OBJECT ORIENTED METHODOLOGY - PROJECT

EXAMINATION RESULT SYSTEM

TEAM MEMBERS:

1.	ADVAIT SONI	- LCS2020004
2.	NITESH SINGH	- LCS2020063
3.	NITTALA RAMA KIRAN	- LCI2020006
4.	VIVEK VARDHAN KOVA	- LCI2020078

Introduction:

Examinations are an essential part of education and student lifecycle. Traditional examination and result computation system involve huge volumes of answer-paper scripts that become difficult to manage and are vulnerable to risks that include damaged and misplaced answer scripts.

The solution lies in switching over to a **digital** and **automated** student result processing system from the current manual result processing system.

About the project:

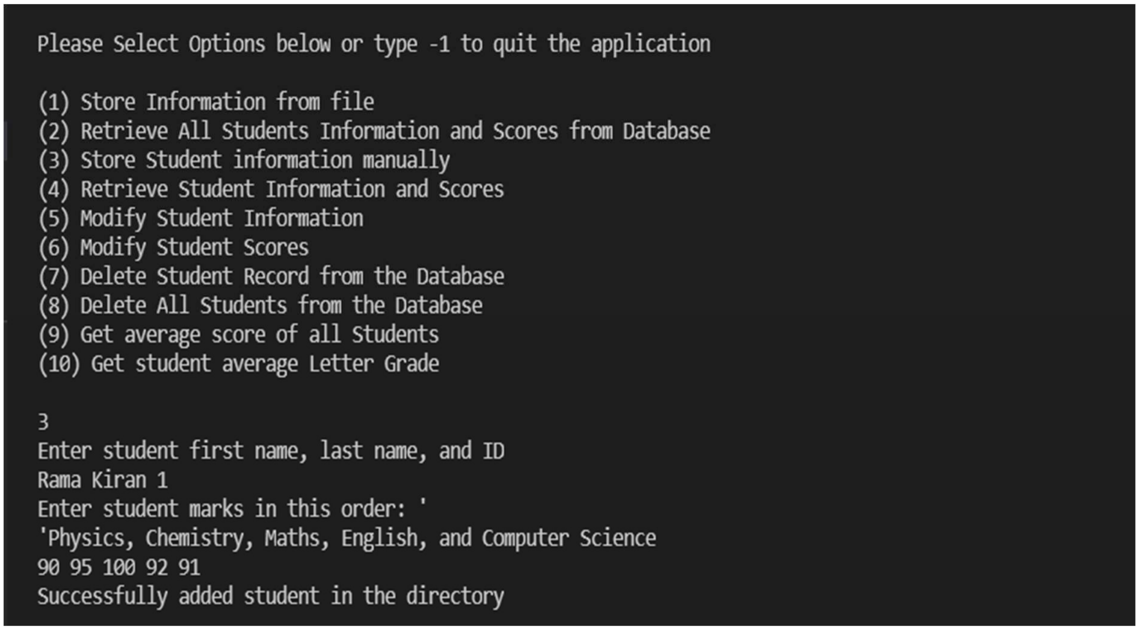
We have created a system such that a teacher or an evaluator can have access to store the student details and their marks in a database and after that they can also get access to modify the student details and marks such that if they find any mistake they can use the modify option to change the required details. There is also another option to delete the student details and marks. Here we considered 5 subject marks (Maths, English, Computer Science, Physics, Chemistry) out of 100 marks and accordingly total marks of the 5 subjects (out of 500) will be evaluated and get displayed.

The grades will also be calculated according to your scores and get awarded (like for 90 – 100% of total marks they will be awarded A++ …). Student have the access to see their results by giving their details.

Teacher can also have the access to see the average scores of total students in a particular course.

