OBE IMPLEMENTATION:UNIVERSITY SETTING

by

Devasish Viswanadh Kolla[AP22110010310]
Nishanth Bhuvaneswer P[AP22110010321]
Rukmini Bhandara Myla[AP22110010315]
Samba Shiva Rao T[AP22110010324]
Avinash M[AP22110010291]

A report for the CS307:Mobile Application Development using JAVA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SRM UNIVERSITY AP::AMARAVATI

INDEX

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:

1.Blooms Level setting

2.Program Level Objective Setting

3.University

4.Schools

5.Department

6.Programs

7.Courses

8.Course objective setting

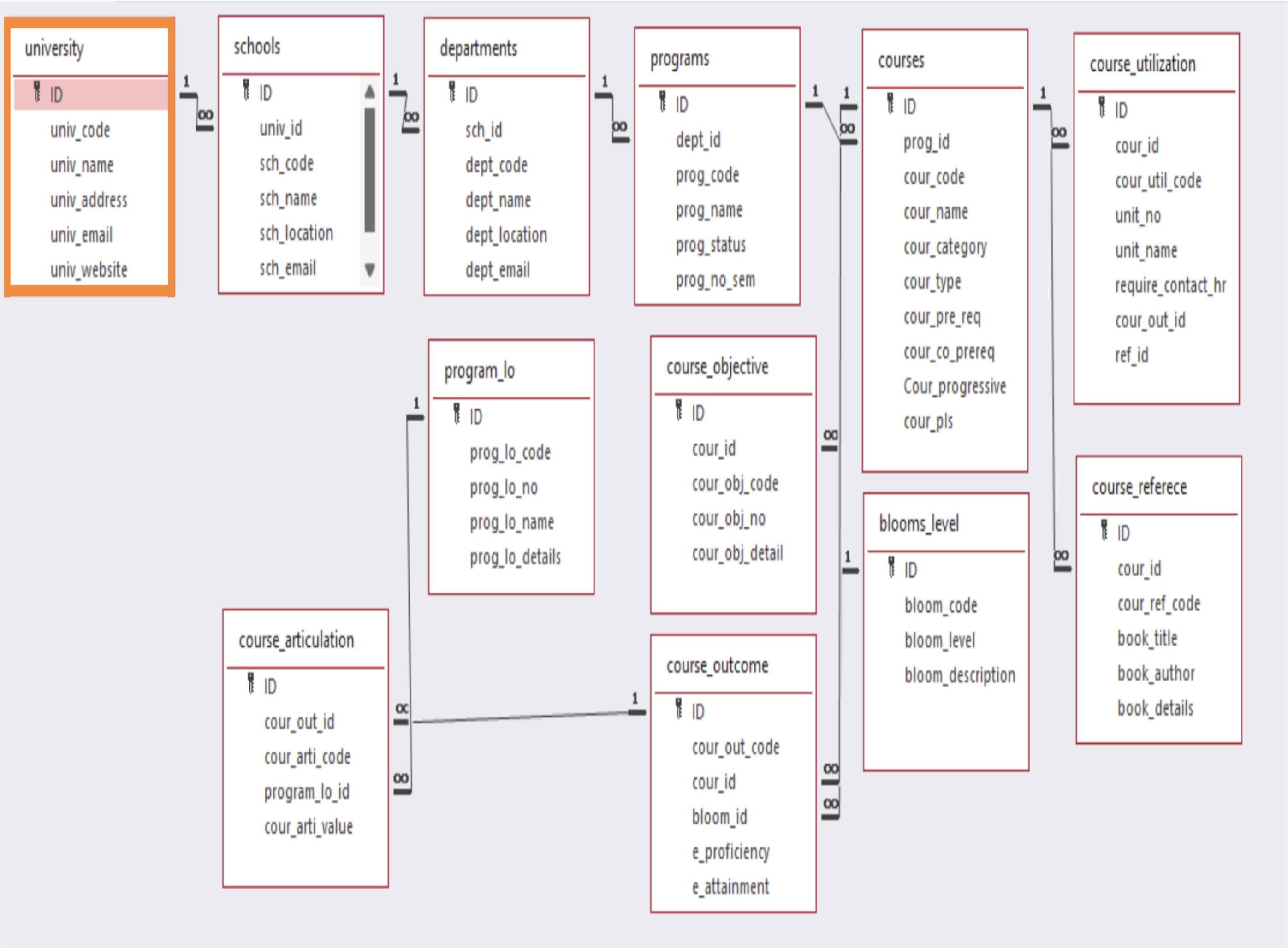
9.Course Outcome Setting

10.Course Articulation matrix Setting

11.Course Utilization Setting

12.Course Reference Setting.

Architecture Diagram



Module Description

Module Name: University

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:**MeenaRashi_University
- **Function/method name**
    - **Create:**AP22110010310_University_create
    - **Update:**AP22110010310_Universityupdate
    - **Retrieve:**AP22110010310_University_retrive
    - **Delete:**AP22110010310_University_delete

Table details:

Field Name	**Data type**
id	integer
univ_code	String
univ_name	String
univ_address	String
univ_email	String
univ_website	String

Source Code

//javaapp.db

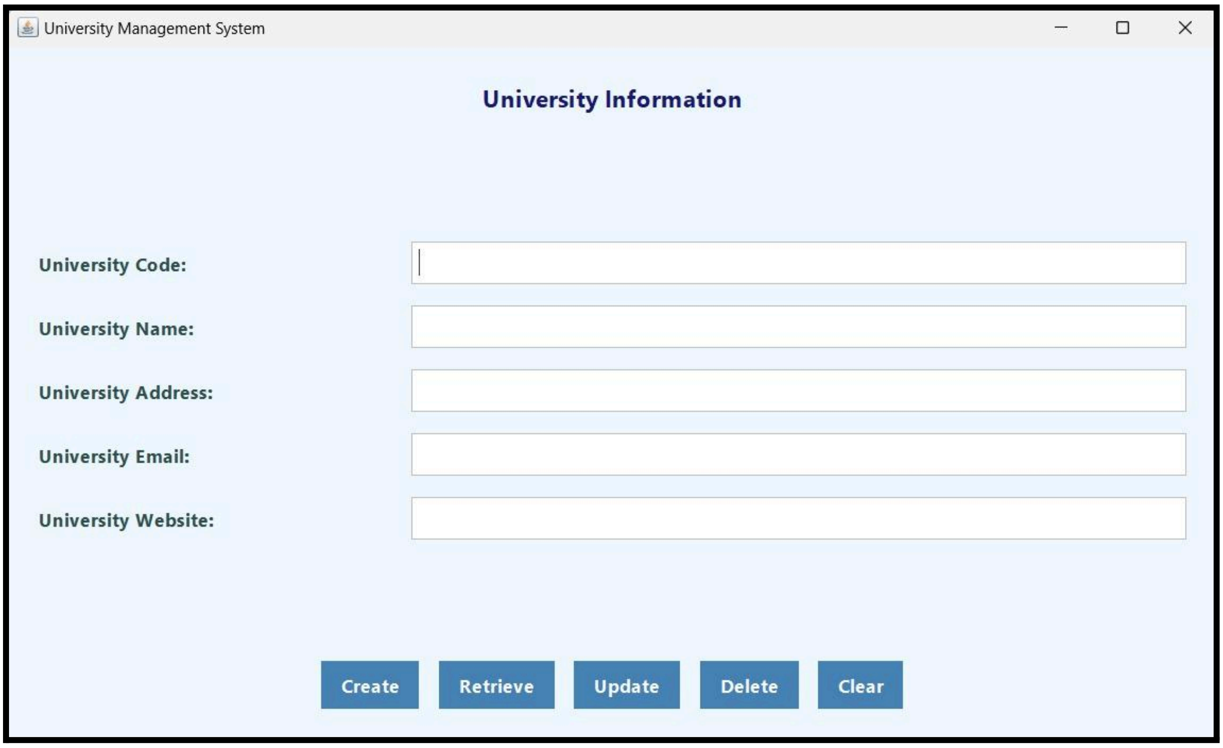
CREATE TABLE university (
    id integer primary key,
    univ_code text,
    univ_name text,
    univ_address text,
    univ_email text,
    univ_website text
)

//UniversityApp.java

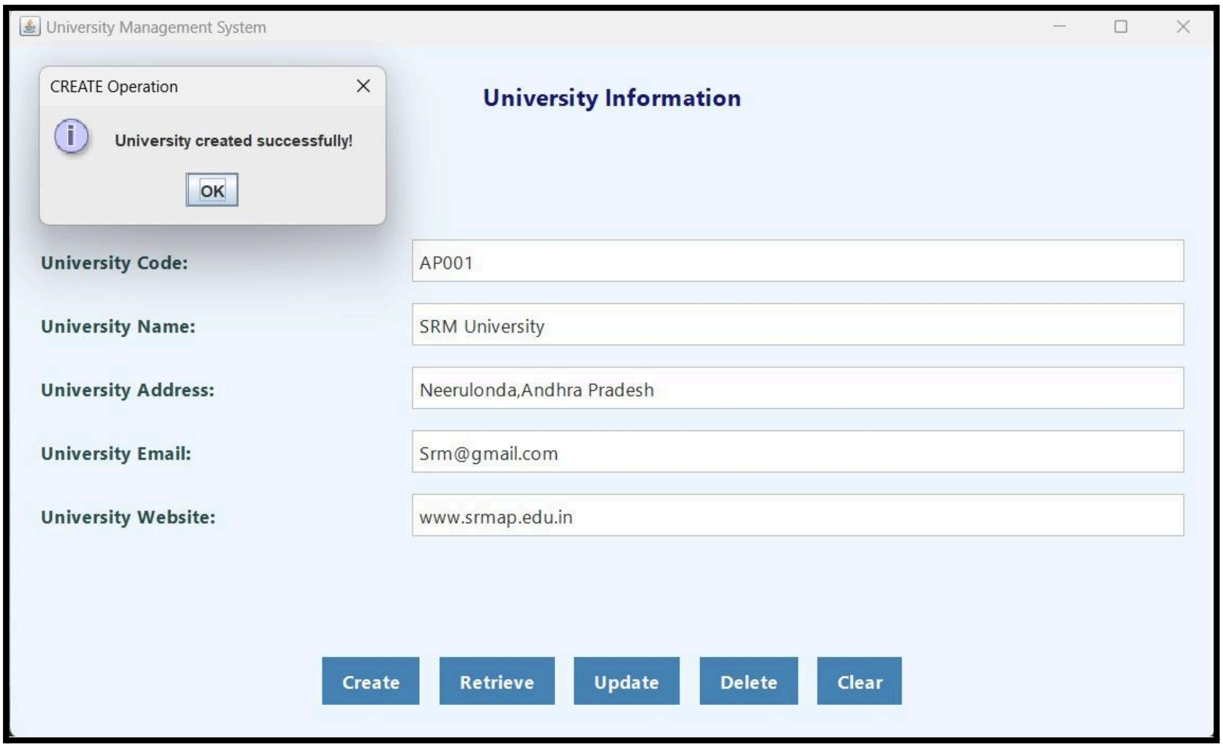
```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
import javax.swing.border.EmptyBorder;

class SQLDB {
    public static Connection conn = null;
    public static Statement stmt = null;
    public static ResultSet rset = null;

    public static void connect(String dbpath) {
        try {
            Class.forName("org.sqlite.JDBC");
            conn = DriverManager.getConnection("jdbc:sqlite:" + dbpath);
            stmt = conn.createStatement();
            initializeDatabase();
        } catch (Exception e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(null, "Database connection failed:
" + e.getMessage(),
                                        "Error", JOptionPane.ERROR_MESSAGE);
        }
    }

    public static void initializeDatabase() {
        try {
            String createTableSQL = "CREATE TABLE IF NOT EXISTS university ("
+
                                   "univ_code TEXT PRIMARY KEY, " +
                                   "univ_name TEXT, " +
                                   "univ_address TEXT, " +
                                   "univ_email TEXT, " +
                                   "univ_website TEXT)";
            stmt.executeUpdate(createTableSQL);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void execute(String query) {
        try {
            rset = stmt.executeQuery(query);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void update(String query) {
        try {
            stmt.executeUpdate(query);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

class MeenaRashi_University {
    private JFrame f = new JFrame("University Management System");
    private Color bgColor = new Color(240, 248, 255);
    private Color buttonColor = new Color(70, 130, 180);
    private Font labelFont = new Font("Segoe UI", Font.BOLD, 14);
    private Font textFont = new Font("Segoe UI", Font.PLAIN, 14);

    // Form components
    private JLabel titleLabel = new JLabel("University Information",
JLabel.CENTER);
    private JLabel[] labels = {
        new JLabel("University Code:"),
        new JLabel("University Name:"),
        new JLabel("University Address:"),
        new JLabel("University Email:"),
        new JLabel("University Website:")
    };

    private JTextField[] fields = {
        new JTextField(20),
        new JTextField(20),
        new JTextField(20),
        new JTextField(20),
        new JTextField(20)
    };

    private JButton[] buttons = {
        new JButton("Create"),
        new JButton("Retrieve"),
        new JButton("Update"),
        new JButton("Delete"),
        new JButton("Clear")
    };

    public MeenaRashi_University() {
        configureUI();
        setupLayout();
        addEventListeners();
    }

    private void configureUI() {
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setSize(600, 500);
        f.setLocationRelativeTo(null);
        f.getContentPane().setBackground(bgColor);

        titleLabel.setFont(new Font("Segoe UI", Font.BOLD, 18));
        titleLabel.setBorder(BorderFactory.createEmptyBorder(15, 0, 15, 0));
        titleLabel.setForeground(new Color(25, 25, 112));

        for (int i = 0; i < labels.length; i++) {
            labels[i].setFont(labelFont);
            labels[i].setForeground(new Color(47, 79, 79));
            fields[i].setFont(textFont);
            fields[i].setBorder(BorderFactory.createCompoundBorder(
                BorderFactory.createLineBorder(new Color(192, 192, 192)),
                BorderFactory.createEmptyBorder(5, 5, 5, 5)
            ));
        }

        for (JButton button : buttons) {
            button.setFont(new Font("Segoe UI", Font.BOLD, 14));
            button.setBackground(buttonColor);
            button.setForeground(Color.WHITE);
            button.setFocusPainted(false);
            button.setBorder(BorderFactory.createEmptyBorder(8, 15, 8, 15));
        }
    }

    private void setupLayout() {
        JPanel mainPanel = new JPanel(new BorderLayout(10, 10));
        mainPanel.setBorder(new EmptyBorder(10, 15, 15, 15));
        mainPanel.setBackground(bgColor);

        mainPanel.add(titleLabel, BorderLayout.NORTH);

        JPanel formPanel = new JPanel(new GridBagLayout());
        formPanel.setBackground(bgColor);
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.anchor = GridBagConstraints.WEST;
        gbc.insets = new Insets(8, 8, 8, 8);

        for (int i = 0; i < labels.length; i++) {
            gbc.gridx = 0;
            gbc.gridy = i;
            gbc.weightx = 0.3;
            formPanel.add(labels[i], gbc);

            gbc.gridx = 1;
            gbc.weightx = 0.7;
            gbc.fill = GridBagConstraints.HORIZONTAL;
            formPanel.add(fields[i], gbc);
        }

        mainPanel.add(formPanel, BorderLayout.CENTER);

        JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 15,
10));
        buttonPanel.setBackground(bgColor);
        for (JButton button : buttons) {
            buttonPanel.add(button);
        }
        mainPanel.add(buttonPanel, BorderLayout.SOUTH);

        f.add(mainPanel);
        f.setVisible(true);
    }

    private void addEventListeners() {
        buttons[0].addActionListener(e -> AP22110010310_University_create());
        buttons[1].addActionListener(e ->
AP22110010310_University_retrive());
        buttons[2].addActionListener(e -> AP22110010310_Universityupdate());
        buttons[3].addActionListener(e -> AP22110010310_University_delete());
        buttons[4].addActionListener(e -> clearFields());
    }

    private void clearFields() {
        for (JTextField field : fields) {
            field.setText("");
        }
    }

    private void AP22110010310_University_create() {
        String code = fields[0].getText();
        String name = fields[1].getText();
        String address = fields[2].getText();
        String email = fields[3].getText();
        String website = fields[4].getText();

        if (validateInputs()) {
            String query = "INSERT INTO university (univ_code, univ_name,
univ_address, univ_email, univ_website) VALUES (?, ?, ?, ?, ?)";
            try (PreparedStatement pst = SQLDB.conn.prepareStatement(query))
{
                pst.setString(1, code);
                pst.setString(2, name);
                pst.setString(3, address);
                pst.setString(4, email);
                pst.setString(5, website);
                pst.executeUpdate();
                showOperationResult("CREATE Operation", "University created
successfully!", true);
            } catch (SQLException e) {
                showOperationResult("Error", "Error creating university: " +
e.getMessage(), false);
            }
        }
    }

    private void AP22110010310_Universityupdate() {
        String code = fields[0].getText();
        String name = fields[1].getText();
        String address = fields[2].getText();
        String email = fields[3].getText();
        String website = fields[4].getText();

        if (validateInputs()) {
            String query = "UPDATE university SET univ_name = ?, univ_address
= ?, univ_email = ?, univ_website = ? WHERE univ_code = ?";
            try (PreparedStatement pst = SQLDB.conn.prepareStatement(query))
{
                pst.setString(1, name);
                pst.setString(2, address);
                pst.setString(3, email);
                pst.setString(4, website);
                pst.setString(5, code);
                int rowsAffected = pst.executeUpdate();

                if (rowsAffected > 0) {
                    showOperationResult("UPDATE Operation", "University
updated successfully!", true);
                } else {
                    showOperationResult("UPDATE Operation", "No university
found with code: " + code, false);
                }
            } catch (SQLException e) {
                showOperationResult("Error", "Error updating university: " +
e.getMessage(), false);
            }
        }
    }
    private void AP22110010310_University_retrive() {
        String code = fields[0].getText();
        if (code.isEmpty()) {
            JOptionPane.showMessageDialog(f, "Please enter a university code
to retrieve",
                                        "Input Error",
JOptionPane.WARNING_MESSAGE);
            return;
        }

        String query = "SELECT * FROM university WHERE univ_code = ?";
        try (PreparedStatement pst = SQLDB.conn.prepareStatement(query)) {
            pst.setString(1, code);
            ResultSet rs = pst.executeQuery();

            if (rs.next()) {
                fields[1].setText(rs.getString("univ_name"));
                fields[2].setText(rs.getString("univ_address"));
                fields[3].setText(rs.getString("univ_email"));
                fields[4].setText(rs.getString("univ_website"));

                showOperationResult("RETRIEVE Operation",
getUniversityDetails(rs), true);
            } else {
                showOperationResult("RETRIEVE Operation", "No university
found with code: " + code, false);
            }
        } catch (SQLException e) {
            showOperationResult("Error", "Error retrieving university: " +
e.getMessage(), false);
        }
    }

    private void AP22110010310_University_delete() {
        String code = fields[0].getText();
        if (code.isEmpty()) {
            JOptionPane.showMessageDialog(f, "Please enter a university code
to delete",
                                        "Input Error",
JOptionPane.WARNING_MESSAGE);
            return;
        }
        int confirm = JOptionPane.showConfirmDialog(f,
            "Are you sure you want to delete university with code: " + code +
"?",
            "Confirm Deletion", JOptionPane.YES_NO_OPTION);

        if (confirm == JOptionPane.YES_OPTION) {
            String query = "DELETE FROM university WHERE univ_code = ?";
            try (PreparedStatement pst = SQLDB.conn.prepareStatement(query))
{
                pst.setString(1, code);
                int rowsAffected = pst.executeUpdate();

                if (rowsAffected > 0) {
                    showOperationResult("DELETE Operation", "University
deleted successfully!", true);
                    clearFields();
                } else {
                    showOperationResult("DELETE Operation", "No university
found with code: " + code, false);
                }
            } catch (SQLException e) {
                showOperationResult("Error", "Error deleting university: " +
e.getMessage(), false);
            }
        }
    }

    private boolean validateInputs() {
        if (fields[0].getText().isEmpty()) {
            JOptionPane.showMessageDialog(f, "University code cannot be
empty",
                                        "Validation Error",
JOptionPane.WARNING_MESSAGE);
            fields[0].requestFocus();
            return false;
        }
        return true;
    }

    private String getUniversityDetails(ResultSet rs) throws SQLException {
        return "University Details:\n\n" +
               "Code: " + rs.getString("univ_code") + "\n" +
               "Name: " + rs.getString("univ_name") + "\n" +
               "Address: " + rs.getString("univ_address") + "\n" +
               "Email: " + rs.getString("univ_email") + "\n" +
               "Website: " + rs.getString("univ_website");
    }

    private void showOperationResult(String title, String message, boolean
isSuccess) {
        JOptionPane.showMessageDialog(f, message, title,
                                    isSuccess ?
JOptionPane.INFORMATION_MESSAGE : JOptionPane.ERROR_MESSAGE);
    }
}

public class UniversityApp {
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            try {

SQLDB.connect("C:\\Users\\viswa\\OneDrive\\Desktop\\Apps\\javaapp.db");
                new MeenaRashi_University();
            } catch (Exception e) {
                JOptionPane.showMessageDialog(null, "Failed to initialize
application: " + e.getMessage(),
                                            "Error",
JOptionPane.ERROR_MESSAGE);
            }
        });
    }
}
```

