

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



SRM UNIVERSITY AP::AMARAVATI

INDEX

Introduction	2
• Project Modules	
Architecture Diagram	3
Module Description	4
• Programming Details	
• Table Details	
Source Code	5
Screen Shots	11
Conclusion	13

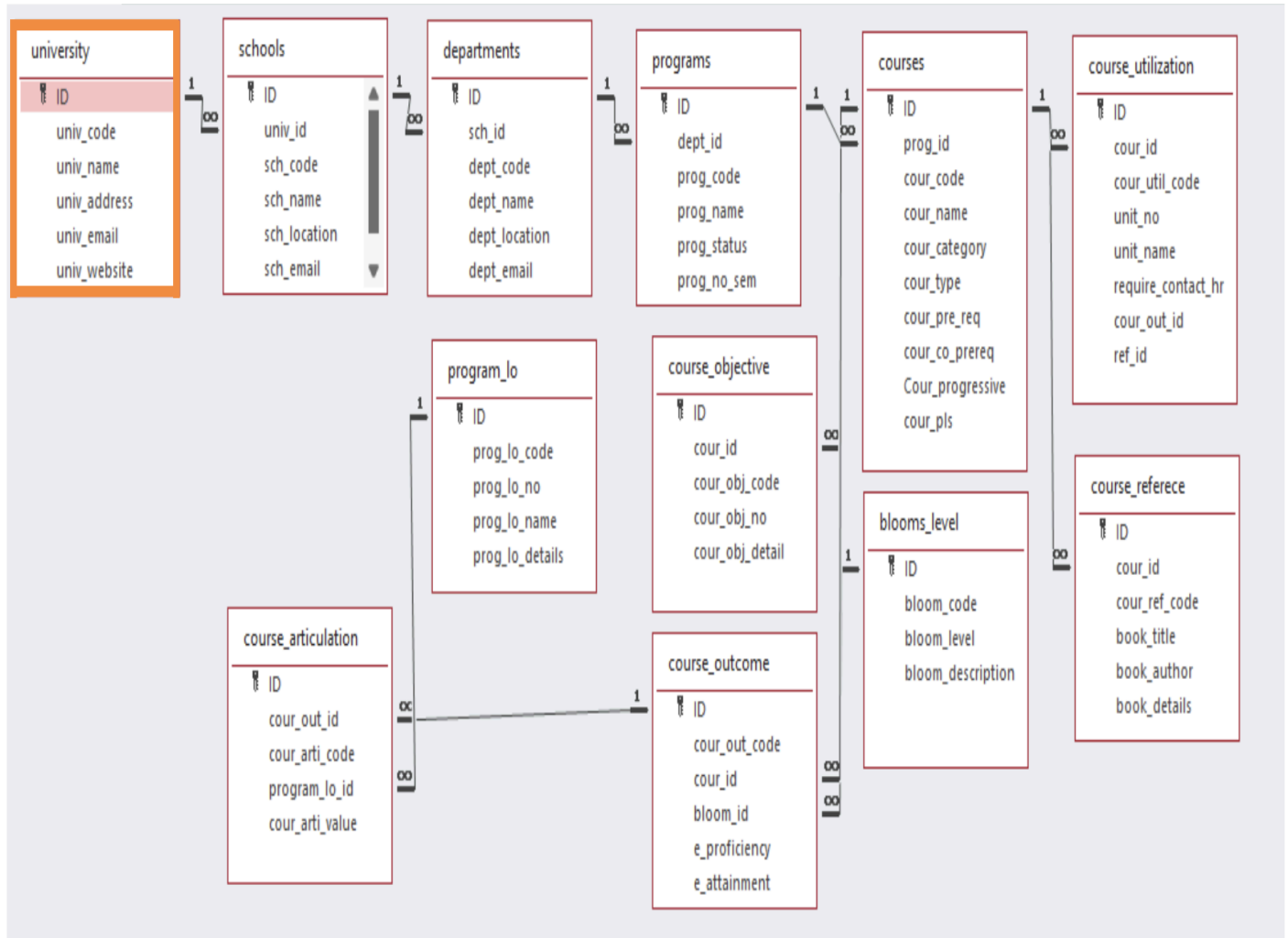
Introduction

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_retrieve
 - **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.*;  
  