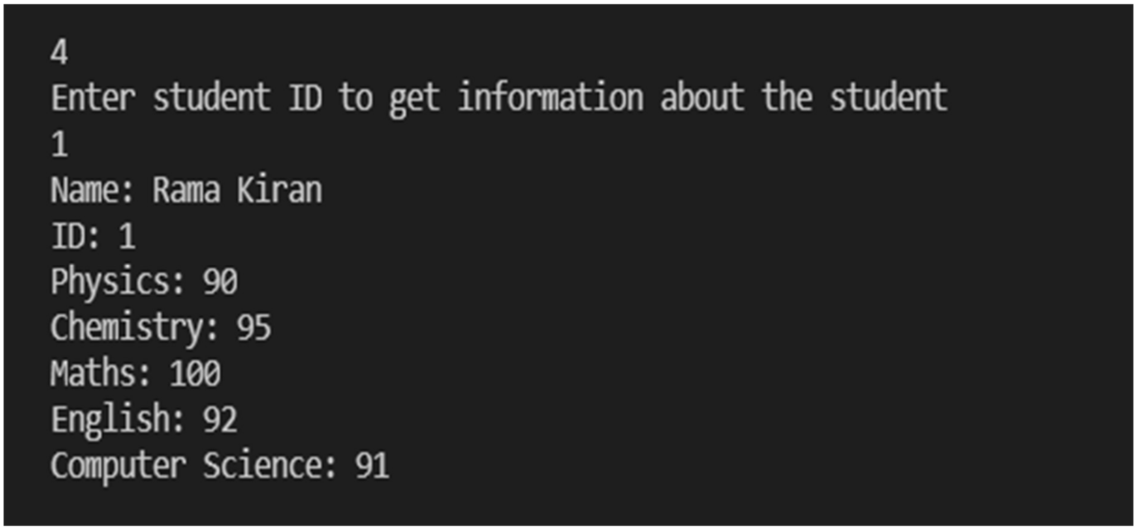
Running code on Terminal:

Storing Student details in database:



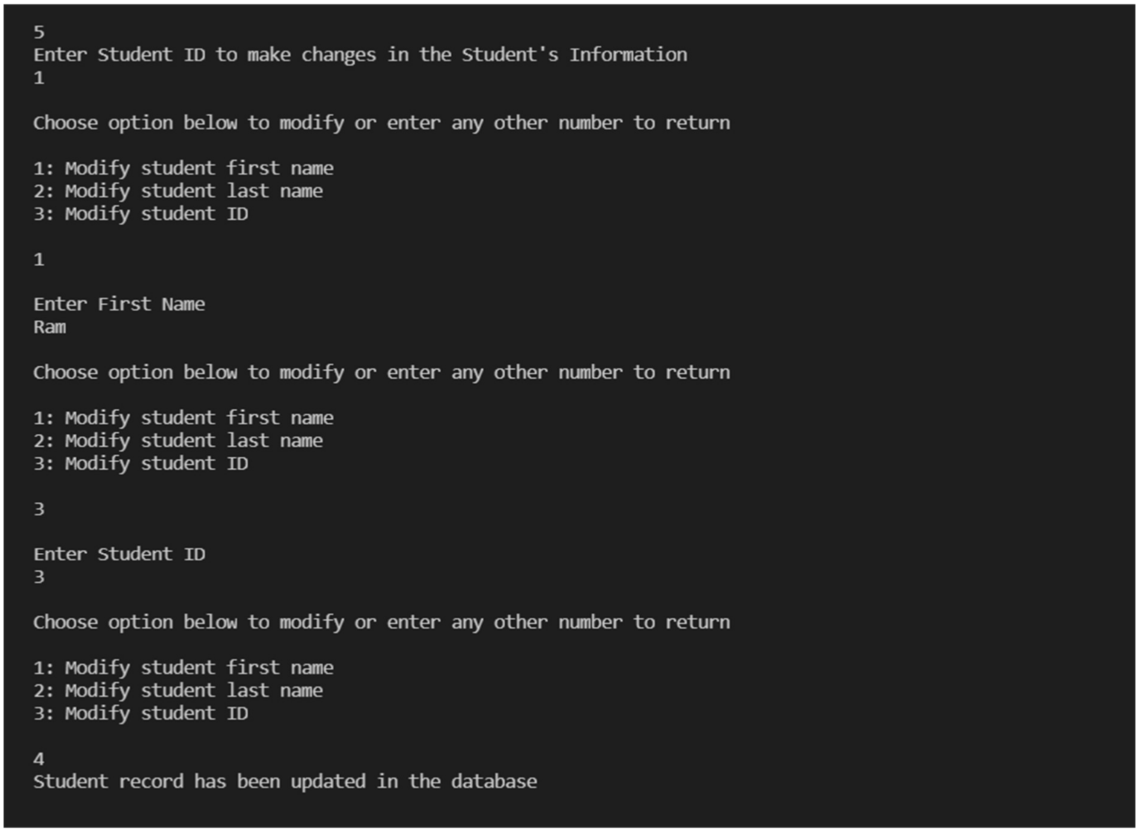
```
Please Select Options below or type -1 to quit the application

(1) Store Information from file
(2) Retrieve All Students Information and Scores from Database
(3) Store Student information manually
(4) Retrieve Student Information and Scores
(5) Modify Student Information
(6) Modify Student Scores
(7) Delete Student Record from the Database
(8) Delete All Students from the Database
(9) Get average score of all Students
(10) Get student average Letter Grade

3
Enter student first name, last name, and ID
Rama Kiran 1
Enter student marks in this order: '
'Physics, Chemistry, Maths, English, and Computer Science
90 95 100 92 91
Successfully added student in the directory
```

