**OBE IMPLEMENTATION:COURSE UTILIZATION SETTING**

***by***

**Chelikani Aakash [AP22110010225]**

**Mopuru Tharun [AP22110010230]**

**Potugari Venkata Ravi Teja [AP22110010256]**

**Tentu Navadeep Ram [AP22110010265]**

**Kocherlakota Sundaram [AP22110010266]**

*A report for the CS307:Mobile Application Development using JAVA*

**A logo of a tree

AI-generated content may be incorrect.**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**SRM UNIVERSITY AP::AMARAVATI**

**INDEX**

Contents

[Introduction 3](#_Toc195520801)

[Project Modules: 3](#_Toc195520802)

[Various Modules available in the project are: 3](#_Toc195520803)

[Architecture Diagram 4](#_Toc195520804)

[Module Description 4](#_Toc195520805)

[Programming Details naming conventions to be used: 4](#_Toc195520806)

[Table details: 5](#_Toc195520807)

[Source Code 5](#_Toc195520808)

[Class1: narst\_course\_utilisation\_setting.java; 5](#_Toc195520809)

[Class2: SQLDB.java; 9](#_Toc195520810)

[Screen Shots 10](#_Toc195520811)

[Conclusion 14](#_Toc195520813)

# 

# Introduction

Our university (here with 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

**A screenshot of a computer

AI-generated content may be incorrect.**



# Module Description

**Module Name:** Course Utilization Setting

**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(Sqlite3).

## Programming Details naming conventions to be used:

* **Class name/Activity name:** narst\_ CourseUtilizationSetting
* **Function/method name:**
  + **Create:** narst\_ CourseUtilizationSetting \_create
  + **Update:** narst\_ CourseUtilizationSetting \_update
  + **Retrieve:** narst\_ CourseUtilizationSetting \_retrive
  + **Delete:** narst\_ CourseUtilizationSetting \_delete

## Table details:

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Id | Integer |
| Cour\_id | String |
| Cour\_util\_code | String |
| Unit\_no | String |
| Unit\_name | String |
| Require\_contact\_hr | String |
| Cour\_out\_id | String |
| Ref\_id | String |

# Source Code

## Class1: narst\_course\_utilisation\_setting.java;

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package narst\_course\_utilisation\_setting;

/\*\*

\*

\* @author tharu

\*/

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

import javax.swing.table.DefaultTableModel;

public class narst\_course\_utilisation\_setting extends JFrame implements ActionListener {

private JLabel[] labels = new JLabel[8];

private JTextField[] fields = new JTextField[8];

private String[] fieldNames = {"ID", "Course ID", "Course Utilization Code", "Unit Number", "Unit Name","Required Contact Hours", "Course Outcome ID", "Reference ID"};

private JButton insertBtn = new JButton("Insert");

private JButton updateBtn = new JButton("Update");

private JButton deleteBtn = new JButton("Delete");

private JButton retrieveBtn = new JButton("Retrieve");

private JTable table;

private DefaultTableModel tableModel;

public narst\_course\_utilisation\_setting() {

setTitle("Course Utilisation Setting");

setSize(1000, 600);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setLayout(new BorderLayout(10, 10));

JPanel inputPanel = new JPanel(new GridLayout(8, 2, 5, 5));

for (int i = 0; i < 8; i++) {

labels[i] = new JLabel(fieldNames[i]);

fields[i] = new JTextField();

inputPanel.add(labels[i]);

inputPanel.add(fields[i]);

}

JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 10, 10));

buttonPanel.add(insertBtn);

buttonPanel.add(updateBtn);

buttonPanel.add(deleteBtn);

buttonPanel.add(retrieveBtn);

String[] columnNames = {"ID", "Course ID", "Course Utilization Code", "Unit Number", "Unit Name","Required Contact Hours", "Course Outcome ID", "Reference ID"};

tableModel = new DefaultTableModel(columnNames, 0);

table = new JTable(tableModel);

JScrollPane tableScroll = new JScrollPane(table);

JPanel mainPanel = new JPanel(new BorderLayout(10, 10));

mainPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

mainPanel.add(inputPanel, BorderLayout.NORTH);

mainPanel.add(buttonPanel, BorderLayout.CENTER);

mainPanel.add(tableScroll, BorderLayout.SOUTH);

insertBtn.addActionListener(this);

updateBtn.addActionListener(this);

deleteBtn.addActionListener(this);

retrieveBtn.addActionListener(this);

table.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

int row = table.getSelectedRow();

if (row >= 0) {

for (int i = 0; i < fields.length; i++) {

fields[i].setText(tableModel.getValueAt(row, i).toString());

}

}

}

});

add(mainPanel);

SQLDB.connect("C:\\Users\\tharu\\OneDrive\\Desktop\\Apps\\javaapp.db");

refreshTable();

}

private void refreshTable() {

try {

tableModel.setRowCount(0);

ResultSet rs = SQLDB.executeQuery("SELECT \* FROM course\_utilization");

while (rs.next()) {

Object[] row = {

rs.getString("ID"),

rs.getString("cour\_id"),

rs.getString("cour\_util\_code"),

rs.getString("unit\_no"),

rs.getString("unit\_name"),

rs.getString("require\_contact\_hr"),

rs.getString("cour\_out\_id"),

rs.getString("ref\_id")

};

tableModel.addRow(row);

}

} catch (SQLException ex) {

JOptionPane.showMessageDialog(this, "Error loading data: " + ex.getMessage());

}

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == insertBtn) {

narst\_CourseUtilizationSetting\_create();

