AMITY UNIVERSITY, PATNA

AMITY INSTITUTE OF INFORMATION TECHNOLOGY

Advanced Java Lab

Name: Kamini Chourasia
Enroll.no:A45304821033
Program:BCA–6(A)
Submitted to:Dr.Naveen Kumar Singh

CRUD OPERATIONS

Problem description :

Develop a simple Java application that utilizes JDBC (Java Database Connectivity) to establish a connection with a relational database system and perform basic CRUD (Create, Read, Update, Delete) operations on a specified database table. The application should:

1. Provide options to perform CRUD operations including inserting new records into the database table, retrieving existing records from the table based on specified criteria, updating records in the table and deleting records from the table.

2. Implement error handling to manage connection failures and database operation exceptions gracefully.

The application should focus on simplicity and functionality, serving as a basic template for JDBC usage in CRUD operations

DESIGN DESCRIPTION

The design of the problem statement for creating a simple Java application that establishes JDBC connection and performs CRUD operations involves several key components and considerations:

1. User Interface Design :

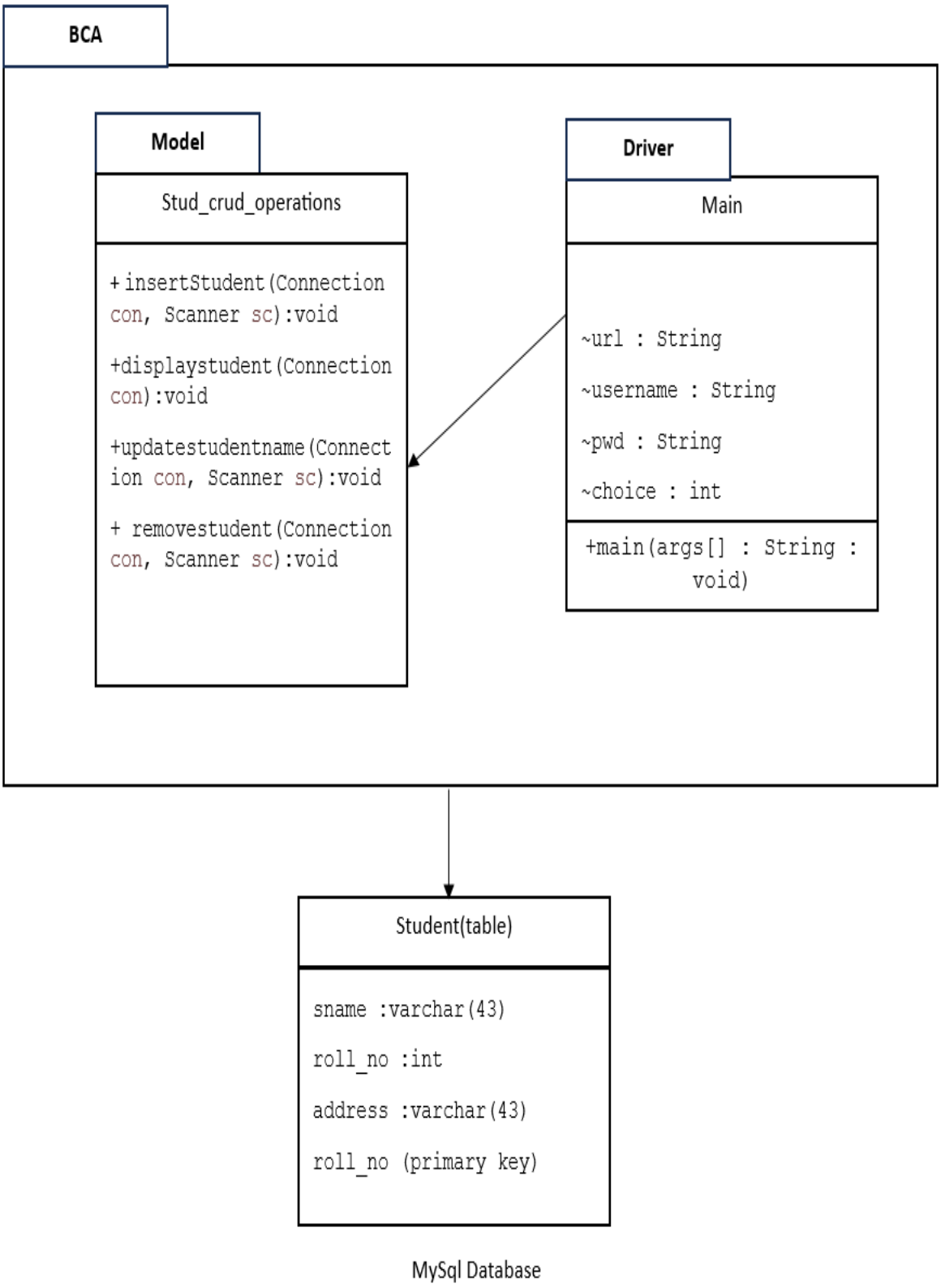
Upon running the application, users will be presented with a menu containing 5 options, with 4 of them representing crud operations and the last option for exiting the application gracefully. Based on the user's choice, the application will invoke the appropriate method from the Student class to perform the CRUD operation. This Java program establishes a connection to a MySQL database and implements a simple console-based CRUD (Create, Read, Update, Delete) system for managing student records. It utilizes a menu-driven user interface with options to add a new student, display all students, update student details, delete a student, or exit the program. The program uses a student class to encapsulate database operations, and a Scanner for user input. The main method operates in an infinite loop, continuously presenting the user with the menu until they choose to exit. While the program serves as a basic framework, further details about the implementation of the student class and its methods are necessary for a complete understanding. Additionally, considerations for error handling and proper closure of database connections could enhance the program's robustness.

