```java
/*
 * David Lim
 * 1/11/22
 * Strings
 */

//import statements

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Map;
import java.util.List;
import java.util.stream.Collectors;
import java.awt.event.ActionEvent;

public class Main extends JFrame implements ActionListener {
    JTextArea outputArea = new JTextArea("Enter size of String array.", 10, 15);
    JButton btnSize = new JButton("Set Size");
    JButton btnAdd = new JButton("Add Text");
    JTextField input = new JTextField(10);
    String text;
    String[] array;
    int count;

    public static void main(String[] args) {
        Main frame = new Main();
        frame.setSize(600, 500);
        frame.setVisible(true);
    }

    public Main() {
        // application bar name
        super("Strings");
        setDefaultCloseOperation(EXIT_ON_CLOSE);

        // layout manager
        setLayout(new FlowLayout());
        add(input);
        add(outputArea);
        add(btnSize);
        add(btnAdd);
        // add listener for button
        btnSize.addActionListener(this);
        btnAdd.addActionListener(this);

        btnSize.setEnabled(true);
        btnAdd.setEnabled(false);
        // System.out.println("is this working");
    }

    public void actionPerformed(ActionEvent event) {
        Object objSource = event.getSource();
        // int arraySize = Integer.parseInt(input.getText());
        if (objSource == btnSize) {
            text = input.getText();
            array = new String[Integer.parseInt(text)];
            outputArea.setText(String.format("The array has a size of %s. Please enter text.",
text));
            btnSize.setEnabled(false);
            btnAdd.setEnabled(true);
            input.setText("");
            input.requestFocusInWindow();
        }
        if (objSource == btnAdd) {
            text = input.getText();
            if (count < array.length) {
                array[count] = text;
                count += 1;
                // System.out.println(Arrays.toString(array));
                input.setText("");
                input.requestFocusInWindow();
            }
            if (count >= array.length) {
                count = 0;
                btnAdd.setEnabled(false);
                input.setText("");
                ArrayList<String> uppercase = new ArrayList<String>();
                ArrayList<String> lowercase = new ArrayList<String>();
                String maxString = array[0];
                String minString = array[0];
                int max = array[0].length();
                int min = array[0].length();
                for (int i = 0; i < array.length; i++) {

                    if (array[i].length() > max) {
                        max = array[i].length();
                        maxString = array[i];
                    }
                    if (array[i].length() < min) {
                        min = array[i].length();
                        minString = array[i];
                    }
                    uppercase.add(array[i].toUpperCase());
                    lowercase.add(array[i].toLowerCase());

                }
                // group each string in the array by chracter length using maps and lists
                // lengthgroups is a dictionary whose keys are the different character lengths
                // and whose values
                // are each string that has the corresponding character length
                // actually you can just nested for loop this. this is overcomplicating things
                Map<Integer, List<String>> lengthGroups = Arrays.stream(array)
                        .collect(Collectors.groupingBy(String::length));
                // int matches = 0;
                // for (int key : lengthGroups.keySet()) {
                //     if (lengthGroups.get(key).size() > 1) {
                //         matches += lengthGroups.get(key).size();
                //     }
                // }
                String shortestAndLongest = maxString + minString;
                outputArea.setText(String.format("All CAPS: %s", uppercase) + "\n" +
                        String.format("All lowercase: %s", lowercase) + "\n" +
                        String.format("Longest String: %s", maxString) + "\n"
                        + String.format("Matches: %s", lengthGroups) + "\n" +
                        String.format("Shortest String: %s", minString) + "\n" +
                        String.format("Shortest and Longest String Concatenated: %s",
shortestAndLongest));
                btnSize.setEnabled(true);
                input.setText("");
                input.requestFocusInWindow();
            }
        }
    }
}
```