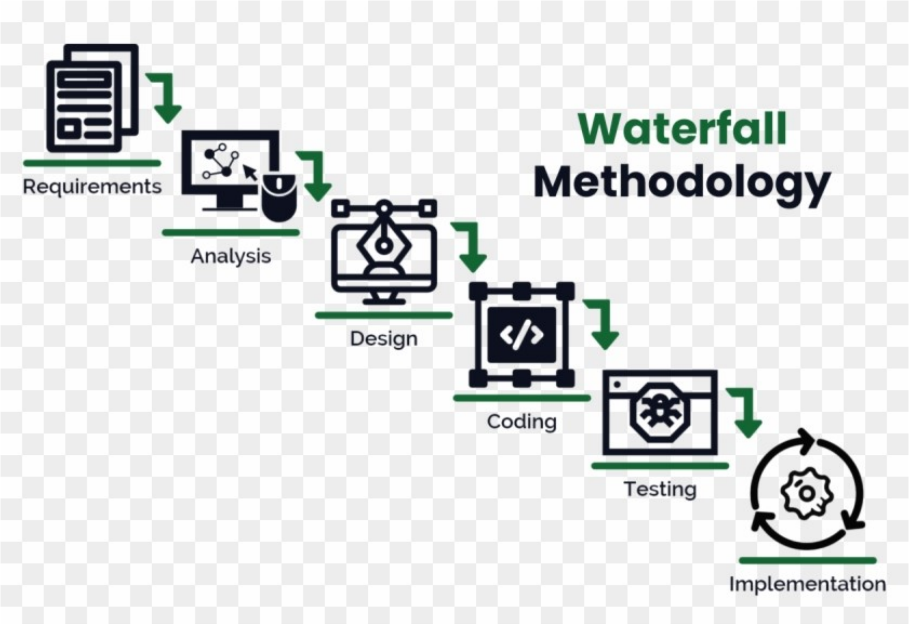


Figure 2.1: Figure of Waterfall Methodology Stages

2.1.1 Requirement:

All the requirement of the software program which goes to be evolved is accumulated on this phase from the client that how the software should look and what are the functionality should be like. A basic dummy GUI will also be made in this phase and this kind of project will look alike almost.

2.1.2 Analysis:

In this stage, a thorough look-out is done on the project like how the program should be made. All the Object-oriented features such as how many classes are going to be made and what will be their properties and features will be decided in this stage or if it is going to be made on Database then this step involves the analysis of the Data Model that is going to be made and store different things.

2.1.3 System Design:

When the analysis stage is done then we move to the design stage. In this stage, the main design of the program is implemented. The wire frames which were provided in the requirement stage are improved and enhanced by placing the different modules in suitable places.

2.1.4 Coding:

According to the inputs from the graphical user interface, the program is divided into small units such as modules. These modules are coded in this phase and then tested for their functionality.

2.1.5 Integration:

After all the units are made and tested in their respective phases then these units are integrated together to make the proper software that was required.

2.1.6 Testing:

When the implementation stage ends, then the developers will start testing on the software in real-time in the real world to find out all the bugs present in the program and then resolve them one by one. Thousands of tests are made on a single program to make it perfect.

2.2 Technology Stack:

Technology stack means the language and the platform used to make the program and the IDE in which program will be implemented.

Table 2.1: Used to show the language and technology to implement the program.

Language	C-Sharp
Plateform	Desktop
Frontend Technology	Win-Frames
IDEs	Visual Studio
Database	Microsoft SQL Database
Database Langauge	T-SQL

Chapter 3

Project Completion Details

3.1 Major Milestone:

This table shows the expected duration of time which is decided to complete a specific phase of waterfall model.

Table 3.1: Used to show the expected and completion date of each milestones

Milestones	Expected Completion Date	Completion Date
Requirement	17-February-2021	18-February-2021
Analysis	18-February-2021	22-February-2021
System Design	22-February-2021	23-February-2021
Coding	28-February-2021	03-March-2021
Integration	2-February-2021	04-March-2021
Testing	4-February-2021	05-March-2021

3.2 File Management:

The whole project is made in such a way that every coder will find it easy to work with. Every function is well commented as well all the files have been saved properly is their respective folders as shown below:

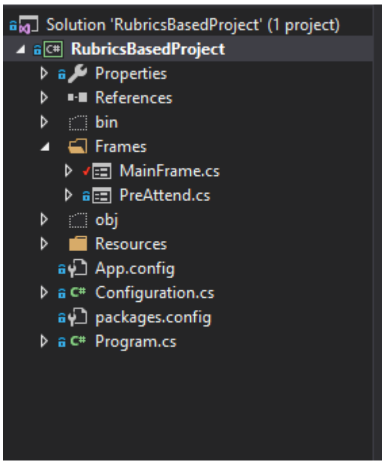


FIGURE 3.1: Source Folder of the program

3.2.1 Frame Folder:

This folder contains all the frames that have been used in this program.

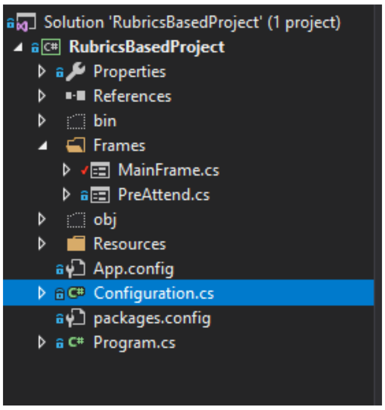
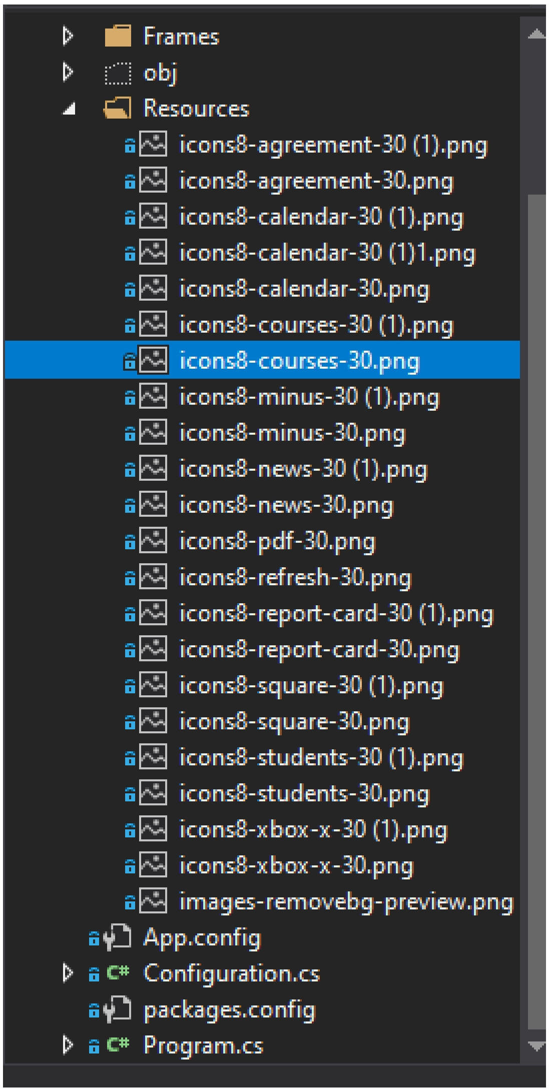


FIGURE 3.2: Folder of Frames

3.2.2 Resources folder:

This folder contain all the pictures or other file formats that have been imported into the program and have been used to display something.

