OBE IMPLEMENTATION:UNIVERSITY SETTING

by

A. Pradeep Kumar [AP22110010005]
P. Avinash Reddy [AP22110010006]
Sk. Jeelani basha [AP22110010013]
A. Maydhazo Dhanyan [AP22110010032]
T. Hemanjali [AP22110010047]
K. Rahul Chandra [AP22110010052]

A report for the CS307:Mobile Application Development using JAVA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SRM UNIVERSITY AP::AMARAVATI

INDEX

Introduction

The implementation of Outcome-Based Education (OBE) at SRM-AP requires a structured approach to defining and managing Course Objectives (COs). These objectives outline the skills and knowledge that students are expected to acquire upon completing a course. As part of the OBE framework, clear and measurable COs are essential for aligning teaching methods with desired learning outcomes. To facilitate this process, a CRUD (Create, Retrieve, Update, Delete) system is being developed to streamline the management of these objectives.

The proposed system will serve as a centralized platform where faculty members and academic administrators can efficiently input, modify, and retrieve Course Objectives. It ensures that all relevant stakeholders have access to accurate and up-to-date information regarding the learning goals for each course. By providing an intuitive interface for managing COs, the system reduces administrative overhead and enhances the efficiency of the academic process, making it easier to track course outcomes.

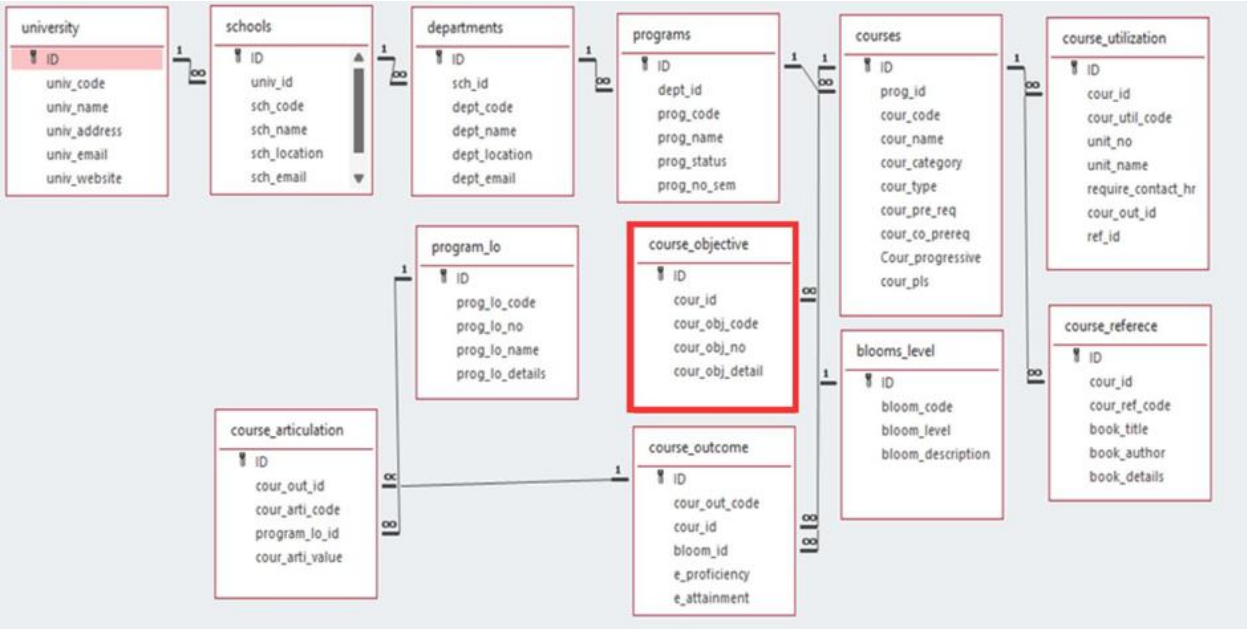
Moreover, the CRUD-based system aims to improve the traceability and transparency of Course Objectives. Faculty will be able to update objectives as needed, while administrators can monitor changes and maintain a comprehensive history of CO modifications. This system will also support seamless integration with other academic processes, ensuring that the implementation of OBE is consistent across various courses and departments, ultimately helping SRM-AP achieve its educational goals.

