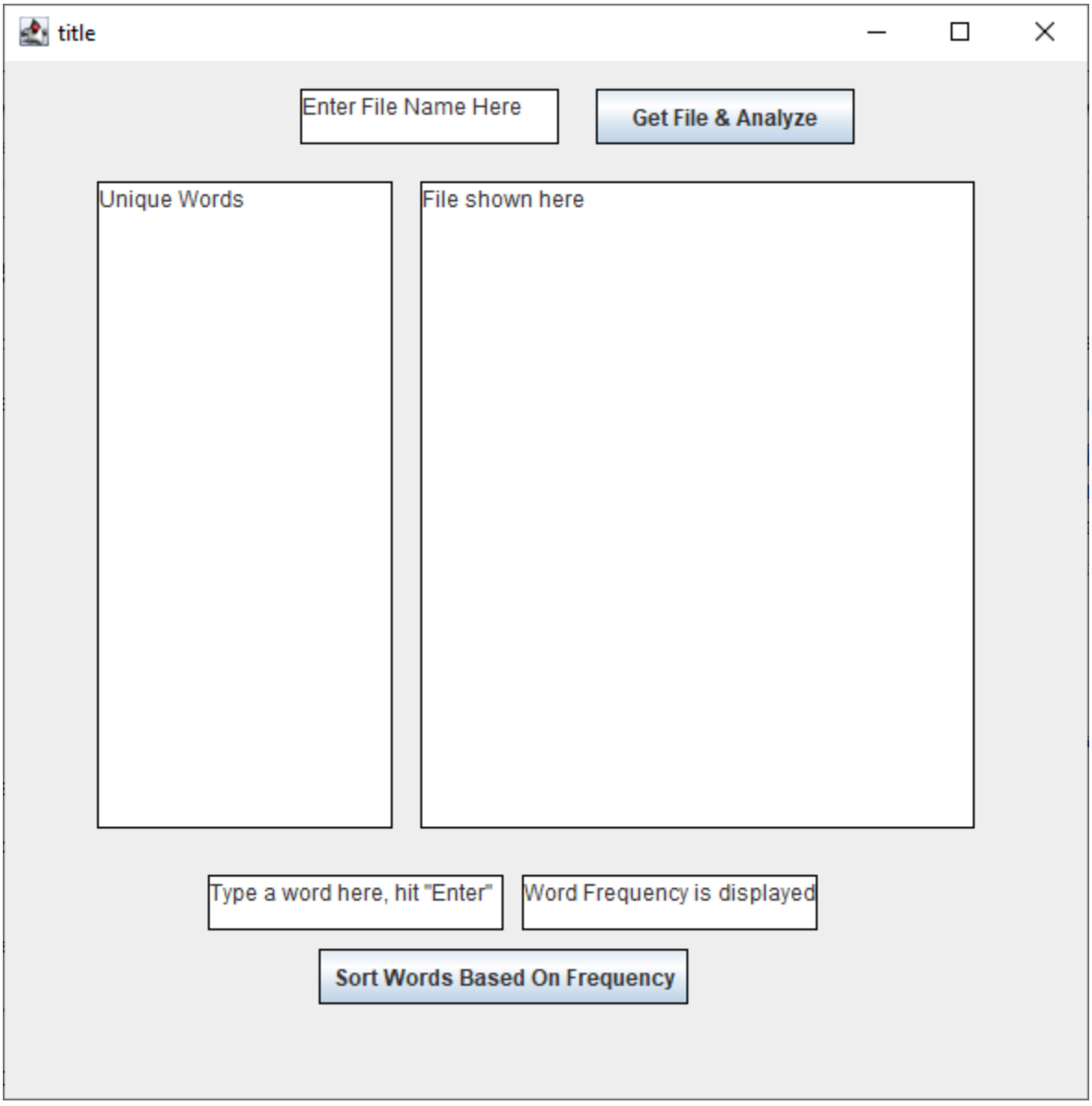
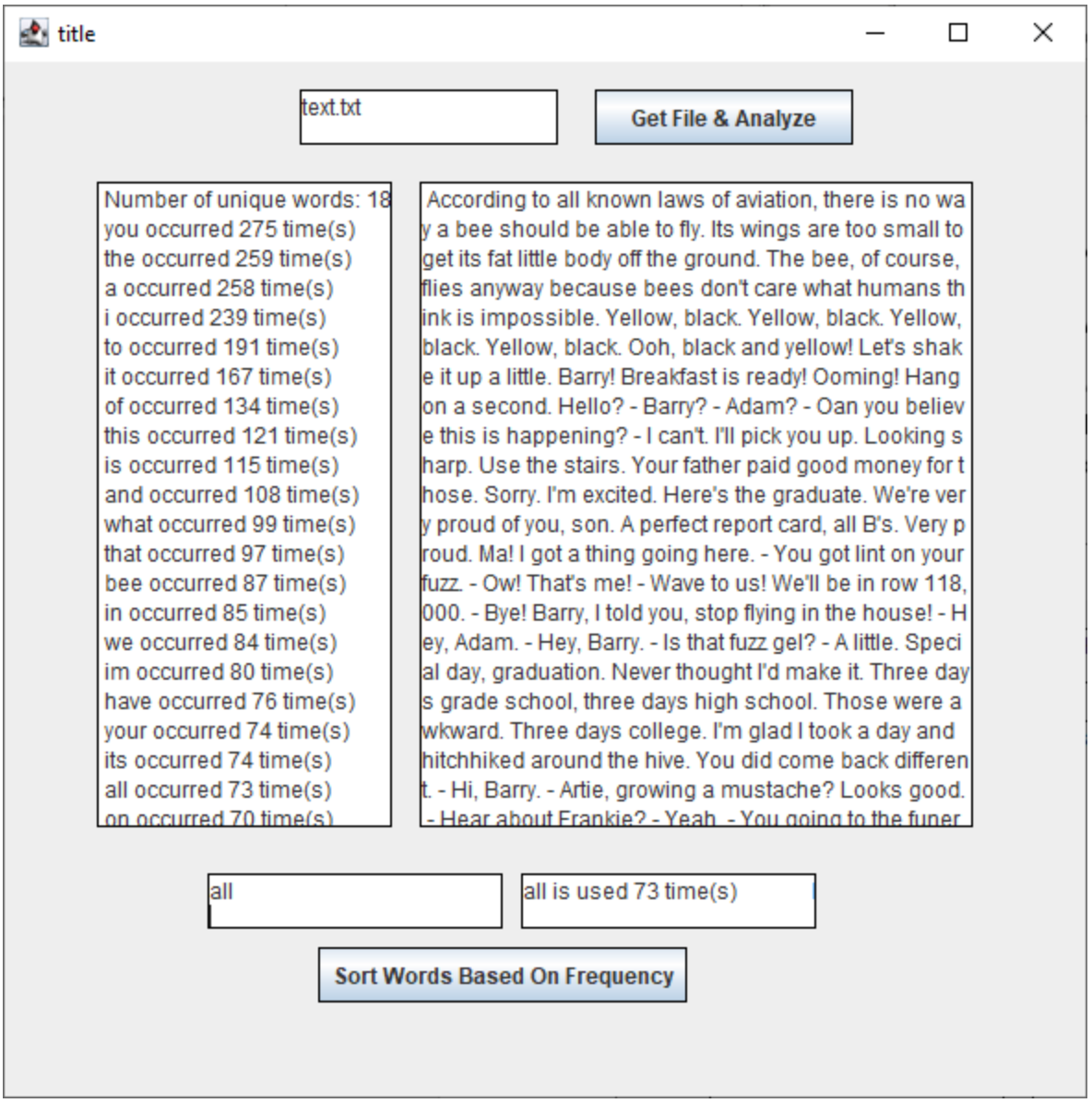
Noah Streveler

Assignment 8






```
import java.awt.*;

import java.awt.event.*;

import java.awt.geom.*;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;


import javax.swing.*;
```

```
import javax.swing.border.Border;

import java.util.*;

public class Frame extends JFrame{

	public Frame(String title) {
		super(title);

		this.setSize(600, 600);
		this.setLocation(300, 100);
		this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		this.setVisible(true);
		this.setLayout(new BorderLayout(15, 15));

	}
}

class Panel extends JPanel{
	JPanel panel1, panel2, panel3;
	JTextArea fileName, displayFile, unique, wordFinder, frequency;
	JButton analyzer, sorting;
	ArrayList<String> fileWord = new ArrayList<String>();
	ArrayList<Integer> wordCounter = new ArrayList<Integer>();

	public Panel(String title) {
```

```
super();
this.setBorder(BorderFactory.createTitledBorder("Panel"));
this.setLayout(new BorderLayout());

Frame frame = new Frame("title");

panel1 = new JPanel(null);
panel1.setPreferredSize(new Dimension(100,100));
panel1.setLayout(new BorderLayout());

Border border = BorderFactory.createLineBorder(Color.BLACK, 1);

//JScrollPane scroll1 = new JScrollPane(unique);

//JScrollPane scroll2 = new JScrollPane(unique);

fileName = new JTextArea("Enter File Name Here", 40, 25);
fileName.setBounds(160, 15, 140, 30);
fileName.setBorder(border);

analyzer = new JButton("Get File & Analyze");
analyzer.setBounds(320, 15, 140, 30);
analyzer.setBorder(border);

displayFile = new JTextArea("File shown here", 40, 25);
displayFile.setBounds(225, 65, 300, 350);
displayFile.setBorder(border);
```

```
unique = new JTextArea("Unique Words", 40, 25);
unique.setBounds(50, 65, 160, 350);
unique.setBorder(border);

wordFinder = new JTextArea("Type a word here, hit \"Enter\"", 40, 25);
wordFinder.setBounds(110, 440, 160, 30);
wordFinder.setBorder(border);

frequency = new JTextArea("Word Frequency is displayed here", 40, 25);
frequency.setBounds(280, 440, 160, 30);
frequency.setBorder(border);

sorting = new JButton("Sort Words Based On Frequency");
sorting.setBounds(170, 480, 200, 30);
sorting.setBorder(border);

//frame.add(scroll1, BorderLayout.CENTER);
frame.add(fileName);
frame.add(analyzer);
frame.add(displayFile);
frame.add(unique);
frame.add(wordFinder);
frame.add(frequency);
frame.add(sorting);
frame.add(panel1);
```

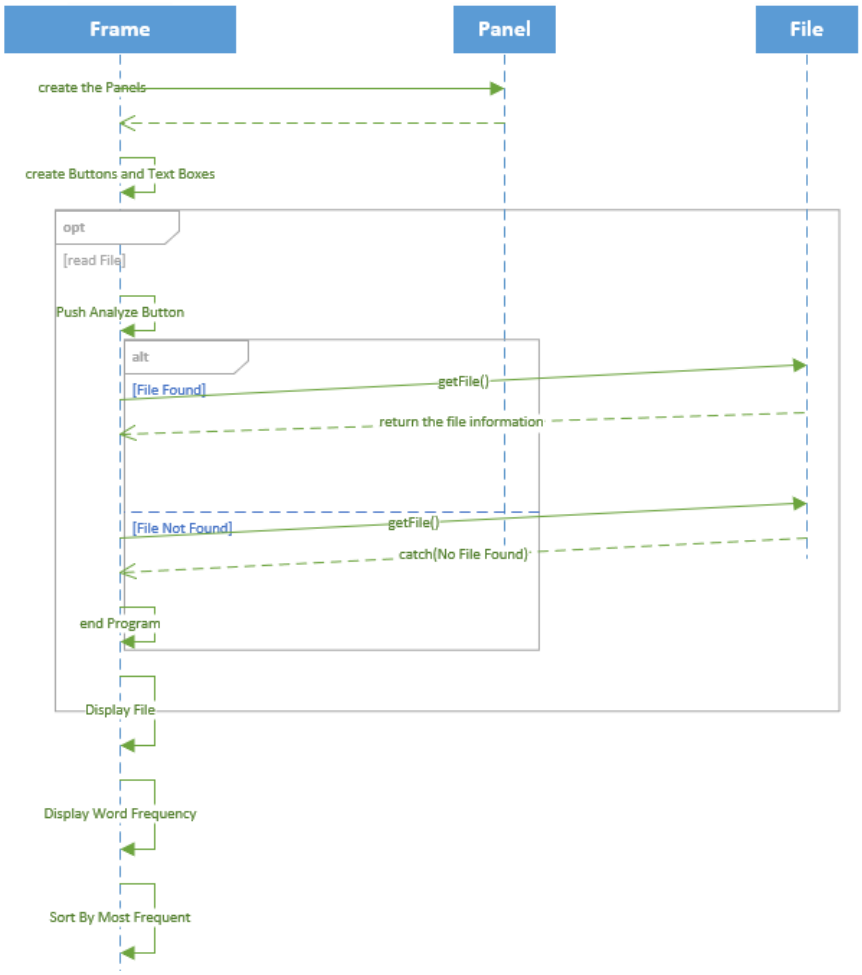
```
    analyzer.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e)
            {
                    try {
                            analyzeFile(getFile());
                    } catch (FileNotFoundException e1) {
                            e1.printStackTrace();
                    }
            }

    });

    sorting.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e)
            {
                    sortFileFrequency();
            }

    });

    wordFinder.addKeyListener(new KeyAdapter() {
            @Override
public void keyPressed(KeyEvent e) {
```

```
            if(e.getKeyCode() == KeyEvent.VK_ENTER)
            {
                                    getWordFrequency(getWord());
            }
}


             });
 }



 public String getWord() {
             return (wordFinder.getText());
 }

 public String getFile() {
             return (fileName.getText());
 }

 public void sortFileFrequency(){


 unique.setText(" ");
 for(int i = 0; i < fileWord.size(); i++) {
             for(int j = i ; j < fileWord.size(); j++) {
             if(wordCounter.get(i) < wordCounter.get(j)) {
                         int temp = wordCounter.get(i);
                         wordCounter.set(i, wordCounter.get(j));
                         wordCounter.set(j, temp);
```

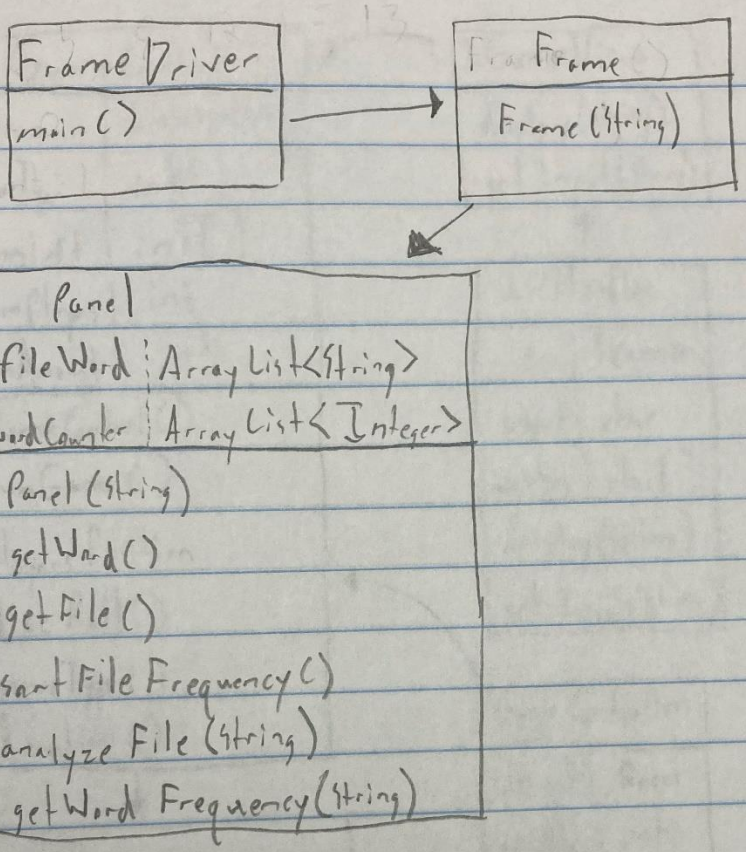
```
                String temp2 = fileWord.get(i);
                fileWord.set(i, fileWord.get(j));
                fileWord.set(j, temp2);
            }
            }
    }

unique.append("Number of unique words: " + wordCounter.size());
for(int i = 0; i < wordCounter.size(); i++) {
    unique.append(" " + fileWord.get(i) + " occurred " + wordCounter.get(i) + " time(s)\n");
    System.out.print(fileWord.get(i) + " occurred " + wordCounter.get(i) + " time(s)\n");
}
    }

    public void analyzeFile(String file) throws FileNotFoundException {
    FileInputStream fileIn = new FileInputStream(file);
    Scanner input = new Scanner(fileIn);
    int counter = 0;

    displayFile.setText(" ");
    unique.setText(" ");
while(input.hasNext()) {
    String word = input.next();
    displayFile.setLineWrap(true);
    displayFile.append(word + " ");
```

```
                    String newWord = word.toLowerCase().replaceAll("\\p{Punct}", "");

                    if (fileWord.contains(newWord))
                    {
                            int index = fileWord.indexOf(newWord);
                            wordCounter.set(index, wordCounter.get(index) + 1);
                    }
                    else
                    {
                            fileWord.add(newWord);
                            wordCounter.add(1);
                            counter++;
                    }
}

input.close();
            //fileIn.close();

unique.append("Number of unique words: " + counter);
for(int i = 0; i < wordCounter.size(); i++) {
    unique.append(" " + fileWord.get(i) + " occurred " + wordCounter.get(i) + " time(s)\n");
}


    }

    public void getWordFrequency(String word) {
```

```
                if (fileWord.contains(word))
                {
                        int location = fileWord.indexOf(word);
                        int amount = wordCounter.get(location);
                        frequency.setText(word + " is used " + amount + " time(s)");
                }
        }
}
```



# UML


FrameDriver
main()
Frame
Frame(String)
Panel
fileWord : ArrayList<String>
wordCounter : ArrayList<Integer>
Panel(String)
getWord()
getFile()
sortFileFrequency()
analyzeFile(String)
getWordFrequency(String)