package com.mycompany.phonebook1;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

public class Phonebook1 extends JFrame {

private JTable table;

private DefaultTableModel model;

private JTextField nameField;

private JTextField phoneField;

private JTextField searchField;

public Phonebook1() {

setTitle("Phonebook");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(600, 400);

setLocationRelativeTo(null);

// Set a background color for the main frame

getContentPane().setBackground(new Color(220, 220, 220)); // Light grey background

// Create table model and table

model = new DefaultTableModel(new String[]{"Name", "Phone Number"}, 0);

table = new JTable(model);

table.setBackground(Color.WHITE); // Table background color

table.setForeground(Color.BLACK); // Table text color

table.setFont(new Font("Arial", Font.PLAIN, 14)); // Table font

// Enable sorting by column

table.setAutoCreateRowSorter(true); // Enable sorting by column

JScrollPane scrollPane = new JScrollPane(table);

// Create input fields

nameField = new JTextField(15);

phoneField = new JTextField(15);

searchField = new JTextField(15); // Search field

// Create buttons

JButton insertButton = new JButton("Insert");

JButton deleteButton = new JButton("Delete Contact");

JButton updateButton = new JButton("Update Contact");

JButton searchButton = new JButton("Search");

// Set button colors

Color buttonColor = new Color(100, 149, 237); // Cornflower blue

Color textColor = Color.WHITE; // White text

insertButton.setBackground(buttonColor);

insertButton.setForeground(textColor);

deleteButton.setBackground(buttonColor);

deleteButton.setForeground(textColor);

updateButton.setBackground(buttonColor);

updateButton.setForeground(textColor);

searchButton.setBackground(buttonColor);

searchButton.setForeground(textColor);

// Add action listeners

insertButton.addActionListener(e -> insertContact());

deleteButton.addActionListener(e -> deleteContact());

updateButton.addActionListener(e -> updateContact());

searchButton.addActionListener(e -> searchContact());

// Create a panel for input fields and buttons

JPanel panel = new JPanel();

panel.setBackground(new Color(220, 220, 220)); // Same background color as the frame

panel.add(new JLabel("Name:"));

panel.add(nameField);

panel.add(new JLabel("Phone:"));

panel.add(phoneField);

panel.add(insertButton);

panel.add(deleteButton);

panel.add(updateButton);

panel.add(new JLabel("Search:"));

panel.add(searchField);

panel.add(searchButton);

// Add components to frame

add(scrollPane, BorderLayout.CENTER);

add(panel, BorderLayout.SOUTH);

}

private void insertContact() {

String name = nameField.getText();

String phone = phoneField.getText();

if (!name.isEmpty() && !phone.isEmpty()) {

if (!isPhoneUnique(phone)) {

JOptionPane.showMessageDialog(this, "Phone number already exists.");

} else {

model.addRow(new Object[]{name, phone});

nameField.setText("");

phoneField.setText("");

}

} else {

JOptionPane.showMessageDialog(this, "Please enter both name and phone number.");

}

}

private void deleteContact() {

int selectedRow = table.getSelectedRow();

if (selectedRow != -1) {

int confirm = JOptionPane.showConfirmDialog(

this,

"Are you sure you want to delete this contact?",

"Confirm Delete",

JOptionPane.YES\_NO\_OPTION,

JOptionPane.QUESTION\_MESSAGE

);

if (confirm == JOptionPane.YES\_OPTION) {

model.removeRow(selectedRow);

}

} else {

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

}

}

private void updateContact() {

int selectedRow = table.getSelectedRow();

if (selectedRow != -1) {

String name = nameField.getText();

String phone = phoneField.getText();

if (!name.isEmpty() && !phone.isEmpty()) {

if (!isPhoneUnique(phone) && !phone.equals(model.getValueAt(selectedRow, 1))) {

JOptionPane.showMessageDialog(this, "Phone number already exists.");

} else {

model.setValueAt(name, selectedRow, 0);

model.setValueAt(phone, selectedRow, 1);

nameField.setText("");

phoneField.setText("");

}

} else {

JOptionPane.showMessageDialog(this, "Please enter both name and phone number.");

}

} else {

JOptionPane.showMessageDialog(this, "Please select a contact to update.");

}

}

private void searchContact() {

String searchName = searchField.getText();

if (!searchName.isEmpty()) {

boolean found = false;

for (int i = 0; i < model.getRowCount(); i++) {

String name = (String) model.getValueAt(i, 0);

if (name.equalsIgnoreCase(searchName)) {

table.setRowSelectionInterval(i, i);

found = true;

break;

}

}

if (!found) {

JOptionPane.showMessageDialog(this, "Contact not found.");

}

} else {

JOptionPane.showMessageDialog(this, "Please enter a name to search.");

}

}

private boolean isPhoneUnique(String phone) {

for (int i = 0; i < model.getRowCount(); i++) {

String existingPhone = (String) model.getValueAt(i, 1);

if (existingPhone.equals(phone)) {

return false;

}

}

return true;

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> {

Phonebook1 phonebook1 = new Phonebook1();

phonebook1.setVisible(true);

});

}

}