OBE Implementation

Module-1: Programs

Submitted By

Likkith Reddy Chirasani [AP22110010179]
Surya Teja Mannava [AP22110010154]
Abhilash Thota [AP22110010140]
Vignesh Maddela [AP22110010185]
Manoj Chandu [AP22110010085]
Akbar Sherif [AP22110010113]

Introduction to Project

SRM University-AP is adopting the Outcome Based Education (OBE) framework to enhance the quality and effectiveness of its academic programs. As part of this initiative, a dedicated software system is required to manage and maintain academic program details in a structured and efficient manner.

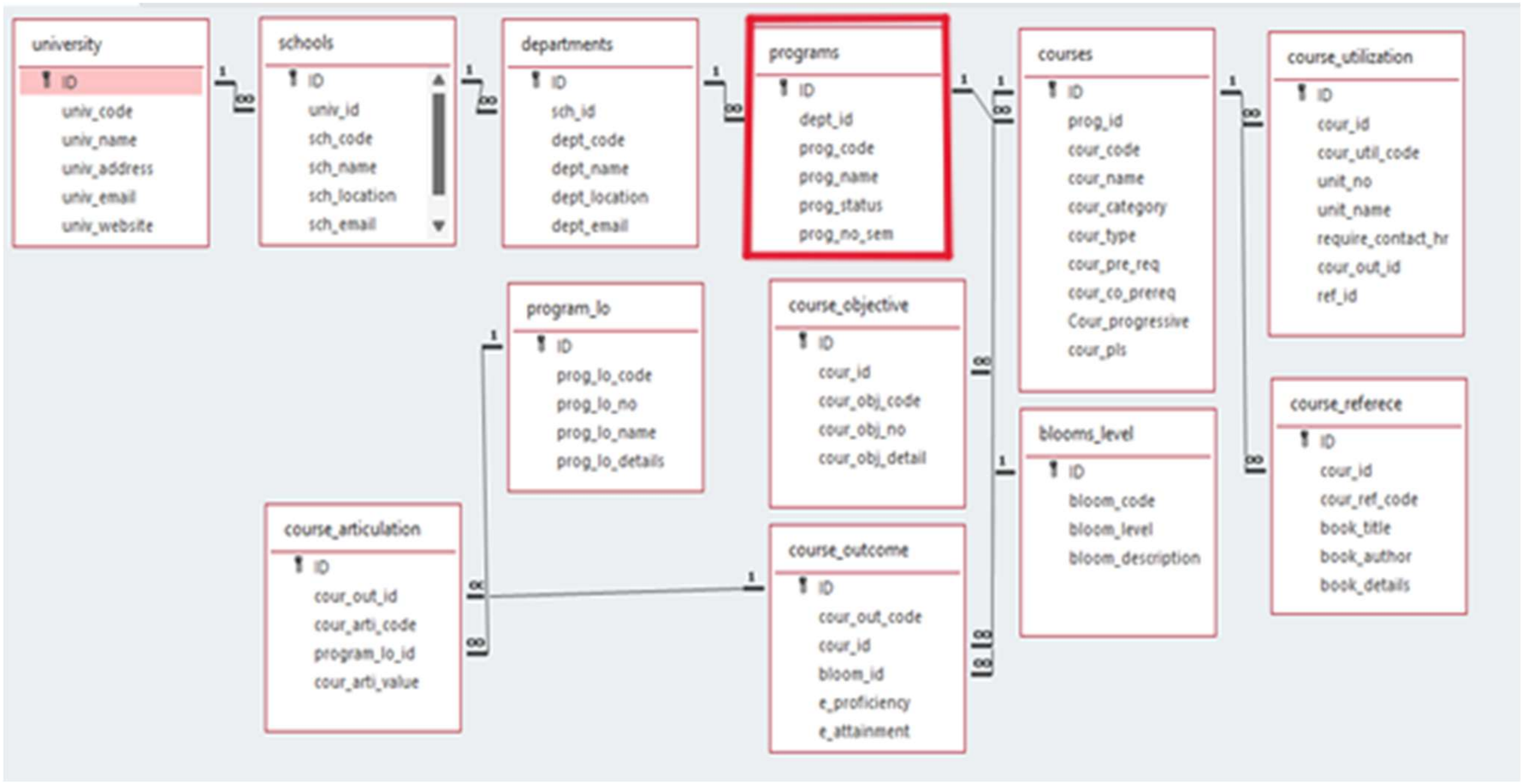
To support this transformation, I have been assigned the task of developing a **CRUD (Create, Retrieve, Update, and Delete)** application using **Java** for the Windows platform and **Android Studio** for mobile devices. This application allows administrators and academic coordinators to efficiently manage the university's program records such as department IDs, program codes, names, statuses, and associated semesters.

