Name: Josaiah Murfeal Dkhar(2447125)

# **LAB EXERCISE 10**

**Question**:.

How does the system provide feedback to users?

A: Through a status bar and dialog boxes for reports and errors.

What validation mechanisms are in place?

A: Basic field validation ensuring no empty fields are submitted.

**Case Study**: Smart Waste Disposal System

A mid-sized city needs to modernize its waste management system to improve efficiency and reduce operational costs.

**Implementation**

1. **Initial Setup** 
   * Deploy system across 100 waste bins
   * Train waste management staff on system usage
   * Establish monitoring protocols
2. **Operational Phase** 
   * Daily monitoring of bin status
   * Weekly route optimization based on fill levels
   * Monthly maintenance scheduling
3. **Results** 
   * 30% reduction in unnecessary collection trips
   * Improved maintenance scheduling
   * Better resource allocation

**Challenges**

1. Staff training requirements
2. Initial data entry workload
3. Need for system customization

.

**Summary:**

The Smart Waste Disposal System is a Java-based desktop application that implements a waste management tracking system using Swing for the graphical user interface. The system allows users to manage and monitor waste bins, track their status, and generate reports. It provides a comprehensive interface for waste management personnel to maintain records of multiple waste bins and their various attributes.

# **Diagram:**

# **A screenshot of a computer Description automatically generated**

# **Code:**

import javax.swing.\*;  
import javax.swing.table.DefaultTableModel;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class SmartWasteDisposalSystem extends JFrame {  
 private JTextField binIdField;  
 private JTextField binStatusField;  
 private JTextField fillLevelField;  
 private JTextField lastEmptiedField;  
 private JTextField locationField;  
 private JTextField wasteTypeField;  
 private JTextField maintenanceStatusField;  
 private JTable binTable;  
 private DefaultTableModel tableModel;  
 private JLabel statusBar;  
  
 public SmartWasteDisposalSystem() {  
 setTitle("Advanced Smart Waste Disposal Management System");  
 setSize(800, 600);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setLocationRelativeTo(null);  
  
 // Create menu bar  
 JMenuBar menuBar = new JMenuBar();  
 JMenu fileMenu = new JMenu("File");  
 JMenuItem exitMenuItem = new JMenuItem("Exit");  
 exitMenuItem.addActionListener(e -> System.*exit*(0));  
 fileMenu.add(exitMenuItem);  
 menuBar.add(fileMenu);  
 setJMenuBar(menuBar);  
  
 // Create components  
 JLabel binIdLabel = new JLabel("Bin ID:");  
 binIdField = new JTextField(10);  
 binIdField.setToolTipText("Enter the unique ID of the bin");  
  
 JLabel binStatusLabel = new JLabel("Bin Status:");  
 binStatusField = new JTextField(10);  
 binStatusField.setToolTipText("Enter the current status of the bin");  
  
 JLabel fillLevelLabel = new JLabel("Fill Level:");  
 fillLevelField = new JTextField(10);  
 fillLevelField.setToolTipText("Enter the fill level (e.g., empty, half-full, full)");  
  
 JLabel lastEmptiedLabel = new JLabel("Last Emptied:");  
 lastEmptiedField = new JTextField(10);  
 lastEmptiedField.setToolTipText("Enter the last emptied date (e.g., 2023-10-01)");  
  
 JLabel locationLabel = new JLabel("Location:");  
 locationField = new JTextField(10);  
 locationField.setToolTipText("Enter the location of the bin");  
  
 JLabel wasteTypeLabel = new JLabel("Type of Waste:");  
 wasteTypeField = new JTextField(10);  
 wasteTypeField.setToolTipText("Enter the type of waste (e.g., recyclable, non-recyclable, organic)");  
  
 JLabel maintenanceStatusLabel = new JLabel("Maintenance Status:");  
 maintenanceStatusField = new JTextField(10);  
 maintenanceStatusField.setToolTipText("Enter the maintenance status (e.g., needs maintenance, good condition)");  
  
 JButton addButton = new JButton("Add Bin");  
 JButton generateReportButton = new JButton("Generate Report");  
  
 // Create table for bin data  
 String[] columnNames = {"Bin ID", "Status", "Fill Level", "Last Emptied", "Location", "Type of Waste", "Maintenance Status"};  
 tableModel = new DefaultTableModel(columnNames, 0);  
 binTable = new JTable(tableModel);  
 JScrollPane tableScrollPane = new JScrollPane(binTable);  
  
 // Create status bar  
 statusBar = new JLabel("Ready");  
 statusBar.setBorder(BorderFactory.*createEtchedBorder*());  
  
 // Set layout and add components  
 JPanel inputPanel = new JPanel(new GridLayout(8, 2));  
 inputPanel.add(binIdLabel);  
 inputPanel.add(binIdField);  
 inputPanel.add(binStatusLabel);  
 inputPanel.add(binStatusField);  
 inputPanel.add(fillLevelLabel);  
 inputPanel.add(fillLevelField);  
 inputPanel.add(lastEmptiedLabel);  
 inputPanel.add(lastEmptiedField);  
 inputPanel.add(locationLabel);  
 inputPanel.add(locationField);  
 inputPanel.add(wasteTypeLabel);  
 inputPanel.add(wasteTypeField);  
 inputPanel.add(maintenanceStatusLabel);  
 inputPanel.add(maintenanceStatusField);  
 inputPanel.add(addButton);  
 inputPanel.add(generateReportButton);  
  
 add(inputPanel, BorderLayout.*NORTH*);  
 add(tableScrollPane, BorderLayout.*CENTER*);  
 add(statusBar, BorderLayout.*SOUTH*);  
  
 // Add action listeners  
 addButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 addBin();  
 }  
 });  
  
 generateReportButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 generateReport();  
 }  
 });  
 }  
  
 private void addBin() {  
 String binId = binIdField.getText();  
 String binStatus = binStatusField.getText();  
 String fillLevel = fillLevelField.getText();  
 String lastEmptied = lastEmptiedField.getText();  
 String location = locationField.getText();  
 String wasteType = wasteTypeField.getText();  
 String maintenanceStatus = maintenanceStatusField.getText();  
  
 if (!binId.isEmpty() && !binStatus.isEmpty() && !fillLevel.isEmpty() && !lastEmptied.isEmpty() && !location.isEmpty() && !wasteType.isEmpty() && !maintenanceStatus.isEmpty()) {  
 tableModel.addRow(new Object[]{binId, binStatus, fillLevel, lastEmptied, location, wasteType, maintenanceStatus});  
 binIdField.setText("");  
 binStatusField.setText("");  
 fillLevelField.setText("");  
 lastEmptiedField.setText("");  
 locationField.setText("");  
 wasteTypeField.setText("");  
 maintenanceStatusField.setText("");  
 statusBar.setText("Bin added successfully");  
 } else {  
 statusBar.setText("Please enter all fields");  
 }  
 }  
  
 private void generateReport() {  
 StringBuilder report = new StringBuilder();  
 for (int i = 0; i < tableModel.getRowCount(); i++) {  
 report.append("Bin ID: ").append(tableModel.getValueAt(i, 0))  
 .append(", Status: ").append(tableModel.getValueAt(i, 1))  
 .append(", Fill Level: ").append(tableModel.getValueAt(i, 2))  
 .append(", Last Emptied: ").append(tableModel.getValueAt(i, 3))  
 .append(", Location: ").append(tableModel.getValueAt(i, 4))  
 .append(", Type of Waste: ").append(tableModel.getValueAt(i, 5))  
 .append(", Maintenance Status: ").append(tableModel.getValueAt(i, 6)).append("\n");  
 }  
 JOptionPane.*showMessageDialog*(this, report.toString(), "Report", JOptionPane.*INFORMATION\_MESSAGE*);  
 }  
  
 public static void main(String[] args) {  
 SwingUtilities.*invokeLater*(new Runnable() {  
 @Override  
 public void run() {  
 new SmartWasteDisposalSystem().setVisible(true);  
 }  
 });  
 }  
}

[Scroll down for Output]

OUTPUT:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Inference**

** System Strengths**

* Intuitive user interface
* Comprehensive bin tracking
* Flexible reporting system
* Easy to extend functionality

** Areas for Improvement**

* Data persistence
* Advanced validation
* Authentication system
* Real-time monitoring capabilities

 **Recommendations**

* Implement database backend
* Add user authentication
* Develop mobile companion app
* Add automated alerts system