2. Database Connection Management:

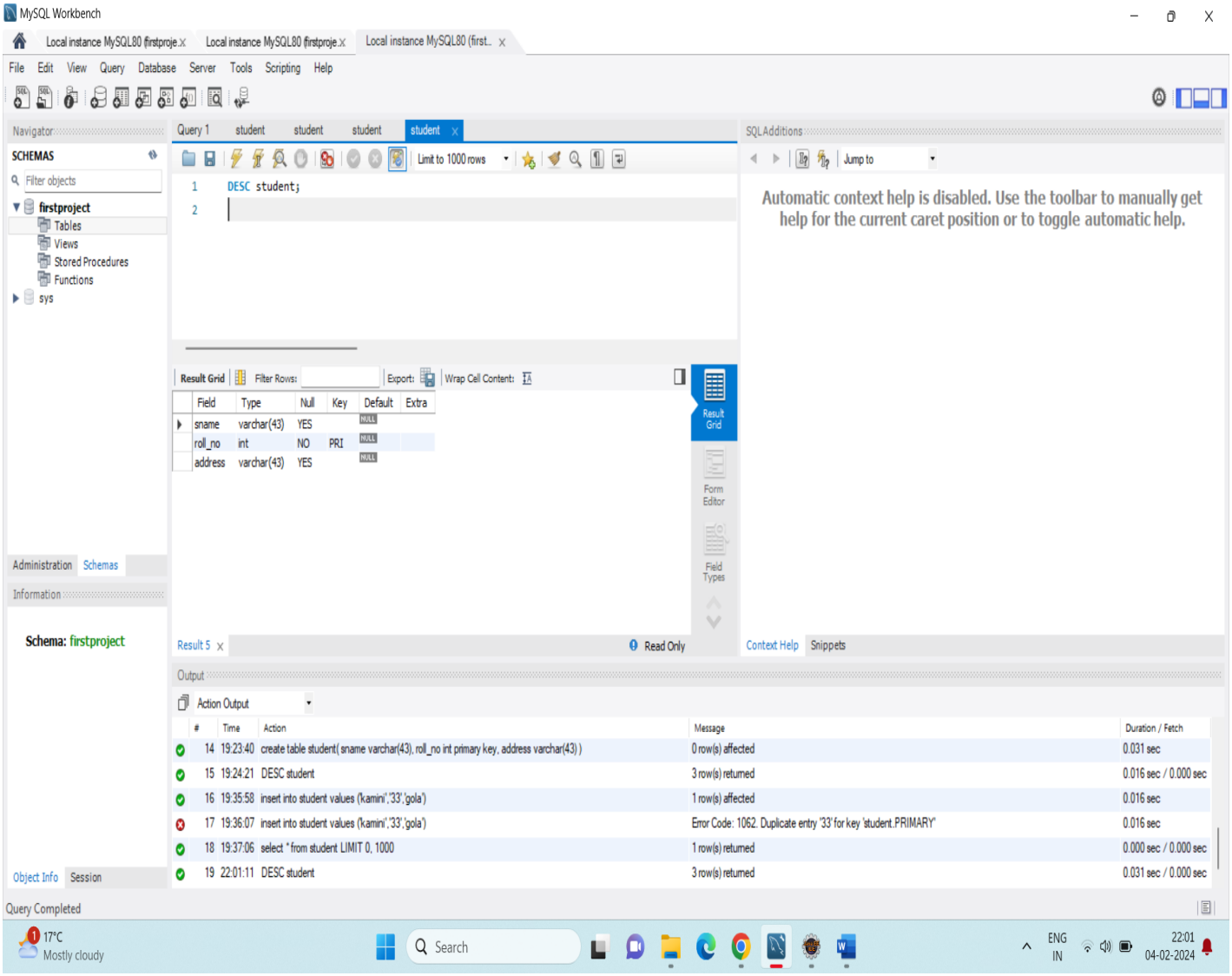
The application needs to establish a JDBC connection with the relational database system using the correct connection details.

