(Assignment 1) Linked List of People:

In this assignment you are going to implement an application that will make use of Linked Lists.

The application should allow users to add, remove and search names.

Below you see a possible GUI, although you are encouraged to make something more user-friendly.

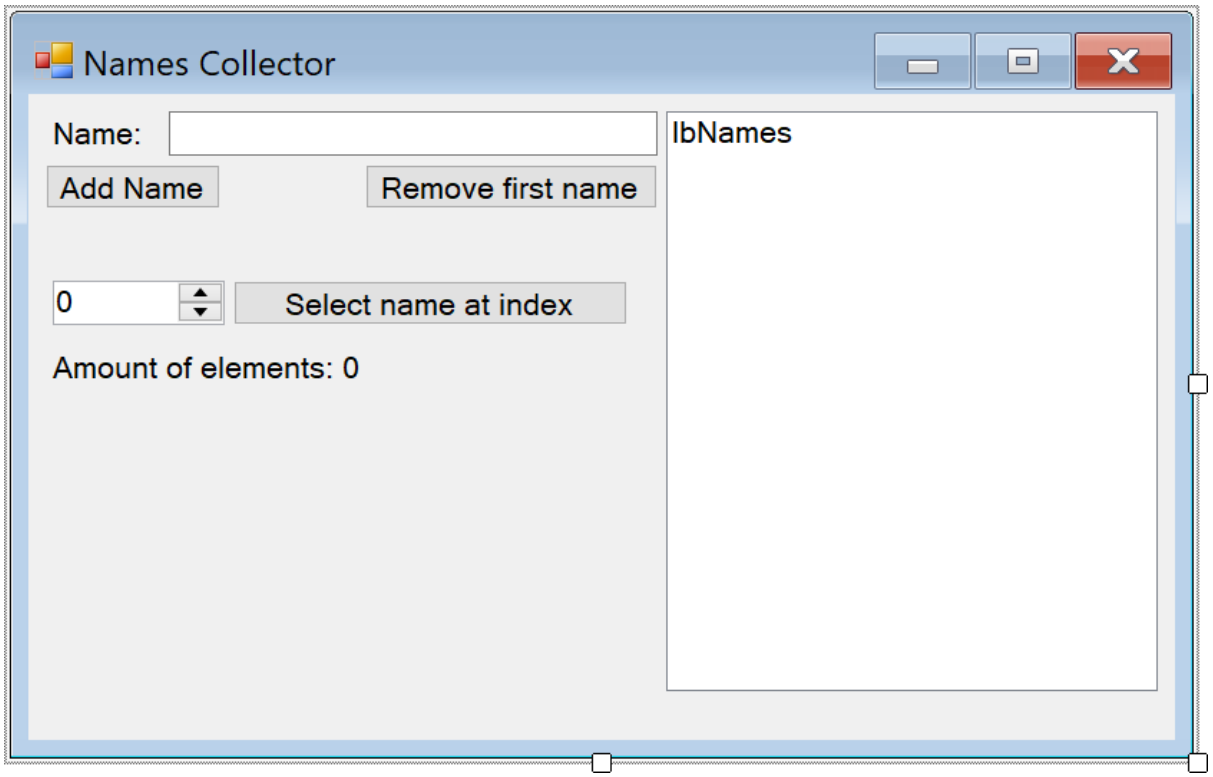

Figure 1: Possible graphical user interface for the names collector form

When the user wants to add a name they can do this by typing the name into the textbox and press the button “Add Name”. This will add the name to the linked list. At all times, the listbox should be updated to reflect what information the linked list contains. When the user wants to remove a name, they can use the same textbox and press the button “Remove first name”, which will remove the first occurrence of the name from the linked list (so if there are two or more instances, the list should contain only one less instance of the name!). The user should also be able to select a specific name based on its index by selecting the correct value in the numeric up down and clicking on the button “Select name at index”. You can show a messagebox with the selected element value.

Last at all time you should display the (up-to-date) amount of elements from the linked list.

Requirements:

FR-01: Add a name.
FR-02: Remove the first instance of a name.
By value.
FR-3: Select a name.
select by index, display in a messagebox.

(Assignment 2) Linked List Queue or Stack:

Queues and stacks are abstract data-structures. This means that their behavior is described, but not how they should be implemented. This gives programming languages (and developers) the freedom to decide what the best implementation is for their purposes. Effectively this means that you could implement a stack or queue with an array to store the elements, but you could also use a Linked List.

Revisit Assignment 2 (Stack) or Assignment 3 (Queue) and replace the Stack or Queue with a linked list that behaves in the same way.