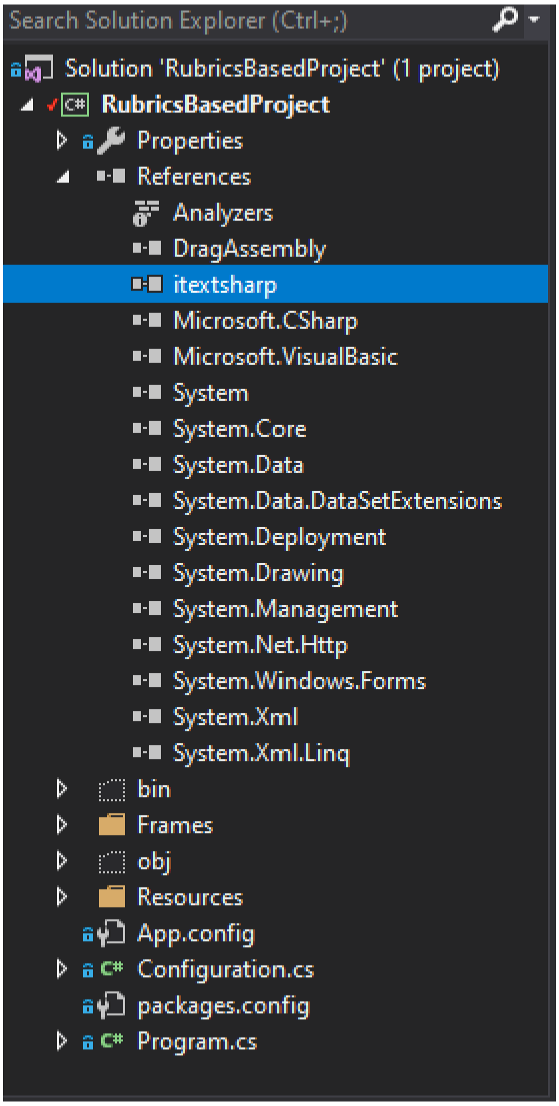


Frames
obj
Resources
icons8-agreement-30 (1).png
icons8-agreement-30.png
icons8-calendar-30 (1).png
icons8-calendar-30 (1)1.png
icons8-calendar-30.png
icons8-courses-30 (1).png
icons8-courses-30.png
icons8-minus-30 (1).png
icons8-minus-30.png
icons8-news-30 (1).png
icons8-news-30.png
icons8-pdf-30.png
icons8-refresh-30.png
icons8-report-card-30 (1).png
icons8-report-card-30.png
icons8-square-30 (1).png
icons8-square-30.png
icons8-students-30 (1).png
icons8-students-30.png
icons8-xbox-x-30 (1).png
icons8-xbox-x-30.png
images-removebg-preview.png
App.config
Configuration.cs
packages.config
Program.cs

FIGURE 3.3: Folder of Resources

3.2.3 References Folder:

This folder contain all the packages that have been used in the program to make the program quite smooth and a better UX experience.



```
Search Solution Explorer (Ctrl+;)
Solution 'RubricsBasedProject' (1 project)
  ◢ RubricsBasedProject
      ▷ Properties
      ◢ References
            Analyzers
            DragAssembly
            itextsharp
            Microsoft.CSharp
            Microsoft.VisualBasic
            System
            System.Core
            System.Data
            System.Data.DataSetExtensions
            System.Deployment
            System.Drawing
            System.Management
            System.Net.Http
            System.Windows.Forms
            System.Xml
            System.Xml.Linq
      ▷ bin
      ▷ Frames
      ▷ obj
      ▷ Resources
        App.config
      ▷ Configuration.cs
        packages.config
      ▷ Program.cs
```

FIGURE 3.4: Data structures implemented into the program

3.3 Road Map:

Here is a road map for using the program from scratch. Follow the following steps to use the program efficiently and smoothly:

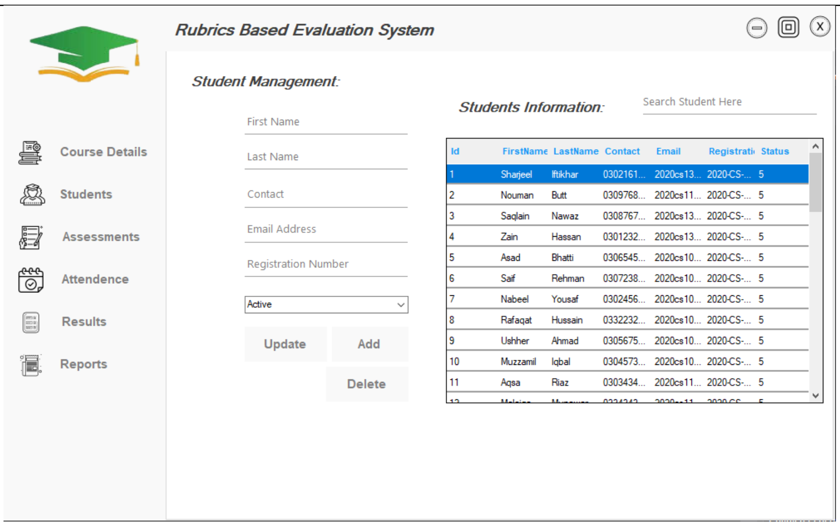
1. Start the program by clicking on the .exe file.
2. Now click on Student Section button and then add students into the system. You can also edit,view and delete students from here.
3. After adding the student go to the CLO portion by clicking on the CLO button and add CLO and perfrom CRUD on it.
4. Now Add Rubrics and rubric levels and perform CRUD operations.
5. Now Add Assessment and perform CRUD on it.
6. After that add assessment component.
7. Now Evaluate the students against each assessment component based on rubric and rubric levels.
8. Teacher can also mark attendance of the students on daily basis.
9. Teacher can also view different reports such as CLO report, Attendance report, Assessment report and Student report.

Chapter 4

Use Cases

4.1 Use Case 1(Student CRUD):

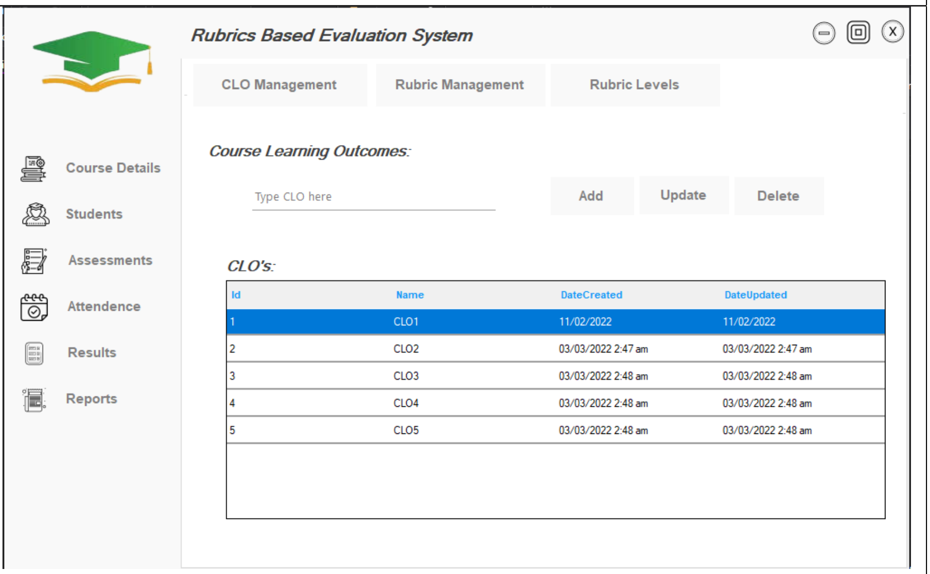
Table 4.1: Used to describe use-case 1 of Student CRUD

