OBE IMPLEMENTATION: UNIVERSITY SETTING

by

Geetesh Kotturu - AP22110010062
Gali Lakshmi Pujitha - AP22110010206
Sai Charitha Yella - AP22110010213
Satwick Gummadi - AP22110010214
Geethika Gangireddy - AP22110010259

A report for the CS307:Mobile Application Development using JAVA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SRM UNIVERSITY AP::AMARAVATI

INDEX

INTRODUCTION

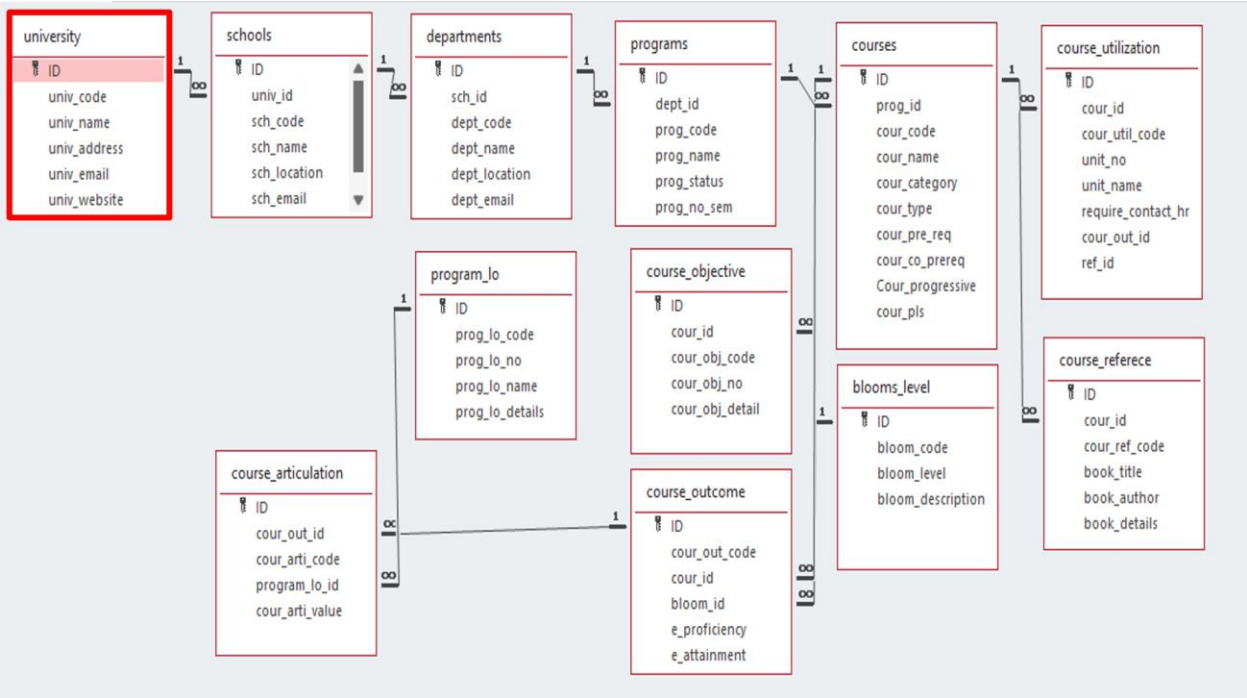
Our University (herewith considered as SRM-AP) is going to implement OBE(Outcome Based Education) in their university and you are assigned in the project to develop a CURD(Create,Update,Retrieve and Delete) windows and mobile application using JAVA programming and Android studio for the same.

Project Modules:

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: university
Module Description:

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

Programming Details naming conventions to be used:

- **class name/activity name:**yourteamname_module_name
- **Function/method name**
  - **Create:** university_TeamA_create
  - **Update:** university_TeamA_update
  - **Retrieve:** university_TeamA_retrive
  - **Delete:** university_TeamA_delete

