Build a GUI that has the following

1. A JTextField where the user can enter a name
2. A JLabel instructing the user to enter a name
3. A JList that starts out empty
4. A JButton that adds a name to the list
5. A JButton that removes the selected name from the list
6. A JButton that clears the contents of the list

Interactions

1. When "Add Name" is clicked the text in the JTextField is added to the JList
   * If the JTextField is empty nothing is added.
     + You can check if it is empty by checking calling the getText().length() within the JTextField and seeing if it is 0
   * Clear the text in the JTextField by setting it back to ""
2. When "Remove Name" is clicked the selected item in the JList is removed from the list
   * If nothing is selected then nothing should happen
     + You can check if something is selected by calling getSelectedIndex() within the JList and seeing if it is -1
3. When "Clear List" is clicked the JList is cleared

**Advice**

Look at slides **18 and 19** for examples of how to work with a JList and its ListModel. This is how you can add to the list, remove from the list, and clear the list.

You need to **access GUI elements from within an event listener**. Look at slides **43, 44, 45, and 46**for options on how to do this. Pages 368-369 of Chapter 16 also show some useful ways to implement this functionality.

You may go with either option shown in the slides above:

* Option A: Create a separate **ActionListener** class that takes your JFrame as its input and access public objects within it.
* Option B: Define the **ActionListener** class and its method when you call the .**addActionListener** method for the JButton.

Build ActionListeners for each JButton. Since each button is doing a different operation you will need a different ActionListener for each one.

**My Layout**

If you want the same layout I used, it was a Grid Layout with 2 rows and 1 column

* Row 1 - A JPanel that contains a Grid Layout with 1 row with 3 columns, then a JLabel, the JTextField and the JList
* Row 2 - A JPanel that has the three buttons added to it

**Screenshots**

**After typing initial name, but no buttons have been clicked:**

A screenshot of a computer

Description automatically generated

**After adding some names using the "Add Name" button:**

A screenshot of a computer

Description automatically generated

**Selecting a name to be deleted:**

A screenshot of a computer

Description automatically generated

**After clicking "Remove Name" with Bob selected:**

A screenshot of a computer

Description automatically generated

**After clicking the "Clear List" button:**

A screenshot of a computer

Description automatically generated

**Ideas for extra work:**

1. Make the input box smaller, just for a name.
2. Be able to print the list or do an export of the output to txt file, pdf, or docx (ms word).
3. Make the items in the list sortable, updatable, and searchable
4. Drag and drop capabilities,
5. Add images: use icons such as a right arrow to add, (maybe) left arrow to delete.
6. Make a menu on top with submenus, **bold** the first letter (A, S, R, E & P) and when the Alt key is pressed, they should be underlined.
   1. Add to **L**ist
   2. **R**emove from List
   3. **C**lear List
   4. Search or **F**ind
   5. Sort (**A**sc, **D**esc)
   6. **E**xport (txt, html, pdf, docx ). Create an API that will send the list, as JSON or XML, to a viewable document in a modal, or separate window, or pop up as a modal webpage.
   7. **P**rint.
7. Connect it to a Database, like MySQL, with something like ODBC.
8. Make it like a shopping cart
9. May be implement an **inventory** system or a **stock trading** platform.
10. Make it “**Distributable**” somehow: run on different computers and at different times.
11. Finding the message on different folders or machines (RPC: Remote Procedural Call), saving the process on a queue and validating the data.

I wanted to start from scratch and develop several versions, but as I read chapter 16 I realized that that would be a waste of time and I decided to concentrate on developing the final version. So, there is version 1 (the homework – bare minimum) and my version, that I hope I can connect later to a database instead of a text file.