Use Case ID	U01
Name	Student CRUD
Actor	Teacher
Description	Teacher will be able to add students and will be able to perform different operations such as search, view and delete.
Implemented GUI	 Figure 4.1: GUI frame of Student CRUD

Id	FirstName	LastName	Contact	Email	Registrati	Status
1	Sharjeel	Iftikhar	0302161...	2020cs13...	2020-CS-...	5
2	Nouman	Butt	0309768...	2020cs11...	2020-CS-...	5
3	Saqlain	Nawaz	0308767...	2020cs13...	2020-CS-...	5
4	Zain	Hassan	0301232...	2020cs13...	2020-CS-...	5
5	Asad	Bhatti	0306545...	2020cs10...	2020-CS-...	5
6	Saif	Rehman	0307238...	2020cs10...	2020-CS-...	5
7	Nabeel	Yousaf	0302456...	2020cs10...	2020-CS-...	5
8	Rafaqat	Hussain	0332232...	2020cs10...	2020-CS-...	5
9	Ushher	Ahmad	0305675...	2020cs10...	2020-CS-...	5
10	Muzzamil	Iqbal	0304573...	2020cs10...	2020-CS-...	5
11	Aqsa	Riaz	0303434...	2020cs11...	2020-CS-...	5

4.2 Use Case 2(CLO):

Table 4.2: Used to describe use-case 2 of CLO

Use Case ID	U02
Name	CLO
Actor	Teacher
Description	Teacher can add CLOs into his course and can edit, view and delete according to its need.
Implemented GUI	 Figure 4.2: GUI frame of CLO

Rubrics Based Evaluation System

Course Learning Outcomes:

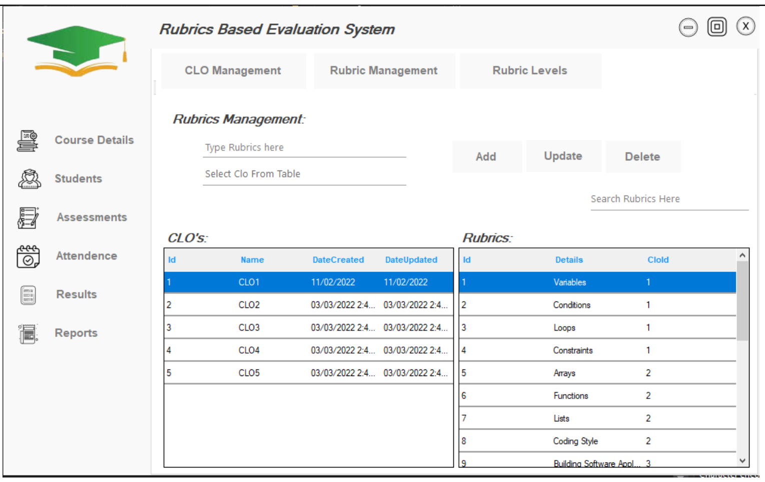
Type CLO here

CLO's:

Id	Name	DateCreated	DateUpdated
1	CLO1	11/02/2022	11/02/2022
2	CLO2	03/03/2022 2:47 am	03/03/2022 2:47 am
3	CLO3	03/03/2022 2:48 am	03/03/2022 2:48 am
4	CLO4	03/03/2022 2:48 am	03/03/2022 2:48 am
5	CLO5	03/03/2022 2:48 am	03/03/2022 2:48 am

4.3 Use Case 3(Rubrics):

Table 4.3: Used to describe use-case 3 of Rubrics

Use Case ID	U03
Name	Rubrics
Actor	Teacher
Description	Teacher will b adding the Rubrics of each Clo in this case. He will also perform CRUD Operations on rubrics as per need.
Implemented GUI	 Figure 4.3: GUI frame of Rubrics

Rubrics Management:

Type Rubrics here

Select Clo From Table

Search Rubrics Here

CLO's:

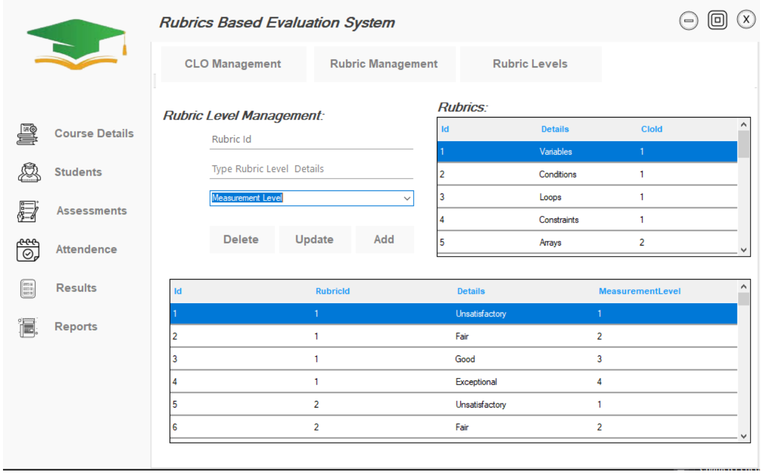
Id	Name	DateCreated	DateUpdated
1	CLO1	11/02/2022	11/02/2022
2	CLO2	03/03/2022 2:4...	03/03/2022 2:4...
3	CLO3	03/03/2022 2:4...	03/03/2022 2:4...
4	CLO4	03/03/2022 2:4...	03/03/2022 2:4...
5	CLO5	03/03/2022 2:4...	03/03/2022 2:4...

Rubrics:

Id	Details	CloId
1	Variables	1
2	Conditions	1
3	Loops	1
4	Constraints	1
5	Arrays	2
6	Functions	2
7	Lists	2
8	Coding Style	2
9	Building Software Appl...	3

4.4 Use Case 4(Rubrics Level):

Table 4.4: Used to describe use-case 4 of Rubrics Level

Use Case ID	U04
Name	Rubrics Level
Actor	Teacher
Description	Teacher will be adding Rubrics Level of each Rubrics and will be performing CRUD operations on rubric Level.
Implemented GUI	 Figure 4.4: GUI frame of Rubrics Level

Rubric Level Management:

Rubric Id

Type Rubric Level Details

Measurement Level

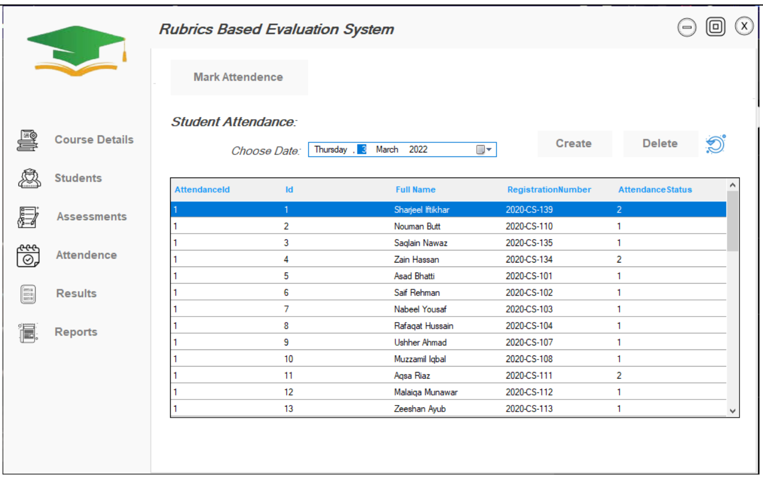
Rubrics:

Id	Details	CloId
1	Variables	1
2	Conditions	1
3	Loops	1
4	Constraints	1
5	Arrays	2

Id	RubricId	Details	MeasurementLevel
1	1	Unsatisfactory	1
2	1	Fair	2
3	1	Good	3
4	1	Exceptional	4
5	2	Unsatisfactory	1
6	2	Fair	2

4.5 Use Case 5(Attendance):

TABLE 4.5: Used to describe use-case 5 of Attendance

Use Case ID	U05
Name	Attendance
Actor	Teacher
Description	Teacher will be able or edit to mark attendance of the students
Implemented GUI	 FIGURE 4.5: GUI frame of Attendance.

