AMITY INSTITUTE OF INFORMATION TECHNOLOGY

LAB-1

Name- Vaishnawi Ranjan

Enrollment No- A45304821028

Program- BCA

Semester- VIth [A]

Subject- Advanced Java

Submitted to:

Dr.Naveen Kumar Singh

Program Discription:

Write a Java application utilizing JDBC (Java Database Connectivity) to establish a connection with a relational database and execute fundamental CRUD (Create, Read, Update, Delete) operations on a designated table within the database.

The application's objectives are as follows:

- Offer functionalities to perform CRUD operations, such as inserting new records into the specified table, retrieving existing records based on specified criteria, updating records, and deleting records.

- Implement robust error handling mechanisms to gracefully manage connection failures and exceptions occurring during database operations.

Emphasizing simplicity and functionality, the application serves as a basic framework for leveraging JDBC in CRUD operations.

Design Description:

The plan for creating a simple Java application that connects to a database via JDBC and performs CRUD operations involves several important aspects:

User Interface Design:

When the program starts, users are presented with a menu featuring five options.

Four of these options correspond to CRUD operations (Create, Read, Update, Delete), while the fifth option allows the user to exit the application gracefully.

Based on the user's selection, the program calls the appropriate method from the Student class to execute the desired operation.

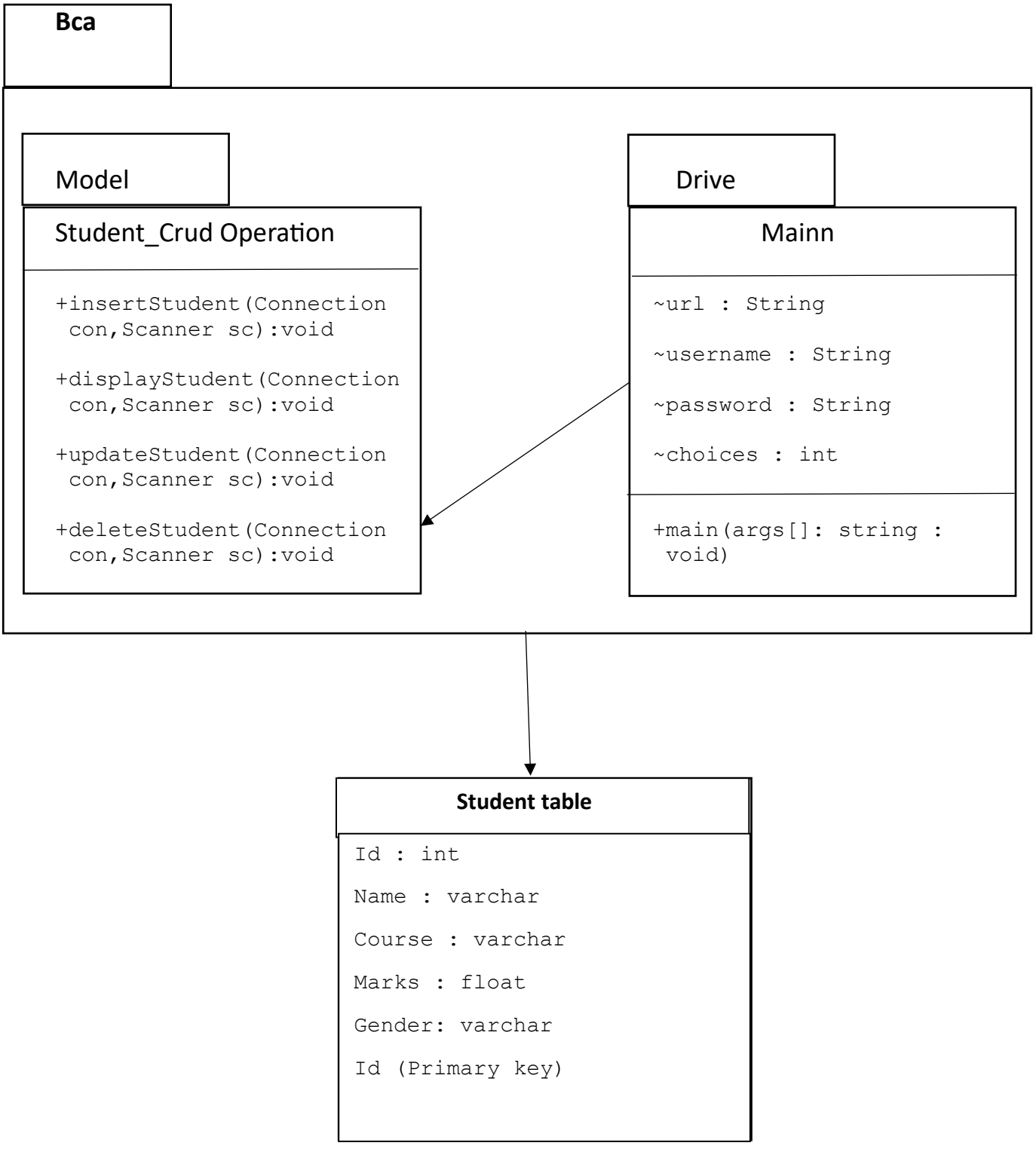
Database Connection Management:

