**Dr. Mahalingam College of Engineering and Technology, Pollachi-642003**

**Department of Information Technology**

**Academic Year: 2024 - 2025**

**TRL and SDG Certificate**

**Project Title :** **Telecommunication Management System**

**Course Code & Name : 23ITI402 Database Management Systems**

**Department and Semester: Information Technology & IV SEM**

**Technology Readiness Level (TRL) of the Project : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Sustainability Development Goals (SDG)-Goal Name : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Names and Roll Numbers of Student** | **Signature of the Student** |
| 01. | Nishaanth.R |  |

|  |  |
| --- | --- |
| **Preparation (25 marks)** |  |
| **Interpretation (30 marks)** |  |
| **Output & Result (15 marks)** |  |
| **Viva (5 marks)** |  |
| **Total (75 marks)** |  |
| **Signature of the faculty incharge** |  |

1. **Objective of the Project (1 page)**

The primary objective of this project is to design and develop a desktop-based *Telecommunication Management System* that allows administrators to manage customer data efficiently and effectively. With the growing number of telecom users, maintaining accurate and up-to-date records has become a significant challenge for service providers. This system aims to address that challenge by providing a user-friendly graphical interface built using Java Swing and backed by a robust MySQL database.Additionally, the system includes intelligent features such as:

* Live Search and Filtering: Enables real-time search based on name, mobile number, or status.
* Status Highlighting: Automatically highlights customers with "Suspended" or "Inactive" status using color-coded rows (e.g., red and gray) for easy visual tracking.
* Sorting and Validation: Ensures accurate data entry and helps in managing subscribers based on plans, status, or other attributes.

By integrating Java Swing for the front-end and JDBC for the database connectivity, the system provides a reliable platform for telecom administrators to perform operations without needing deep technical knowledge of databases. This project not only demonstrates a practical application of database management concepts but also reinforces principles of software design, user interface development, and system integration.

1. **Entire Project Code**

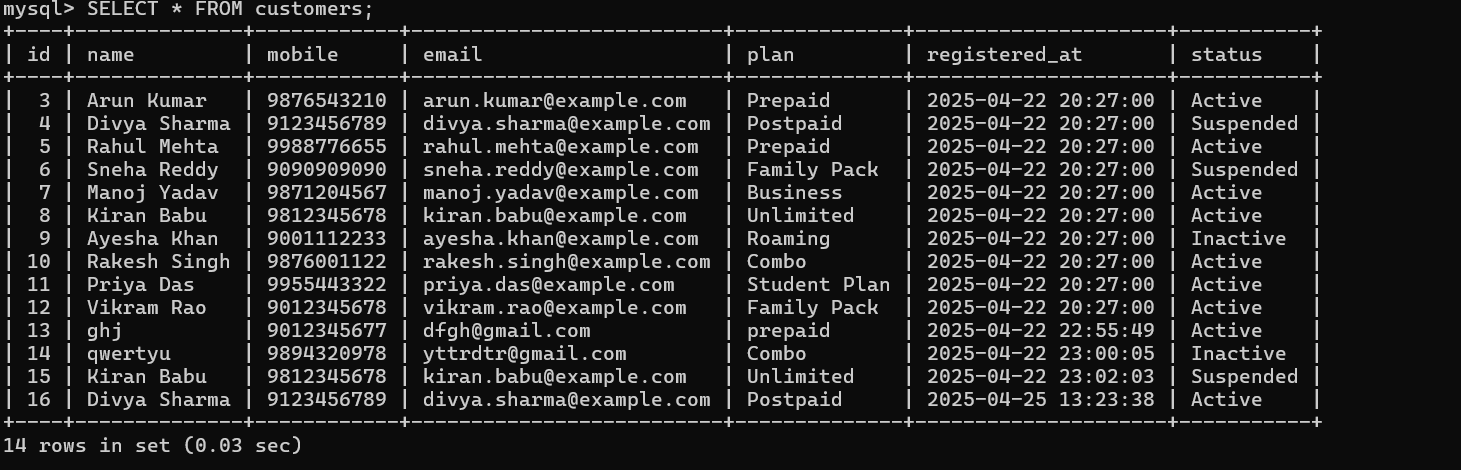
* **Login Page**

import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.\*;  
import java.sql.\*;  
  
public class LoginPage extends JFrame {  
 private JTextField usernameField;  
 private JPasswordField passwordField;  
  
 private final String DB\_URL = "jdbc:mysql://localhost:3306/telecom";  
 private final String DB\_USER = "root";  
 private final String DB\_PASS = "dbms";  
  
 public LoginPage() {  
 setTitle("Login - Telecommunication System");  
 setSize(400, 300);  
 setLocationRelativeTo(null);  
 setDefaultCloseOperation(*EXIT\_ON\_CLOSE*);  
 setResizable(false);  
  