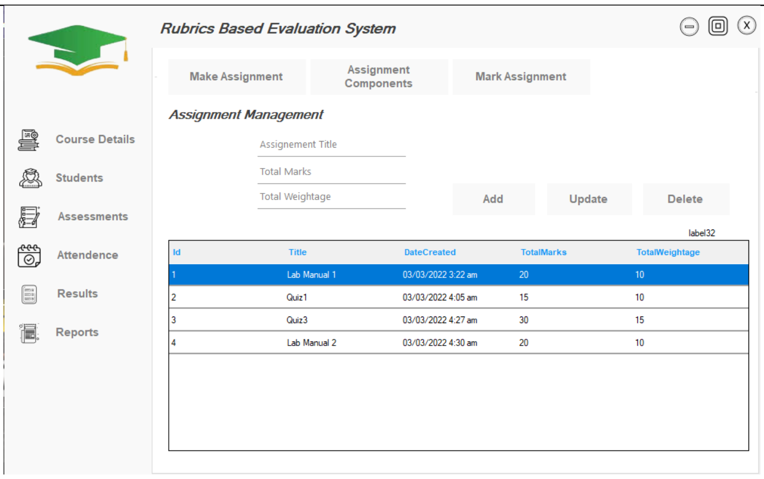
Student Attendance:

Choose Date: Thursday , 3 March 2022

AttendanceId	Id	Full Name	RegistrationNumber	AttendanceStatus
1	1	Sharjeel Iftikhar	2020-CS-139	2
1	2	Nouman Butt	2020-CS-110	1
1	3	Saqlain Nawaz	2020-CS-135	1
1	4	Zain Hassan	2020-CS-134	2
1	5	Asad Bhatti	2020-CS-101	1
1	6	Saif Rehman	2020-CS-102	1
1	7	Nabeel Yousaf	2020-CS-103	1
1	8	Rafaqat Hussain	2020-CS-104	1
1	9	Ushher Ahmad	2020-CS-107	1
1	10	Muzzamil Iqbal	2020-CS-108	1
1	11	Aqsa Riaz	2020-CS-111	2
1	12	Malaiqa Munawar	2020-CS-112	1
1	13	Zeeshan Ayub	2020-CS-113	1

4.6 Use Case 6(Make Assessment):

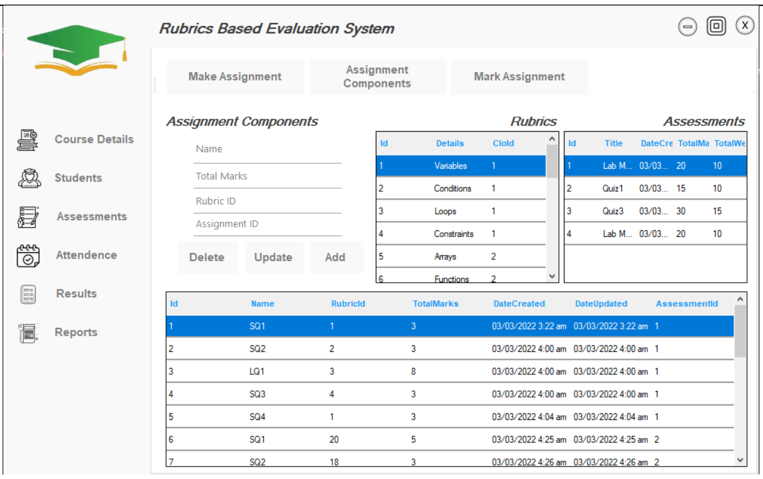
Table 4.6: Used to describe use-case 6 of Make Assessment

Use Case ID	U06
Name	Make Assessment
Actor	Teacher
Description	He will be able to make and perform CRUD operations on assessments.
Implemented GUI	 Figure 4.6: GUI frame of Make Assessment

Id	Title	DateCreated	TotalMarks	TotalWeightage
1	Lab Manual 1	03/03/2022 3:22 am	20	10
2	Quiz1	03/03/2022 4:05 am	15	10
3	Quiz3	03/03/2022 4:27 am	30	15
4	Lab Manual 2	03/03/2022 4:30 am	20	10

4.7 Use Case 7(CRUD Assessment Components):

Table 4.7: Used to describe use-case 7 of CRUD Assessment Components

Use Case ID	U07
Name	CRUD Assessment Components
Actor	Teacher
Description	Teacher will be able to add assessment components of each assessment and can perform CRUD on it.
Implemented GUI	 Figure 4.7: GUI frame of CRUD Assessment Components

Rubrics

Id	Details	CloId
1	Variables	1
2	Conditions	1
3	Loops	1
4	Constraints	1
5	Arrays	2
6	Functions	2

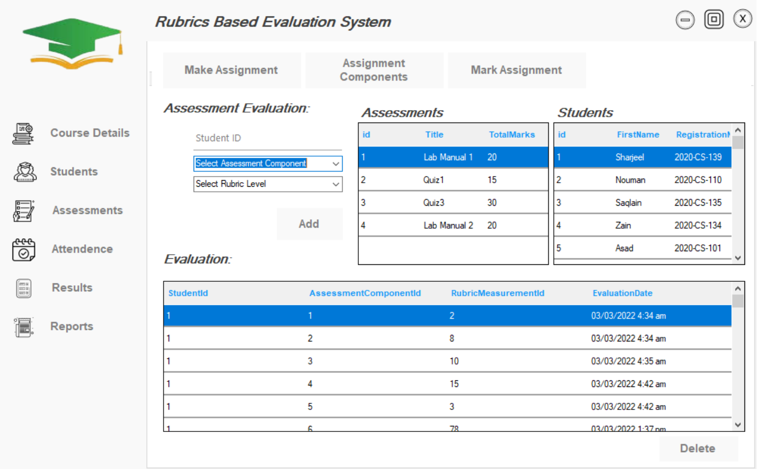
Assessments

Id	Title	DateCre	TotalMa	TotalWe
1	Lab M...	03/03...	20	10
2	Quiz1	03/03...	15	10
3	Quiz3	03/03...	30	15
4	Lab M...	03/03...	20	10

Id	Name	RubricId	TotalMarks	DateCreated	DateUpdated	AssessmentId
1	SQ1	1	3	03/03/2022 3:22 am	03/03/2022 3:22 am	1
2	SQ2	2	3	03/03/2022 4:00 am	03/03/2022 4:00 am	1
3	LQ1	3	8	03/03/2022 4:00 am	03/03/2022 4:00 am	1
4	SQ3	4	3	03/03/2022 4:00 am	03/03/2022 4:00 am	1
5	SQ4	1	3	03/03/2022 4:04 am	03/03/2022 4:04 am	1
6	SQ1	20	5	03/03/2022 4:25 am	03/03/2022 4:25 am	2
7	SQ2	18	3	03/03/2022 4:26 am	03/03/2022 4:26 am	2