The application must establish a JDBC connection with the database using the correct connection details, such as the URL, username, and password.

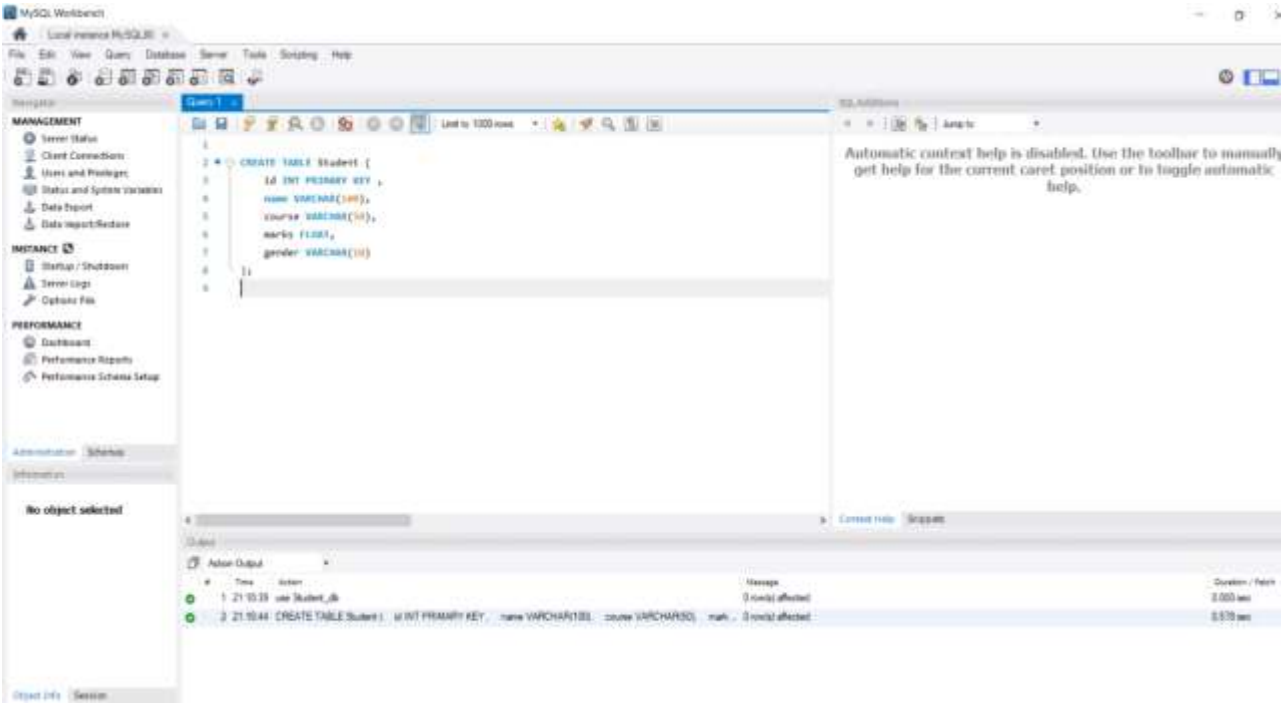
Class Diagram:

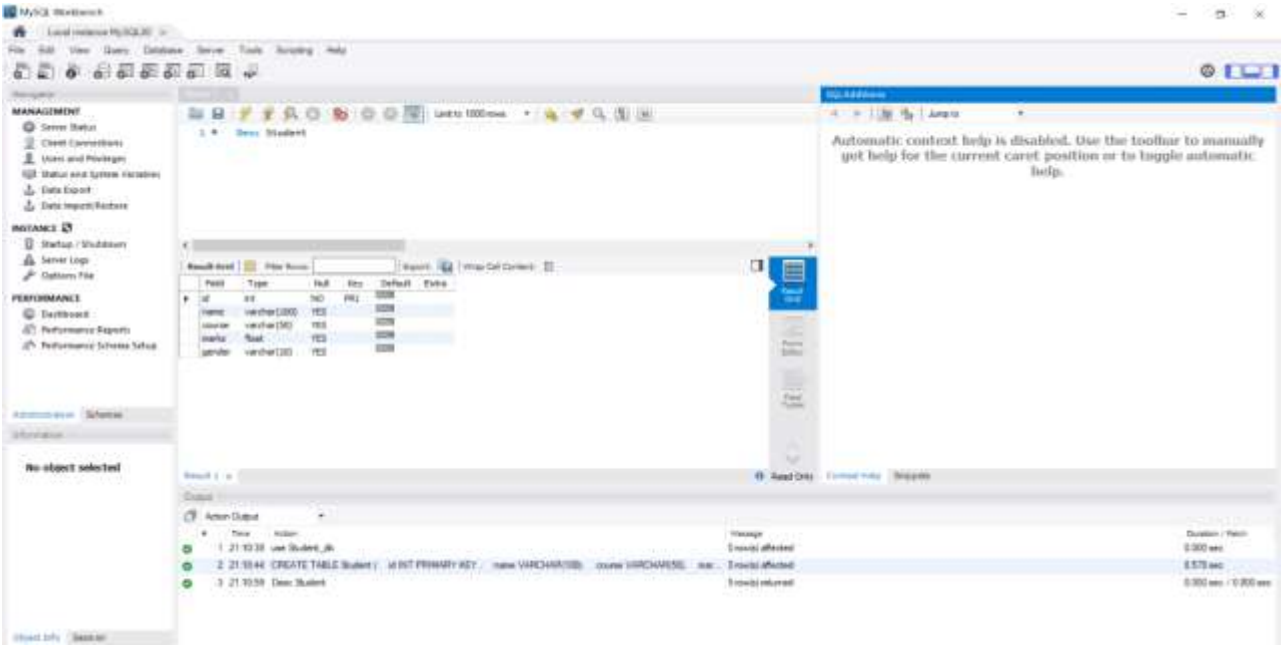
A class diagram is essential for visually representing the structure, relationships, and behavior of classes within the system.

It helps in organizing software components, facilitating communication among developers, guiding implementation, and ensuring consistency and scalability throughout the design process.


Bca
Model
Student_Crud Operation
+insertStudent(Connection con,Scanner sc):void
+displayStudent(Connection con,Scanner sc):void
+updateStudent(Connection con,Scanner sc):void
+deleteStudent(Connection con,Scanner sc):void
Drive
Mainn
~url : String
~username : String
~password : String
~choices : int
+main(args[]: string : void)
Student table
Id : int
Name : varchar
Course : varchar
Marks : float
Gender: varchar
Id (Primary key)

Schema:


MySQL Workbench
CREATE TABLE Student (
id INT PRIMARY KEY ,
name VARCHAR(100),
course VARCHAR(50),
marks FLOAT,
gender VARCHAR(10)
);
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
No object selected


MySQL Workbench
Desc Student
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
No object selected

Code:

Student.java

```
package bca.model;


import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Scanner;


public class Student {


	public Student() {

		super();

		// TODO Auto-generated constructor stub

	}

	public void addStudent(Connection con, Scanner sc) throws
SQLException {

		//create statement

		Statement st = con.createStatement();


		//read student details

		System.out.println("Enter Student Id: ");

		int id = sc.nextInt();


		System.out.println("Enter Student Name: ");

		String name = sc.next();


		System.out.println("Enter Student Course: ");

		String course = sc.next();


		System.out.println("Enter Student Marks: ");

		double marks = sc.nextDouble();
```

```
            System.out.println("Enter Student Gender: ");
            String gender = sc.next();

            //create sql squery string
            String query = String.format("Insert Into student
values(%d,'%s', '%s', %f, '%s') ", id, name,course, marks, gender);

            //execute sql query
            int rows = st.executeUpdate(query);

            System.out.println(rows + " record inserted!!!");

      }

      public void displayStudents(Connection con) throws SQLException {
            Statement st = con.createStatement();

            ResultSet rs = st.executeQuery("select * from student");

            while(rs.next()) {
                  System.out.println(rs.getInt(1)+ "\t"+rs.getString(2)+
"\t"+rs.getString(3)+ "\t"+ "\t"+rs.getDouble(4)+"\t"+rs.getString(5));
            }
      }

      public void updateStudentName(Connection con, Scanner sc) throws
SQLException {
            Statement st = con.createStatement();
            System.out.println("Enter Student ID: ");
            int id = sc.nextInt();
            System.out.println("Enter Student New Name: ");
            String name = sc.next();

            String query = String.format("update student set name='%s'
where id = %d", name, id);
            int rowsAffected = st.executeUpdate(query);
```

```
            System.out.println(rowsAffected+" recored updated!!!");


      }


      public void deleteStudent(Connection con, Scanner sc) throws
SQLException {
            Statement st = con.createStatement();
            System.out.println("Enter Student ID: ");
            int id = sc.nextInt();


            int rowAffected = st.executeUpdate("delete from student where
id = "+id);
            System.out.println(rowAffected + " recored deleted!!!");
      }
}
```

Mainn.java

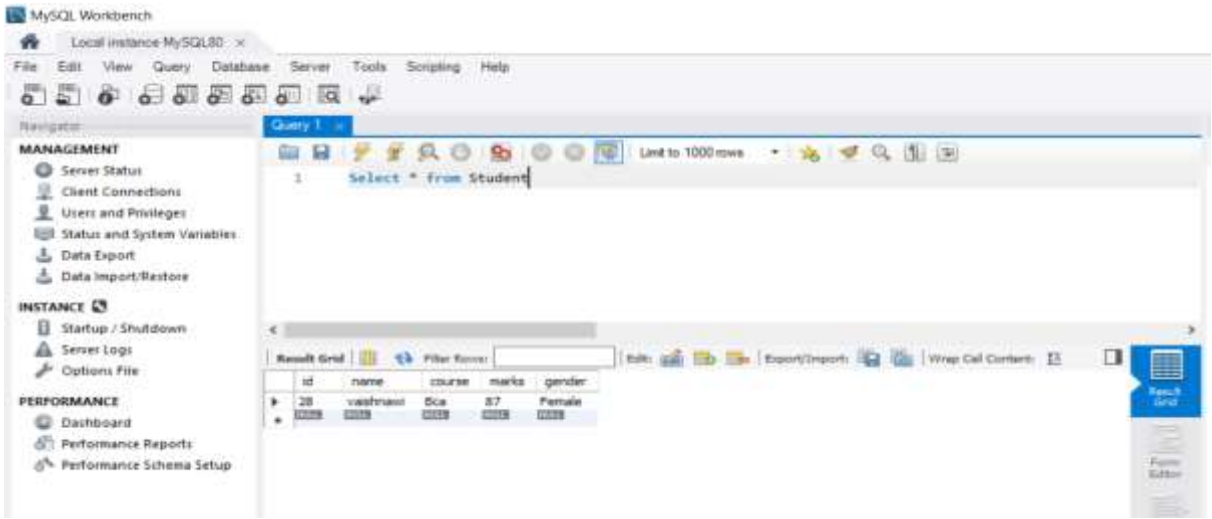
```
package bca.drive;


import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Scanner;


import bca.model.Student;


public class Mainn {


      public static void main(String[] args) throws ClassNotFoundException,
SQLException {
            // TODO Auto-generated method stub
            //1. load and register
            Class.forName("com.mysql.cj.jdbc.Driver");
```

```
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/Student_db","root"
,"vaishnawi@123");


          Scanner sc = new Scanner(System.in);
            Student student = new Student();



            while(true) {
                  menu();
                  int choice = sc.nextInt();
                  switch(choice) {
                  case 1: student.addStudent(con, sc);
                        break;

                  case 2: student.displayStudents(con);
                        break;

                  case 3: student.updateStudentName(con, sc);
                        break;

                  case 4: student.deleteStudent(con, sc);
                        break;

                  case 5:
                        System.out.println("Exit the Application! Thank
You");
                        System.exit(0);
                  default:
                        System.out.println("Invalid Choice!!");
                  }
            }

      }

      public static void menu() {
```

```
            System.out.println("------------Menu-----------");
            System.out.println("1. Add New Student");
            System.out.println("2. Display All Students");
            System.out.println("3. Update Name of Student");
            System.out.println("4. Delete a Student");
            System.out.println("5. Exit");
            System.out.println("Your Choice...");
        }

}
```