Retrieving Student details from database:



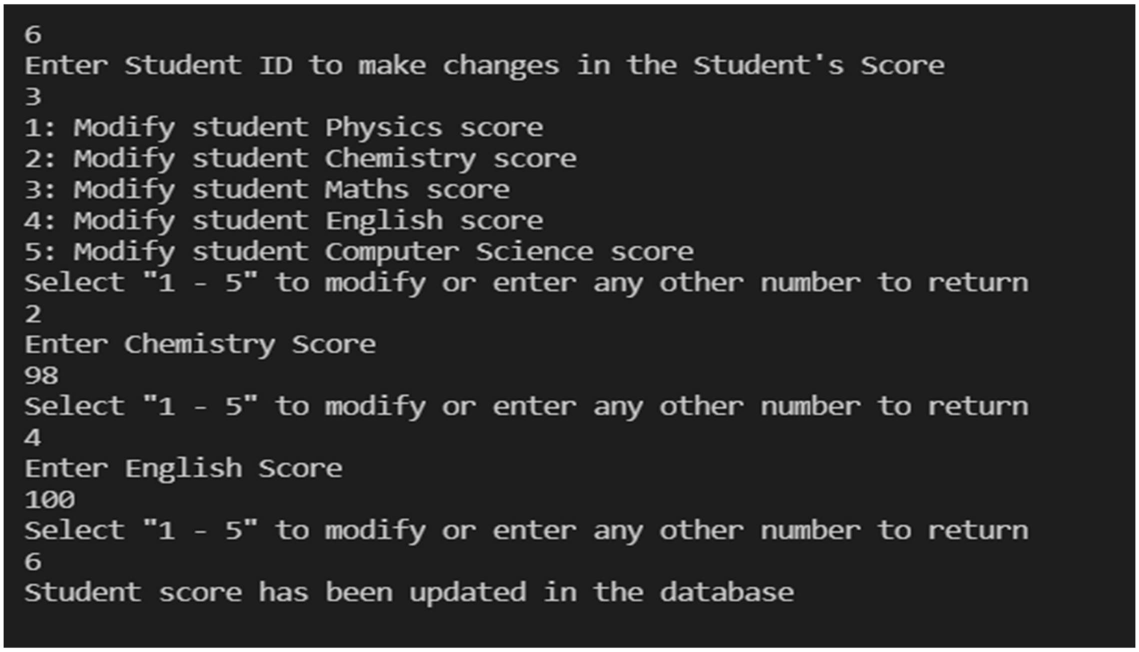
```
4
Enter student ID to get information about the student
1
Name: Rama Kiran
ID: 1
Physics: 90
Chemistry: 95
Maths: 100
English: 92
Computer Science: 91
```