3. Class Diagram:

A class diagram is crucial for design purposes as it visually illustrates the structure, relationships, and behavior of classes within a system. It aids in organizing and conceptualizing software components, facilitating communication among developers, guiding implementation, and ensuring consistency and scalability. throughout the design process. Here’s a class diagram demonstrating our problem statement.


BCA
Model
Stud_crud_operations
+ insertStudent(Connection con, Scanner sc):void
+displaystudent(Connection con):void
+updatestudentname(Connection con, Scanner sc):void
+ removestudent(Connection con, Scanner sc):void
Driver
Main
~url : String
~username : String
~pwd : String
~choice : int
+main(args[] : String : void)
Student(table)
sname :varchar(43)
roll_no :int
address :varchar(43)
roll_no (primary key)
MySql Database

SCHEMA


MySQL Workbench
Local instance MySQL80 (firstproje...
Local instance MySQL80 (firstproje...
Local instance MySQL80 (first...
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
firstproject
Tables
Views
Stored Procedures
Functions
sys
Administration Schemas
Information
Schema: firstproject
Object Info Session
Query 1 student student student student
Limit to 1000 rows
1 DESC student;
2
Result Grid Filter Rows: Export: Wrap Cell Content:
Field Type Null Key Default Extra
sname varchar(43) YES NULL
roll_no int NO PRI NULL
address varchar(43) YES NULL
Result Grid
Form Editor
Field Types
Result 5
Read Only
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help Snippets
Output
Action Output
Time Action Message Duration / Fetch
14 19:23:40 create table student(sname varchar(43), roll_no int primary key, address varchar(43)) 0 row(s) affected 0.031 sec
15 19:24:21 DESC student 3 row(s) returned 0.016 sec / 0.000 sec
16 19:35:58 insert into student values ('kamini','33','gola') 1 row(s) affected 0.016 sec
17 19:36:07 insert into student values ('kamini','33','gola') Error Code: 1062. Duplicate entry '33' for key 'student.PRIMARY' 0.016 sec
18 19:37:06 select * from student LIMIT 0, 1000 1 row(s) returned 0.000 sec / 0.000 sec
19 22:01:11 DESC student 3 row(s) returned 0.031 sec / 0.000 sec
Query Completed
17°C Mostly cloudy
Search
ENG IN
22:01 04-02-2024

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();

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

// Create statement
Statement st = con.createStatement();
//read student details
System.out.println("Enter the Student Name:");
String sname = sc.next();
System.out.println("Enter the student Roll Number:");
int roll_no = sc.nextInt();
System.out.println("Enter Student Address: ");
String address = sc.next();
// create sql query string
//create sql squery string
String query = String.format("Insert Into student values('%s', %d, '%s') ", sname,
roll_no, address);
//excecute sql quesry
int rows = st.executeUpdate(query);

