**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**](#_h1yjool1sr53)

[Project Modules: 3](#_cefhhkyf9g38)

[**Architecture Diagram 3**](#_jzd73axxtfz9)

[**Module Description 4**](#_ex6roxy7xo5e)

[Programming Details naming conventions to be used: 5](#_2f7xyttubwzj)

[Table details:(eg university)[you consider you module ] 5](#_43k63vpbidp5)

[**Source Code 5**](#_f0tt1ko3he94)

[**Screen Shots 6**](#_wnrc3pboqjjv)

[**Conclusion 7**](#_4i4wvhnin50l)

# 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

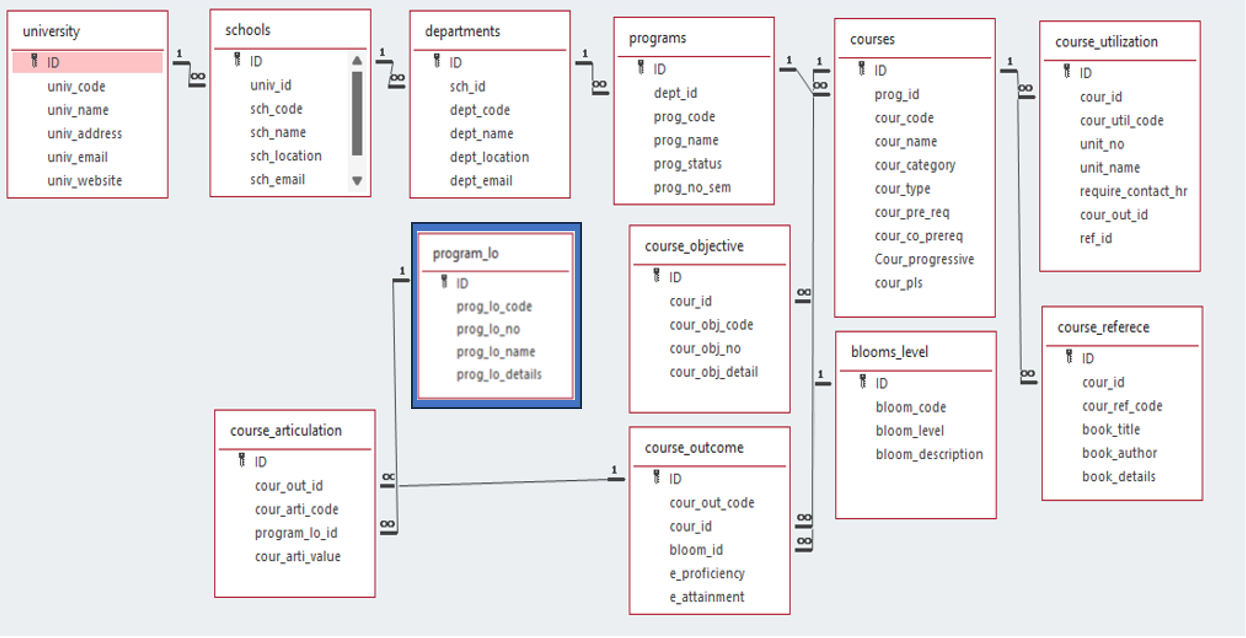
10.Course Articulation matrix Setting

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) {

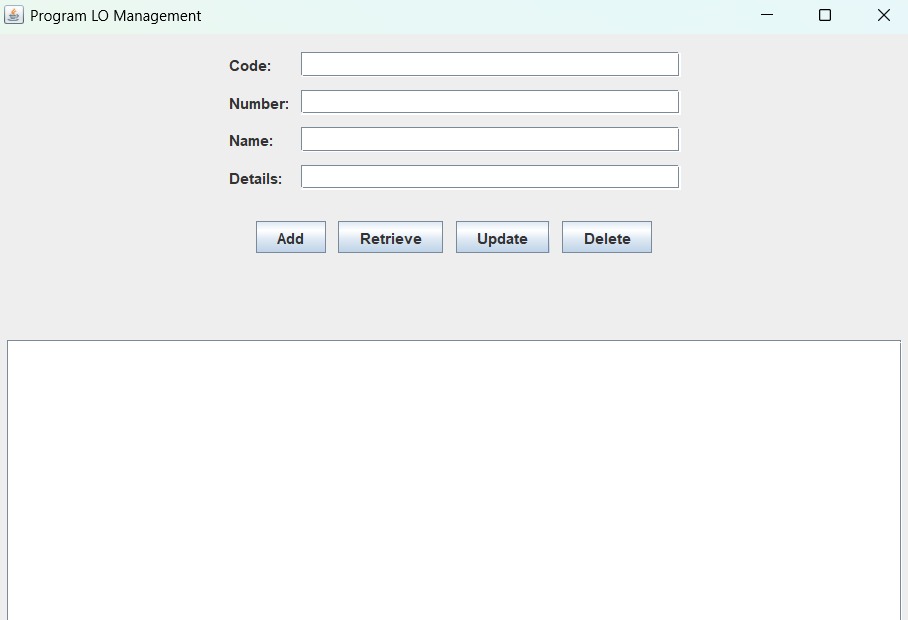
SwingUtilities.invokeLater(Program\_lo::new);