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

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

Source Code

TeamA_University.java :

```
package teama_university;

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.sql.*;

public class TeamA_University extends JFrame {
    Connection conn;
    JTextField tfCode, tfName, tfAddress, tfEmail, tfWebsite;

    public TeamA_University() {
        conn = Dbconn.connectDB();
        setTitle("University CRUD - TeamA");
        setSize(500, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        initUI();
    }
    private void initUI() {
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(5, 1, 10, 10));
        panel.setBorder(BorderFactory.createEmptyBorder(30, 50, 30, 50));
```

```
        JLabel title = new JLabel("University CRUD - TeamA", SwingConstants.CENTER);
        title.setFont(new Font("SansSerif", Font.BOLD, 18));
        add(title, BorderLayout.NORTH);

        JButton btnCreate = new JButton("Create University");
        JButton btnRetrieve = new JButton("Retrieve All");
        JButton btnUpdate = new JButton("Update University");
        JButton btnDelete = new JButton("Delete University");
        JButton btnExit = new JButton("Exit");

        btnCreate.addActionListener(e -> university_TeamA_create());
        btnRetrieve.addActionListener(e -> university_TeamA_retrieve());
        btnUpdate.addActionListener(e -> university_TeamA_update());
        btnDelete.addActionListener(e -> university_TeamA_delete());
        btnExit.addActionListener(e -> System.exit(0));

        panel.add(btnCreate);
        panel.add(btnRetrieve);
        panel.add(btnUpdate);
        panel.add(btnDelete);
        panel.add(btnExit);

        add(panel, BorderLayout.CENTER);
    }
//CREATE
    public void university_TeamA_create() {
        JPanel panel = new JPanel(new GridLayout(5, 2, 10, 10));
        tfCode = new JTextField();
        tfName = new JTextField();
        tfAddress = new JTextField();
        tfEmail = new JTextField();
        tfWebsite = new JTextField();

        panel.add(new JLabel("Code:")); panel.add(tfCode);
        panel.add(new JLabel("Name:")); panel.add(tfName);
        panel.add(new JLabel("Address:")); panel.add(tfAddress);
        panel.add(new JLabel("Email:")); panel.add(tfEmail);
        panel.add(new JLabel("Website:")); panel.add(tfWebsite);

        int result = JOptionPane.showConfirmDialog(this, panel, "Create University",
JOptionPane.OK_CANCEL_OPTION);
        if (result == JOptionPane.OK_OPTION) {
```

```
        if (tfCode.getText().isEmpty() || tfName.getText().isEmpty() ||
tfAddress.getText().isEmpty()
                || tfEmail.getText().isEmpty() || tfWebsite.getText().isEmpty()) {
            JOptionPane.showMessageDialog(this, "All fields must be filled.");
            return;
        }
        try {
            String sql = "INSERT INTO university(univ_code, univ_name, univ_address,
univ_email, univ_website) VALUES (?, ?, ?, ?, ?)";
            PreparedStatement pst = conn.prepareStatement(sql);
            pst.setString(1, tfCode.getText());
            pst.setString(2, tfName.getText());
            pst.setString(3, tfAddress.getText());
            pst.setString(4, tfEmail.getText());
            pst.setString(5, tfWebsite.getText());
            pst.executeUpdate();
            JOptionPane.showMessageDialog(this, "University created successfully!");
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Insert error: " + e.getMessage());
        }
    }
}
```

//RETRIEVE

```
    public void university_TeamA_retrieve() {
        try {
            String sql = "SELECT * FROM university";
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(sql);
            DefaultTableModel model = new DefaultTableModel(
                    new Object[]{"ID", "Code", "Name", "Address", "Email", "Website"}, 0
            );
            while (rs.next()) {
                model.addRow(new Object[]{
                        rs.getInt("id"),
                        rs.getString("univ_code"),
                        rs.getString("univ_name"),
                        rs.getString("univ_address"),
                        rs.getString("univ_email"),
                        rs.getString("univ_website")
                });
            }
            JTable table = new JTable(model);
            JScrollPane scrollPane = new JScrollPane(table);
            scrollPane.setPreferredSize(new Dimension(700, 200));
```

```
            JOptionPane.showMessageDialog(this, scrollPane, "University Records",
JOptionPane.INFORMATION_MESSAGE);
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Read error: " + e.getMessage());
        }
    }
```