Modifying Student details in database:



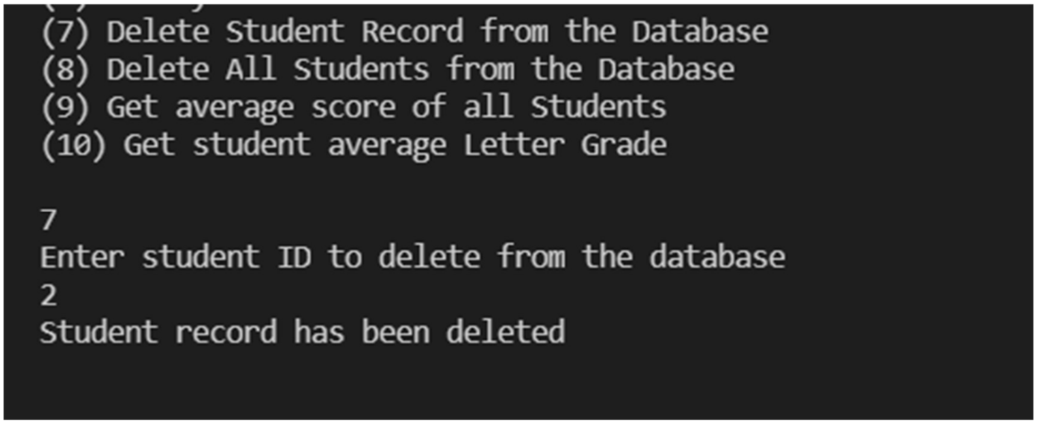
```
5
Enter Student ID to make changes in the Student's Information
1

Choose option below to modify or enter any other number to return

1: Modify student first name
2: Modify student last name
3: Modify student ID

1

Enter First Name
Ram

Choose option below to modify or enter any other number to return

1: Modify student first name
2: Modify student last name
3: Modify student ID

3

Enter Student ID
3

Choose option below to modify or enter any other number to return

1: Modify student first name
2: Modify student last name
3: Modify student ID

4
Student record has been updated in the database
```

Modifying Student marks in database:



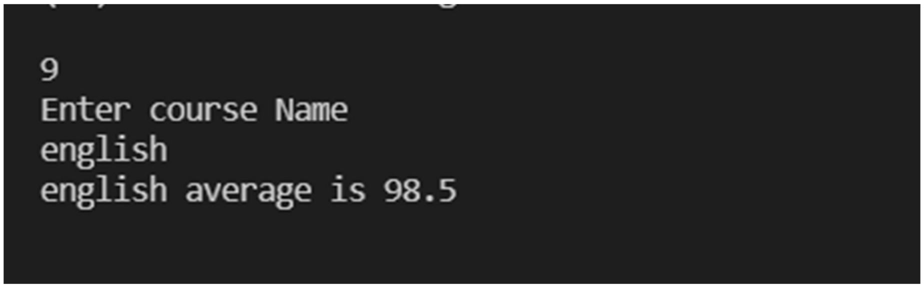
```
6
Enter Student ID to make changes in the Student's Score
3
1: Modify student Physics score
2: Modify student Chemistry score
3: Modify student Maths score
4: Modify student English score
5: Modify student Computer Science score
Select "1 - 5" to modify or enter any other number to return
2
Enter Chemistry Score
98
Select "1 - 5" to modify or enter any other number to return
4
Enter English Score
100
Select "1 - 5" to modify or enter any other number to return
6
Student score has been updated in the database
```