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

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

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

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

}
public void updatestudentname(Connection con, Scanner sc) throws SQLException {
Statement st = con.createStatement();
System.out.println("Enter Student Roll: ");
int roll_no = sc.nextInt();
System.out.println("Enter Student New Name: ");
String sname = sc.next();

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

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

```
}
public void removestudent(Connection con, Scanner sc) throws SQLException {
Statement st = con.createStatement();
System.out.println("Enter Student Roll: ");
int roll_no = sc.nextInt();

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


}

}
```

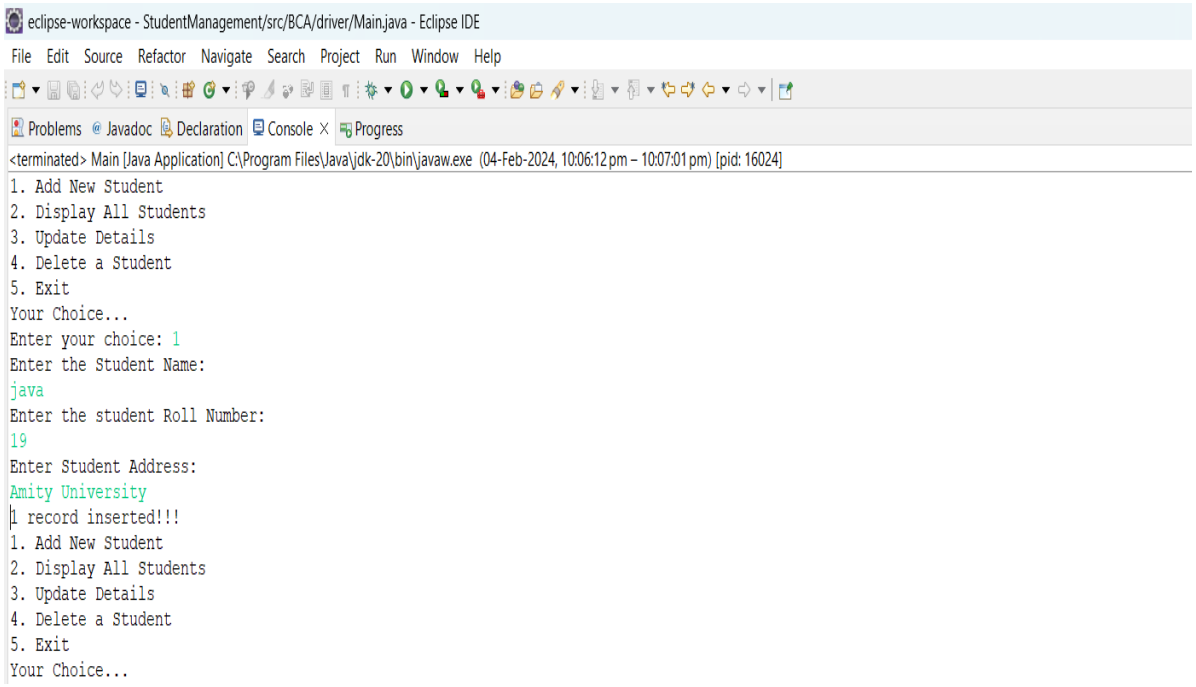
Main.java

```
package BCA.driver;

import java.sql.*;
import java.util.Scanner;
import BCA.model.Student;

public class Main {

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");
//1
String url = "jdbc:mysql://localhost:3306/firstproject";
String username = "root";
String pwd = "12345";
Connection con = DriverManager.getConnection(url, username, pwd);
Scanner sc = new Scanner(System.in);

Student stud = new Student();
while(true) {
menu();
System.out.print("Enter your choice: ");
int choice = sc.nextInt();
switch (choice) {
case 1:
stud.insertStudent(con, sc);
break;
case 2:
stud.displaystudent(con);
break;
case 3:
stud.updatestudentname(con, sc);
break;
case 4:
stud.removestudent(con, sc);
break;
case 5:
System.out.println("Exiting program. Goodbye!");
System.exit(0);
break;
default:
System.out.println("Invalid choice. Please enter a number between 1 and 5.");
}
```