//UPDATE

```
    public void university_TeamA_update() {
        String id = JOptionPane.showInputDialog(this, "Enter University ID to update:");
        if (id != null && !id.isEmpty()) {
            try {
                String checkSql = "SELECT COUNT(*) FROM university WHERE id=?";
                PreparedStatement checkStmt = conn.prepareStatement(checkSql);
                checkStmt.setInt(1, Integer.parseInt(id));
                ResultSet rs = checkStmt.executeQuery();
                if (rs.next() && rs.getInt(1) == 0) {
                    JOptionPane.showMessageDialog(this, "University with ID " + id + " does not
exist.");
                    return;
                }
            } catch (SQLException e) {
                JOptionPane.showMessageDialog(this, "Check error: " + e.getMessage());
                return;
            }

            JPanel panel = new JPanel(new GridLayout(4, 2, 10, 10));
            tfName = new JTextField();
            tfAddress = new JTextField();
            tfEmail = new JTextField();
            tfWebsite = new JTextField();

            panel.add(new JLabel("Name:")); panel.add(tfName);
            panel.add(new JLabel("Address:")); panel.add(tfAddress);
            panel.add(new JLabel("Email:")); panel.add(tfEmail);
            panel.add(new JLabel("Website:")); panel.add(tfWebsite);

            int result = JOptionPane.showConfirmDialog(this, panel, "Update University",
JOptionPane.OK_CANCEL_OPTION);
            if (result == JOptionPane.OK_OPTION) {
                if (tfName.getText().isEmpty() || tfAddress.getText().isEmpty()
                        || tfEmail.getText().isEmpty() || tfWebsite.getText().isEmpty()) {
                    JOptionPane.showMessageDialog(this, "All fields must be filled.");
                    return;
                }
```

```
            try {
                String sql = "UPDATE university SET univ_name=?, univ_address=?,
univ_email=?, univ_website=? WHERE id=?";
                PreparedStatement pst = conn.prepareStatement(sql);
                pst.setString(1, tfName.getText());
                pst.setString(2, tfAddress.getText());
                pst.setString(3, tfEmail.getText());
                pst.setString(4, tfWebsite.getText());
                pst.setInt(5, Integer.parseInt(id));
                pst.executeUpdate();
                JOptionPane.showMessageDialog(this, "University updated successfully!");
            } catch (SQLException e) {
                JOptionPane.showMessageDialog(this, "Update error: " + e.getMessage());
            }
        }
    }
  }
```

//DELETE

```
  public void university_TeamA_delete() {
    String id = JOptionPane.showInputDialog(this, "Enter University ID to delete:");
    if (id != null && !id.isEmpty()) {
      try {
        String checkSql = "SELECT COUNT(*) FROM university WHERE id=?";
        PreparedStatement checkStmt = conn.prepareStatement(checkSql);
        checkStmt.setInt(1, Integer.parseInt(id));
        ResultSet rs = checkStmt.executeQuery();
        if (rs.next() && rs.getInt(1) == 0) {
          JOptionPane.showMessageDialog(this, "University with ID " + id + " does not
exist.");
          return;
        }

        String sql = "DELETE FROM university WHERE id=?";
        PreparedStatement pst = conn.prepareStatement(sql);
        pst.setInt(1, Integer.parseInt(id));
        pst.executeUpdate();
        JOptionPane.showMessageDialog(this, "University deleted successfully!");
      } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Delete error: " + e.getMessage());
      }
    }
  }

  public static void main(String[] args) {
```