4.8 Use Case 8(Mark Assessments):

TABLE 4.8: Used to describe use-case 8 of Mark Assessments

Use Case ID	U08
Name	Mark Assessments
Actor	Teacher
Description	Teacher will be evaluating all the students against each assessment component based on rubrics and rubric level and can delete it as well.
Implemented GUI	 FIGURE 4.8: GUI frame of Mark Assessments

Rubrics Based Evaluation System

Assessment Evaluation:

Student ID

Select Assessment Component

Select Rubric Level

Assessments

id	Title	TotalMarks
1	Lab Manual 1	20
2	Quiz1	15
3	Quiz3	30
4	Lab Manual 2	20

Students

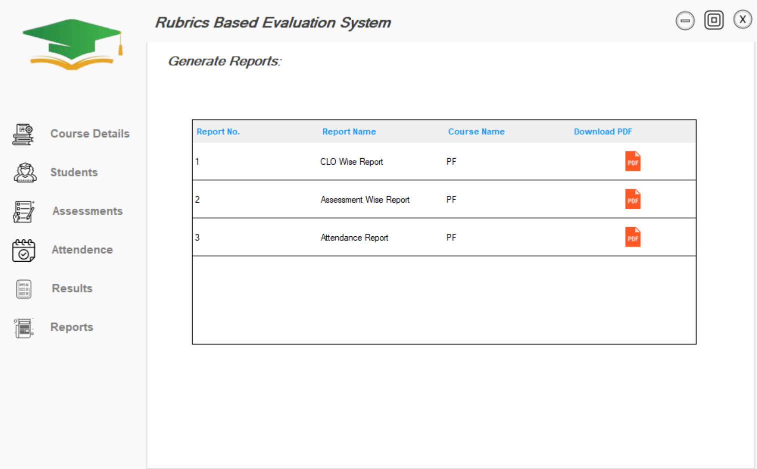
id	FirstName	Registration
1	Sharjeel	2020-CS-139
2	Nouman	2020-CS-110
3	Saqlain	2020-CS-135
4	Zain	2020-CS-134
5	Asad	2020-CS-101

Evaluation:

StudentId	AssessmentComponentId	RubricMeasurementId	EvaluationDate
1	1	2	03/03/2022 4:34 am
1	2	8	03/03/2022 4:34 am
1	3	10	03/03/2022 4:35 am
1	4	15	03/03/2022 4:42 am
1	5	3	03/03/2022 4:42 am
1	6	78	03/03/2022 1:37 pm

4.9 Use Case 9(Reports):

Table 4.9: Used to describe use-case 9 of Reports

<table>
<tr><td>Use Case ID</td><td>U09</td></tr>
<tr><td>Name</td><td>Reports</td></tr>
<tr><td>Actor</td><td>Teacher</td></tr>
<tr><td>Description</td><td>Teacher will be able to see different reports that are shown in the figure below..</td></tr>
<tr><td>Proposed Wire Frame</td><td>


Generate Reports:

<table>
<tr><th>Report No.</th><th>Report Name</th><th>Course Name</th><th>Download PDF</th></tr>
<tr><td>1</td><td>CLO Wise Report</td><td>PF</td><td>PDF</td></tr>
<tr><td>2</td><td>Assessment Wise Report</td><td>PF</td><td>PDF</td></tr>
<tr><td>3</td><td>Attendance Report</td><td>PF</td><td>PDF</td></tr>
</table>

Figure 4.9: Wire frame analysis report of a particular person
</td></tr>
<tr><td>Implemented GUI</td><td>We are not able to implement this use case in the program. So, there is no UI implementation for this use case.</td></tr>
</table>

Chapter 5

User Interface Detail

In this section, fill the table for summary that which use case will have the required component. Inside each box, write the counts for each component. If component is not used, write zero.

Table 5.1: Used to show the details of components through which GUI is made.

Use Case ID	Textbox	Dropdown	Label	DataGrid	Data Fields	Buttons	Radio Button	Text Fields
U01	6	1	3	1	0	3	0	0
U02	1	0	2	1	0	3	0	0
U03	0	0	6	0	0	1	12	0
U04	0	0	12	0	0	6	0	0
U05	9	1	3	1	0	7	0	0
U06	0	0	3	1	0	7	0	0
U07	2	3	4	5	6	7	8	9
U08	0	0	0	12	7	3	4	5
U09	0	0	0	4	3	6	6	1

Chapter 6

Database Design

6.1 Tables:

This section will show the way tables are gonna be made adn how data is gonna be stored in it.

Table 6.1: Used to show the details of all the classes used in the program

Table Name	**Total Number of Attributes**	**Identity**	**No. of Primary Keys**	**No. of Foreign Keys**
Assessment	5	Yes	1	0
Assessment Components	7	Yes	1	2
StudentResult	4	No	2	3
Rubric Level	4	Yes	1	1
Rubric	3	No	1	1
CLO	4	Yes	1	0
Student	7	Yes	1	0
Lookup	3	Yes	1	0
StudentAttendance	3	No	2	2
ClassAttendance	2	Yes	1	0

6.2 Database Design:

Below Image shows the database design of rubric based evaluation system.

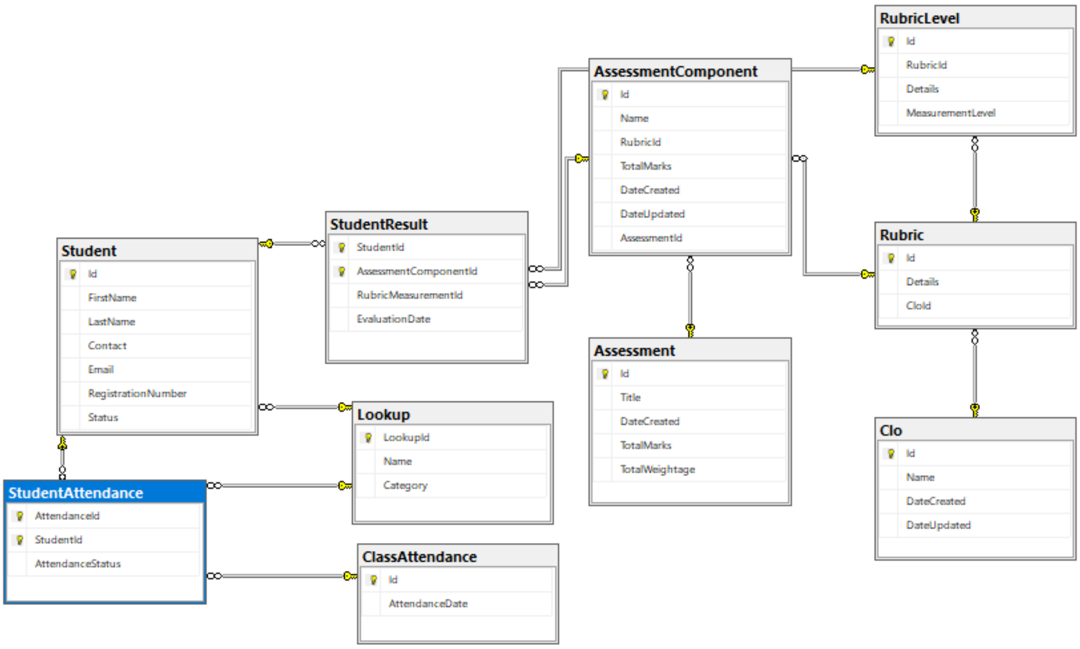


Figure 6.1: Database design of project

Chapter 7

Reports

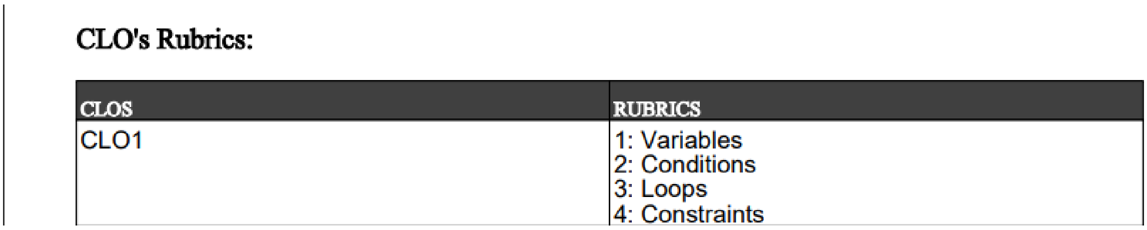
In this project 4 reports are being generated based on the requirements showing the total attendance and the obtained marks of each student in CLO or assessments.