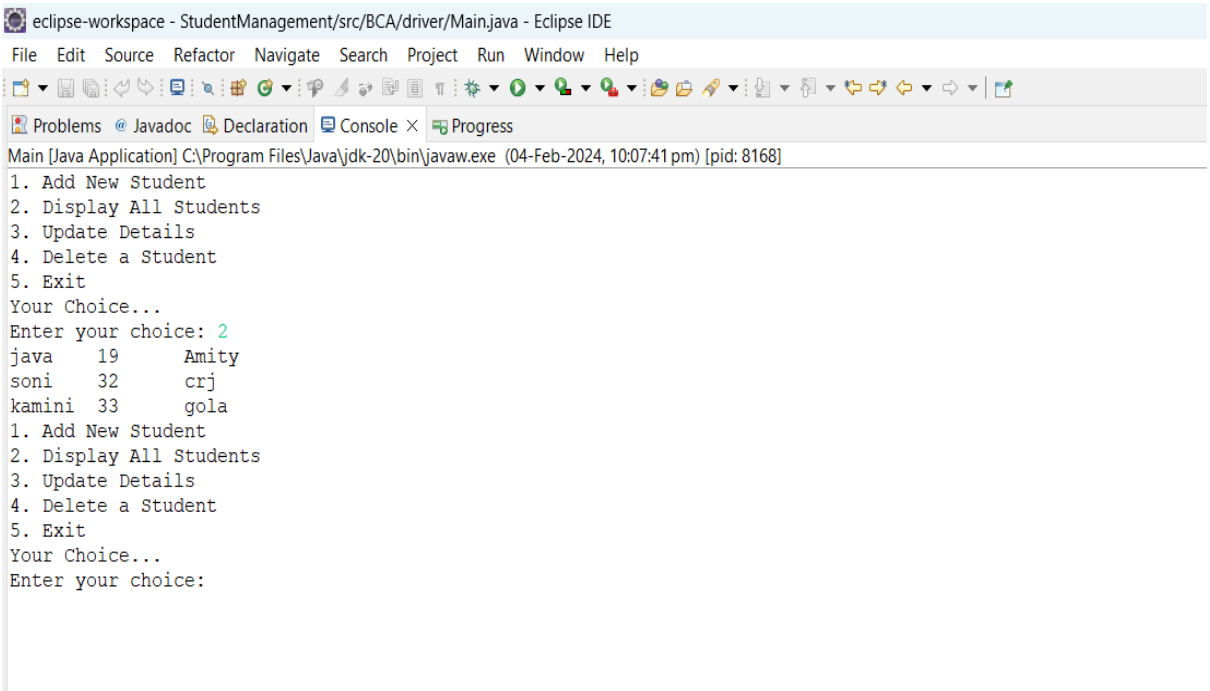
```
}

}
public static void menu() {
System.out.println("1. Add New Student");
System.out.println("2. Display All Students");
System.out.println("3. Update Details");
System.out.println("4. Delete a Student");
System.out.println("5. Exit");
System.out.println("Your Choice...");


}

}
```

INPUT/OUTPUT

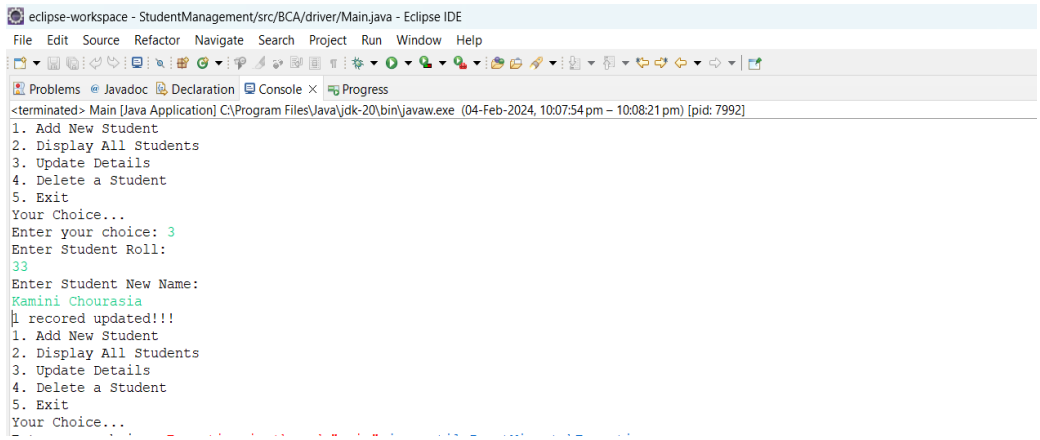
Insert operation :