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();  
        } 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_module {  
    Frame f = new Frame("University Module");
```

```

Label l1 = new Label("University Code");
Label l2 = new Label("University Name");
Label l3 = new Label("University Address");
Label l4 = new Label("University Email");
Label l5 = new Label("University Website");

TextField t1 = new TextField();
TextField t2 = new TextField();
TextField t3 = new TextField();
TextField t4 = new TextField();
TextField t5 = new TextField();

Button b1 = new Button("Create");
Button b2 = new Button("Retrieve");
Button b3 = new Button("Update");
Button b4 = new Button("Delete");
Button b5 = new Button("Minimize");
Button b6 = new Button("Close");

meenarashi_university_module() {
    f.setLayout(new GridBagLayout());
    GridBagConstraints gbc = new GridBagConstraints();
    gbc.fill = GridBagConstraints.HORIZONTAL;
    gbc.insets = new Insets(10, 10, 10, 10);

    f.setSize(500, 350);
    f.setVisible(true);

    f.setBackground(new Color(255, 255, 255));
    b1.setBackground(new Color(100, 149, 237));
    b2.setBackground(new Color(100, 149, 237));
    b3.setBackground(new Color(100, 149, 237));
    b4.setBackground(new Color(100, 149, 237));
    b5.setBackground(new Color(255, 69, 0));
    b6.setBackground(new Color(255, 69, 0));
    b1.setFont(new Font("Arial", Font.PLAIN, 12));
    b2.setFont(new Font("Arial", Font.PLAIN, 12));
    b3.setFont(new Font("Arial", Font.PLAIN, 12));
    b4.setFont(new Font("Arial", Font.PLAIN, 12));
    b5.setFont(new Font("Arial", Font.PLAIN, 12));
    b6.setFont(new Font("Arial", Font.PLAIN, 12));

    gbc.gridx = 0; gbc.gridy = 0; gbc.gridwidth = 2;
    f.add(l1, gbc);

    gbc.gridx = 0; gbc.gridy = 1; gbc.gridwidth = 2;
    f.add(t1, gbc);

```

```

gbc.gridx = 0; gbc.gridy = 2; gbc.gridwidth = 2;
f.add(l2, gbc);

gbc.gridx = 0; gbc.gridy = 3; gbc.gridwidth = 2;
f.add(t2, gbc);

gbc.gridx = 0; gbc.gridy = 4; gbc.gridwidth = 2;
f.add(l3, gbc);

gbc.gridx = 0; gbc.gridy = 5; gbc.gridwidth = 2;
f.add(t3, gbc);

gbc.gridx = 0; gbc.gridy = 6; gbc.gridwidth = 2;
f.add(l4, gbc);

gbc.gridx = 0; gbc.gridy = 7; gbc.gridwidth = 2;
f.add(t4, gbc);

gbc.gridx = 0; gbc.gridy = 8; gbc.gridwidth = 2;
f.add(l5, gbc);

gbc.gridx = 0; gbc.gridy = 9; gbc.gridwidth = 2;
f.add(t5, gbc);

gbc.gridx = 0; gbc.gridy = 10; gbc.gridwidth = 1;
f.add(b1, gbc);

gbc.gridx = 1; gbc.gridy = 10; gbc.gridwidth = 1;
f.add(b2, gbc);

gbc.gridx = 0; gbc.gridy = 11; gbc.gridwidth = 1;
f.add(b3, gbc);

gbc.gridx = 1; gbc.gridy = 11; gbc.gridwidth = 1;
f.add(b4, gbc);

gbc.gridx = 0; gbc.gridy = 12; gbc.gridwidth = 1;
f.add(b5, gbc);

gbc.gridx = 1; gbc.gridy = 12; gbc.gridwidth = 1;
f.add(b6, gbc);

b1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        AP22110010310_university_create();
    }
}

```



```

    });

    b2.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
            AP22110010310_university_retrieve();
        }
    });

    b3.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
            AP22110010310_university_update();
        }
    });

    b4.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
            AP22110010310_university_delete();
        }
    });

    b5.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
            f.setState(Frame.ICONIFIED);
        }
    });

    b6.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
            f.dispose();
        }
    });

    f.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent we) {
            f.dispose();
        }
    });
}

public void AP22110010310_university_create() {
    String code = t1.getText();
    String name = t2.getText();
    String address = t3.getText();
    String email = t4.getText();
    String website = t5.getText();

```