7.1 CLO Wise Reports:

CLO wise report basically shows the performance of students in each CLO weather it is in same assessment or not but it should be in same CLO and if it has passed the CLO or not. There are basically 2 tables in CLO report.

7.1.1 CLO Rubrics:

This table contain the following columns:



CLO's Rubrics:

CLOS	RUBRICS
CLO1	1: Variables 2: Conditions 3: Loops 4: Constraints

Figure 7.1: CLO wise report preview of first table

It Shows that in each CLO there are multiple number of rubrics that are added and these rubrics will also contain 4 rubric levels each. such as in this example CLO1 contain 4 rubrics.

7.1.1.1 Col1 SQL Query:

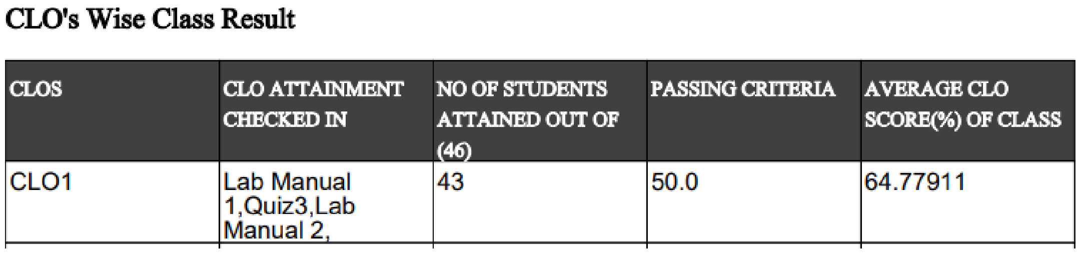
```
SELECT Name
from Clo
where id = @id
```

7.1.1.2 Col2 SQL Query:

```
Select R.Details
From Rubric R
Where R.CloId = @CloId
```

7.1.2 CLO Result:

This table contain the following columns:



CLO's Wise Class Result

CLOS	CLO ATTAINMENT CHECKED IN	NO OF STUDENTS ATTAINED OUT OF (46)	PASSING CRITERIA	AVERAGE CLO SCORE(%) OF CLASS
CLO1	Lab Manual 1,Quiz3,Lab Manual 2,	43	50.0	64.77911

Figure 7.2: CLO wise report preview of second table

It Shows that in each CLO e.g CLO1 that it has been used in the following assessments and in 3rd columns it shows that how many students have passed passed the CLO and then in 4th column it shows the passing percentage of the student and in 5th column it shows the percentage of the total marks obtained in this CLO from the total marks of assessments components.

7.1.2.1 Col1 SQL Query:

```
SELECT Name
from Clo
where id = @id
```

7.1.2.2 Col2 SQL Query:

```
Select Title
From Assessment
where Id in
(select AssessmentId from AssessmentComponent where RubricId in
(select Id from Rubric where CloId=@CloId))
```

7.1.2.3 Col5 SQL Query:

From the below query the obtained marks of all students in the CLO is get and then perform the same query with a little change to get full marks then divide both of them to get average marks of CLO.

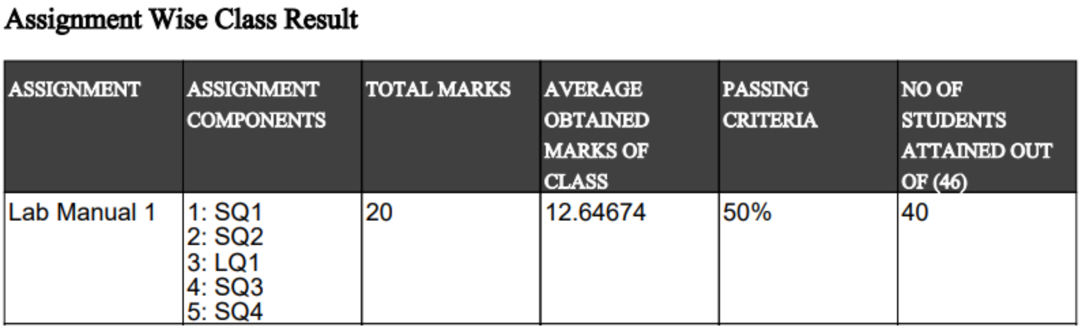
```
Select sum(N.Obtained)
From(Select(cast(T.[Student obtained rub leve] as float) / 4) * T.TotalMarks
as [Obtained]
From(Select S.StudentId, A.Name, A.RubricId, A.TotalMarks, (Select
MeasurementLevel From RubricLevel R where R.Id = S.RubricMeasurementId)
as [Student obtained rub leve]
From StudentResult S Join AssessmentComponent A on S.AssessmentComponentId
= A.Id
where A.RubricId IN(Select R.Id From Rubric R Where R.CloId =
@CloId)) as T) AS N
```

7.2 Assessment Report:

Assessment wise report basically shows the performance of students in each assessment weather it is in same Same CLO or not.
There are basically 2 tables in Assessment report.

7.2.1 Assessment Result:

This table contain the following columns:



Assignment Wise Class Result

ASSIGNMENT	ASSIGNMENT COMPONENTS	TOTAL MARKS	AVERAGE OBTAINED MARKS OF CLASS	PASSING CRITERIA	NO OF STUDENTS ATTAINED OUT OF (46)
Lab Manual 1	1: SQ1 2: SQ2 3: LQ1 4: SQ3 5: SQ4	20	12.64674	50%	40

FIGURE 7.3: Assessment wise report preview of first table

First column in table shoes the Assessment name while the 2nd column contain the name of assessment components in each assessment, 3rd column contain the total marks of each assessment, 4th column shows the average marks obtained in that that CLO, 5th column shows the passing percentage while 6th column shows the average obtained percentage of CLO.