The project leverages **Java Swing** for the desktop application with **SQLite** as the backend database to ensure a lightweight and responsive experience. The mobile version developed in **Android Studio** aims to provide similar CRUD functionality on the go, ensuring accessibility and usability across devices.

- By combining a structured approach with modern UI practices, this system lays a foundational step toward digital transformation in line with SRM-AP’s goal of successfully implementing the OBE model.

SRM UNIVERSITY AP
Amaravati

Architecture Diagram



Module Description : University Setting

This module is designed to perform **Create, Update, Retrieve, and Delete (CRUD)** operations on academic program details as part of the Outcome Based Education (OBE) system implementation at SRM University-AP. The module allows users to manage key information such as **Department ID, Program Code, Program Name, Program Status**, and **Number of Semesters** associated with each program.

The application is developed using **Java (Swing)** for the Windows platform and is integrated with a **MySQL database** to ensure reliable and efficient data storage. Users can interact with a user-friendly graphical interface to perform CRUD operations, and all updates made through the application are directly reflected in the MySQL table.

- This module serves as a core component of the larger academic management system, providing a structured and consistent way to handle program-related data and supporting SRM-AP's transition to a more outcome-oriented education model.

University Setting:Field/table details

Field Name	Data type
id	integer
dept_id	integer
prog_code	String
porg_name	String
prog_status	String
prog_no_sem	integer

University Setting:Programming Details

- **File name:** Legends_Programs
- **Function/method name**
  - **Create:**AP22110010140_Programs_create
  - **Update:**AP22110010179_Programs _update
  - **Retrieve:**AP22110010154_Programs _retrive
  - **Delete:**AP22110010185_Programs _delete
  - **GUI**: AP22110010085 and AP22110010113

Sample Source Code

```
Creating a Program:

private void AP22110010140_Programs_create(int deptId, String progCode, String progName, String progStatus, int
     progNoSem) {
        try (PreparedStatement ps = conn.prepareStatement(
                "INSERT INTO programs (dept_id, prog_code, prog_name, prog_status, prog_no_sem) VALUES (?, ?, ?, ?,
     ?)")) {
            ps.setInt(1, deptId);
            ps.setString(2, progCode);
            ps.setString(3, progName);
            ps.setString(4, progStatus);
            ps.setInt(5, progNoSem);
            ps.executeUpdate();
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(frame, "❌ Error inserting program.");
            e.printStackTrace();
        }
    }
```

Sample Source Code

```
Retrieving the data:

private void AP22110010154_Programs_retrieve() {
        model.setRowCount(0);
        try (Statement stmt = conn.createStatement(); ResultSet rs = stmt.executeQuery("SELECT * FROM programs")) {
            while (rs.next()) {
                model.addRow(new Object[]{
                        rs.getInt("ID"), rs.getInt("dept_id"), rs.getString("prog_code"),
                        rs.getString("prog_name"), rs.getString("prog_status"), rs.getInt("prog_no_sem")
                });
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
```

Sample Source Code

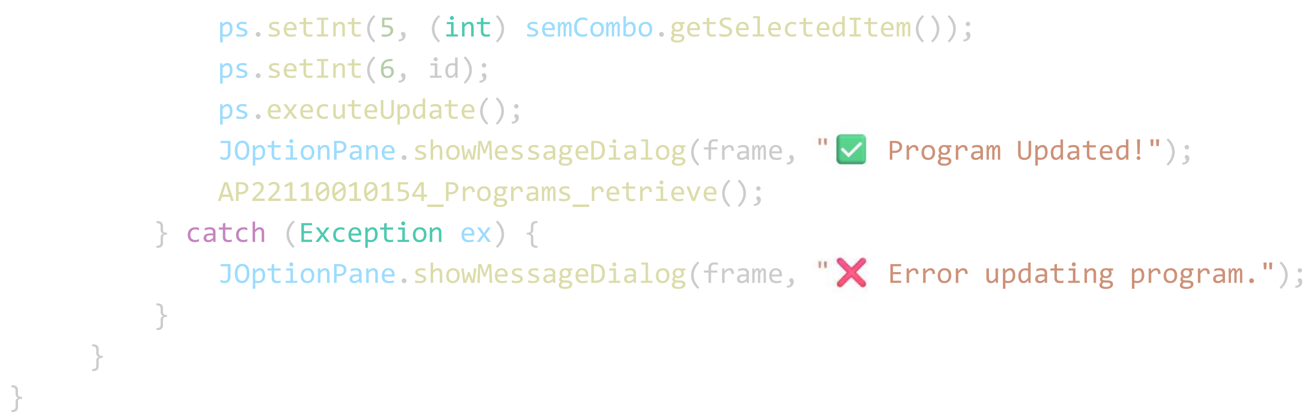
Updating a Program:

```
private void AP22110010179_Programs_update() {
        int row = table.getSelectedRow();
        if (row == -1) {
            JOptionPane.showMessageDialog(frame, "❌ Select a program to update.");
            return;
        }
             int id = (int) table.getValueAt(row, 0);
        JTextField deptField = new JTextField(table.getValueAt(row, 1).toString());
        JTextField codeField = new JTextField((String) table.getValueAt(row, 2));
        JTextField nameField = new JTextField((String) table.getValueAt(row, 3));
        JTextField statusField = new JTextField((String) table.getValueAt(row, 4));
        JComboBox<Integer> semCombo = new JComboBox<>();
        for (int i = 1; i <= 8; i++) semCombo.addItem(i);
        semCombo.setSelectedItem(table.getValueAt(row, 5));
```

Sample Source Code

```
JPanel panel = new JPanel(new GridLayout(6, 2, 5, 5));
        panel.add(new JLabel("Dept ID:")); panel.add(deptField);
        panel.add(new JLabel("Code:")); panel.add(codeField);
        panel.add(new JLabel("Name:")); panel.add(nameField);
        panel.add(new JLabel("Status:")); panel.add(statusField);
        panel.add(new JLabel("Semester:")); panel.add(semCombo);

             int result = JOptionPane.showConfirmDialog(frame, panel, "Update Program",
     JOptionPane.OK_CANCEL_OPTION);

             if (result == JOptionPane.OK_OPTION) {
            try (PreparedStatement ps = conn.prepareStatement(
                    "UPDATE programs SET dept_id=?, prog_code=?, prog_name=?, prog_status=?, prog_no_sem=? WHERE
     ID=?")) {
                ps.setInt(1, Integer.parseInt(deptField.getText().trim()));
                ps.setString(2, codeField.getText().trim());
                ps.setString(3, nameField.getText().trim());
                ps.setString(4, statusField.getText().trim());
```

Sample Source Code



```
                ps.setInt(5, (int) semCombo.getSelectedItem());
                ps.setInt(6, id);
                ps.executeUpdate();
                JOptionPane.showMessageDialog(frame, "✅ Program Updated!");
                AP22110010154_Programs_retrieve();
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(frame, "❌ Error updating program.");
            }
        }
    }
```

Sample Source Code

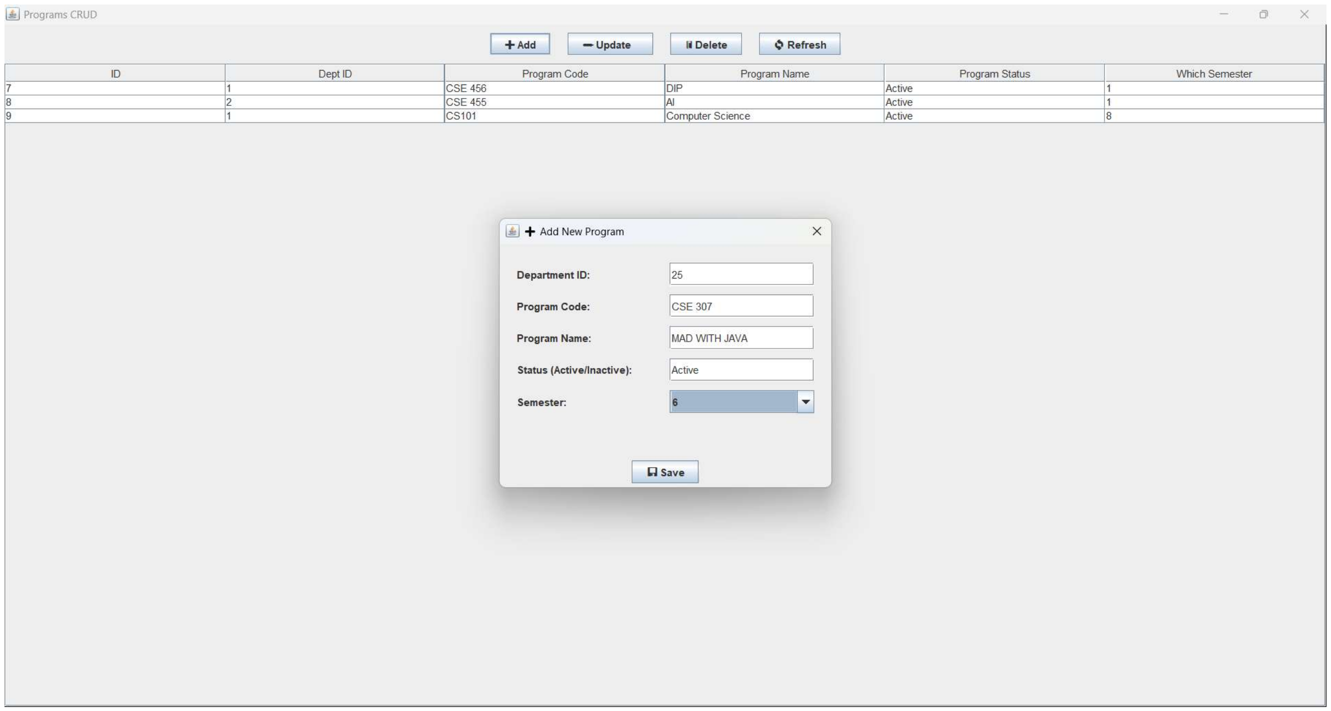
```
Deleting a Program:

private void AP221100101185_Programs_delete() {
        int row = table.getSelectedRow();
        if (row == -1) {
            JOptionPane.showMessageDialog(frame, "❌ Select a row to delete.");
            return;
        }
             int id = (int) table.getValueAt(row, 0);
        int confirm = JOptionPane.showConfirmDialog(frame, "Are you sure?", "Delete Program",
     JOptionPane.YES_NO_OPTION);
        if (confirm == JOptionPane.YES_OPTION) {
            try (PreparedStatement ps = conn.prepareStatement("DELETE FROM programs WHERE ID=?")) {
                ps.setInt(1, id);
                    }
```

Sample Source Code

```
ps.executeUpdate();
                JOptionPane.showMessageDialog(frame, "✅ Program Deleted!");
                AP22110010154_Programs_retrieve();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
```

Sample Screen Shots[*Screenshot of CRUD)]

Adding or creating a Program:


Programs CRUD

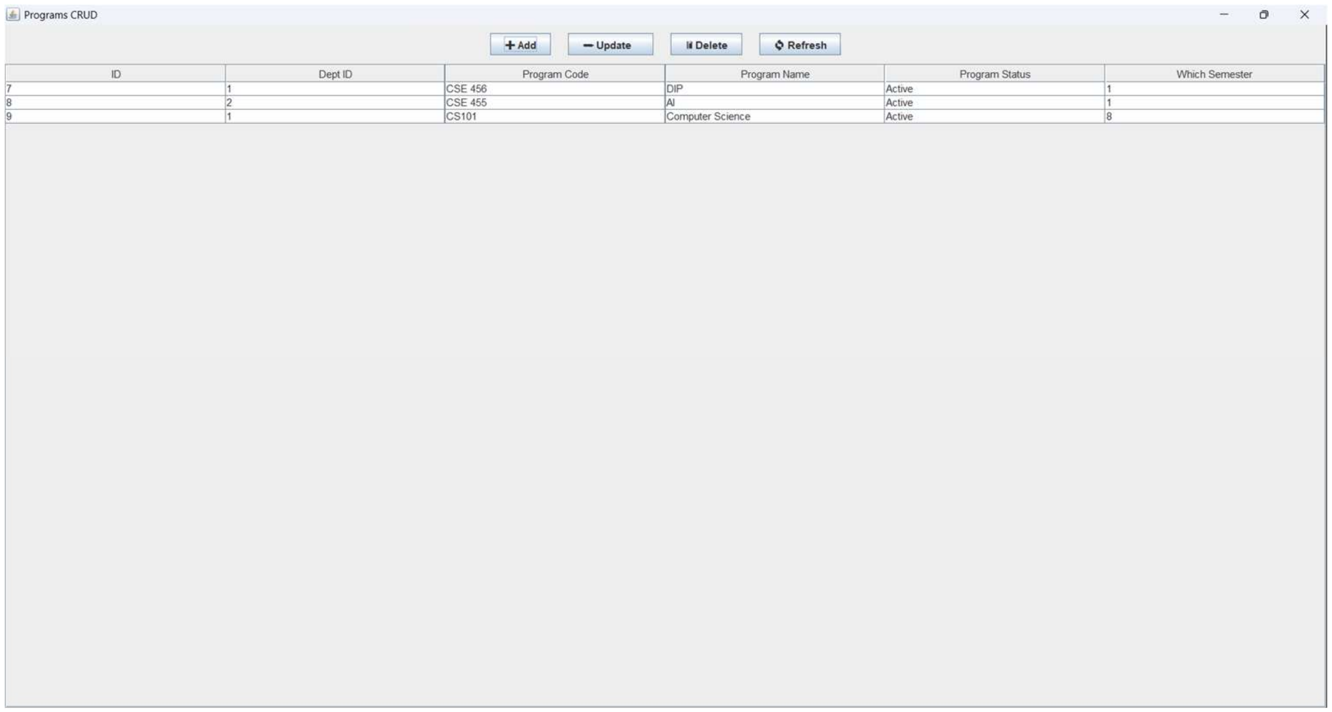
ID	Dept ID	Program Code	Program Name	Program Status	Which Semester
7	1	CSE 456	DIP	Active	1
8	2	CSE 455	AI	Active	1
9	1	CS101	Computer Science	Active	8

+ Add New Program

Department ID:	25
Program Code:	CSE 307
Program Name:	MAD WITH JAVA
Status (Active/Inactive):	Active
Semester:	6

Sample Screen Shots[*Screenshot of CRUD)]

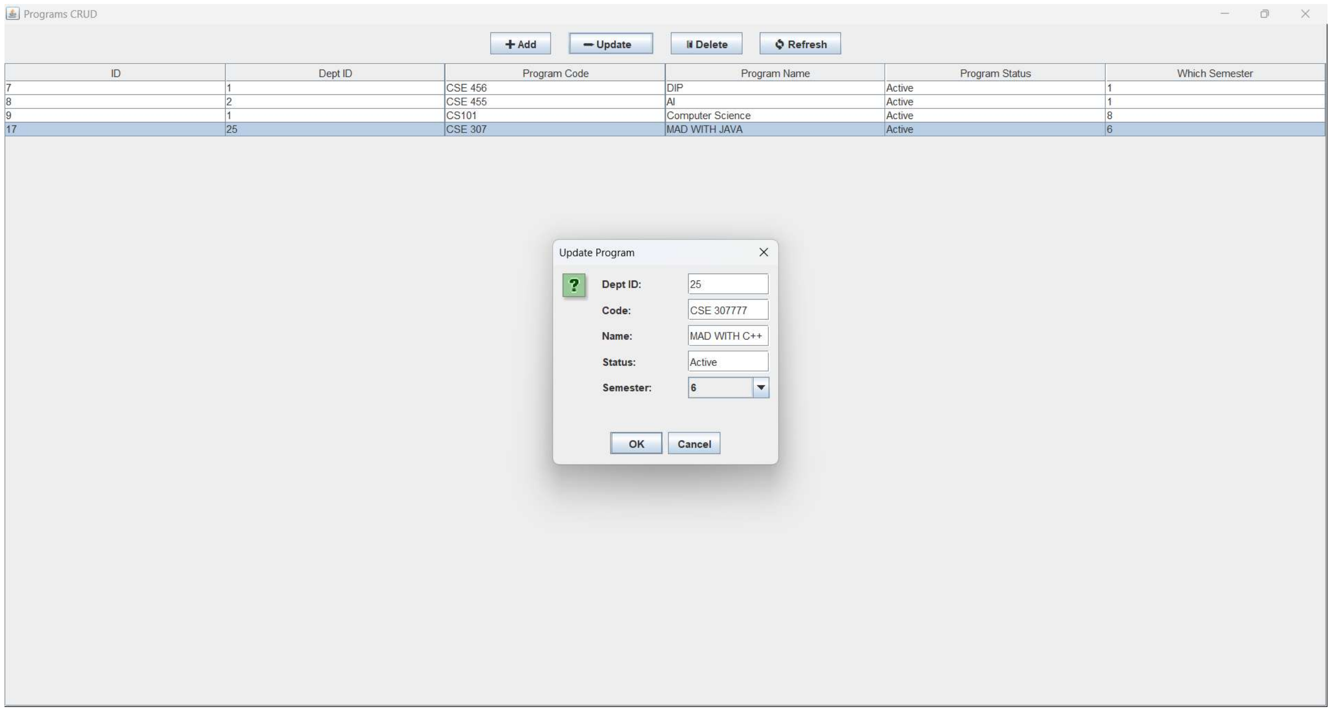
Retrieving all the Programs data from database:



ID	Dept ID	Program Code	Program Name	Program Status	Which Semester
7	1	CSE 456	DIP	Active	1
8	2	CSE 455	AI	Active	1
9	1	CS101	Computer Science	Active	8

Sample Screen Shots[*Screenshot of CRUD)]

Updating a Program:

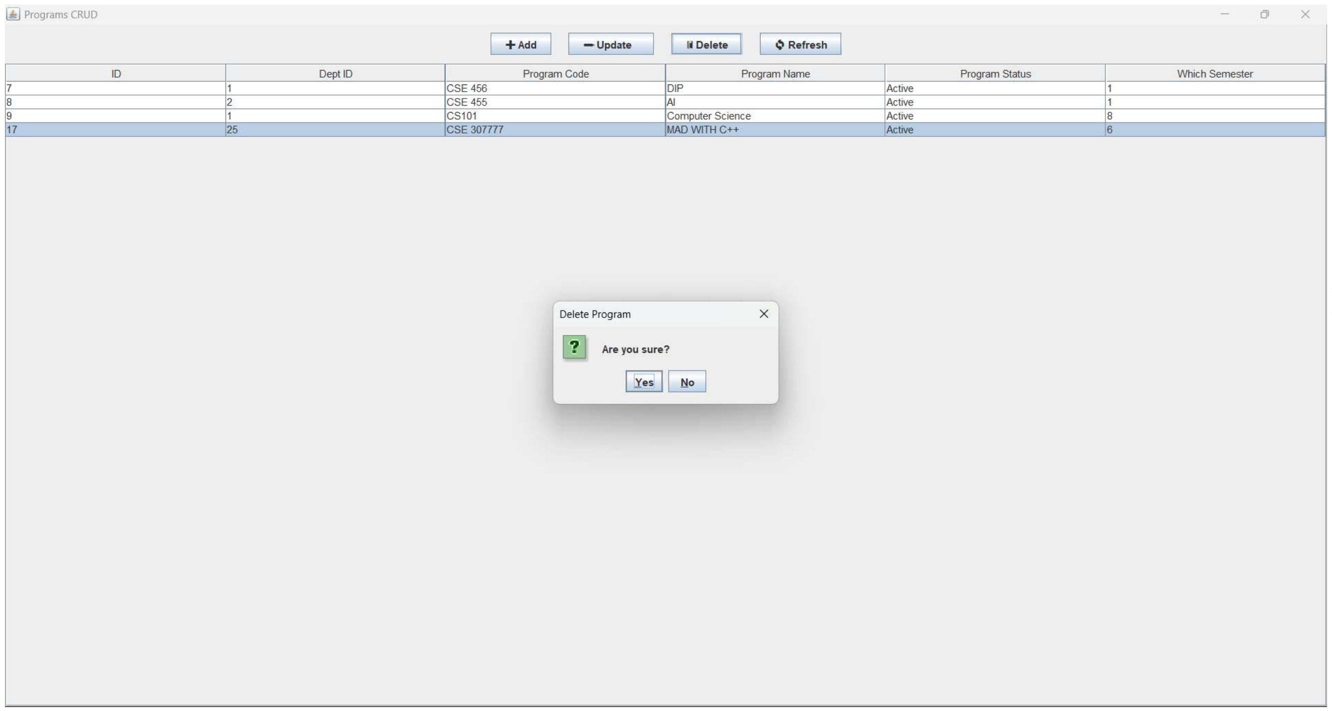


ID	Dept ID	Program Code	Program Name	Program Status	Which Semester
7	1	CSE 456	DIP	Active	1
8	2	CSE 455	AI	Active	1
9	1	CS101	Computer Science	Active	8
17	25	CSE 307	MAD WITH JAVA	Active	6

Dept ID:	25
Code:	CSE 307777
Name:	MAD WITH C++
Status:	Active
Semester:	6

Sample Screen Shots[*Screenshot of CRUD)]

Deleting a Program:



ID	Dept ID	Program Code	Program Name	Program Status	Which Semester
7	1	CSE 456	DIP	Active	1
8	2	CSE 455	AI	Active	1
9	1	CS101	Computer Science	Active	8
17	25	CSE 307777	MAD WITH C++	Active	6

);

Conclusion

The *Programs CRUD Management System* has been successfully developed as a robust and user-friendly desktop application using Java Swing and SQLite. It enables efficient management of academic programs by providing core functionalities such as creating, reading, updating, and deleting program records.

- This project demonstrates practical implementation of GUI design, database connectivity using JDBC, and structured modular coding practices. By integrating intuitive user interfaces with backend operations, the system offers a seamless experience for administrators to manage program data with accuracy and ease.

SRM UNIVERSITY AP Amaravati

Thank You