} else if (e.getSource() == updateBtn) {

narst\_CourseUtilizationSetting\_update();

} else if (e.getSource() == deleteBtn) {

narst\_CourseUtilizationSetting\_delete();

} else if (e.getSource() == retrieveBtn) {

narst\_CourseUtilizationSetting\_retrieve();

}

refreshTable();

}

private void narst\_CourseUtilizationSetting\_create(){

try (PreparedStatement pstmt = SQLDB.conn.prepareStatement(

"INSERT INTO course\_utilization VALUES(?,?,?,?,?,?,?,?)")) {

for (int i = 0; i < 8; i++) {

pstmt.setString(i + 1, fields[i].getText());

}

pstmt.executeUpdate();

JOptionPane.showMessageDialog(this, "Record inserted successfully!");

clearFields();

} catch (SQLException ex) {

JOptionPane.showMessageDialog(this, "Insert error: " + ex.getMessage());

}

}

private void narst\_CourseUtilizationSetting\_update() {

String id = fields[0].getText();

if (id.isEmpty()) {

JOptionPane.showMessageDialog(this, "Enter Data to Update!");

return;

}

try (PreparedStatement pstmt = SQLDB.conn.prepareStatement(

"UPDATE course\_utilization SET cour\_id=?, cour\_util\_code=?, unit\_no=?, " +

"unit\_name=?, require\_contact\_hr=?, cour\_out\_id=?, ref\_id=? WHERE ID=?")) {

for (int i = 0; i < 7; i++) {

pstmt.setString(i + 1, fields[i+1].getText());

}

pstmt.setString(8, fields[0].getText());

pstmt.executeUpdate();

JOptionPane.showMessageDialog(this, "Record updated successfully!");

clearFields();

} catch (SQLException ex) {

JOptionPane.showMessageDialog(this, "Update error: " + ex.getMessage());

}

}

private void narst\_CourseUtilizationSetting\_delete() {

String id = fields[0].getText();

if (id.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please select a record to delete!");

return;

}

try (PreparedStatement pstmt = SQLDB.conn.prepareStatement(

"DELETE FROM course\_utilization WHERE ID=?")) {

pstmt.setString(1, id);

pstmt.executeUpdate();

JOptionPane.showMessageDialog(this, "Record deleted successfully!");

clearFields();

} catch (SQLException ex) {

JOptionPane.showMessageDialog(this, "Delete error: " + ex.getMessage());

}

}

private void narst\_CourseUtilizationSetting\_retrieve() {

try {

tableModel.setRowCount(0);

ResultSet rs = SQLDB.executeQuery("SELECT \* FROM course\_utilization");

while (rs.next()) {

Object[] row = {

rs.getString("ID"),

rs.getString("cour\_id"),

rs.getString("cour\_util\_code"),

rs.getString("unit\_no"),

rs.getString("unit\_name"),

rs.getString("require\_contact\_hr"),

rs.getString("cour\_out\_id"),

rs.getString("ref\_id")

};

tableModel.addRow(row);

}

JOptionPane.showMessageDialog(this, "Data retrieved successfully!");

} catch (SQLException ex) {

JOptionPane.showMessageDialog(this, "Retrieve error: " + ex.getMessage());

}

}

private void clearFields() {

for (JTextField field : fields) {

field.setText("");

}

}

public static void main(String[] args) {

new narst\_course\_utilisation\_setting().setVisible(true);

}

}

## Class2: SQLDB.java;

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package narst\_course\_utilisation\_setting;

/\*\*

\*

\* @author tharu

\*/

import java.sql.\*;

public 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 ResultSet executeQuery(String query) throws SQLException {

Statement stmt = conn.createStatement();

return stmt.executeQuery(query);

}

public static void update(String query) {

try {

stmt.executeUpdate(query);

} catch (Exception e) {

e.printStackTrace();

}

}

}

# Screen Shots

#### Home Page:

A screenshot of a computer

AI-generated content may be incorrect.

#### Insertion:

Entering Data to Insert:

A screenshot of a computer

AI-generated content may be incorrect.

After Entering Data Press Insert:

A screenshot of a computer

AI-generated content may be incorrect.

After Data has been Inserted into the table:

A screenshot of a computer

AI-generated content may be incorrect.

### Update:

Select a Row the data will be displayed in the form then Update the Data and then Press UPDATE button.

A screenshot of a computer

AI-generated content may be incorrect.

After Updating:

A screenshot of a computer

AI-generated content may be incorrect.

#### Delete:

Select a Row the data will be displayed in the form then press DELETE.

A screenshot of a computer

AI-generated content may be incorrect.

Deletion:

A screenshot of a computer

AI-generated content may be incorrect.

Table After Deletion:

A screenshot of a computer

AI-generated content may be incorrect.

#### Retrieve:

Press Retrieve Button to Retrieve all Data.

A screenshot of a computer

AI-generated content may be incorrect.

# Conclusion

In conclusion, the Course Utilization Setting module successfully demonstrates the effective implementation of an OBE (Outcome Based Education) framework within the SRM-AP university setting. The module is a robust solution for managing key aspects of course data through its comprehensive CURD operations Create, Update, Retrieve, and Delete. By integrating a structured SQLite3 database with Java-based application programming, the system ensures efficient data management, thereby promoting reliability and user accessibility.

This project not only highlights clear module architecture and well-documented code samples but also provides detailed visual evidence through screen shots for each stage of the module's functionality. The careful design of field/table details, along with the modular programming approach, underscores the system’s scalability and adaptability to evolving academic requirements.