package searchapp;

import java.awt.event.ActionEvent;

import java.awt.BorderLayout;

import java.awt.Button;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.FlowLayout;

import java.awt.Font;

import java.awt.Toolkit;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.awt.event.TextEvent;

import java.awt.event.TextListener;

import javax.swing.ButtonGroup;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JMenu;

import javax.swing.JMenuBar;

import javax.swing.JMenuItem;

import javax.swing.JPanel;

import javax.swing.JRadioButton;

import javax.swing.JTextArea;

import javax.swing.JTextField;

import javax.swing.event.DocumentEvent;

import javax.swing.event.DocumentListener;

//this class contains the constructor, events, and properties all relating to the main application window of the program

public class MainWindow extends BaseWindow{

JFrame frame = new JFrame("Search Engine");

JLabel label;

JTextField textField;

SearchType currentSearchType = SearchType.AND;

//we establish the index window and about window before hand so that we know this is the only instance of the window running

IndexSetupWindow isw = new IndexSetupWindow();

AboutWindow aw = new AboutWindow();

public MainWindow(){

initWindow();

}

public void initWindow(){

//initialize the components

final JTextArea searchArea= new JTextArea("Type in the search bar to get started!");

// make frame..

JFrame.setDefaultLookAndFeelDecorated(true);

frame.setSize(700,400);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setBackground(Color.RED);

frame.setBounds(100,90,300,120);

frame.setLayout(new BorderLayout());

//frame.setSize(200, 200);

frame.setVisible(true);

// make a panel for all search related functionality

JPanel searchPanel = new JPanel();

searchPanel.setLayout(new BorderLayout());

//for better organization purposes

JPanel searchTop = new JPanel();

JPanel searchBottom = new JPanel();

//initialize the label

this.label = new JLabel();

label.setText("Search: ");

searchTop.add(label);

// make a text field

this.textField = new JTextField("",50);

searchTop.add(this.textField);

//we want to listen for changes in the text to start searching

this.textField.addKeyListener(new KeyListener() {

@Override

public void keyTyped(KeyEvent arg0) {

/\*\*\*

\*

\*

\*

\* ADD THE SEARCH INDEXING LOGIC HERE... ASSUMING IT'S NOT TOO SLOW!

\*

\*

\*

\*/

System.out.println("we are receiving input!");

searchArea.setText("I'm sorry, but no matching files were found");

//use IndexSearch.searchForFiles here!

}

@Override

public void keyReleased(KeyEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void keyPressed(KeyEvent arg0) {

// TODO Auto-generated method stub

}

});

//setting up the menu bar

JMenuBar mb = new JMenuBar();

JMenu file = new JMenu("File");

mb.add(file);

JMenu edit = new JMenu("Edit");

mb.add(edit);

JMenuItem undo= new JMenu("Undo");

edit.add(undo);

JMenu Name = new JMenu("About");

mb.add(Name);

JMenuItem proj= new JMenuItem("Maintain Index");

Name.add(proj);

JMenuItem exit= new JMenuItem("Exit");

file.add(exit);

JMenuItem Save= new JMenuItem("Save as");

file.add(Save);

frame.setJMenuBar(mb);

//Create Radio buttons

JRadioButton c = new JRadioButton("AND");

c.setBackground(Color.green);

c.setLocation(210, 200);

c.setSize(50, 50);

c.setSelected(true);

c.addMouseListener(new MouseListener() {

@Override

public void mouseReleased(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mousePressed(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mouseExited(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mouseEntered(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mouseClicked(MouseEvent arg0) {

//the radio button will set the search type

currentSearchType = SearchType.AND;

System.out.println(currentSearchType.toString());

}

});

searchBottom.add(c,BorderLayout.PAGE\_END);

JRadioButton d = new JRadioButton("OR");

d.setBackground(Color.green);

d.setLocation(210, 200);

d.setSize(50, 50);

d.setSelected(false);

d.addMouseListener(new MouseListener() {

@Override

public void mouseReleased(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mousePressed(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mouseExited(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mouseEntered(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mouseClicked(MouseEvent arg0) {

//the radio button will set the search type

currentSearchType = SearchType.OR;

System.out.println(currentSearchType.toString());

}

});

searchBottom.add(d,BorderLayout.PAGE\_END);

JRadioButton f = new JRadioButton("PHRASE");

f.setBackground(Color.green);

f.setLocation(210, 200);

f.setSize(50, 50);

f.setSelected(false);

f.addMouseListener(new MouseListener() {

@Override

public void mouseReleased(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mousePressed(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mouseExited(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mouseEntered(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mouseClicked(MouseEvent arg0) {

//the radio button will set the search type

currentSearchType = SearchType.PHRASE;

System.out.println(currentSearchType.toString());

}

});

searchBottom.add(f,BorderLayout.PAGE\_END);

//to make sure that the radio buttons are multiple-exclusive, we will add a button group

ButtonGroup searchOptions = new ButtonGroup();

searchOptions.add(f);

searchOptions.add(d);

searchOptions.add(c);

searchPanel.add(searchTop,BorderLayout.PAGE\_START);

searchPanel.add(searchBottom,BorderLayout.PAGE\_END);

frame.add(searchPanel,BorderLayout.PAGE\_START);

//setting the window dimensions

int frameWidth = 700;

int frameHeight = 600;

Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

frame.setBounds((int) screenSize.getWidth() - frameWidth, 0, frameWidth, frameHeight);

frame.setVisible(true);

//creating the search area

searchArea.setSelectionColor(Color.BLUE);

searchArea.setFont(new Font("Serif", Font.ITALIC, 16));

searchArea.setSize(200,200);

searchArea.setEditable(false);

frame.add(searchArea,BorderLayout.CENTER);

//creating the buttons

//to organize the buttons at the bottom

JPanel buttonPanel = new JPanel();

//Creating the maintain index button

Button indxBtn = new Button("Maintain Index");

indxBtn.setBounds(334, 126, 90, 25);

indxBtn.addMouseListener(new MouseListener() {

@Override

public void mouseReleased(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mousePressed(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseExited(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseEntered(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseClicked(MouseEvent e) {

// TODO Auto-generated method stub

isw.setVisibility(true);

}

});

buttonPanel.add(indxBtn);

//creating the about us button

Button aboutBtn = new Button("About us");

aboutBtn.setBounds(334, 126, 90, 25);

aboutBtn.addMouseListener(new MouseListener() {

@Override

public void mouseReleased(MouseEvent e) {

}

@Override

public void mousePressed(MouseEvent e) {

}

@Override

public void mouseExited(MouseEvent e) {

}

@Override

public void mouseEntered(MouseEvent e) {

}

@Override

public void mouseClicked(MouseEvent e) {

aw.setVisibility(true);

}

});

buttonPanel.add(aboutBtn);

frame.add(buttonPanel,BorderLayout.PAGE\_END);

frame.setVisible(true);

}

@Override

public void actionPerformed(ActionEvent e) {

//for general purpose

System.out.println(currentSearchType.toString());

}

public void textChanged(TextEvent e)

{

}

}