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: Course Objective Setting

Module Description:

The Course Objective CURD Module is a core component in the Outcome-Based Education (OBE) implementation framework at SRM University-AP. This module is responsible for managing the lifecycle of course-related data through four primary operations — Create, Update, Retrieve, and Delete (CURD). It provides faculty and academic administrators a user-friendly interface to manage course objectives efficiently and store the data securely in a MySQL database.

Programming Details naming conventions to be used:

- **class name/activity name:**CodeBrew_CourseObjective.java
- **Function/method name**
  - **Create:**CourseObjective_CreateForm
  - **Update:**CourseObjective_Update
  - **Retrieve:**CourseObjective_Retrieve
  - **Delete:**CourseObjective_Delete

Table details:Course Objective Setting

Field Name	**Data type**
id	integer
cour_id	String
cour_obj_code	String
Cour_obj_no	String
Cour_obj_detail	String

Source Code

```
import java.awt.BorderLayout;
import java.awt.Button;
import java.awt.Font;
import java.awt.Frame;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.Panel;
import java.awt.ScrollPane;
import java.awt.TextArea;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

import javax.swing.JOptionPane;

class DB {
  public static Connection conn = null;
  public static PreparedStatement pstmt = null;

  public static void connect(String path) {
    try {
      Class.forName("org.sqlite.JDBC");
      conn = DriverManager.getConnection("jdbc:sqlite:" + path);
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
```

```
  public static void CourseObjective_Create(int id, String code, String
objCode, int no, String detail) {
    try {
      String q = "INSERT INTO courseobjective(ID, cour_id, cour_obj_code,
cour_obj_no, cour_obj_detail) VALUES(?, ?, ?, ?, ?)";
      pstmt = conn.prepareStatement(q);
      pstmt.setInt(1, id);
      pstmt.setString(2, code);
      pstmt.setString(3, objCode);
      pstmt.setInt(4, no);
      pstmt.setString(5, detail);
      pstmt.executeUpdate();
      JOptionPane.showMessageDialog(null, "Record Inserted.");
    } catch (Exception e) {
      JOptionPane.showMessageDialog(null, "Insertion Failed.");
    }
  }
}

class CourseApp extends Frame implements ActionListener {
  Button createBtn = new Button("Create");
  Button updateBtn = new Button("Update");
  Button retrieveBtn = new Button("Retrieve");
  Button deleteBtn = new Button("Delete");

  Panel optionPanel = new Panel();
  Panel actionPanel = new Panel();

  CourseApp() {
    setTitle("Course Objective Manager");
    setSize(800, 700);
    setLayout(new BorderLayout());

    optionPanel.setLayout(new GridLayout(1, 4));
    optionPanel.add(createBtn);
    optionPanel.add(updateBtn);
    optionPanel.add(retrieveBtn);
    optionPanel.add(deleteBtn);

    add(optionPanel, BorderLayout.NORTH);
```

```
    add(actionPanel, BorderLayout.CENTER);

    createBtn.addActionListener(this);
    updateBtn.addActionListener(this);
    retrieveBtn.addActionListener(this);
    deleteBtn.addActionListener(this);

    addWindowListener(new WindowAdapter() {
      public void windowClosing(WindowEvent e) {
        dispose();
      }
    });

    DB.connect("C:\\Users\\lenovo\\OneDrive\\Desktop\\Source
Code\\courseobjective.db");
    setVisible(true);
    CourseObjective_Retrieve(); // Load retrieve view by default
    actionPanel.revalidate();
    actionPanel.repaint();
  }

  public void actionPerformed(ActionEvent e) {
    actionPanel.removeAll();
    if (e.getSource() == createBtn)
      CourseObjective_CreateForm();
    else if (e.getSource() == updateBtn)
      CourseObjective_Update();
    else if (e.getSource() == retrieveBtn)
      CourseObjective_Retrieve();
    else if (e.getSource() == deleteBtn)
      CourseObjective_Delete();
    actionPanel.revalidate();
    actionPanel.repaint();
  }

  void CourseObjective_CreateForm() {
    actionPanel.setLayout(new GridLayout(6, 2));
    TextField id = new TextField();
    TextField tf1 = new TextField();
    TextField tf2 = new TextField();
```

```
  TextField tf3 = new TextField();
  TextField tf4 = new TextField();
  Button save = new Button("Save");

  save.addActionListener(ae -> {
    try {
      DB.CourseObjective_Create(
          Integer.parseInt(id.getText()),
          tf1.getText(),
          tf2.getText(),
          Integer.parseInt(tf3.getText()),
          tf4.getText());
      id.setText("");
      tf1.setText("");
      tf2.setText("");
      tf3.setText("");
      tf4.setText("");
    } catch (Exception ex) {
      JOptionPane.showMessageDialog(null, "Invalid Input.");
    }
  });

  actionPanel.add(new Label("ID:"));
  actionPanel.add(id);
  actionPanel.add(new Label("Course ID:"));
  actionPanel.add(tf1);
  actionPanel.add(new Label("Objective Code:"));
  actionPanel.add(tf2);
  actionPanel.add(new Label("Objective No:"));
  actionPanel.add(tf3);
  actionPanel.add(new Label("Objective Detail:"));
  actionPanel.add(tf4);
  actionPanel.add(new Label(""));
  actionPanel.add(save);
}

void CourseObjective_Update() {
  actionPanel.setLayout(new BorderLayout());

  Panel formPanel = new Panel(new GridLayout(10, 1));
```

```
    TextArea allData = new TextArea(15, 80);
    allData.setFont(new Font("Monospaced", Font.PLAIN, 12));
    ScrollPane scrollPane = new ScrollPane();
    scrollPane.add(allData);

    TextField idField = new TextField();
    TextField tf1 = new TextField();
    TextField tf2 = new TextField();
    TextField tf3 = new TextField();
    TextField tf4 = new TextField();
    Button load = new Button("Load");
    Button update = new Button("Update");

    try {
      Statement stmt = DB.conn.createStatement();
      ResultSet rs = stmt.executeQuery("SELECT * FROM courseobjective");
      StringBuilder sb = new StringBuilder();
      sb.append(String.format("%-4s %-10s %-15s %-10s %-50s\n", "ID",
"cour_id", "obj_code", "obj_no",
          "obj_detail"));
      sb.append("-----------------------------------------------------------
------------------------------\n");
      while (rs.next()) {
        sb.append(String.format(
            "%-4d %-10s %-15s %-10d %-50s\n",
            rs.getInt("ID"),
            rs.getString("cour_id"),
            rs.getString("cour_obj_code"),
            rs.getInt("cour_obj_no"),
            rs.getString("cour_obj_detail")));
      }
      allData.setText(sb.toString());
    } catch (Exception ex) {
    }

    load.addActionListener(ae -> {
      try {
        int id = Integer.parseInt(idField.getText());
        PreparedStatement ps = DB.conn.prepareStatement("SELECT * FROM
courseobjective WHERE ID = ?");
```

```
        ps.setInt(1, id);
        ResultSet rs = ps.executeQuery();
        if (rs.next()) {
          tf1.setText(rs.getString("cour_id"));
          tf2.setText(rs.getString("cour_obj_code"));
          tf3.setText(String.valueOf(rs.getInt("cour_obj_no")));
          tf4.setText(rs.getString("cour_obj_detail"));
        }
      } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, "Load Failed.");
      }
    });

    update.addActionListener(ae -> {
      try {
        int id = Integer.parseInt(idField.getText());
        String q = "UPDATE courseobjective SET cour_id=?, cour_obj_code=?,
cour_obj_no=?, cour_obj_detail=? WHERE ID=?";
        PreparedStatement ps = DB.conn.prepareStatement(q);
        ps.setString(1, tf1.getText());
        ps.setString(2, tf2.getText());
        ps.setInt(3, Integer.parseInt(tf3.getText()));
        ps.setString(4, tf4.getText());
        ps.setInt(5, id);
        ps.executeUpdate();
        JOptionPane.showMessageDialog(null, "Record Updated.");
      } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, "Update Failed.");
      }
    });

    formPanel.add(new Label("Enter ID to Update:"));
    formPanel.add(idField);
    formPanel.add(load);
    formPanel.add(new Label("Course ID:"));
    formPanel.add(tf1);
    formPanel.add(new Label("Objective Code:"));
    formPanel.add(tf2);
    formPanel.add(new Label("Objective No:"));
    formPanel.add(tf3);
```