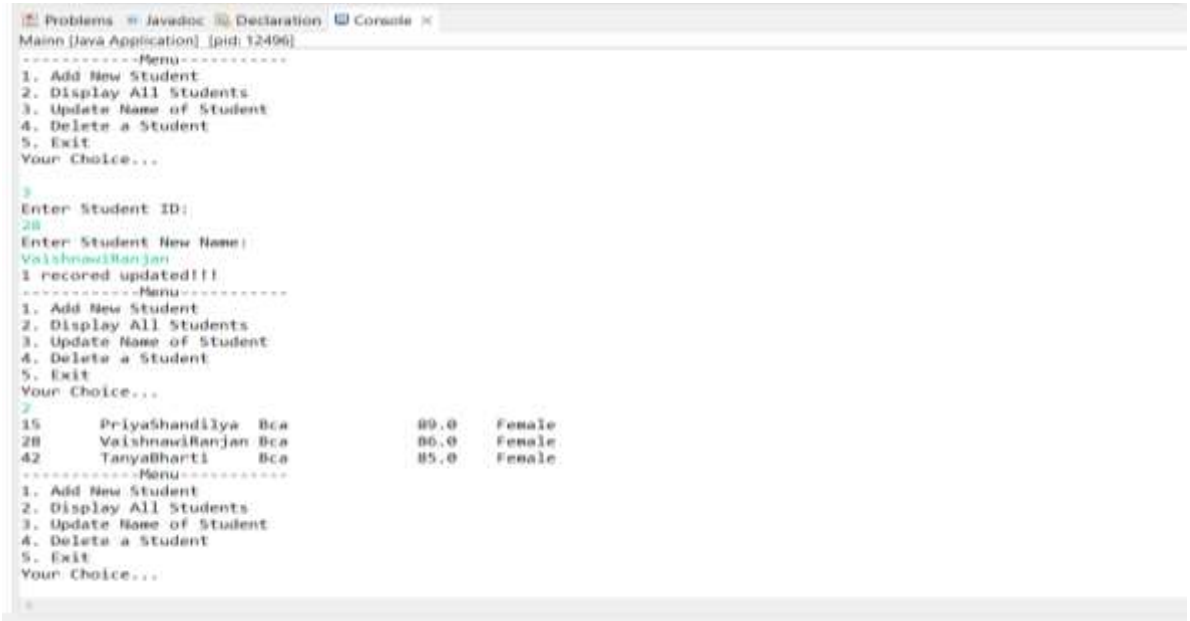
Input/Output:

Insert operation :



Display operation:

Update operation :



Delete Operation:

Exit:

Invalid choice: