OBE IMPLEMENTATION: UNIVERSITY SETTING

by

R. Sri Pardhu - [AP22110010372]

Y. Santhosh - [AP22110010388]

R. Meghana - [AP22110010347]

M. Mose - [AP22110011441]

A report for the CS307: Mobile Application Development using JAVA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SRM UNIVERSITY AP :: AMARAVATI

INDEX

Introduction **2**

Project Modules: 3

Architecture Diagram **3**

Module Description **4**

Programming Details naming conventions to be used 5

Table details 5

Source Code **5**

Screen Shots **6**

Conclusion **7**

Introduction

Our University (herewith considered as SRM-AP) is going to implement OBE(Outcome Based Education) in their university and you are assigned in the project to develop a CURD(Create,Update,Retrieve and Delete) windows and mobile application using JAVA programming and Android studio for the same.

Project Modules:

Various Modules available in the project are

1.Blooms Level setting

2.Program Level Objective Setting

3.University

4.Schools

5.Department

6.Programs

7.Courses

8.Course objective setting

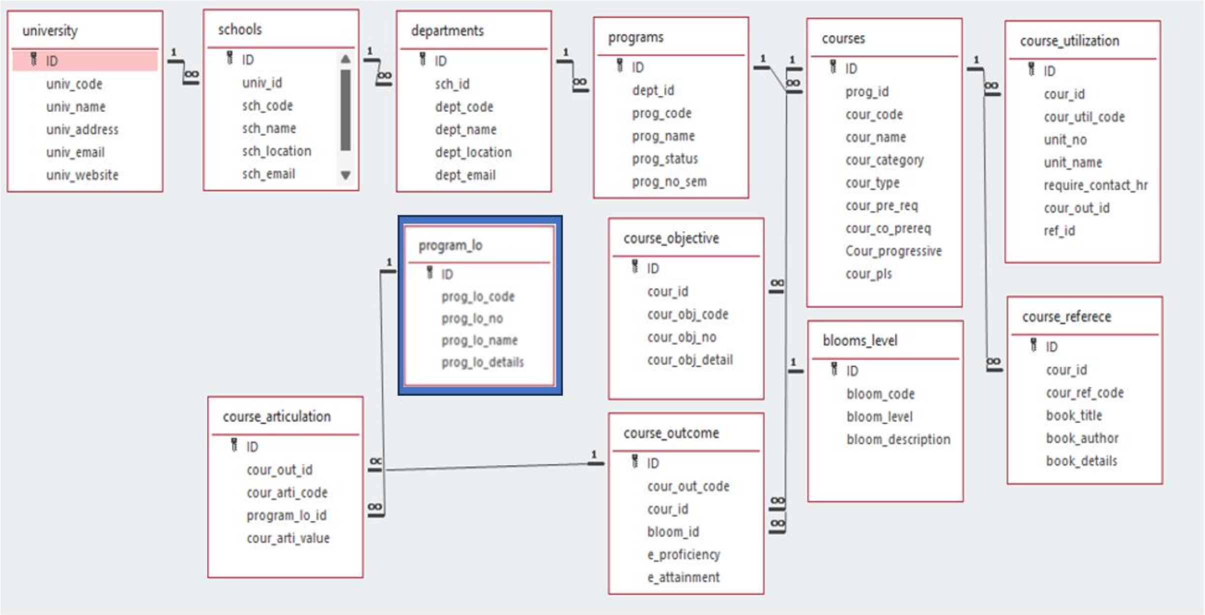
9.Course Outcome Setting

10.Course Articulation matrix Setting

11.Course Utilization Setting

12.Course Reference Setting.

Architecture Diagram



Module Description

Module Name: Programlo
Module Description:

This module is used to create,Update,Retrieve,Delete(hereafter known as CURD) details of the module and storing the details in the database table(eg.MySQL).

Programming Details naming conventions to be used:

- **class name/activity name:**Programlo
- **Function/method name**
  - **Create:**AP22110010372_programlo_create
  - **Update:**AP22110010372_programlo_update
  - **Retrieve:**AP22110010372_programlo_retrive
  - **Delete:**AP22110010372_programlo_delete

Table details:

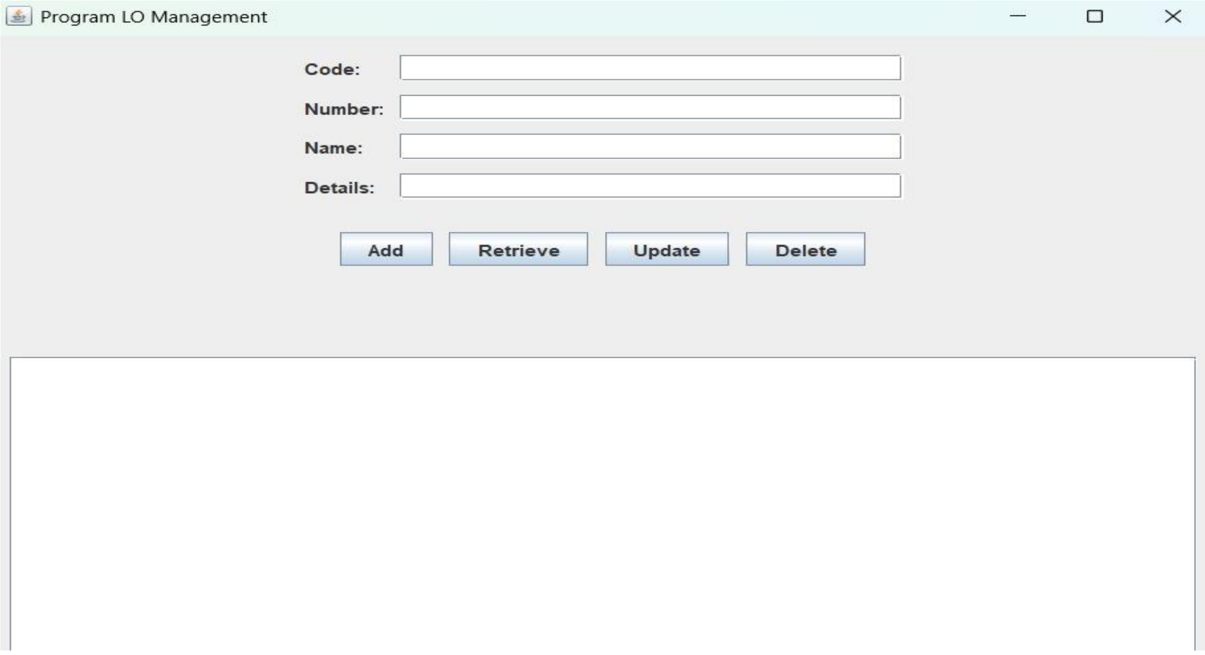
Field Name	**Data type**
ID	integer
proglo_code	String
proglo_number	integer
proglo_name	String
proglo_details	String

Source Code

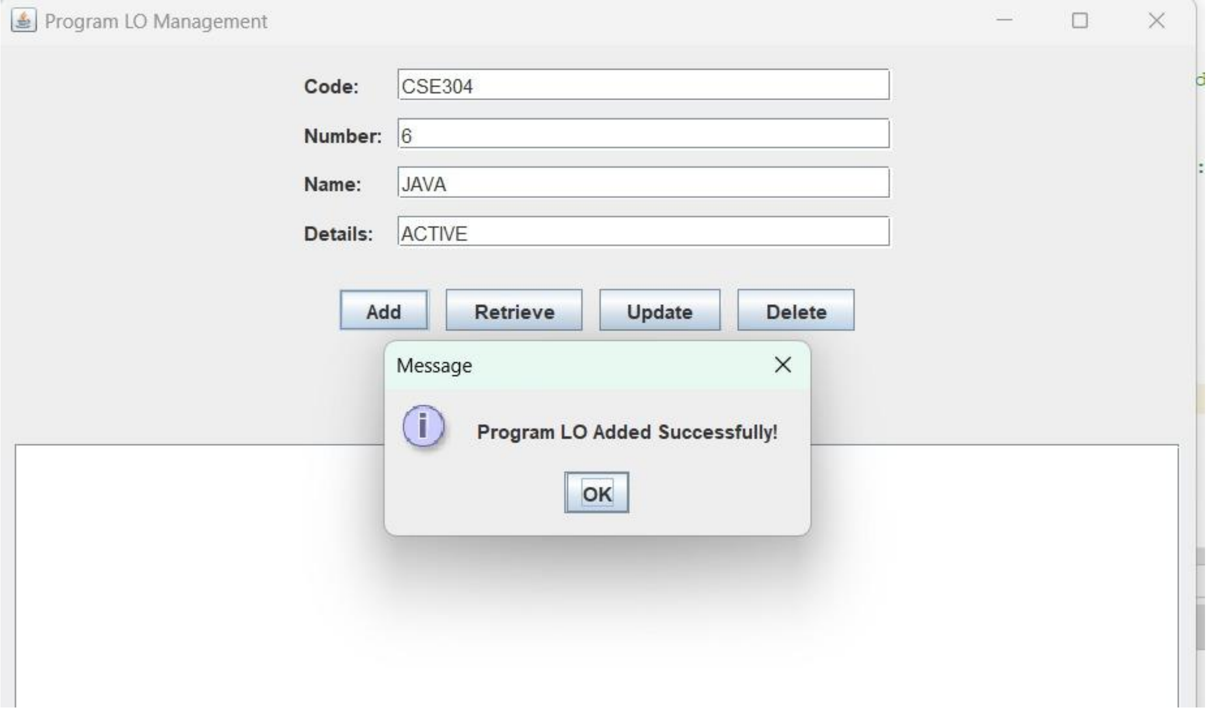
```
package program_lo;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;

public class Program_lo extends JFrame {
    private static final String URL = 
"jdbc:sqlite:C:\\Users\\sripa\\OneDrive\\Desktop\\Apps\\javaapp.db";

    private JTextField txtCode, txtNo, txtName, txtDetails;
    private JTextArea txtDisplay;
    private JButton btnAdd, btnRetrieve, btnUpdate, btnDelete;

    public Program_lo() {
        setTitle("Program LO Management");
        setSize(750, 550);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        JPanel mainPanel = new JPanel(new BorderLayout(10, 10));
        mainPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

        JPanel inputPanel = new JPanel(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(5, 5, 5, 5);
        gbc.fill = GridBagConstraints.HORIZONTAL;

        gbc.gridx = 0;
        gbc.gridy = 0;
        inputPanel.add(new JLabel("Code:"), gbc);
        gbc.gridx = 1;
        txtCode = new JTextField(15);
        inputPanel.add(txtCode, gbc);

        gbc.gridx = 0;
        gbc.gridy = 1;
        inputPanel.add(new JLabel("Number:"), gbc);
        gbc.gridx = 1;
        txtNo = new JTextField(10);
        inputPanel.add(txtNo, gbc);
        gbc.gridx = 0;
        gbc.gridy = 2;
        inputPanel.add(new JLabel("Name:"), gbc);
        gbc.gridx = 1;
        txtName = new JTextField(20);
        inputPanel.add(txtName, gbc);

        gbc.gridx = 0;
        gbc.gridy = 3;
        inputPanel.add(new JLabel("Details:"), gbc);
        gbc.gridx = 1;
        txtDetails = new JTextField(30);
        inputPanel.add(txtDetails, gbc);

        JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER,
10, 10));
        btnAdd = new JButton("Add");
        btnRetrieve = new JButton("Retrieve");
        btnUpdate = new JButton("Update");
        btnDelete = new JButton("Delete");

        buttonPanel.add(btnAdd);
        buttonPanel.add(btnRetrieve);
        buttonPanel.add(btnUpdate);
        buttonPanel.add(btnDelete);

        txtDisplay = new JTextArea(15, 60);
        txtDisplay.setEditable(false);
        txtDisplay.setFont(new Font("Monospaced", Font.PLAIN, 12));
        JScrollPane scrollPane = new JScrollPane(txtDisplay);

        mainPanel.add(inputPanel, BorderLayout.NORTH);
        mainPanel.add(buttonPanel, BorderLayout.CENTER);
        mainPanel.add(scrollPane, BorderLayout.SOUTH);

        add(mainPanel);

        btnAdd.addActionListener(e -> AP22110010372_programlo_create());
        btnRetrieve.addActionListener(e ->
AP22110010372_programlo_Retrieve());
        btnUpdate.addActionListener(e -> AP22110010372_programlo_Update());
        btnDelete.addActionListener(e -> AP22110010372_programlo_Delete());

        setVisible(true);
    }
    private Connection connect() {
        try {
            Class.forName("org.sqlite.JDBC");
            return DriverManager.getConnection(URL);
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "Database Connection Error: " +
e.getMessage());
            return null;
        }
    }

    private void clearFields() {
        txtCode.setText("");
        txtNo.setText("");
        txtName.setText("");
        txtDetails.setText("");
    }

    private void AP22110010372_programlo_create() {
        String sql = "INSERT INTO program_lo (prog_lo_code, prog_lo_no,
prog_lo_name, prog_lo_details) VALUES (?, ?, ?, ?)";
        try (Connection conn = connect(); PreparedStatement pstmt =
conn.prepareStatement(sql)) {
            pstmt.setString(1, txtCode.getText());
            pstmt.setString(2, txtNo.getText());
            pstmt.setString(3, txtName.getText());
            pstmt.setString(4, txtDetails.getText());
            pstmt.executeUpdate();
            JOptionPane.showMessageDialog(this, "Program LO Added
Successfully!");
            clearFields();
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Insert Error: " +
e.getMessage());
        }
    }

    private void AP22110010372_programlo_Retrieve() {
        String sql = "SELECT * FROM program_lo";
        txtDisplay.setText("");
        try (Connection conn = connect(); Statement stmt =
conn.createStatement(); ResultSet rs = stmt.executeQuery(sql)) {
            while (rs.next()) {
                txtDisplay.append("ID: " + rs.getInt("ID") +
                        ", Code: " + rs.getString("prog_lo_code") +
                        ", No: " + rs.getString("prog_lo_no") +
                        ", Name: " + rs.getString("prog_lo_name") +
                        ", Details: " + rs.getString("prog_lo_details") + "\n");
            }
            clearFields();
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Fetch Error: " +
e.getMessage());
        }
    }

    private void AP22110010372_programlo_Update() {
        if (txtCode.getText().trim().isEmpty()) {
            JOptionPane.showMessageDialog(this, "Please enter Code");
            return;
        }

        String sql = "UPDATE program_lo SET prog_lo_no = ?, prog_lo_name =
?, prog_lo_details = ? WHERE prog_lo_code = ?";
        try (Connection conn = connect(); PreparedStatement pstmt =
conn.prepareStatement(sql)) {
            pstmt.setString(1, txtNo.getText());
            pstmt.setString(2, txtName.getText());
            pstmt.setString(3, txtDetails.getText());
            pstmt.setString(4, txtCode.getText());

            int rowsAffected = pstmt.executeUpdate();
            if (rowsAffected > 0) {
                JOptionPane.showMessageDialog(this, "Program LO Updated
Successfully!");
                clearFields();
            } else {
                JOptionPane.showMessageDialog(this, "No record found with Code:
" + txtCode.getText());
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Update Error: " +
e.getMessage());
        }
    }

    private void AP22110010372_programlo_Delete() {
        if (txtCode.getText().trim().isEmpty()) {
            JOptionPane.showMessageDialog(this, "Please enter Code");
            return;
        }

        int confirm = JOptionPane.showConfirmDialog(
                this,
                "Are you sure you want to delete Program LO with Code: " +
txtCode.getText() + "?",
                "Confirm Delete",
                JOptionPane.YES_NO_OPTION
        );
        if (confirm != JOptionPane.YES_OPTION) return;

        String sql = "DELETE FROM program_lo WHERE prog_lo_code = ?";
        try (Connection conn = connect(); PreparedStatement pstmt =
conn.prepareStatement(sql)) {
            pstmt.setString(1, txtCode.getText());
            int rowsAffected = pstmt.executeUpdate();
            if (rowsAffected > 0) {
                JOptionPane.showMessageDialog(this, "Program LO Deleted
Successfully!");
                clearFields();
            } else {
                JOptionPane.showMessageDialog(this, "No record found with Code:
" + txtCode.getText());
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Delete Error: " +
e.getMessage());
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(Program_lo::new);
    }
}
```

Screen shots

Initially:



Create (insert/add):



Code: CSE304

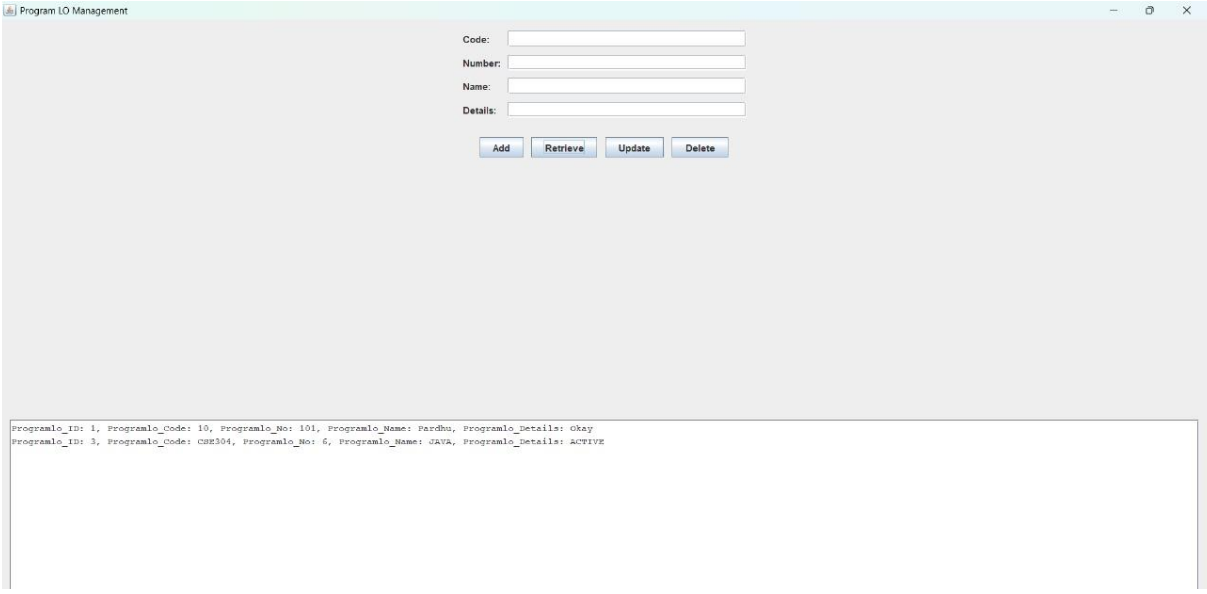
Number: 6

Name: JAVA

Details: ACTIVE

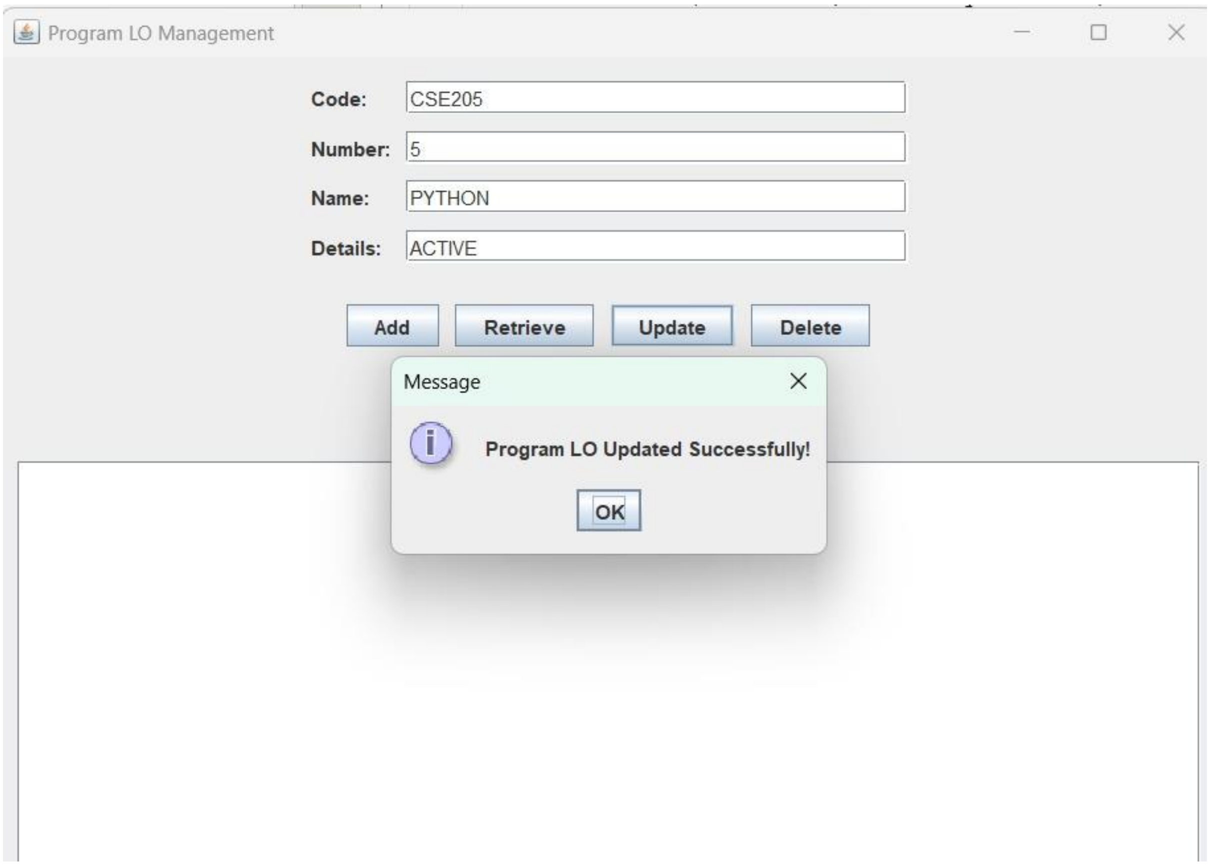
Program LO Added Successfully!

Retrieve:



```
Programlo_ID: 1, Programlo_Code: 10, Programlo_No: 101, Programlo_Name: Pardhu, Programlo_Details: Okay
Programlo_ID: 3, Programlo_Code: CSE304, Programlo_No: 6, Programlo_Name: JAVA, Programlo_Details: ACTIVE
```

Update:



Code: CSE205

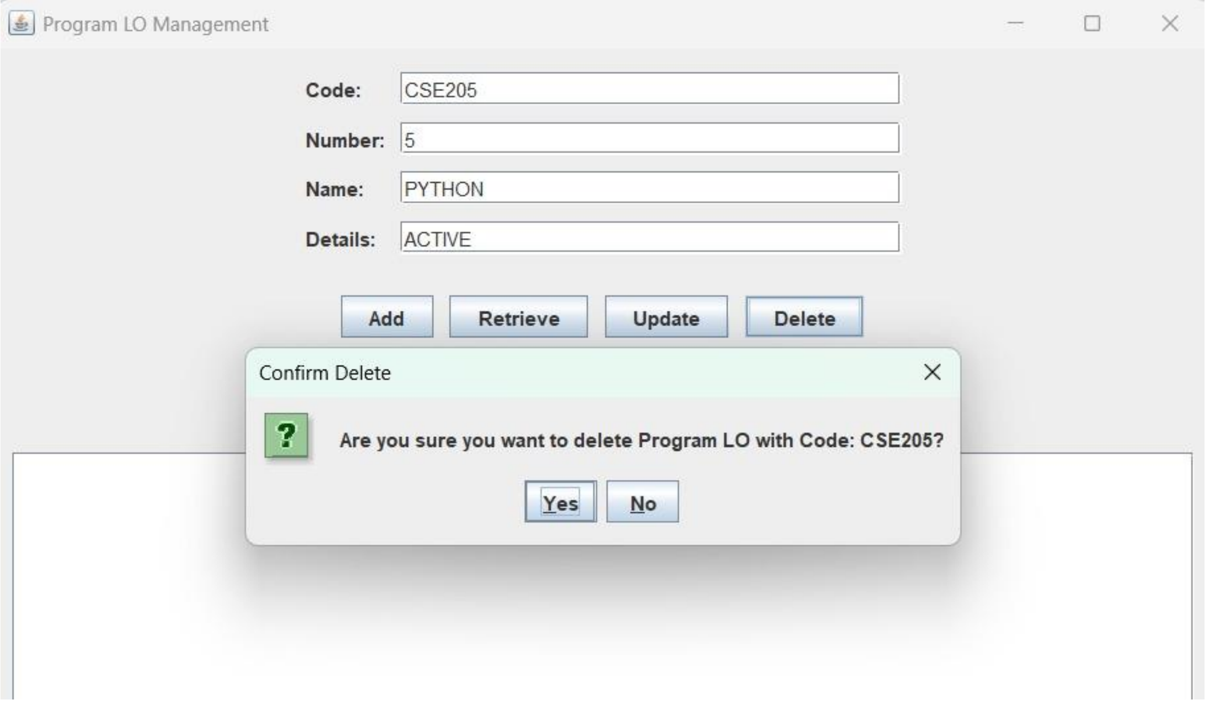
Number: 5

Name: PYTHON

Details: ACTIVE

Program LO Updated Successfully!

Delete:



Program LO Management

Code: CSE205

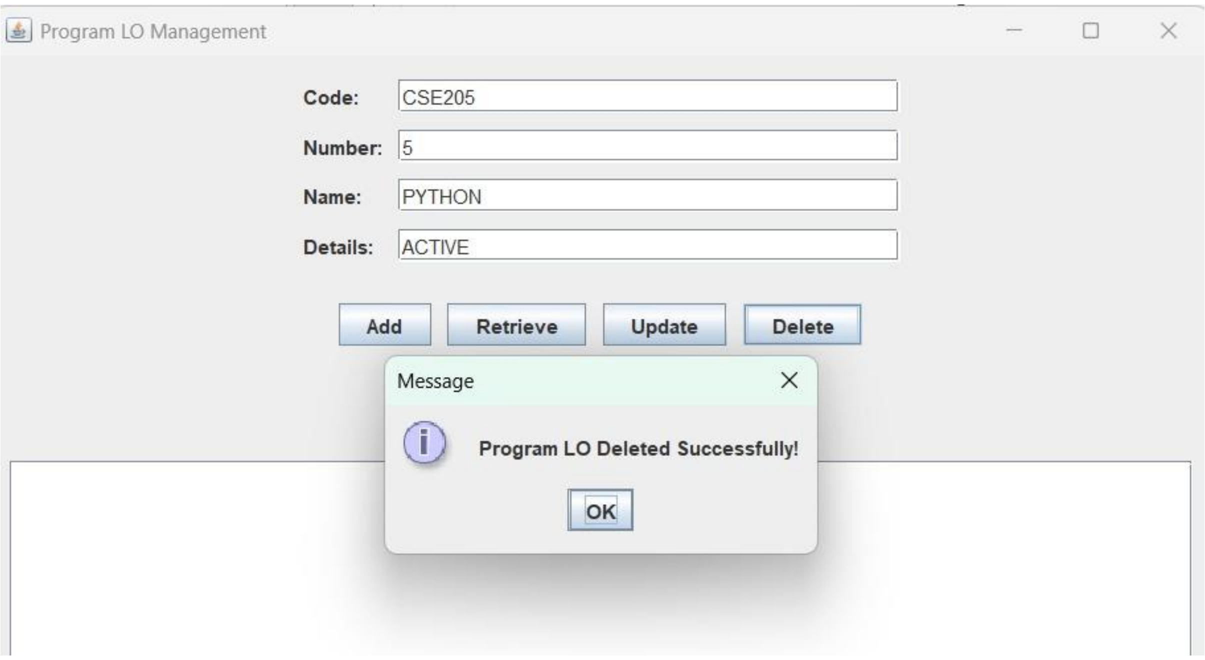
Number: 5

Name: PYTHON

Details: ACTIVE

Confirm Delete

Are you sure you want to delete Program LO with Code: CSE205?



Program LO Management

Code: CSE205

Number: 5

Name: PYTHON

Details: ACTIVE

Message

Program LO Deleted Successfully!

Conclusion

This project successfully implements CRUD (Create, Read, Update, Delete) operations in a Java Swing application using SQLite. The program allows users to manage program records by adding, viewing, updating, and deleting data through a simple graphical interface.

The java file combines both database handling and GUI design, making it a compact and efficient solution. The use of JTable ensures that users can easily view and modify records. Overall, this project provides a basic yet functional database management system that can be improved with additional features like search, validation, and enhanced UI in the future.