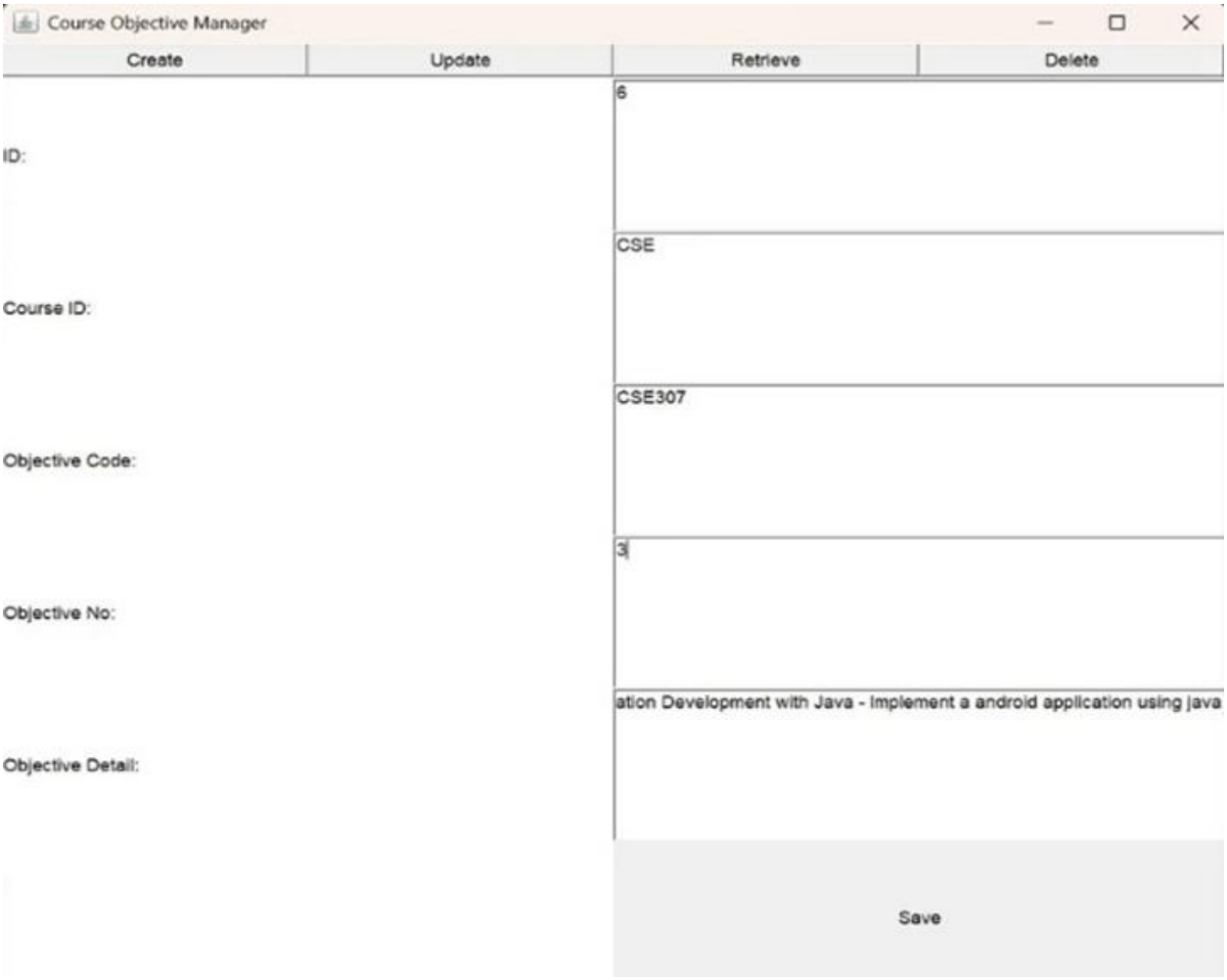
```
    formPanel.add(new Label("Objective Detail:"));
    formPanel.add(tf4);
    formPanel.add(update);

    actionPanel.add(scrollPane, BorderLayout.CENTER);
    actionPanel.add(formPanel, BorderLayout.SOUTH);
  }

  void CourseObjective_Retrieve() {
    actionPanel.setLayout(new BorderLayout());
    TextArea output = new TextArea();
    output.setFont(new Font("Monospaced", Font.PLAIN, 12));
    try {
      Statement stmt = DB.conn.createStatement();
      ResultSet rs = stmt.executeQuery("SELECT * FROM courseobjective");
      StringBuilder sb = new StringBuilder();
      sb.append(String.format("%-4s %-10s %-15s %-10s %-50s\n", "ID",
"cour_id", "obj_code", "obj_no",
          "obj_detail"));
      sb.append("-----------------------------------------------------------
------------------------------\n");
      while (rs.next()) {
        sb.append(String.format(
            "%-4d %-10s %-15s %-10d %-50s\n",
            rs.getInt("ID"),
            rs.getString("cour_id"),
            rs.getString("cour_obj_code"),
            rs.getInt("cour_obj_no"),
            rs.getString("cour_obj_detail")));
      }
      output.setText(sb.toString());
    } catch (Exception ex) {
      JOptionPane.showMessageDialog(null, "Retrieve Failed.");
    }
    actionPanel.add(output, BorderLayout.CENTER);
  }

  void CourseObjective_Delete() {
    actionPanel.setLayout(new BorderLayout());
```

```
    TextArea dataArea = new TextArea(15, 80);
    dataArea.setFont(new Font("Monospaced", Font.PLAIN, 12));
    ScrollPane scroll = new ScrollPane();
    scroll.add(dataArea);

    Panel form = new Panel(new GridLayout(2, 2));
    TextField idField = new TextField();
    Button delete = new Button("Delete");

    try {
      Statement stmt = DB.conn.createStatement();
      ResultSet rs = stmt.executeQuery("SELECT * FROM courseobjective");
      StringBuilder sb = new StringBuilder();
      sb.append(String.format("%-4s %-10s %-15s %-10s %-50s\n", "ID",
"cour_id", "obj_code", "obj_no",
          "obj_detail"));
      sb.append("-----------------------------------------------------------
------------------------------\n");
      while (rs.next()) {
        sb.append(String.format(
            "%-4d %-10s %-15s %-10d %-50s\n",
            rs.getInt("ID"),
            rs.getString("cour_id"),
            rs.getString("cour_obj_code"),
            rs.getInt("cour_obj_no"),
            rs.getString("cour_obj_detail")));
      }
      dataArea.setText(sb.toString());
    } catch (Exception ex) {
    }

    delete.addActionListener(ae -> {
      try {
        int id = Integer.parseInt(idField.getText());
        PreparedStatement ps = DB.conn.prepareStatement("DELETE FROM
courseobjective WHERE ID = ?");
        ps.setInt(1, id);
        ps.executeUpdate();
        JOptionPane.showMessageDialog(null, "Record Deleted.");
        idField.setText("");
```

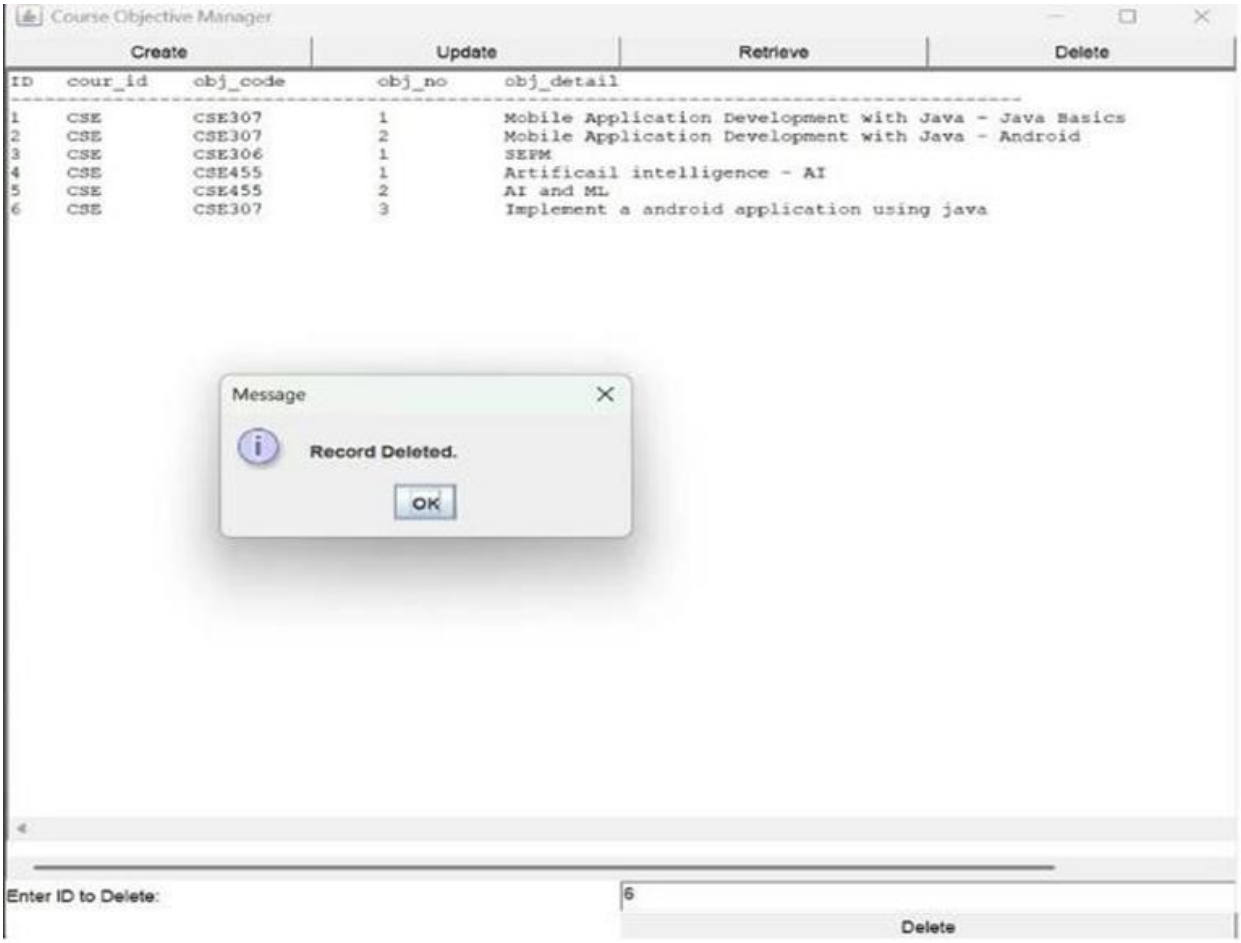
```
      } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, "Delete Failed.");
      }
    });

    form.add(new Label("Enter ID to Delete:"));
    form.add(idField);
    form.add(new Label(""));
    form.add(delete);

    actionPanel.add(scroll, BorderLayout.CENTER);
    actionPanel.add(form, BorderLayout.SOUTH);
  }
}

public class CodeBrew_CourseObjective {
  public static void main(String[] args) {
    new CourseApp();
  }
}
```

Screen Shots

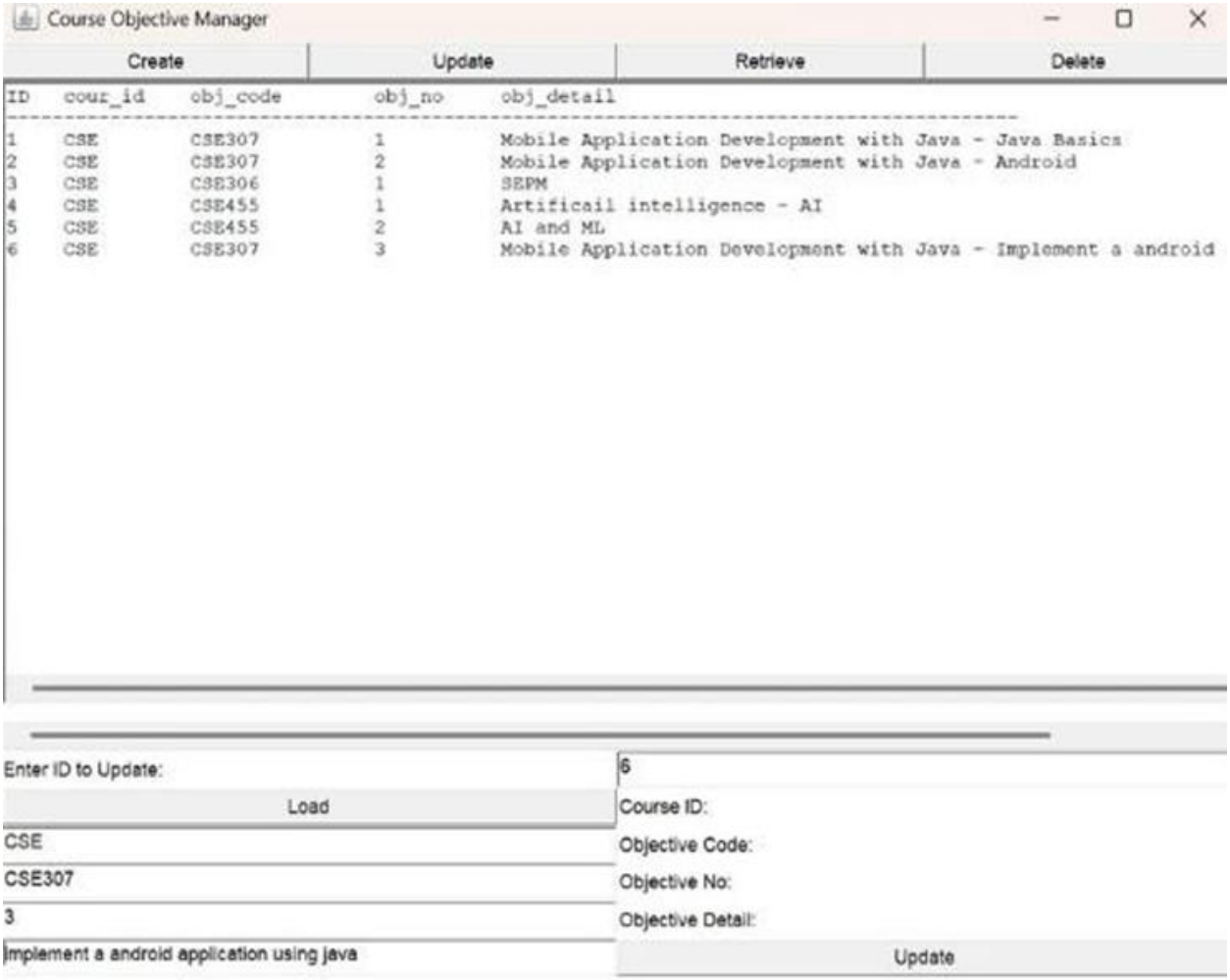
Create:



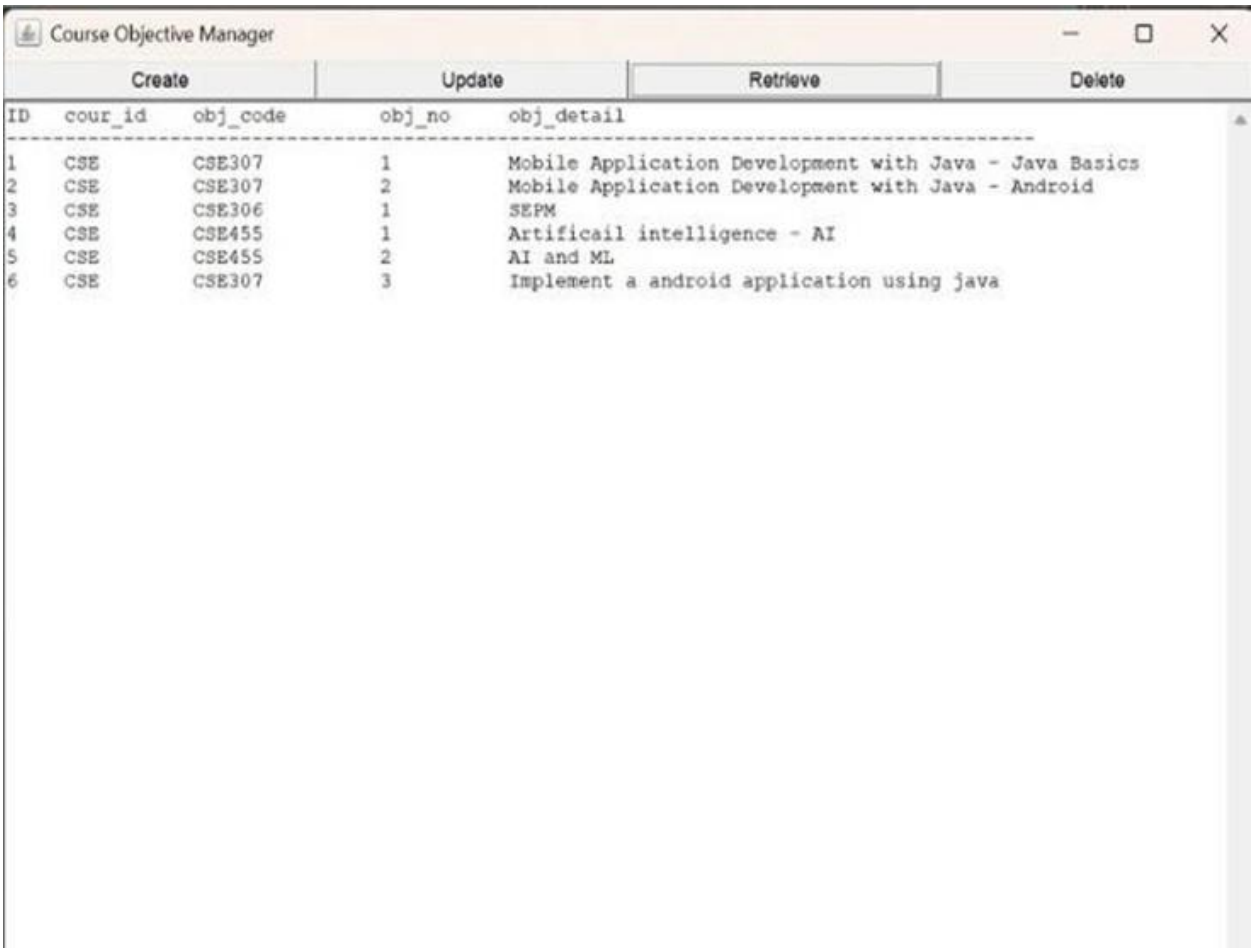
Delete:


Course Objective Manager
Create
Update
Retrieve
Delete
ID cour_id obj_code obj_no obj_detail
1 CSE CSE307 1 Mobile Application Development with Java - Java Basics
2 CSE CSE307 2 Mobile Application Development with Java - Android
3 CSE CSE306 1 SEPM
4 CSE CSE455 1 Artificail intelligence - AI
5 CSE CSE455 2 AI and ML
6 CSE CSE307 3 Implement a android application using java
Message
Record Deleted.
OK
Enter ID to Delete:
6
Delete

Update:


Course Objective Manager
Create
Update
Retrieve
Delete
ID cour_id obj_code obj_no obj_detail
1 CSE CSE307 1 Mobile Application Development with Java - Java Basics
2 CSE CSE307 2 Mobile Application Development with Java - Android
3 CSE CSE306 1 SEPM
4 CSE CSE455 1 Artificail intelligence - AI
5 CSE CSE455 2 AI and ML
6 CSE CSE307 3 Mobile Application Development with Java - Implement a android
Enter ID to Update:
6
Load
Course ID:
CSE
Objective Code:
CSE307
Objective No:
3
Objective Detail:
Implement a android application using java
Update

Retrieve:


Course Objective Manager
Create
Update
Retrieve
Delete
ID cour_id obj_code obj_no obj_detail
1 CSE CSE307 1 Mobile Application Development with Java - Java Basics
2 CSE CSE307 2 Mobile Application Development with Java - Android
3 CSE CSE306 1 SEPM
4 CSE CSE455 1 Artificail intelligence - AI
5 CSE CSE455 2 AI and ML
6 CSE CSE307 3 Implement a android application using java

Conclusion

The *Course Objective Setting* module successfully supports the OBE implementation at SRM University-AP by allowing easy creation, updating, and management of course objectives. With a simple CRUD functionality and integration into the overall academic system, it helps streamline the process and ensures better alignment of course goals with program outcomes.

This module ensures that course objectives are clearly defined and easily maintained, which is essential for continuous improvement in teaching and learning. By integrating it with other academic modules, it strengthens the overall structure of the OBE system and helps faculty align course delivery with desired outcomes.

In addition, the use of a user-friendly interface and structured database support enables academic departments to manage large sets of course-related data efficiently. This not only saves time but also minimizes errors in documentation and reporting, which are key components in accreditation processes and quality assurance.

Overall, this project has contributed to the digitalization of academic workflows at SRM University-AP. By supporting transparency, consistency, and easy accessibility of course objectives, it aids in building a more outcome-focused and student-centric education system.