7.2.1.1 Col1 SQL Query:

```
Select Title
from Assessment
where Id=@id
```

7.2.1.2 Col2 SQL Query:

```
Select AC.Name
From AssessmentComponent AC
Where AC.AssessmentId = @Id
```

7.2.1.3 Col3 SQL Query:

```
Select sum(AC.TotalMarks)
from AssessmentComponent AC
Where AssessmentId = @Id
```

7.2.1.4 Col4 SQL Query:

```
Select sum(M.ObMarks) / count(M.ObMarks)
From(Select B.StudentId, sum(B.Obtained) as obMarks
From(Select T.StudentId, (cast(T.[Student obtained rub leve] as float)
/ 4) * T.TotalMarks as[Obtained]
From(Select S.StudentId, AC.Name, AC.RubricId, AC.TotalMarks, (Select
MeasurementLevel From RubricLevel R where R.Id = S.RubricMeasurementId)
as [Student obtained rub leve]
from AssessmentComponent AC Join StudentResult S on AC.Id =
S.AssessmentComponentId Where AssessmentId = @Id) AS T) as B
Group By B.StudentId)as M
```

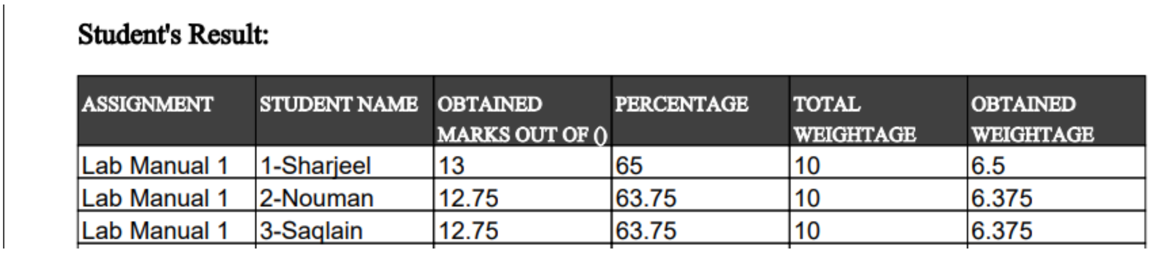
7.2.1.5 Col6 SQL Query:

From the below query the obtained marks of all students in the Assessment is get and then perform the same query with a little change to get full marks then divide both of them to get average marks of Assessment.

```
Select sum(B.Obtained)
From(Select T.StudentId, (cast(T.[Student obtained rub leve] as float)
/ 4) * T.TotalMarks as[Obtained]
From(Select S.StudentId, AC.Name, AC.RubricId, AC.TotalMarks, (Select
MeasurementLevel From RubricLevel R where R.Id = S.RubricMeasurementId)
as [Student obtained rub leve]
from AssessmentComponent AC Join StudentResult S on AC.Id =
S.AssessmentComponentId Where AssessmentId = @Id) AS T)as B
Group By B.StudentId
```

7.2.2 Student wise report:

This table contain the following columns:



Student's Result:

ASSIGNMENT	STUDENT NAME	OBTAINED MARKS OUT OF ()	PERCENTAGE	TOTAL WEIGHTAGE	OBTAINED WEIGHTAGE
Lab Manual 1	1-Sharjeel	13	65	10	6.5
Lab Manual 1	2-Nouman	12.75	63.75	10	6.375
Lab Manual 1	3-Saqlain	12.75	63.75	10	6.375

FIGURE 7.4: Assessment Student wise report preview of second table

First column in table shoes the Assessment name while the 2nd column contain the name of Student and ID , 3rd column contain the total marks obtained of each assessment, 4th column shows the percenatage, 5th column shows the total weightage while 6th column shows the obtained weightage.

7.2.2.1 Col1 SQL Query:

```
Select Title
from Assessment
where Id=@id
```

7.2.2.2 Col2 SQL Query:

```
Select FirstName
from Student
where Id=@id
```

7.2.2.3 Col3 SQL Query:

```
Select sum(B.Obtained)
From(Select T.StudentId, (cast(T.[Student obtained rub leve] as float) / 4) *
T.TotalMarks as[Obtained]
From(Select S.StudentId, AC.Name, AC.RubricId, AC.TotalMarks, (Select MeasurementLevel From RubricLevel R where R.Id = S.RubricMeasurementId) as
[Student obtained rub leve]
```

from AssessmentComponent AC Join StudentResult S on AC.Id = S.AssessmentComponentId
Where AssessmentId = @ass$_idandS.StudentId = @stu_id)AST)ASB$

7.2.2.4 Col4 SQL Query:

Select TotalWeightage
from Assessment
where Id=@id

7.2.2.5 Col6 SQL Query:

From the below query the obtained marks of all students in the Assessment is get and then perform the same query with a little change to get full marks then divide both of them to get average marks of Assessment.

Select (sum(B.Obtained) / sum(B.TotalMarks)) *B.TotalWeightage // From(Select T.StudentId, T.TotalWeightage, T.TotalMarks, (cast(T.[Student obtained rub leve] as float) / 4) * T.TotalMarks as[Obtained] // From(Select S.StudentId, A.TotalWeightage, AC.Name, AC.RubricId, AC.TotalMarks, // (Select MeasurementLevel From RubricLevel R where R.Id = S.RubricMeasurementId) as [Student obtained rub leve] // from AssessmentComponent AC Join StudentResult S on AC.Id = S.AssessmentComponentId Join Assessment A On A.Id = AC.AssessmentId// Where AssessmentId = @ass$_IdandS.StudentI$
@$stu_Id)AST)AsBGroupByB.TotalWeightage$

7.3 Student Report:

Student report will basically show the performance of each student in whole semester that how he has been doing until now.
There are basically 2 tables in Student report.

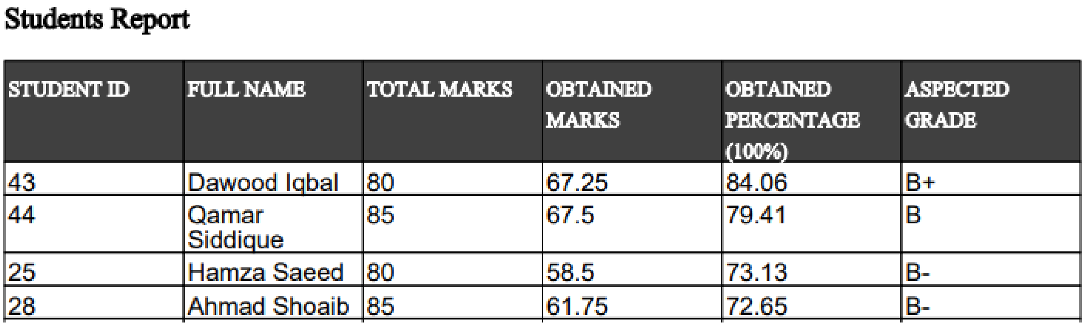
7.3.1 Student Result:

This table contain the following columns:
First column in table shoes the Student Id while the second column shows the name of the student who has achieved highest marks until now. As the table is ordered by obtained marks. 3rd column shoes the total marks of the student while

7.3.1.1 SQL Query:

There is only one query that returns multiple columns in the table that makes the table for student.

Select N.StudentId,N.Name,N.[Total MArks],N.[Total Obtained],cast((N.[Total Obtained] / N.[Total MArks]) *100 as numeric(18,2)) as [Percentage



Students Report

STUDENT ID	FULL NAME	TOTAL MARKS	OBTAINED MARKS	OBTAINED PERCENTAGE (100%)	ASPECTED GRADE
43	Dawood Iqbal	80	67.25	84.06	B+
44	Qamar Siddique	85	67.5	79.41	B
25	Hamza Saeed	80	58.5	73.13	B-
28	Ahmad Shoaib	85	61.75	72.65	B-