 JPanel panel = new JPanel();  
 panel.setLayout(null);  
 panel.setBackground(new Color(230, 240, 255));  
  
 JLabel titleLabel = new JLabel("Telecom Login");  
 titleLabel.setFont(new Font("Arial", Font.*BOLD*, 24));  
 titleLabel.setBounds(110, 20, 200, 30);  
 panel.add(titleLabel);  
  
 JLabel userLabel = new JLabel("Username:");  
 userLabel.setBounds(50, 80, 100, 25);  
 panel.add(userLabel);  
  
 usernameField = new JTextField();  
 usernameField.setBounds(150, 80, 180, 25);  
 panel.add(usernameField);  
  
 JLabel passLabel = new JLabel("Password:");  
 passLabel.setBounds(50, 120, 100, 25);  
 panel.add(passLabel);  
  
 passwordField = new JPasswordField();  
 passwordField.setBounds(150, 120, 180, 25);  
 panel.add(passwordField);  
  
 JButton loginBtn = new JButton("Login");  
 loginBtn.setBounds(140, 180, 100, 30);  
 panel.add(loginBtn);  
  
 loginBtn.addActionListener(e -> authenticate());  
  
 add(panel);  
 }  
  
 private void authenticate() {  
 String username = usernameField.getText();  
 String password = String.*valueOf*(passwordField.getPassword());  
  
 if (username.isEmpty() || password.isEmpty()) {  
 JOptionPane.*showMessageDialog*(this, "Please fill in both fields.");  
 return;  
 }  
  
 try (Connection conn = DriverManager.*getConnection*(DB\_URL, DB\_USER, DB\_PASS)) {  
 String query = "SELECT \* FROM users WHERE username = ? AND password = ?";  
 PreparedStatement stmt = conn.prepareStatement(query);  
 stmt.setString(1, username);  
 stmt.setString(2, password);  
 ResultSet rs = stmt.executeQuery();  
  
 if (rs.next()) {  
 JOptionPane.*showMessageDialog*(this, "Login successful!");  
 this.dispose(); // Close login page  
 new Tele().setVisible(true); // Open the main application  
 } else {  
 JOptionPane.*showMessageDialog*(this, "Invalid username or password.");  
 }  
  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 JOptionPane.*showMessageDialog*(this, "Database error.");  
 }  
 }  
  
 public static void main(String[] args) {  
 SwingUtilities.*invokeLater*(() -> new LoginPage().setVisible(true)); // Show login first  
 }  
}

* **Dashboard Page**

import javax.swing.\*;  
import javax.swing.border.TitledBorder;  
import javax.swing.table.DefaultTableCellRenderer;  
import javax.swing.table.DefaultTableModel;  
import java.awt.\*;  
import java.awt.event.\*;  
import java.sql.\*;  
import java.util.regex.\*;  
import java.util.\*;  
  
public class Tele extends JFrame {  
 private JTextField nameField, mobileField, emailField, planField, searchField;  
 private JComboBox<String> statusComboBox;  
 private JTable table;  
 private DefaultTableModel tableModel;  
 private int selectedCustomerId = -1;  
  
 private JButton addButton, updateButton, deleteButton, clearButton, loadButton, searchButton, sortButton;  
  
 private final String DB\_URL = "jdbc:mysql://localhost:3306/telecom";  
 private final String DB\_USER = "root";  
 private final String DB\_PASS = "dbms";  
  
 public Tele() {  
 setTitle("Telecommunication Management System");  
 setSize(900, 600);  
 setLocationRelativeTo(null);  
 setDefaultCloseOperation(*EXIT\_ON\_CLOSE*);  
  
 JPanel mainPanel = new JPanel(new BorderLayout(10, 10));  
 mainPanel.setBorder(BorderFactory.*createEmptyBorder*(10, 10, 10, 10));  
  
 JPanel customerFormPanel = new JPanel(new GridLayout(7, 2, 10, 10));  
 customerFormPanel.setBorder(BorderFactory.*createTitledBorder*(BorderFactory.*createEtchedBorder*(), "Customer Details", TitledBorder.*LEFT*, TitledBorder.*TOP*));  
  
 nameField = new JTextField();  
 mobileField = new JTextField();  
 emailField = new JTextField();  
 planField = new JTextField();  
 searchField = new JTextField();  
 statusComboBox = new JComboBox<>(new String[]{"Active", "Inactive", "Suspended"});  
  
 customerFormPanel.add(new JLabel("Name:"));  
 customerFormPanel.add(nameField);  
  
 customerFormPanel.add(new JLabel("Mobile:"));  
 customerFormPanel.add(mobileField);  
  
 customerFormPanel.add(new JLabel("Email:"));  
 customerFormPanel.add(emailField);  
  
 customerFormPanel.add(new JLabel("Plan:"));  
 customerFormPanel.add(planField);  
  
 customerFormPanel.add(new JLabel("Status:"));  
 customerFormPanel.add(statusComboBox);  
  
 customerFormPanel.add(new JLabel("Search (Name/Mobile):"));  
 customerFormPanel.add(searchField);  
  
 JPanel buttonPanel = new JPanel(new GridLayout(1, 7, 5, 5));  
 addButton = new JButton("Add");  
 updateButton = new JButton("Update");  
 deleteButton = new JButton("Delete");  
 clearButton = new JButton("Clear");  
 loadButton = new JButton("Load All");  
 searchButton = new JButton("Search");  
 sortButton = new JButton("Sort by Name");  
  
 buttonPanel.add(addButton);  
 buttonPanel.add(updateButton);  
 buttonPanel.add(deleteButton);  
 buttonPanel.add(clearButton);  
 buttonPanel.add(loadButton);  
 buttonPanel.add(searchButton);  
 buttonPanel.add(sortButton);  
  
 customerFormPanel.add(buttonPanel);  
  
 tableModel = new DefaultTableModel(new String[]{"ID", "Name", "Mobile", "Email", "Plan", "Status", "Registered"}, 0);  
 table = new JTable(tableModel);  
  
 // Feature 10: Highlight suspended users in red and inactive users in gray  
 table.setDefaultRenderer(Object.class, new DefaultTableCellRenderer() {  
 @Override  
 public Component getTableCellRendererComponent(JTable table, Object value, boolean isSelected, boolean hasFocus, int row, int column) {  
 Component c = super.getTableCellRendererComponent(table, value, isSelected, hasFocus, row, column);  
 String status = table.getValueAt(row, 5).toString(); // Status column is at index 5  
  
 if (status.equalsIgnoreCase("Suspended")) {  
 c.setForeground(Color.*RED*); // Highlight Suspended status in Red  
 } else if (status.equalsIgnoreCase("Inactive")) {  
 c.setForeground(Color.*GRAY*); // Highlight Inactive status in Gray  
 } else {  
 c.setForeground(Color.*BLACK*); // Default color for other statuses  
 }  
  
 return c;  
 }  
 });  
  
 JScrollPane tableScrollPane = new JScrollPane(table);  
 tableScrollPane.setBorder(BorderFactory.*createTitledBorder*(BorderFactory.*createEtchedBorder*(), "Customer Records", TitledBorder.*LEFT*, TitledBorder.*TOP*));  
  
 mainPanel.add(customerFormPanel, BorderLayout.*NORTH*);  
 mainPanel.add(tableScrollPane, BorderLayout.*CENTER*);  
  
 add(mainPanel);  
  
 setupButtonListeners();  
 setupKeyboardShortcuts(); // Feature 11  
  
 table.addMouseListener(new MouseAdapter() {  
 public void mouseClicked(MouseEvent e) {  
 int row = table.getSelectedRow();  
 if (row >= 0) {  
 selectedCustomerId = (int) tableModel.getValueAt(row, 0);  
 nameField.setText((String) tableModel.getValueAt(row, 1));  
 mobileField.setText((String) tableModel.getValueAt(row, 2));  
 emailField.setText((String) tableModel.getValueAt(row, 3));  
 planField.setText((String) tableModel.getValueAt(row, 4));  
 statusComboBox.setSelectedItem((String) tableModel.getValueAt(row, 5));  
 }  
 }  
 });  
  
 searchField.getDocument().addDocumentListener(new javax.swing.event.DocumentListener() {  
 public void insertUpdate(javax.swing.event.DocumentEvent e) { liveSearch(); }  
 public void removeUpdate(javax.swing.event.DocumentEvent e) { liveSearch(); }  
 public void changedUpdate(javax.swing.event.DocumentEvent e) { liveSearch(); }  
 });  
  
 loadCustomers();  
 }  
  
 private void setupKeyboardShortcuts() {  
 KeyStroke saveKey = KeyStroke.*getKeyStroke*(KeyEvent.*VK\_S*, InputEvent.*CTRL\_DOWN\_MASK*);  
 KeyStroke deleteKey = KeyStroke.*getKeyStroke*(KeyEvent.*VK\_DELETE*, 0);  
  
 getRootPane().getInputMap(JComponent.*WHEN\_IN\_FOCUSED\_WINDOW*).put(saveKey, "save");  
 getRootPane().getActionMap().put("save", new AbstractAction() {  
 public void actionPerformed(ActionEvent e) { addCustomer(); }  
 });  
  
 getRootPane().getInputMap(JComponent.*WHEN\_IN\_FOCUSED\_WINDOW*).put(deleteKey, "delete");  
 getRootPane().getActionMap().put("delete", new AbstractAction() {  
 public void actionPerformed(ActionEvent e) { deleteCustomer(); }  
 });  
 }  
  
 private void setupButtonListeners() {  
 addButton.addActionListener(e -> addCustomer());  
 updateButton.addActionListener(e -> updateCustomer());  
 deleteButton.addActionListener(e -> deleteCustomer());  
 clearButton.addActionListener(e -> clearFields());  
 loadButton.addActionListener(e -> loadCustomers());  
 searchButton.addActionListener(e -> searchCustomers());  
 sortButton.addActionListener(e -> sortCustomersByName());  
 }  
  
 private Connection getConnection() {  
 try {  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
 return DriverManager.*getConnection*(DB\_URL, DB\_USER, DB\_PASS);  
 } catch (Exception e) {  
 e.printStackTrace();  
 JOptionPane.*showMessageDialog*(this, "Database connection failed");  
 return null;  
 }  
 }  
  
 private void addCustomer() {  
 if (!validateFields()) return;  
 try (Connection conn = getConnection()) {  
 String sql = "INSERT INTO customers (name, mobile, email, plan, status, registered\_at) VALUES (?, ?, ?, ?, ?, NOW())";  
 PreparedStatement stmt = conn.prepareStatement(sql);  
 stmt.setString(1, nameField.getText());  
 stmt.setString(2, mobileField.getText());  
 stmt.setString(3, emailField.getText());  
 stmt.setString(4, planField.getText());  
 stmt.setString(5, statusComboBox.getSelectedItem().toString());  
 stmt.executeUpdate();  
 JOptionPane.*showMessageDialog*(this, "Customer added successfully!");  
 clearFields();  
 loadCustomers();  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 JOptionPane.*showMessageDialog*(this, "Error adding customer.");  
 }  
 }  
  
 private void updateCustomer() {  
 if (selectedCustomerId == -1) {  
 JOptionPane.*showMessageDialog*(this, "Select a customer to update.");  
 return;  
 }  
 if (!validateFields()) return;  
 try (Connection conn = getConnection()) {  
 String sql = "UPDATE customers SET name=?, mobile=?, email=?, plan=?, status=? WHERE id=?";  
 PreparedStatement stmt = conn.prepareStatement(sql);  
 stmt.setString(1, nameField.getText());  
 stmt.setString(2, mobileField.getText());  
 stmt.setString(3, emailField.getText());  
 stmt.setString(4, planField.getText());  
 stmt.setString(5, statusComboBox.getSelectedItem().toString());  
 stmt.setInt(6, selectedCustomerId);  
 stmt.executeUpdate();  
 JOptionPane.*showMessageDialog*(this, "Customer updated.");  
 clearFields();  
 loadCustomers();  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 JOptionPane.*showMessageDialog*(this, "Error updating customer.");  
 }  
 }  
  
 private void deleteCustomer() {  
 if (selectedCustomerId == -1) {  
 JOptionPane.*showMessageDialog*(this, "Select a customer to delete.");  
 return;  
 }  
 int confirm = JOptionPane.*showConfirmDialog*(this, "Are you sure you want to delete?", "Confirm Delete", JOptionPane.*YES\_NO\_OPTION*);  
 if (confirm == JOptionPane.*YES\_OPTION*) {  
 try (Connection conn = getConnection()) {  
 String sql = "DELETE FROM customers WHERE id=?";  
 PreparedStatement stmt = conn.prepareStatement(sql);  
 stmt.setInt(1, selectedCustomerId);  
 stmt.executeUpdate();  
 JOptionPane.*showMessageDialog*(this, "Customer deleted.");  
 clearFields();  
 loadCustomers();  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 JOptionPane.*showMessageDialog*(this, "Error deleting customer.");  
 }  
 }  
 }  
  
 private void loadCustomers() {  
 tableModel.setRowCount(0);  
 try (Connection conn = getConnection()) {  
 Statement stmt = conn.createStatement();  
 ResultSet rs = stmt.executeQuery("SELECT \* FROM customers ORDER BY registered\_at DESC");  
  
 while (rs.next()) {  
 tableModel.addRow(new Object[]{  
 rs.getInt("id"),  
 rs.getString("name"),  
 rs.getString("mobile"),  
 rs.getString("email"),  
 rs.getString("plan"),  
 rs.getString("status"),  
 rs.getTimestamp("registered\_at")  
 });  
 }  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 JOptionPane.*showMessageDialog*(this, "Error loading customers.");  
 }  
 }  
  
 private void liveSearch() {  
 String keyword = searchField.getText();  
 tableModel.setRowCount(0);  
 try (Connection conn = getConnection()) {  
 String sql = "SELECT \* FROM customers WHERE name LIKE ? OR mobile LIKE ?";  
 PreparedStatement stmt = conn.prepareStatement(sql);  
 stmt.setString(1, "%" + keyword + "%");  
 stmt.setString(2, "%" + keyword + "%");  
 ResultSet rs = stmt.executeQuery();  
  
 while (rs.next()) {  
 tableModel.addRow(new Object[]{  
 rs.getInt("id"),  
 rs.getString("name"),  
 rs.getString("mobile"),  
 rs.getString("email"),  
 rs.getString("plan"),  
 rs.getString("status"),  
 rs.getTimestamp("registered\_at")  
 });  
 }  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 }  
 }  
  
 private void searchCustomers() {  
 liveSearch();  
 }  
  
 private void sortCustomersByName() {  
 tableModel.setRowCount(0);  
 try (Connection conn = getConnection()) {  
 String sql = "SELECT \* FROM customers ORDER BY name ASC";  
 PreparedStatement stmt = conn.prepareStatement(sql);  
 ResultSet rs = stmt.executeQuery();  
  
 while (rs.next()) {  
 tableModel.addRow(new Object[]{  
 rs.getInt("id"),  
 rs.getString("name"),  
 rs.getString("mobile"),  
 rs.getString("email"),  
 rs.getString("plan"),  
 rs.getString("status"),  
 rs.getTimestamp("registered\_at")  
 });  
 }  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 }  
 }  
  
 private void clearFields() {  
 nameField.setText("");  
 mobileField.setText("");  
 emailField.setText("");  
 planField.setText("");  
 searchField.setText("");  
 statusComboBox.setSelectedIndex(0);  
 selectedCustomerId = -1;  
 table.clearSelection();  
 }  
  
 private boolean validateFields() {  
 if (nameField.getText().isEmpty() || mobileField.getText().isEmpty() || emailField.getText().isEmpty() || planField.getText().isEmpty()) {  
 JOptionPane.*showMessageDialog*(this, "All fields are required.");  
 return false;  
 }  
 if (!Pattern.*matches*("\\d{10}", mobileField.getText())) {  
 JOptionPane.*showMessageDialog*(this, "Mobile number must be 10 digits.");  
 return false;  
 }  
 if (!Pattern.*matches*("[a-zA-Z ]+", nameField.getText())) {  
 JOptionPane.*showMessageDialog*(this, "Name must contain only letters and spaces.");  
 return false;  
 }  
 // Feature 5: Basic email validation  
 if (!Pattern.*matches*("^[\\w.-]+@[\\w.-]+\\.\\w{2,}$", emailField.getText())) {  
 JOptionPane.*showMessageDialog*(this, "Invalid email format.");  
 return false;  
 }  
 return true;  
 }  
  
 public static void main(String[] args) {  
 SwingUtilities.*invokeLater*(() -> new Tele().setVisible(true));  
 }  
}

1. **Database view**

****

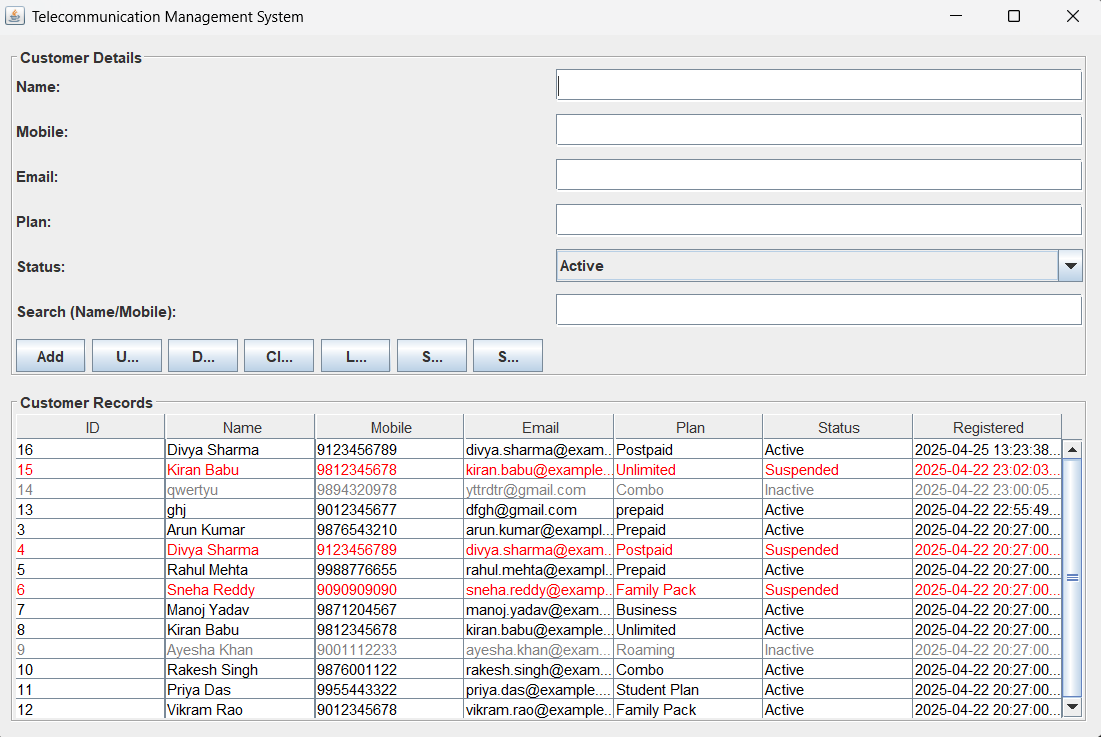
1. **Output screenshots**

**A screenshot of a login screen

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

****

1. **Result**

The **Telecommunication Management System** was successfully designed and implemented using Java (Swing) for the frontend and MySQL for the backend database. The project fulfills all the core functionalities expected of a basic telecom management platform, including:

* Customer registration and management
* Plan selection and tracking
* Status updates (Active/Inactive/Suspended)
* Efficient storage and retrieval of customer data

This system demonstrates effective integration of **Java JDBC with MySQL**, providing a seamless desktop-based interface for telecom operators to manage subscribers. The user interface is intuitive, and the backend ensures data consistency and integrity.

Overall, the project achieves its objective of creating a mini yet scalable system that reflects real-world telecommunication data handling. Future enhancements like billing, login security, and analytics can further enrich the system’s utility.