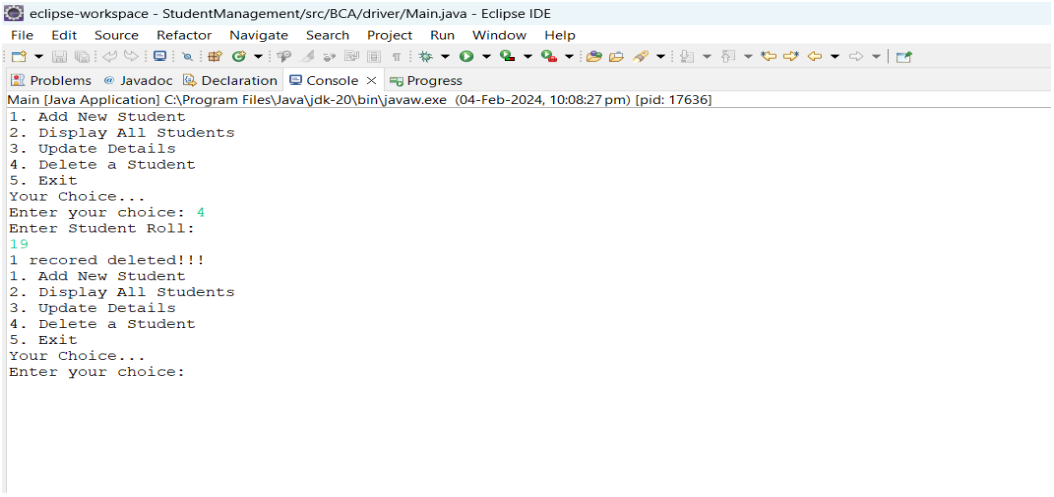
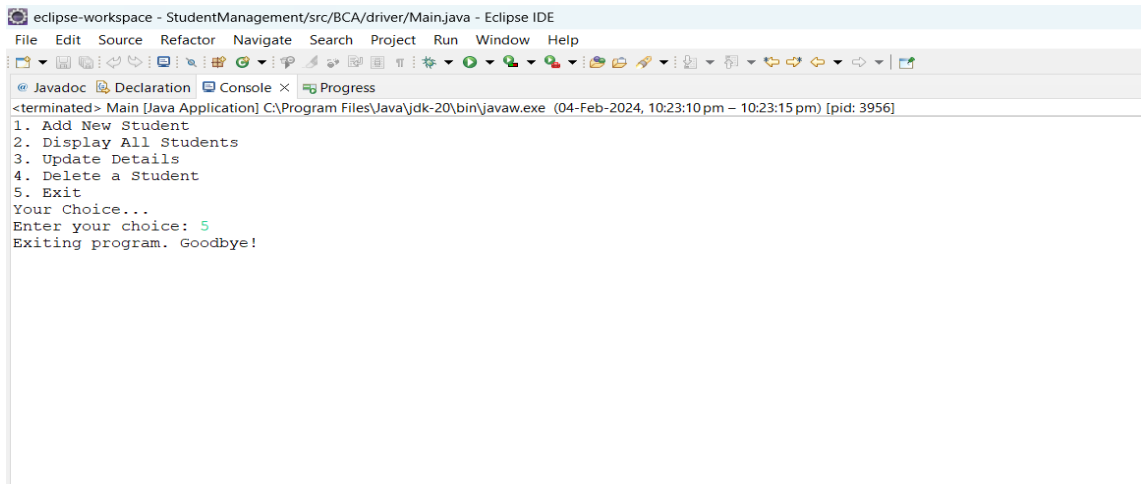
Display operation :



Update operation :



Delete operation :



Edit Operation: