

OBE Implementation

Module-1:University

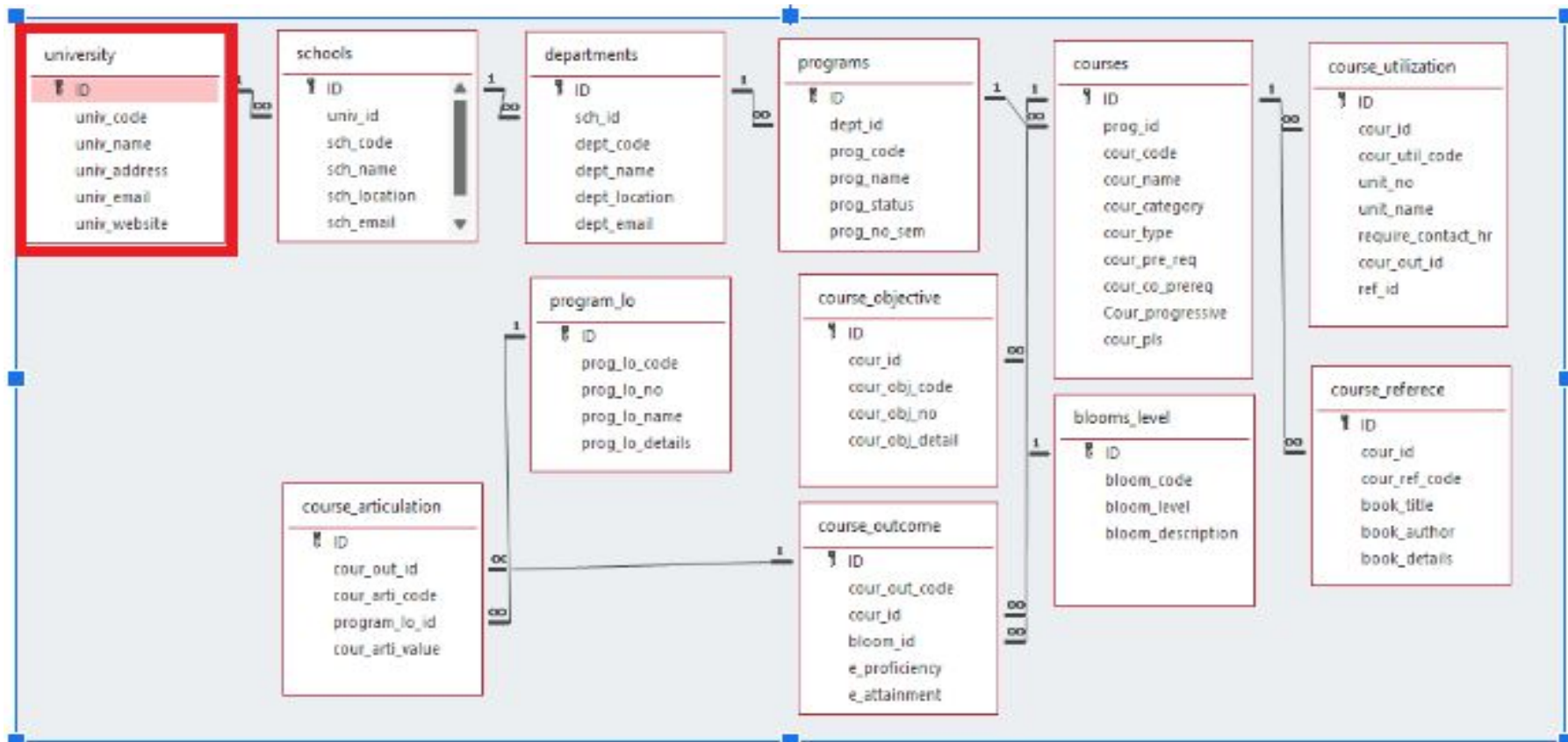
Submitted By

Team Meena Rashi-AP22110010310
AP22110010321
AP22110010315
AP22110010324
AP22110010291

Introduction to Project

SRM-AP University is implementing Outcome-Based Education (OBE), and as part of this initiative, we are tasked with developing a **CURD** (Create, Update, Retrieve, Delete) application for managing university data. The application will be developed using **Java** for the Windows-based desktop version. This project will facilitate efficient management of university-related records and align with the OBE framework by streamlining administrative processes.

Architecture Diagram



Module Description : University Setting

- **Purpose:** Manage university-related data (code, name, address, email, website).
- **Operations:** Supports **Create**, **Retrieve**, **Update**, and **Delete (CRUD)**.
- **Technology:** Built using **Java** and **AWT** for the graphical interface.
- **Database:** Data stored and accessed via **SQLite**.
- **Functionality:** Allows users to add, view, update, and delete university records.

University Setting:Field/table details

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

University Setting: Programming Details

- **File name:** Meenarashi_university
- **Function/method name**
 - **Create:** AP22110010310_university_create
 - **Update:** AP22110010310_university_update
 - **Retrieve:** AP22110010310_university_retrieve
 - **Delete:** AP22110010310_university_delete

Source Code

```
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_retrieve());
    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_retrieve() {
    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:

University Management System

University Information

University Code:

University Name:

University Address:

University Email:

University Website:

[Create](#) [Retrieve](#) [Update](#) [Delete](#) [Clear](#)

Screen Shots

Creation:

University Management System

CREATE Operation

University created successfully!

OK

University Information

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

Create

Retrieve

Update

Delete

Clear

Screen Shots

Retrieve:

University Management System

University Information

University Code:	<input type="text" value="A"/>
University Name:	<input type="text" value="S"/>
University Address:	<input type="text" value="M"/>
University Email:	<input type="text" value="S"/>
University Website:	<input type="text" value="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

Create Retrieve Update Delete Clear

Screen Shots

Update:

University Management System

University Information

University Code:

University Name:

University Address:

University Email:

University Website:

UPDATE Operation

University updated successfully!

OK

Create Retrieve Update Delete Clear

Screen Shots

University Management System

University Information

University Code:


University Name:

University Address:

University Email:

University Website:

Confirm Deletion

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

Screen Shots

University Management System

University Information

University Code:


University Name:

University Address:

University Email:

University Website:

DELETE Operation

 University deleted successfully!

OK

Create Retrieve Update Delete Clear

Conclusion

- The **University Module** enables efficient management of university data (code, name, address, email, and website).
- It supports **CRUD operations** (Create, Retrieve, Update, Delete) through a **Java AWT GUI** and **SQLite database**.
- The module provides a seamless user interface for interacting with the database, ensuring easy management of university records.
- This project demonstrates practical use of **Java** for desktop applications and **SQLite** for data storage, offering a foundation for future enhancements in the OBE (Outcome-Based Education) implementation.

Thank You