Screen Shots

//Main



//Creation



University Management System

University Information

CREATE Operation

University created successfully!

University Code: AP001

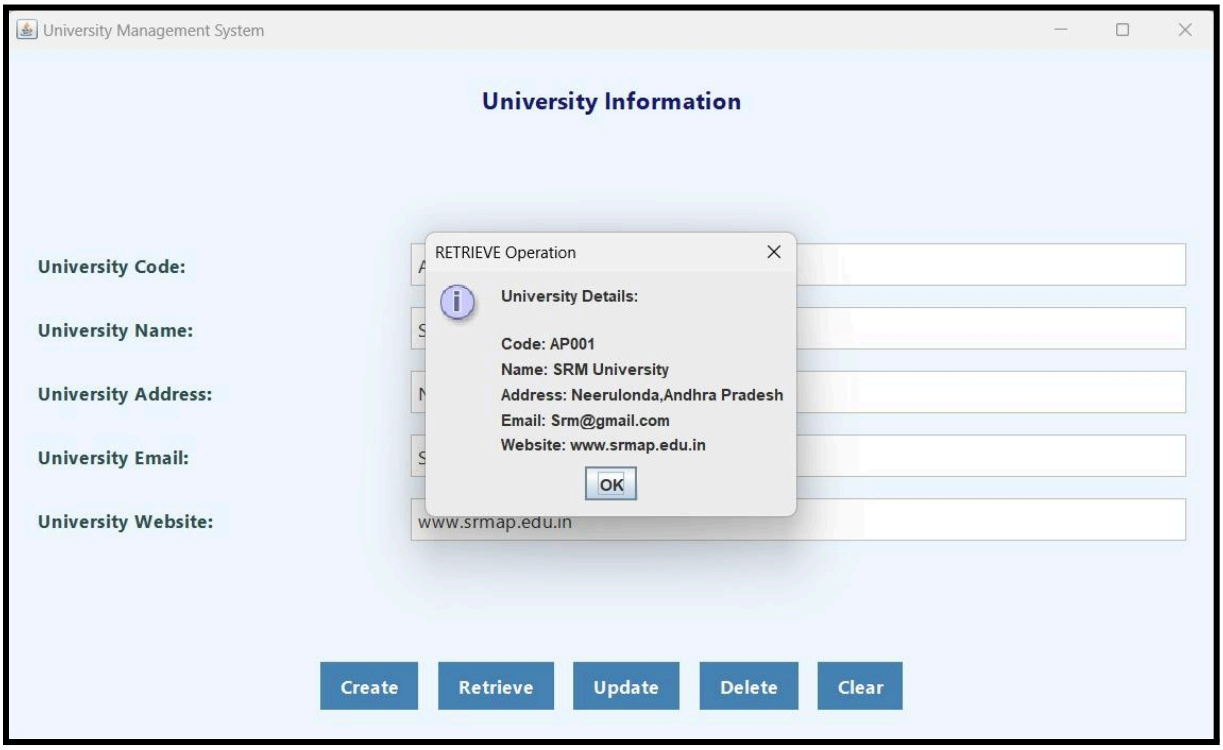
University Name: SRM University

University Address: Neerulonda,Andhra Pradesh

University Email: Srm@gmail.com

University Website: www.srmap.edu.in

//Retrieve