FIGURE 7.5: Student report preview of first table

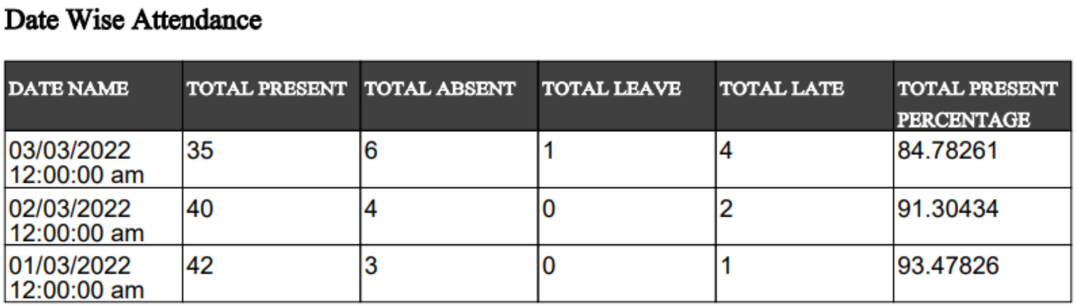
```
obtained]
From(Select B.StudentId, B.Name, sum(B.TotalMarks) as [Total MArks],
Sum(B.[Obtained Marks]) as [Total Obtained]
From(Select T.StudentId, T.Name, T.TotalMarks, (cast(T.[Student obtained rub leve] as float) / 4) * T.TotalMarks as [Obtained Marks]
From(Select S.StudentId, St.FirstName + ' ' + St.LastName as[Name],
A.RubricId, A.TotalMarks,
(Select MeasurementLevel From RubricLevel R where R.Id = S.RubricMeasurementId)
as [Student obtained rub leve] From StudentResult S Join AssessmentComponent A on S.AssessmentComponentId = A.Id join Student ST
on St.Id = S.StudentId) as T) as B
Group by B.StudentId,B.Name) As N order by[Percentage obtained]
DESC
```

7.4 Attendance Report:

Student report will basically show the performance of each student in whole semester that how many times he was present in the class and this will help the teacher to filter those students whose attendance is below average.
There are basically 2 tables in Student report.

7.4.1 Date wise attendance:

This table contain the following columns:
First column in table shows the date while second column shows the total students present on that day while second, third and forth column shows absend, leave and late students respeectively and at the end there is percentage.



Date Wise Attendance

DATE NAME	TOTAL PRESENT	TOTAL ABSENT	TOTAL LEAVE	TOTAL LATE	TOTAL PRESENT PERCENTAGE
03/03/2022 12:00:00 am	35	6	1	4	84.78261
02/03/2022 12:00:00 am	40	4	0	2	91.30434
01/03/2022 12:00:00 am	42	3	0	1	93.47826

FIGURE 7.6: Date wise Attendance preview of first table

7.4.1.1 Col1 SQL Query:

```
Select cast(AttendanceDate as date)
from ClassAttendance
where Id=@id
```

7.4.1.2 Col2 SQL Query:

```
Select distinct (Select count(*)
From StudentAttendance SA
where SA.AttendanceId =C.Id and SA.AttendanceStatus = ’1’) as
[Present Students]
from ClassAttendance C join StudentAttendance S on C.Id = S.AttendanceId
where C.Id = @ID
```

7.4.1.3 Col3 SQL Query:

```
Select distinct (Select count(*)
From StudentAttendance SA
where SA.AttendanceId =C.Id and SA.AttendanceStatus = ’2’) as
[Present Students]
from ClassAttendance C join
StudentAttendance S on C.Id = S.AttendanceId where C.Id = @ID
```

7.4.1.4 Col4 SQL Query:

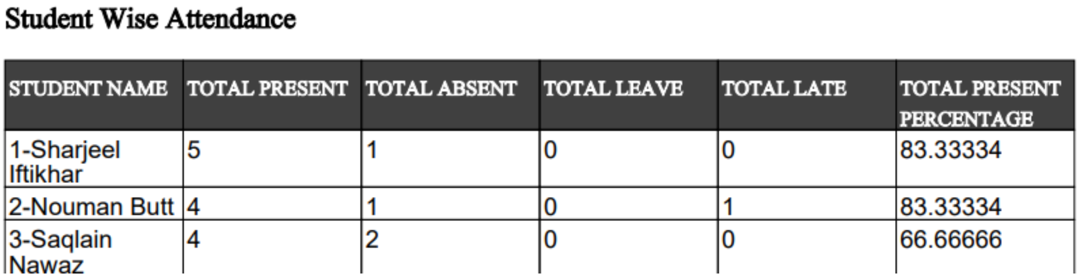
```
Select distinct (Select count(*)
From StudentAttendance SA
where SA.AttendanceId =C.Id and SA.AttendanceStatus = ’3’) as
[Present Students]
from ClassAttendance C join StudentAttendance S
on C.Id = S.AttendanceId where C.Id = @ID
```

7.4.1.5 Col5 SQL Query:

```
Select cast(A.[Present Students] + A.[Late Students] as float) / (A.[Present
Students]+A.[Absent Students]+A.[Late Students]+A.[Leave Students])
*100
From(Select Distinct C.AttendanceDate, (Select count(*)
From StudentAttendance SA where SA.AttendanceId = C.Id and SA.AttendanceStatus
= '1') as [Present Students],
(Select count(*) From StudentAttendance SA where SA.AttendanceId
= C.Id and SA.AttendanceStatus = '2') as [Absent Students],
(Select count(*) From StudentAttendance SA where SA.AttendanceId
= C.Id and SA.AttendanceStatus = '3') as [Leave Students],
(Select count(*) From StudentAttendance SA where SA.AttendanceId
= C.Id and SA.AttendanceStatus = '4') as [Late Students]
from ClassAttendance C join StudentAttendance S on C.Id = S.AttendanceId
where C.Id = @ID ) As A
```

7.4.2 Student wise attendance:

This table contain the following columns:



Student Wise Attendance

STUDENT NAME	TOTAL PRESENT	TOTAL ABSENT	TOTAL LEAVE	TOTAL LATE	TOTAL PRESENT PERCENTAGE
1-Sharjeel Iftikhar	5	1	0	0	83.33334
2-Nouman Butt	4	1	0	1	83.33334
3-Saqlain Nawaz	4	2	0	0	66.66666

Figure 7.7: Date wise Attendance preview of first table

First column in table shows the student name while second column shows the total absents and 3rd column shows the leave and late of the student and last column shows the percentage.

7.4.2.1 Col1 SQL Query:

```
Select S.FirstName + ' ' + S.LastName as [Full Name]
From Student S
where S.Id = @id
```

7.4.2.2 Col2 SQL Query:

```
Select Distinct (Select count(*)
From StudentAttendance SA
where SA.StudentId = @stuID and SA.AttendanceStatus = ’1’)
from StudentAttendance S where S.StudentId = @StuId
```

7.4.2.3 Col3 SQL Query:

```
Select Distinct
(Select count(*) From StudentAttendance SA
where SA.StudentId = @stuID and SA.AttendanceStatus = ’2’)
from StudentAttendance S where S.StudentId = @StuId
```

7.4.2.4 Col4 SQL Query:

```
Select Distinct
(Select count(*) From StudentAttendance SA
where SA.StudentId = @stuID and SA.AttendanceStatus = ’3’)
from StudentAttendance S where S.StudentId = @StuId
```

7.4.2.5 Col5 SQL Query:

```
Select (cast(B.present+B.Late as float) / (B.present+B.Late + B.Absent
+ B.Leave) ) * 100 From(Select Distinct(Select count(*)
From StudentAttendance SA where SA.StudentId = @ID and SA.AttendanceStatus
= ’1’) as present,
(Select count(*) From StudentAttendance SA where SA.StudentId =
@ID and SA.AttendanceStatus = ’2’)
as Absent,(Select count(*) From StudentAttendance SA where SA.StudentId
= @ID and SA.AttendanceStatus = ’3’)
as Leave, (Select count(*) From StudentAttendance SA where SA.StudentId
= @ID and SA.AttendanceStatus = ’4’)
as Late from StudentAttendance S where S.StudentId = @ID ) AS B
```