```

        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();
            showMessage("University created successfully!");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public void AP22110010310_university_update() {
        String code = t1.getText();
        String name = t2.getText();
        String address = t3.getText();
        String email = t4.getText();
        String website = t5.getText();

        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);
            pst.executeUpdate();
            showMessage("University updated successfully!");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public void AP22110010310_university_retrieve() {
        String code = t1.getText();
        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()) {

```

```

        t2.setText(rs.getString("univ_name"));
        t3.setText(rs.getString("univ_address"));
        t4.setText(rs.getString("univ_email"));
        t5.setText(rs.getString("univ_website"));
    } else {
        showMessage("University not found!");
    }
} catch (SQLException e) {
    e.printStackTrace();
}
}

public void AP22110010310_university_delete() {
    String code = t1.getText();
    String query = "DELETE FROM university WHERE univ_code = ?";
    try {
        PreparedStatement pst = SQLDB.conn.prepareStatement(query);
        pst.setString(1, code);
        pst.executeUpdate();
        showMessage("University deleted successfully!");
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public void showMessage(String message) {
    Dialog d = new Dialog(f, "Message", true);
    d.setLayout(new FlowLayout());
    Button okButton = new Button("OK");
    okButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
            d.dispose();
        }
    });
    d.add(new Label(message));
    d.add(okButton);
    d.setSize(300, 100);
    d.setVisible(true);
}
}

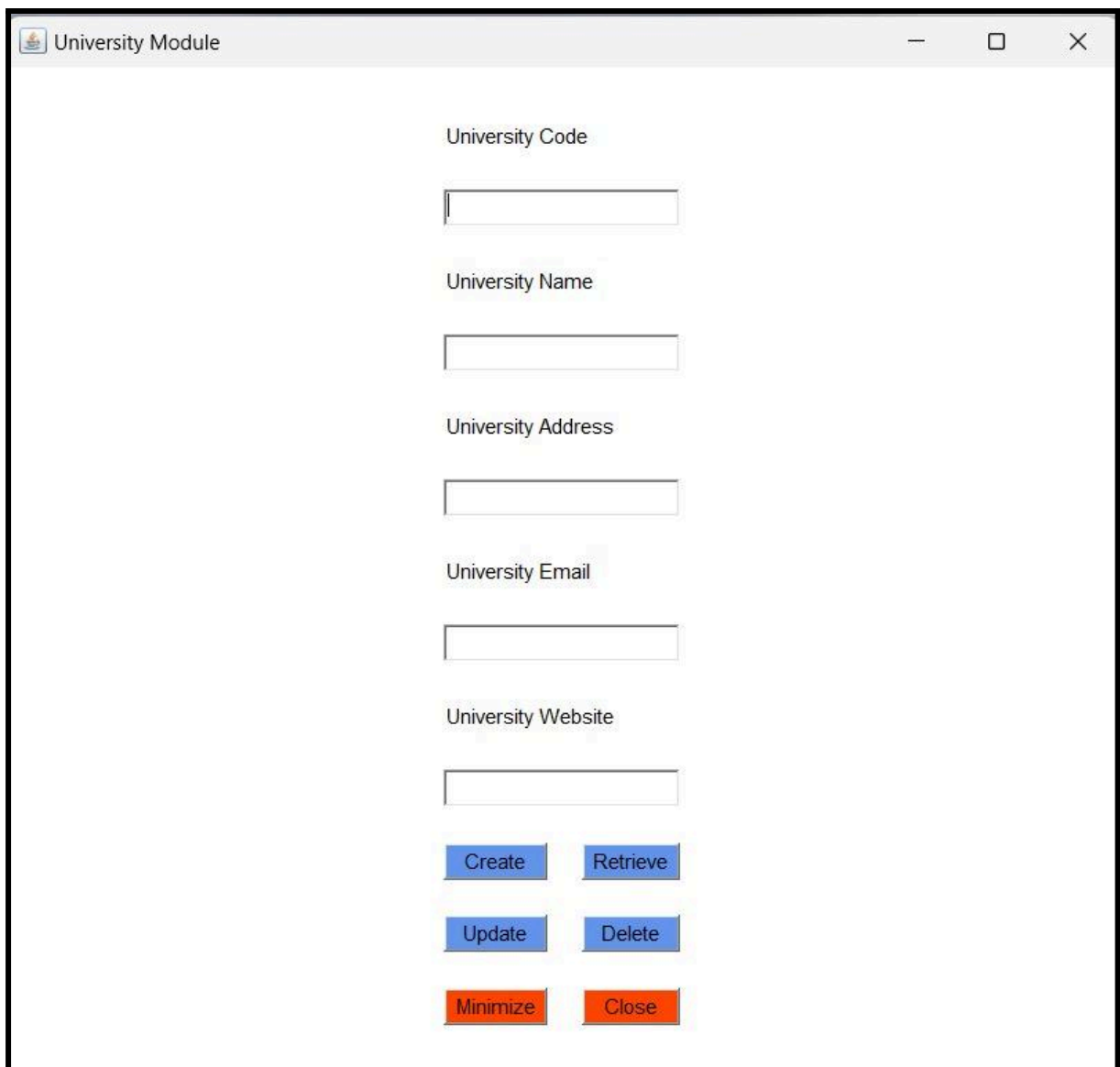
public class UniversityApp {
    public static void main(String[] args) {
        SQLDB.connect("C:\\Users\\viswa\\OneDrive\\Desktop\\Apps\\javaapp.db");