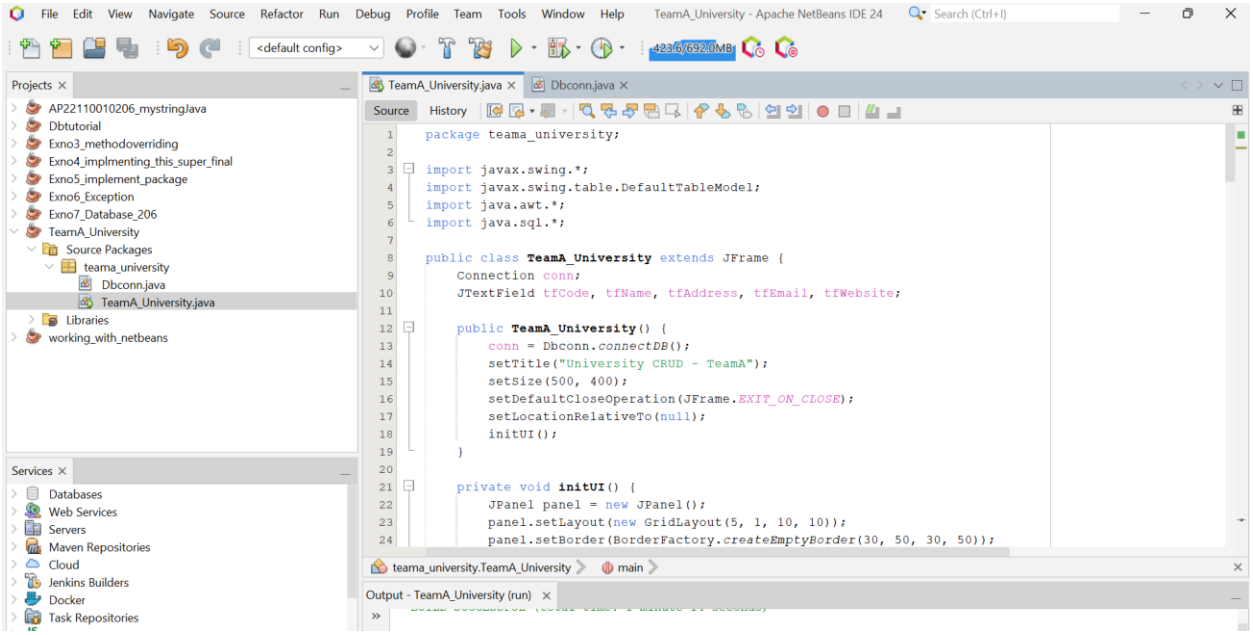
```
        SwingUtilities.invokeLater(() -> new TeamA_University().setVisible(true));
    }
}
```

Dbconn.java :

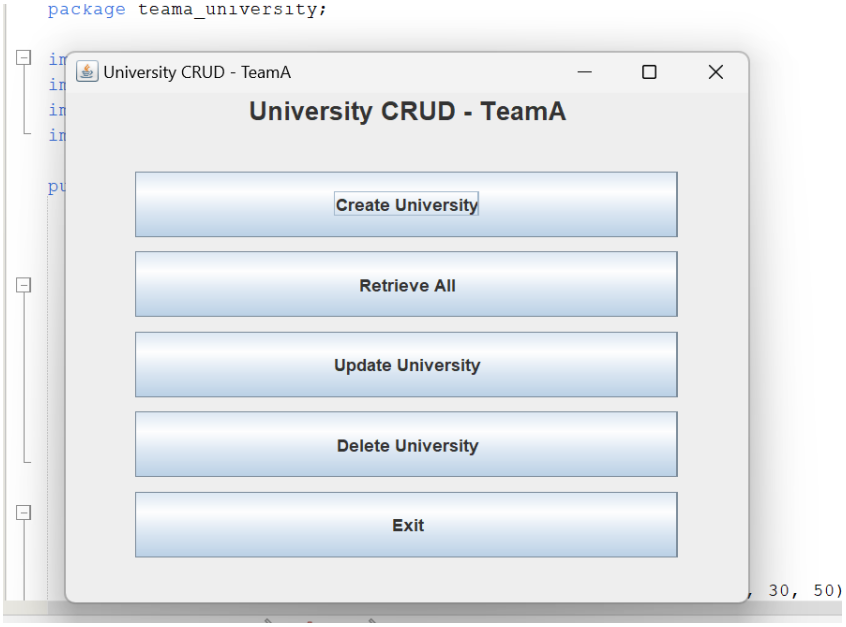
```
package teama_university;

import java.sql.Connection;
import java.sql.DriverManager;

public class Dbconn {
    public static Connection connectDB() {
        try {
            Class.forName("org.sqlite.JDBC");
            Connection con = DriverManager.getConnection("jdbc:sqlite:university.sqlite");
            System.out.println("Connection Successful");
            return con;
        } catch (Exception e) {
            System.out.println("Connection Failed " + e);
            return null;
        }
    }
}
```