Deleting Student details from database:



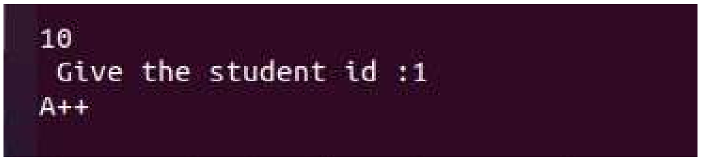
```
(7) Delete Student Record from the Database
(8) Delete All Students from the Database
(9) Get average score of all Students
(10) Get student average Letter Grade

7
Enter student ID to delete from the database
2
Student record has been deleted
```

Displaying average score of total students in a particular course:



```
9
Enter course Name
english
english average is 98.5
```

Displaying the grades of a student:



```
10
 Give the student id :1
A++
```

Requirements:

- How to store student info
  - Student’s name
  - Student’s id
  - Student’s marks

- Modifying of information
  - Student’s id
  - Student’s marks

- Calculating grades

- Deleting the student’s info
  - Student’s name
  - Student’s id
  - Student’s marks

- Displaying result
  - Marks
  - Grade
  - Average

Design:

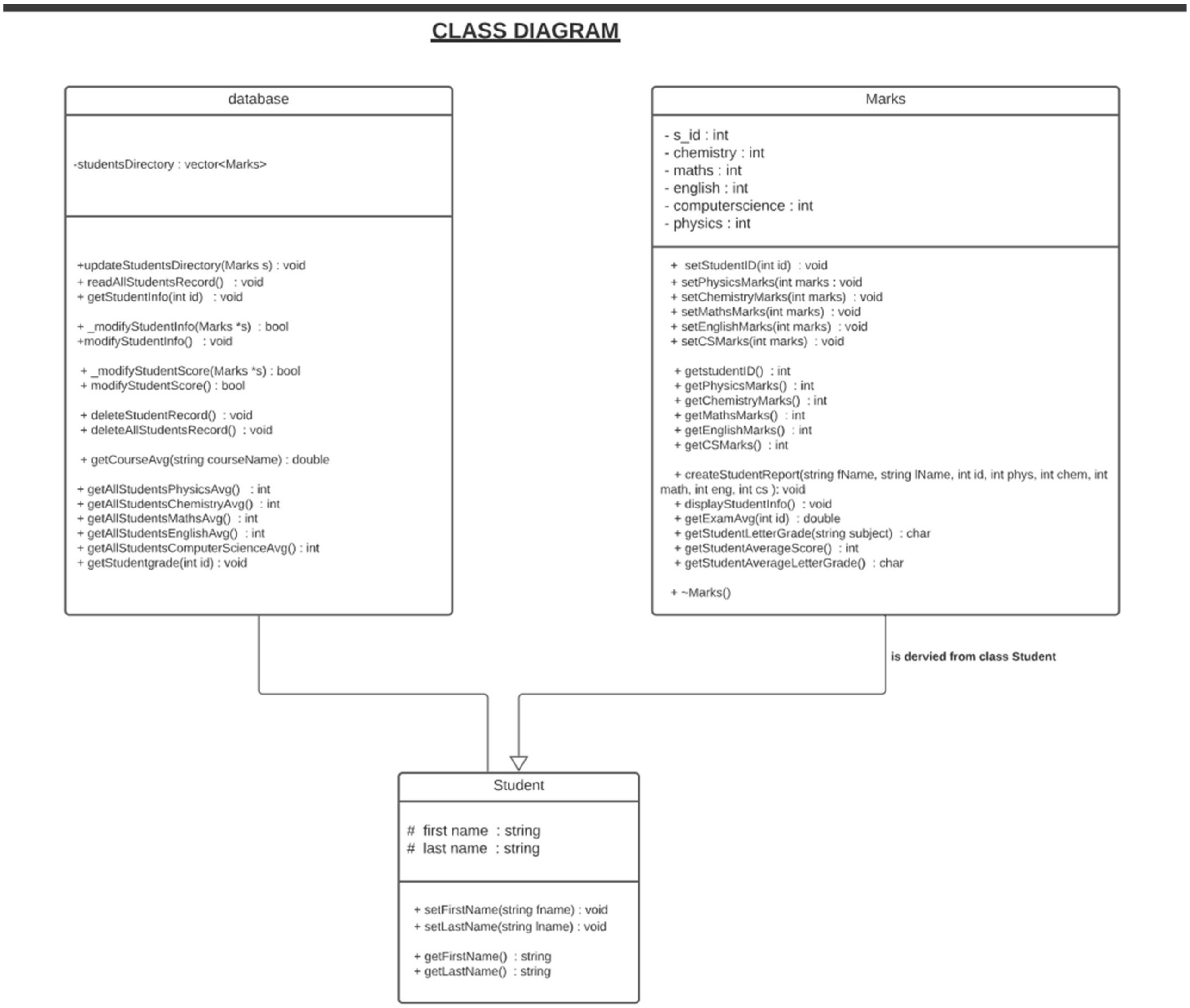
Structural UML Diagram:

CLASS DIAGRAM

This class diagram includes all the declarations of the functions and the variables in the code. This defines the structure of the code.

In our system we have 3 classes Database, Student and Marks. We have inherited Marks class from Student class. It is a single inheritance as only one parent(base class) has only one child(derived class).

to

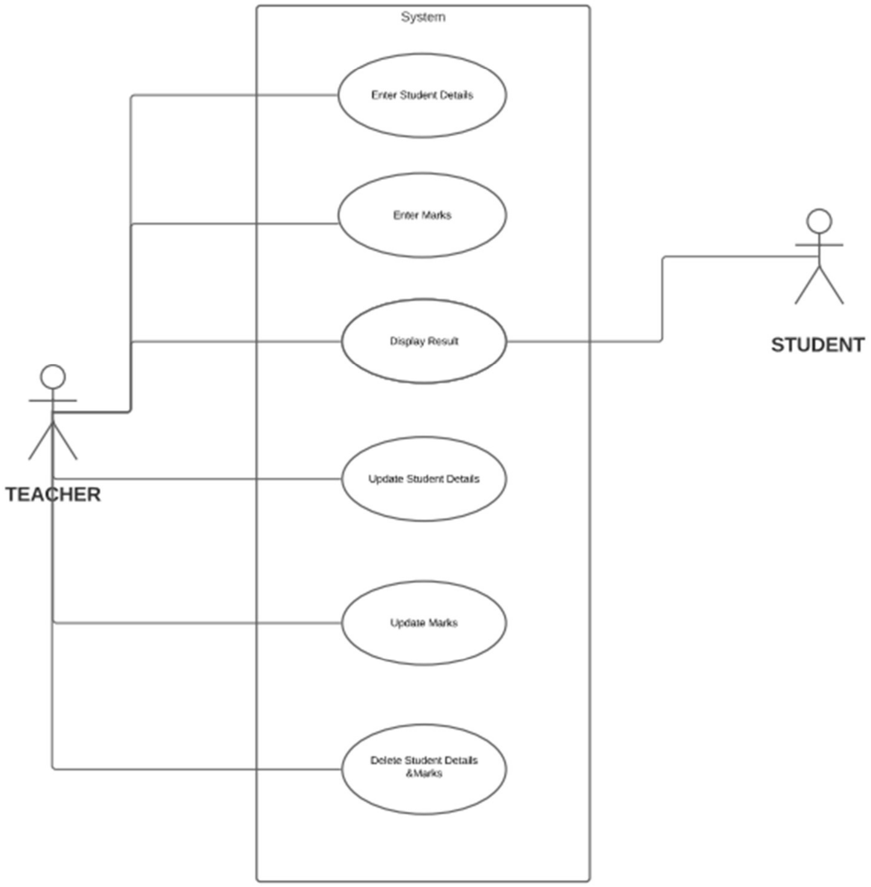

CLASS DIAGRAM
database
-studentsDirectory : vector<Marks>
+updateStudentsDirectory(Marks s) : void
+ readAllStudentsRecord() : void
+ getStudentInfo(int id) : void
+ _modifyStudentInfo(Marks *s) : bool
+modifyStudentInfo() : void
+ _modifyStudentScore(Marks *s) : bool
+ modifyStudentScore() : bool
+ deleteStudentRecord() : void
+ deleteAllStudentsRecord() : void
+ getCourseAvg(string courseName) : double
+ getAllStudentsPhysicsAvg() : int
+ getAllStudentsChemistryAvg() : int
+ getAllStudentsMathsAvg() : int
+ getAllStudentsEnglishAvg() : int
+ getAllStudentsComputerScienceAvg() : int
+ getStudentgrade(int id) : void
Marks
- s_id : int
- chemistry : int
- maths : int
- english : int
- computerscience : int
- physics : int
+ setStudentID(int id) : void
+ setPhysicsMarks(int marks : void
+ setChemistryMarks(int marks) : void
+ setMathsMarks(int marks) : void
+ setEnglishMarks(int marks) : void
+ setCSMarks(int marks) : void
+ getstudentID() : int
+ getPhysicsMarks() : int
+ getChemistryMarks() : int
+ getMathsMarks() : int
+ getEnglishMarks() : int
+ getCSMarks() : int
+ createStudentReport(string fName, string lName, int id, int phys, int chem, int math, int eng, int cs): void
+ displayStudentInfo() : void
+ getExamAvg(int id) : double
+ getStudentLetterGrade(string subject) : char
+ getStudentAverageScore() : int
+ getStudentAverageLetterGrade() : char
+ ~Marks()
is dervied from class Student
Student
# first name : string
# last name : string
+ setFirstName(string fname) : void
+ setLastName(string lname) : void
+ getFirstName() : string
+ getLastName() : string

Behavioral UML Diagram:

1. USE CASE DIAGRAM

It gives the idea of how the user get interacted with system who himself is not a part of the system. Here in this diagram it depicts two actors (Teacher and Student) and how they get related to different use cases.