}

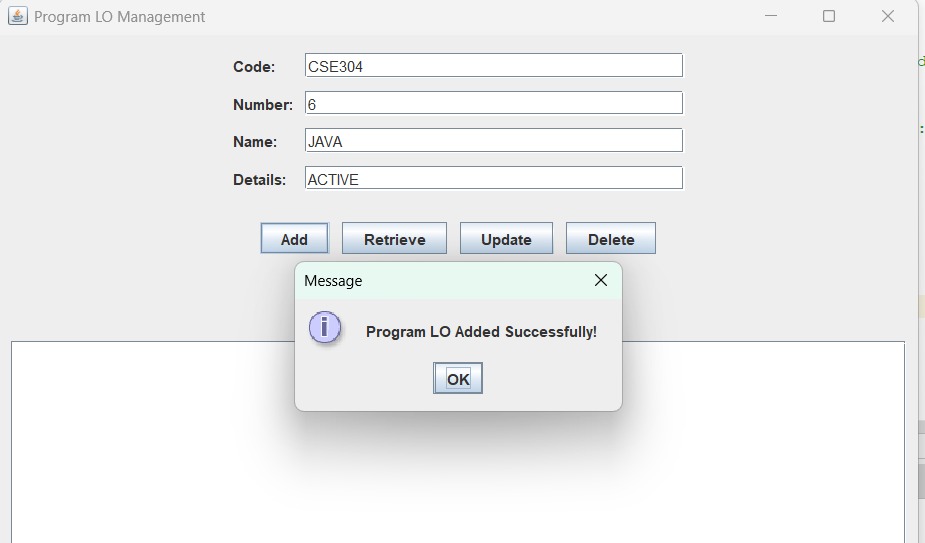
}

# Screen Shots

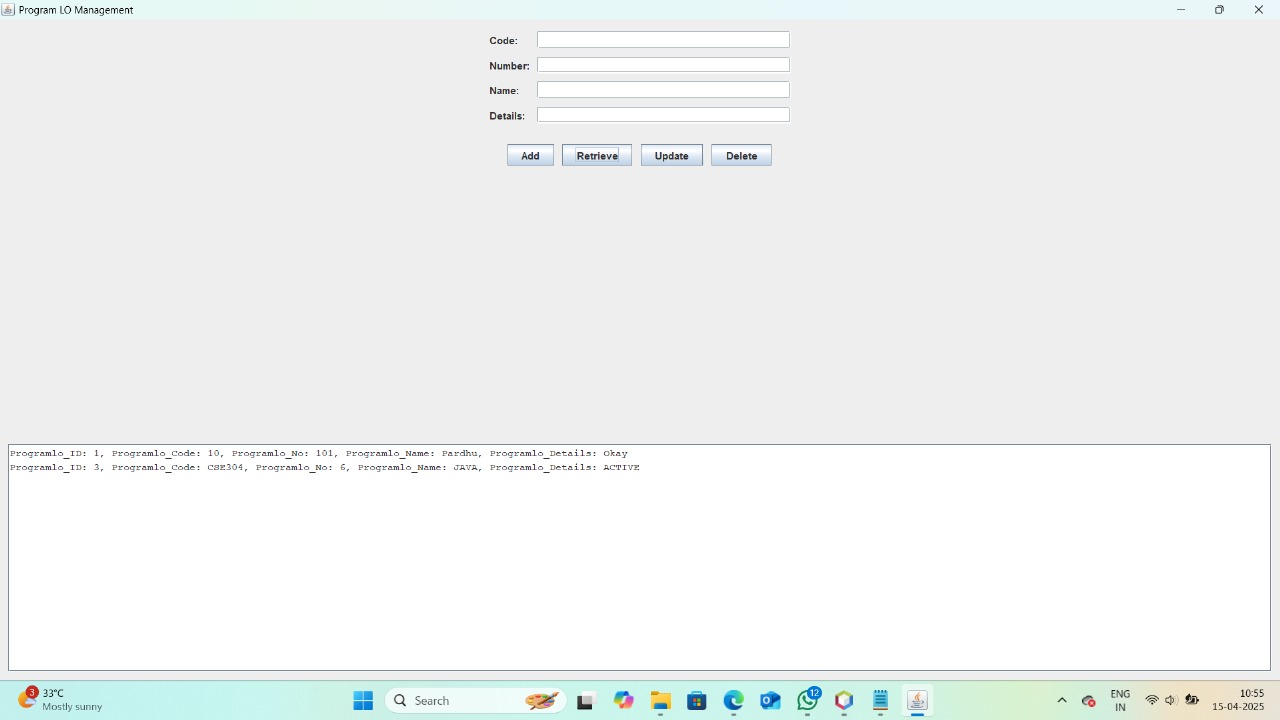
**Initially:**



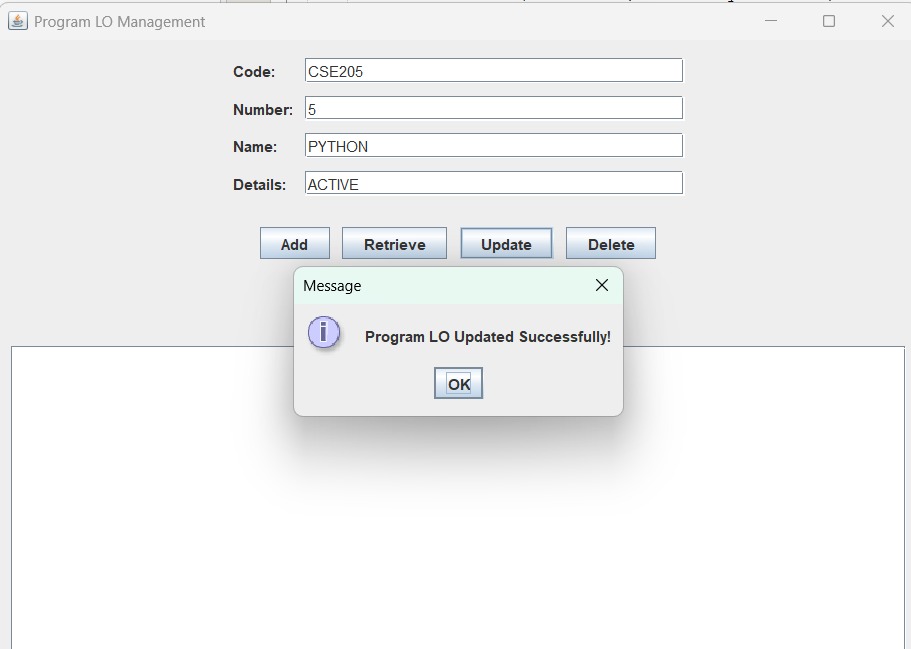
**Create (insert/add):**



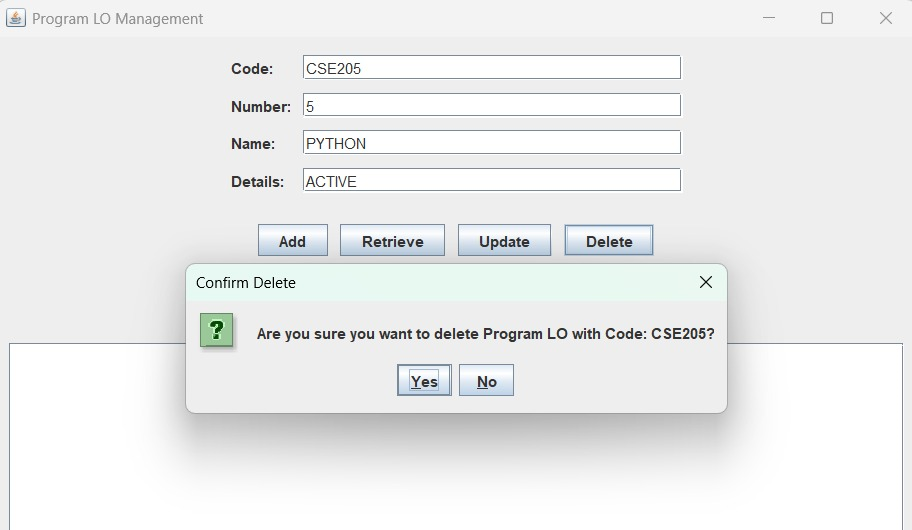
**Retrieve:**

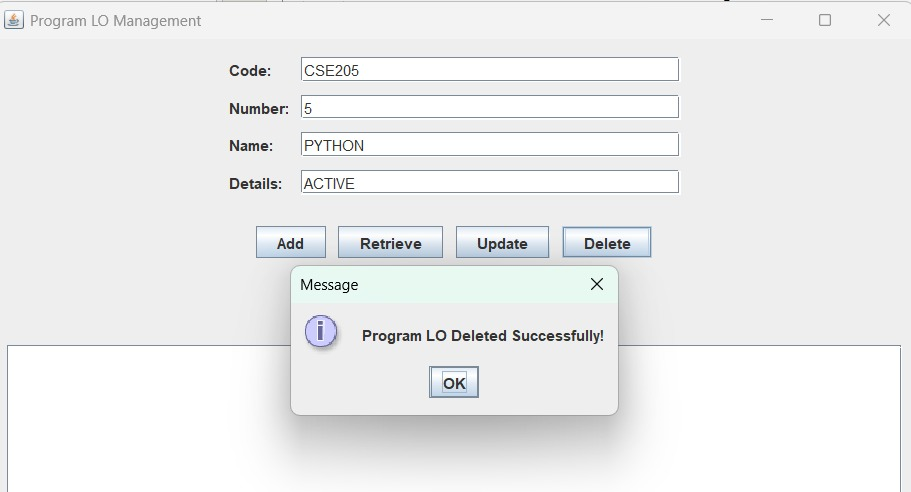


**Update:**



**Delete:**





# 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.