Screen Shots

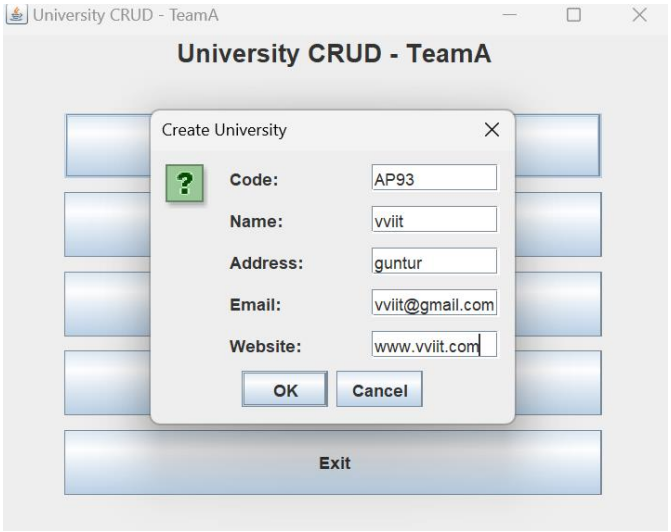
Project setup in netbeans :

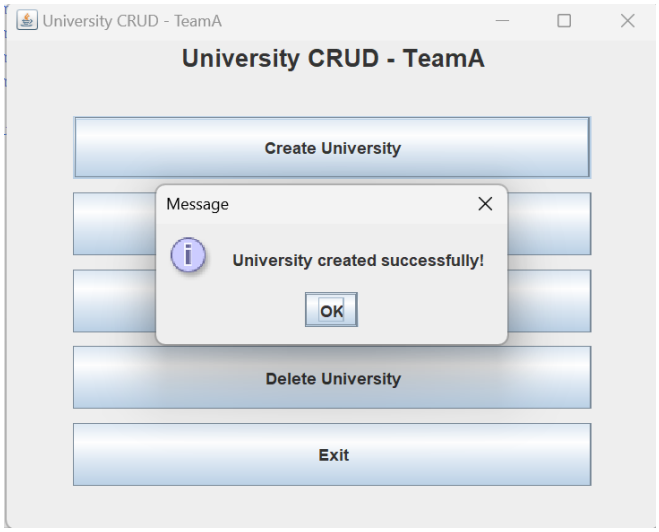


Output :

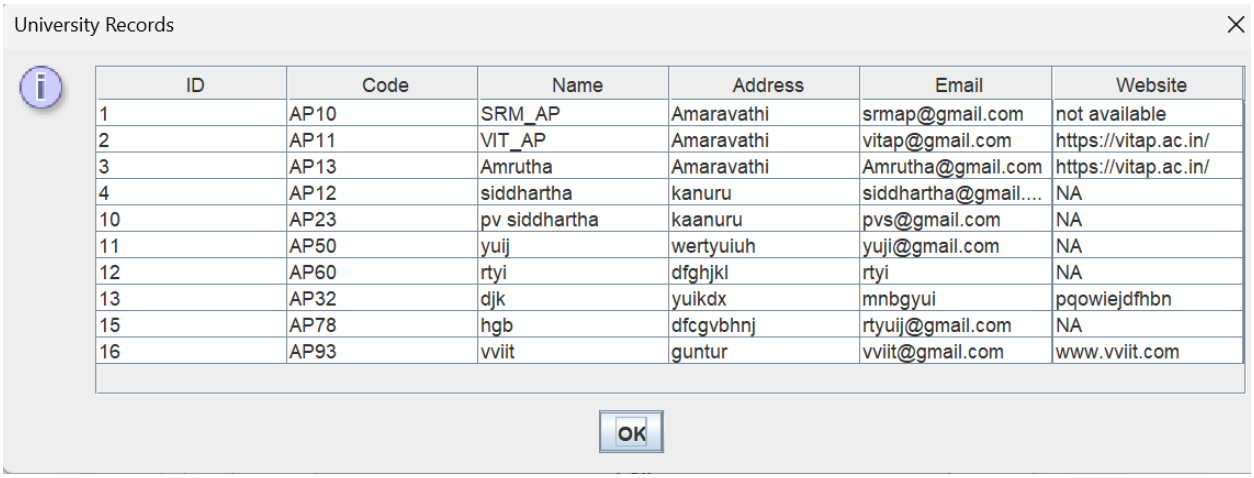


Create university :

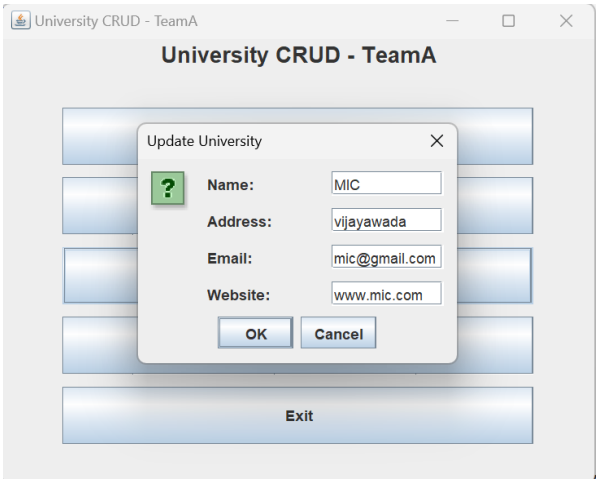


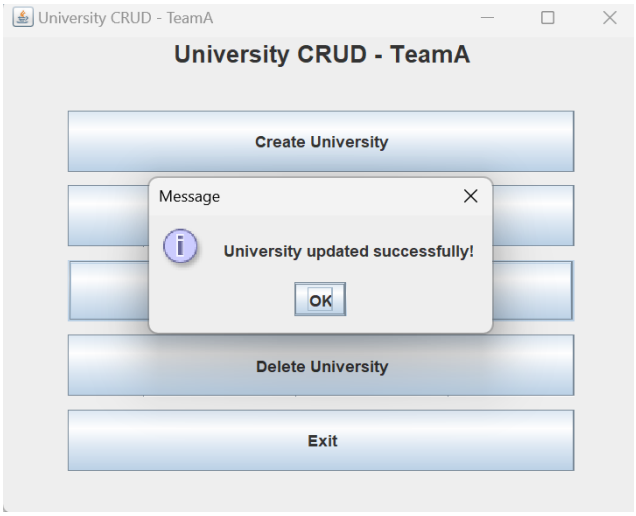


Retrieve Universities :



Update University :





Updation reflection in records :

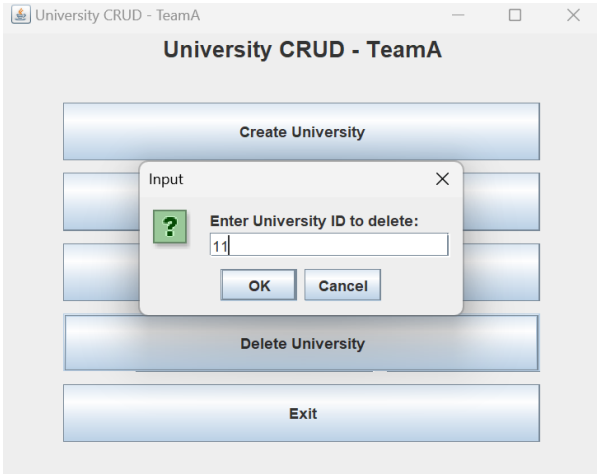
Before updation :

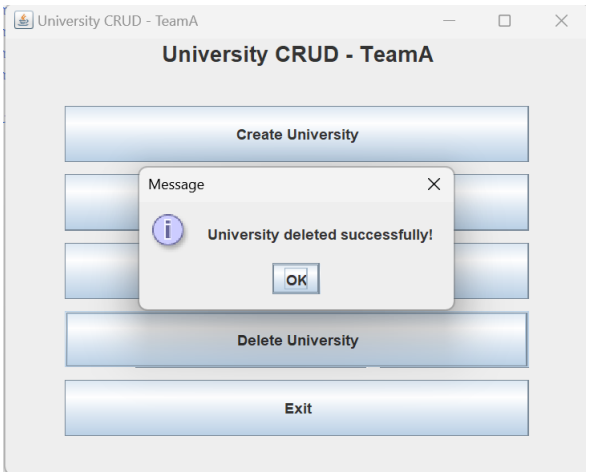
15	AP78	hgb	dfcgvbhnj	rtyuij@gmail.com	NA

After updation :

15	AP78	MIC	vijayawada	mic@gmail.com	www.mic.com

Delete university :





Conclusion

In this project, we created a University module to support the Outcome Based Education (OBE) system. The application was built using Java and SQLite, and it allows users to Create, Read, Update, and Delete (CRUD) university details. We used Java Swing to design a simple and user-friendly interface.

The project followed proper naming rules and organized the code clearly, which makes it easier to understand and improve in the future. This module works well and can be connected with other parts of the OBE system like Schools, Departments, and Courses. Overall, the project was successful and met its goals. It can be used as a base for building a complete OBE application at SRM University–AP.