www.srmap.edu.in

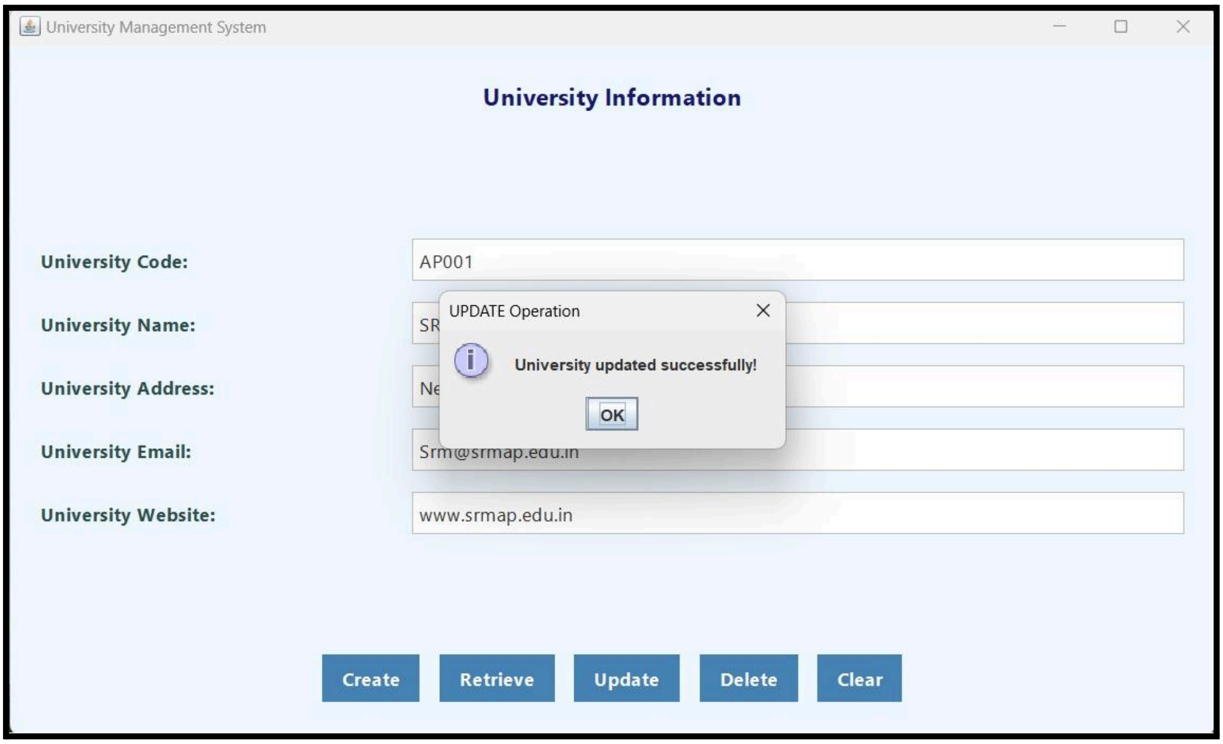
RETRIEVE Operation

University Details:

Code: AP001
Name: SRM University
Address: Neerulonda,Andhra Pradesh
Email: Srm@gmail.com
Website: www.srmap.edu.in

OK

//Updation



AP001

Srm@srmap.edu.in

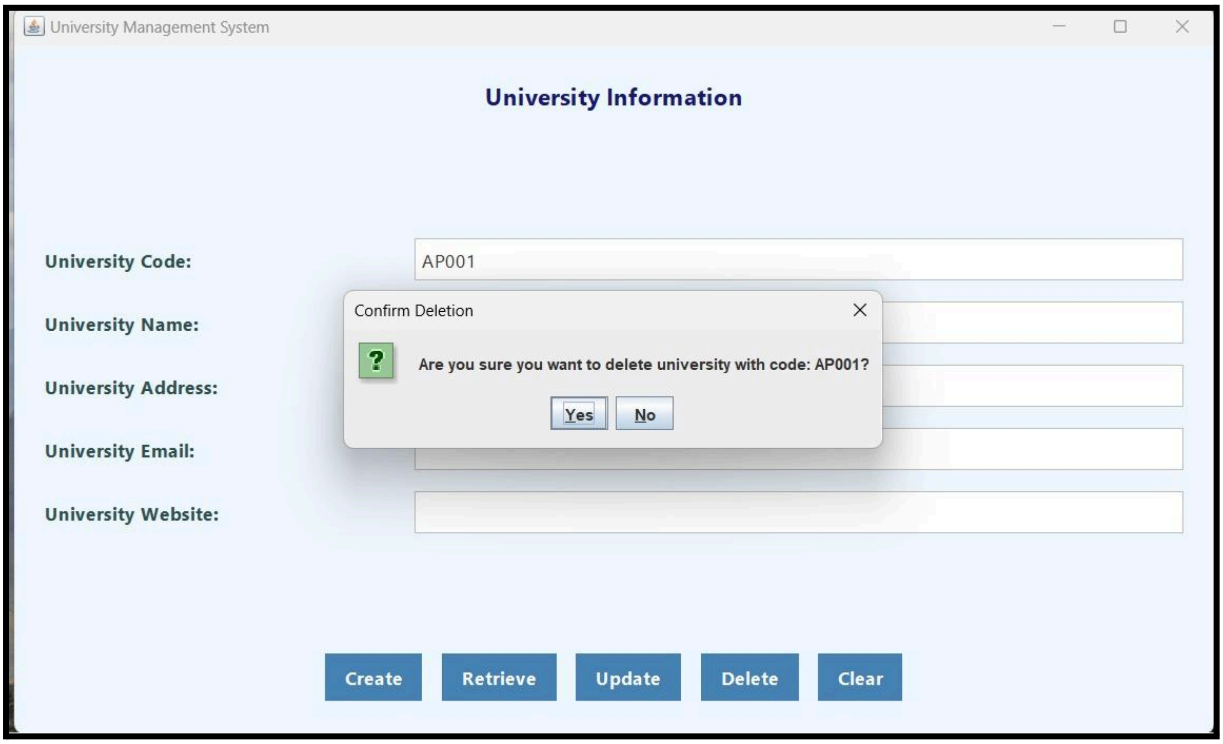
www.srmap.edu.in

UPDATE Operation

University updated successfully!

OK

//Deletion



University Management System

University Information

University Code: AP001

University Name:

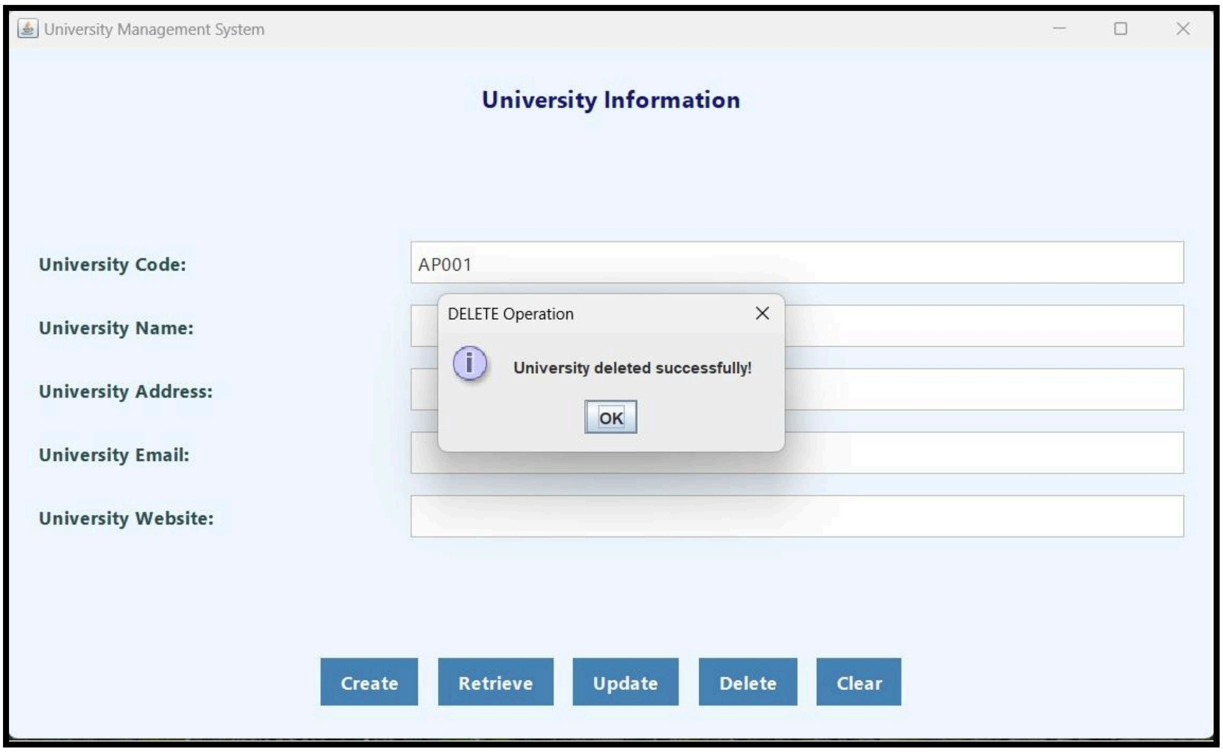
University Address:

University Email:

University Website:

Confirm Deletion

Are you sure you want to delete university with code: AP001?



University Management System

University Information

University Code: AP001

University Name:

University Address:

University Email:

University Website:

DELETE Operation

University deleted successfully!

Conclusion

The "University Module" developed as part of the OBE (Outcome-Based Education) Implementation project successfully demonstrates the use of Java and SQLite in building a functional desktop application with a graphical interface using AWT. This module fulfills the CURD (Create, Update, Retrieve, Delete) operations, allowing users to efficiently manage university information such as code, name, address, email, and website.

Throughout the development process, we applied structured programming practices, adhered to naming conventions, and followed design principles that enhanced both usability and readability. The integration with an SQLite database enabled persistent storage and fast access to records, while the UI components provided an intuitive interface for end-users.

This project not only strengthened our technical skills in Java GUI development and database handling but also gave us practical exposure to software development lifecycles in academic settings. Moving forward, this module can be extended or integrated with other components like Departments and Programs to build a complete OBE management system.