AMITY UNIVERSITY, PATNA

AMITY INSTITUTE OF INFORMATION TECHNOLOGY

Advanced Java Lab

Name : Keshav Kant Mishra
Program/Semester : BCA – 6 ‘A’
Enroll. Number : A45304821002
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 :

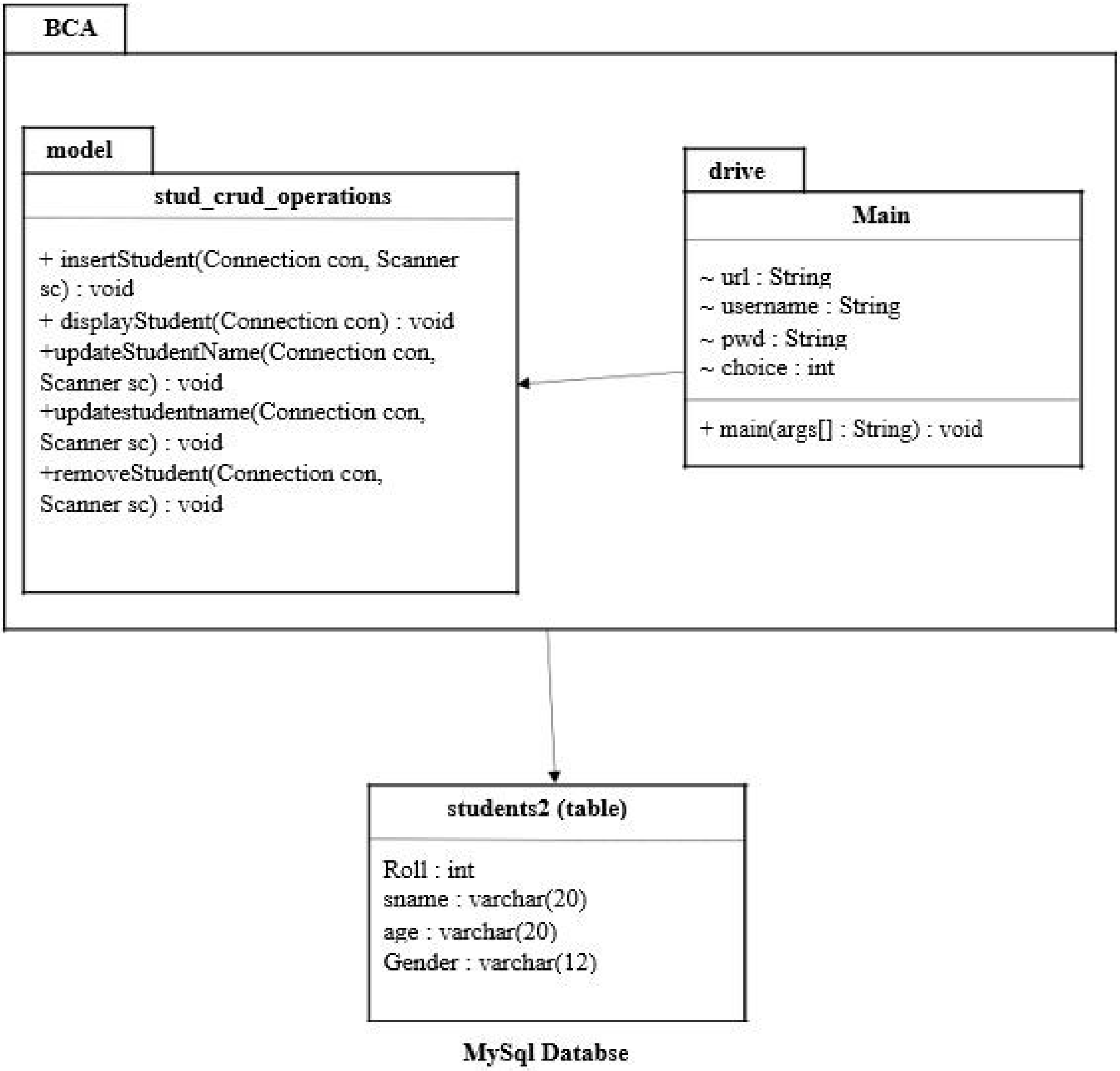
Upon running the application, users will be presented with a menu containing 5 options, with 4 of them representing crud operations and the last option for exiting the application gracefully. Based on the user's choice, the application will invoke the appropriate method from the Student class to perform 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:

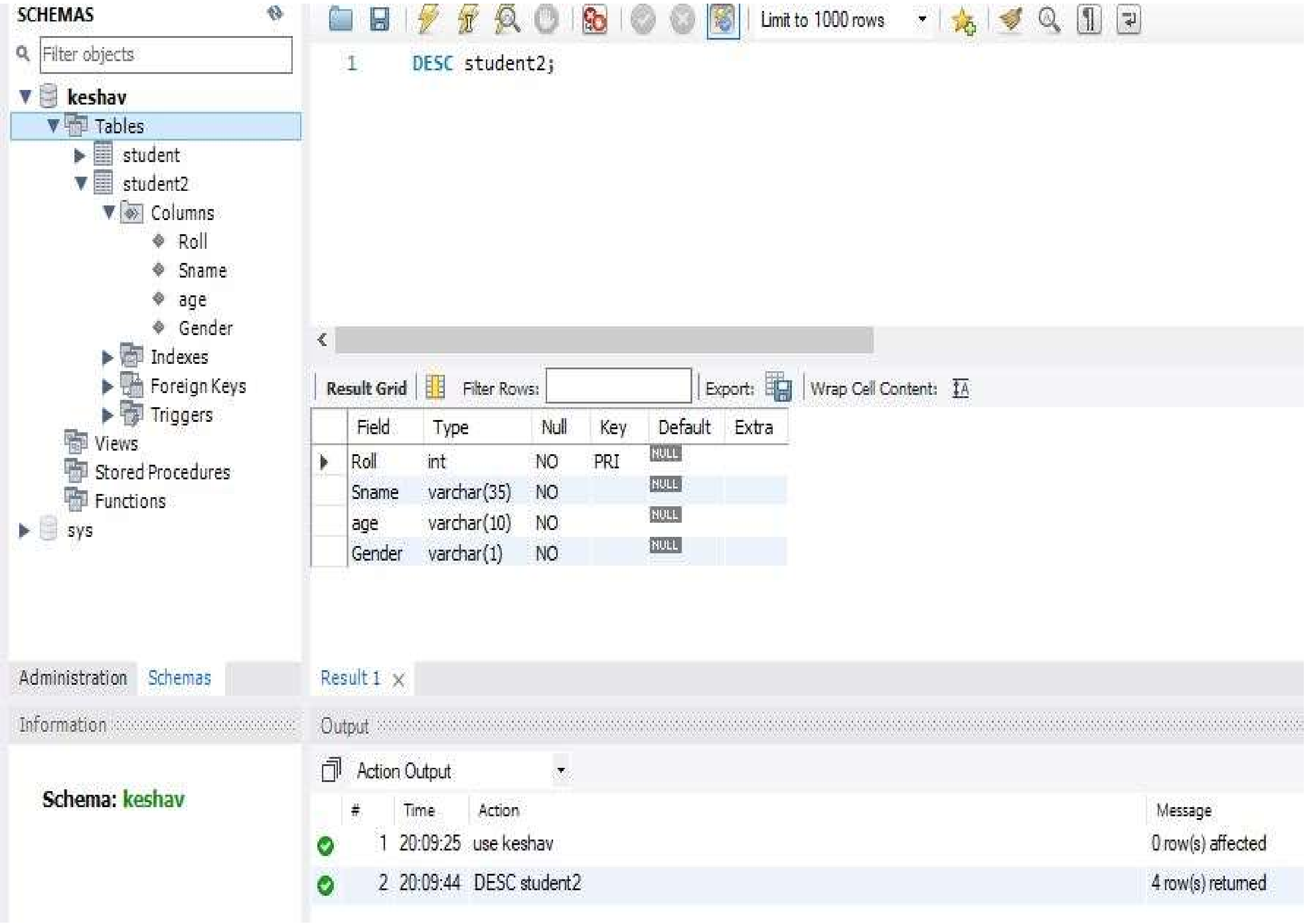
The application needs to establish a JDBC connection with the relational database system using the correct connection details.

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
+updatestudentname(Connection con, Scanner sc) : void
+removeStudent(Connection con, Scanner sc) : void
drive
Main
~ url : String
~ username : String
~ pwd : String
~ choice : int
+ main(args[] : String) : void
students2 (table)
Roll : int
sname : varchar(20)
age : varchar(20)
Gender : varchar(12)
MySql Databse

SCHEMA



```
DESC student2;
```

Field	Type	Null	Key	Default	Extra
Roll	int	NO	PRI	NULL	
Sname	varchar(35)	NO		NULL	
age	varchar(10)	NO		NULL	
Gender	varchar(1)	NO		NULL	

#	Time	Action	Message
1	20:09:25	use keshav	0 row(s) affected
2	20:09:44	DESC student2	4 row(s) returned

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 Roll Number:");
          int Roll = sc.nextInt();
          System.out.println("Enter the Student Name:");
          String Sname = sc.next();
          System.out.println("Enter Student Age: ");
          String age = sc.next();
          System.out.println("Enter Student Gender: ");
          String Gender = sc.next();

          // create sql query string
          //create sql squery string
          String query = String.format("Insert Into student2
values(%d, '%s', %s, '%s') ", Roll, Sname, age, Gender);
          //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
student2");
```

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

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

          String query = String.format("update student2 set
Sname='%s' where Roll = %d", Sname, Roll);
          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 = sc.nextInt();

          int rowAffected = st.executeUpdate("delete from student2
where Roll = "+Roll);
          System.out.println(rowAffected + " recored deleted!!!");


     }

}
```

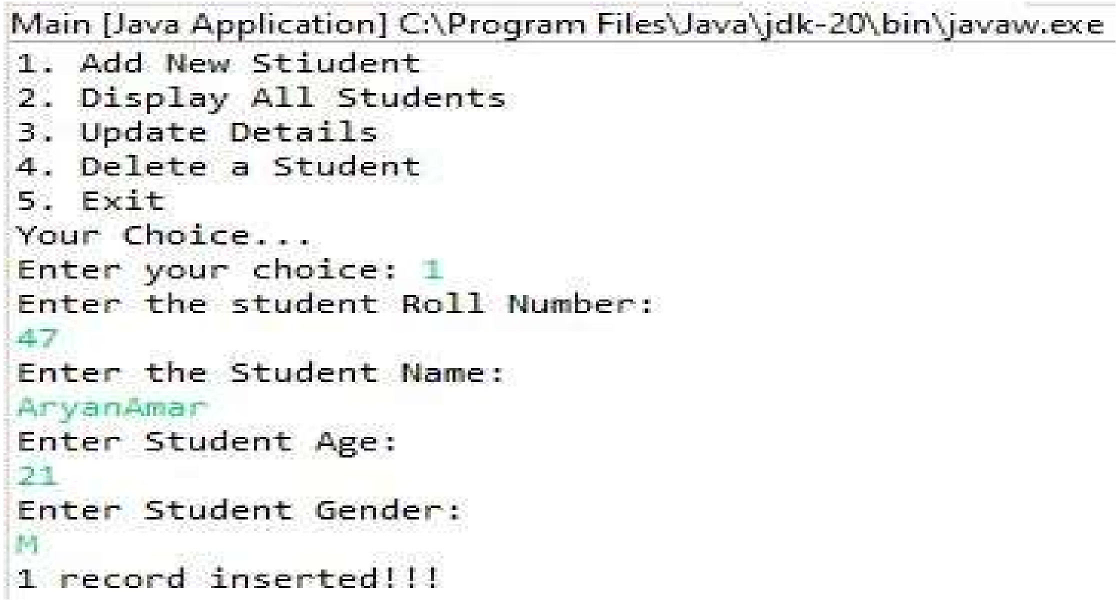
Main.java

```
package BCA.drive;

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/keshav";
         String username = "root";
         String pwd = "gautam7488";
         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 Stiudent");
         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...");}}
```

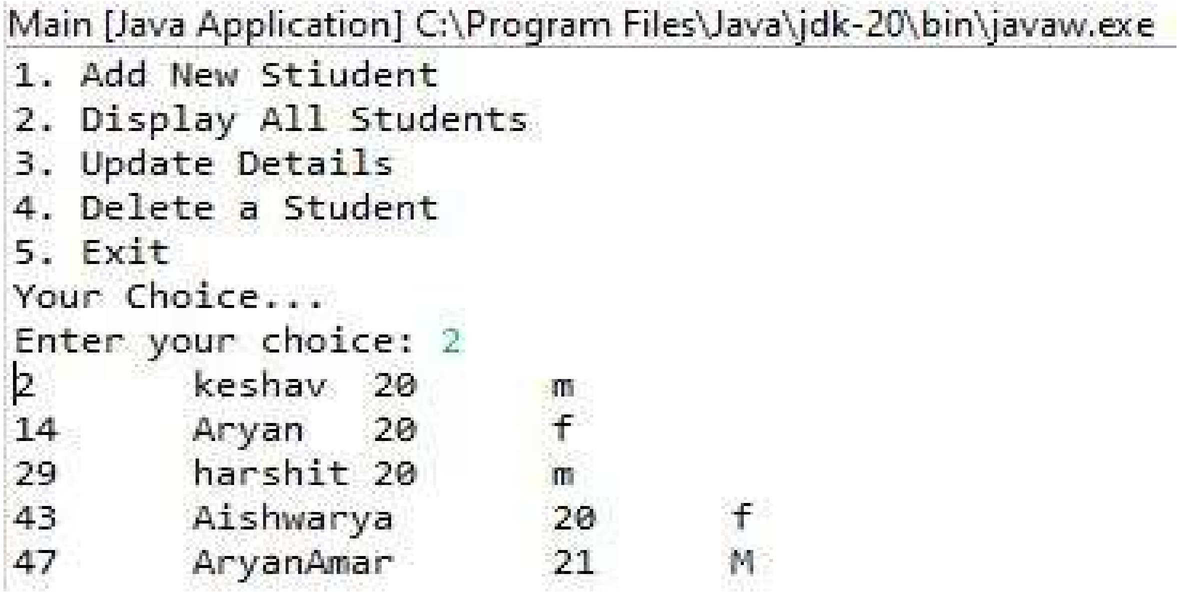
INPUT/OUTPUT

Insert operation :



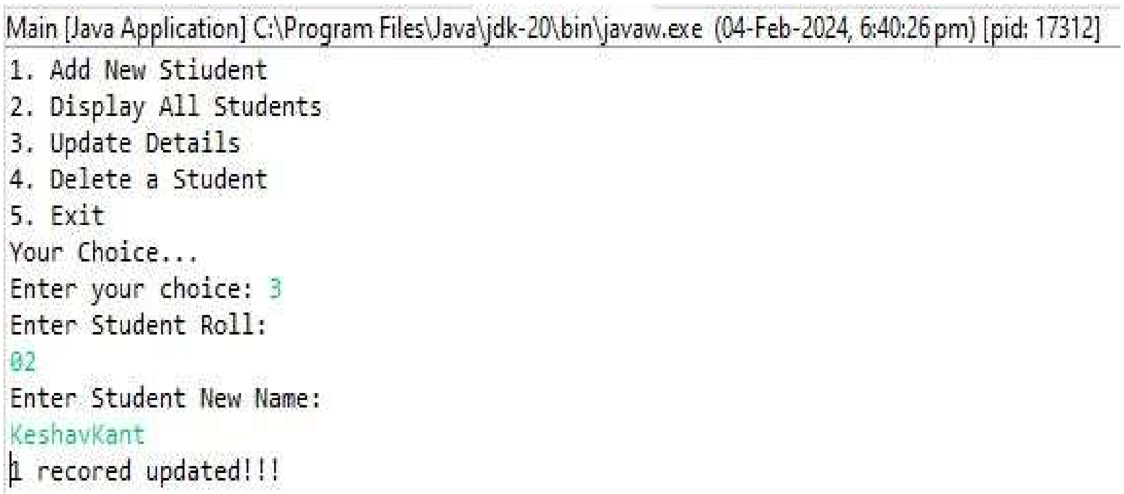
```
Main [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe
1. Add New Stiudent
2. Display All Students
3. Update Details
4. Delete a Student
5. Exit
Your Choice...
Enter your choice: 1
Enter the student Roll Number:
47
Enter the Student Name:
AryanAmar
Enter Student Age:
21
Enter Student Gender:
M
1 record inserted!!!
```

Display operation :



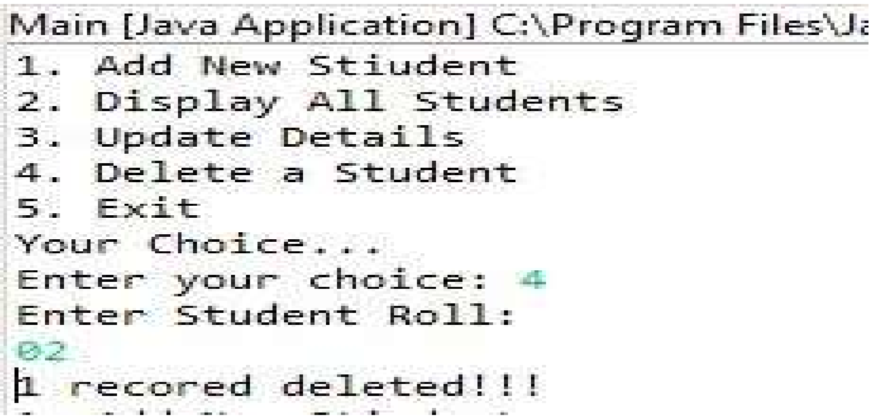
```
Main [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe
1. Add New Stiudent
2. Display All Students
3. Update Details
4. Delete a Student
5. Exit
Your Choice...
Enter your choice: 2
2       keshav  20      m
14      Aryan   20      f
29      harshit 20      m
43      Aishwarya       20      f
47      AryanAmar       21      M
```

Update operation :



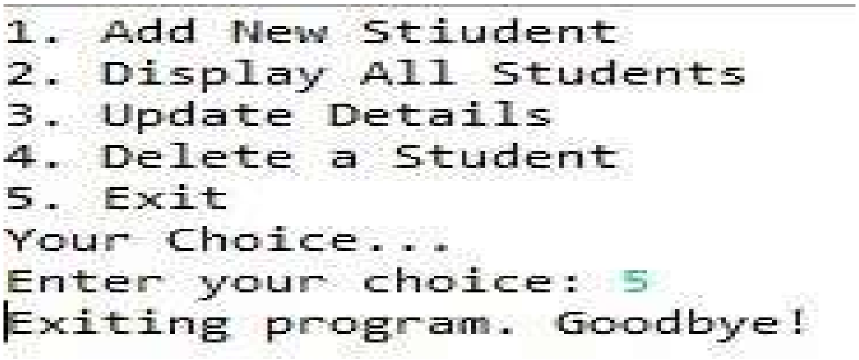
```
Main [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe  (04-Feb-2024, 6:40:26 pm) [pid: 17312]
1. Add New Stiudent
2. Display All Students
3. Update Details
4. Delete a Student
5. Exit
Your Choice...
Enter your choice: 3
Enter Student Roll:
02
Enter Student New Name:
KeshavKant
1 recored updated!!!
```

Delete operation :



```
Main [Java Application] C:\Program Files\Ja
1. Add New Stiudent
2. Display All Students
3. Update Details
4. Delete a Student
5. Exit
Your Choice...
Enter your choice: 4
Enter Student Roll:
02
1 recored deleted!!!
```

Exit operation :



```
1. Add New Stiudent
2. Display All Students
3. Update Details
4. Delete a Student
5. Exit
Your Choice...
Enter your choice: 5
Exiting program. Goodbye!
```