USE CASE DIAGRAM

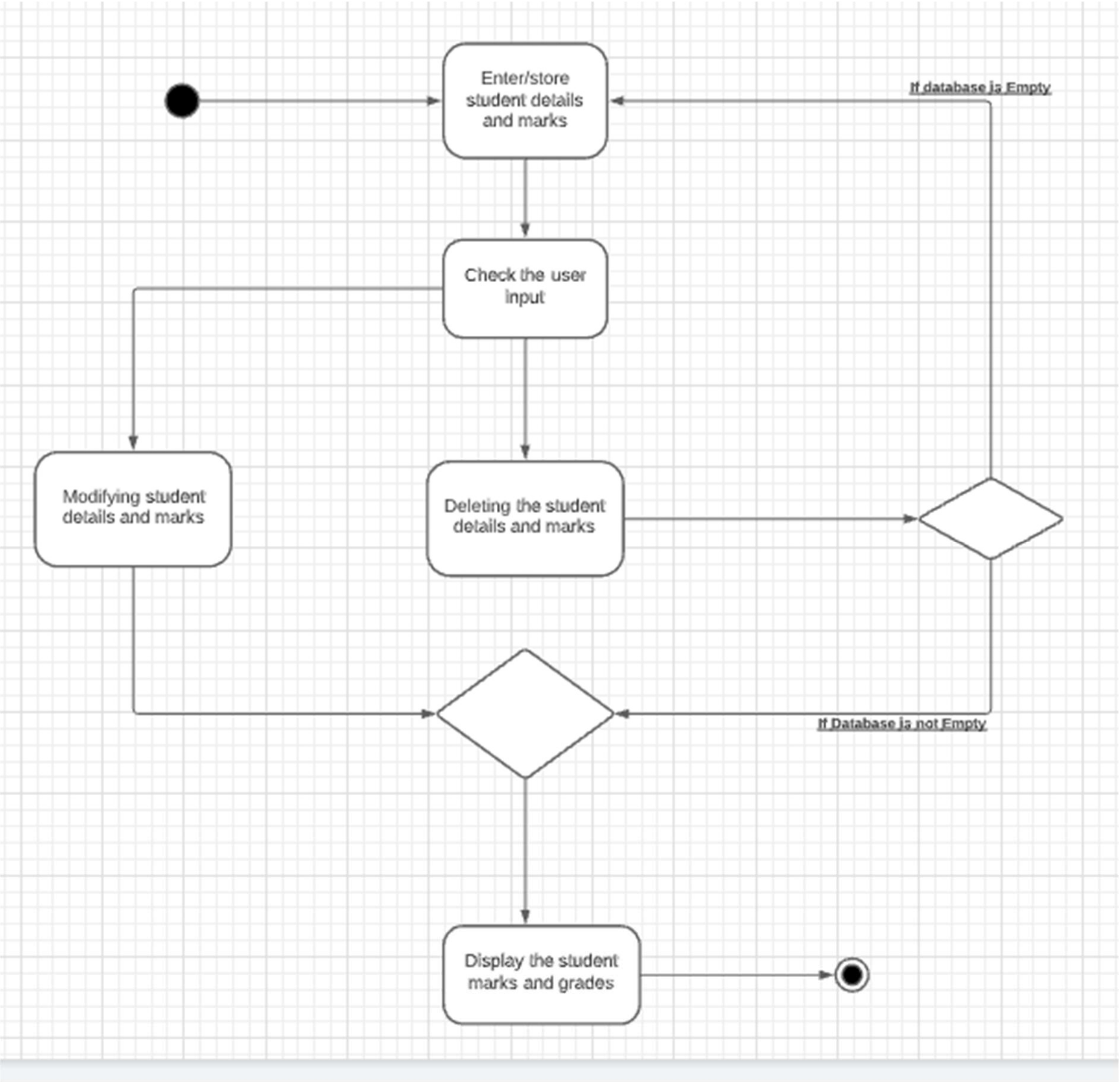


2.Activity Diagram:

At first database is empty and we have to enter our student details and marks to store them, then according to the user input we can have two options either modify or to delete. After modification it displays the students marks and grades but coming to deletion we have 2 ways:

If after the deletion the database became empty it goes to the initial stage that to enter or store the student details and marks.

If after the deletion the database did not became empty then it display the remaining student marks and grades corresponding to the remained student details.


Enter/store student details and marks
If database is Empty
Check the user input
Modifying student details and marks
Deleting the student details and marks
If Database is not Empty
Display the student marks and grades

Conclusion:

- A database has been created such that it can store all the student details (their names and ID's) and the marks scored by the respective students.

- Functions or methods have been created such that they can access the student details and can modify them.

- Total marks and the average scores are also be calculated and the grades are calculated accordingly.

- Deletion of students' detail can also be achieved similar to modify students' detail part.

- A system is been created to display the total marks and grades of the students.

In this way, we have achieved our requirements for the **Examination result system**.

In this project we learnt,

We learnt some of the system building skills along with our coding skills ang got to know that how important to learn these system building skills along with our coding skills.

We learnt about what is **Object oriented methodology** and what are the benefits in preferring OOM that Structure oriented programming.

We learnt how to make classes and objects and make use of concepts of inheritance and polymorphism practically with our project.

We also got to know the importance of group discussions which includes sharing of each other ideas and ultimately which leads to understand the importance of team work for the project.