        new AP22110010310_meenarashi_university_module();
    }
}
}

```

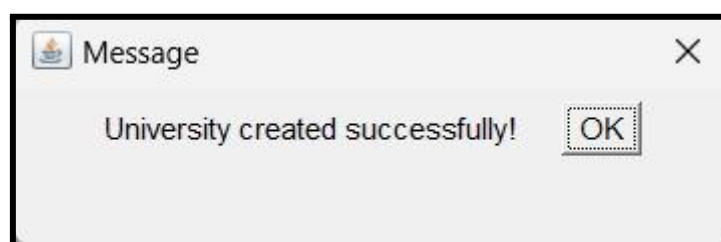
Screen Shots

//Main Page



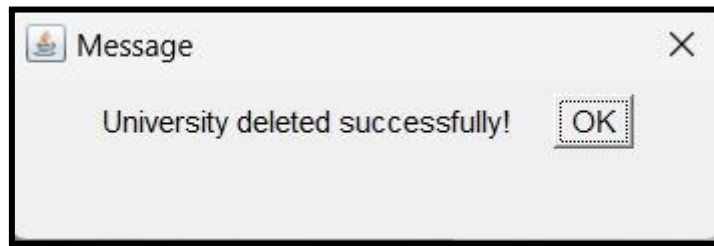
The screenshot shows a window titled "University Module" with a standard Windows-style title bar (minimize, maximize, close buttons). The main content area contains five text input fields, each preceded by a label: "University Code", "University Name", "University Address", "University Email", and "University Website". Below these fields are six buttons arranged in three rows: "Create" and "Retrieve" in the first row, "Update" and "Delete" in the second row, and "Minimize" and "Close" in the third row. The "Minimize" and "Close" buttons are red, while the others are blue.

//Dialogue when data Created



The screenshot shows a small dialog box titled "Message" with a close button (X) in the top right corner. The main text inside the dialog box reads "University created successfully!". To the right of the text is an "OK" button.

/Dialogue when data Deleted



/Dialogue when data Updated



//Data Table in sqlite3

```
C:\Users\viswa\OneDrive\Desktop\Apps>sqlite3 javaapp.db
SQLite version 3.49.1 2025-02-18 13:38:58
Enter ".help" for usage hints.
sqlite> .tables
university
sqlite> .headers on
sqlite> .mode column
sqlite> SELECT*FROM university;
id  univ_code  univ_name      univ_address  univ_email    univ_website
--  -
1   SRM01      SRM University AP Andhra Pradesh info@srm.ap   www.srm.ap.edu
2
3   SRM001     SRM             TN             srm@gmail.com www.srm.com
4   SRM001     SRM             Ap             srm@gmail.com www.srm.com
sqlite> |
```

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 CRUD (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.