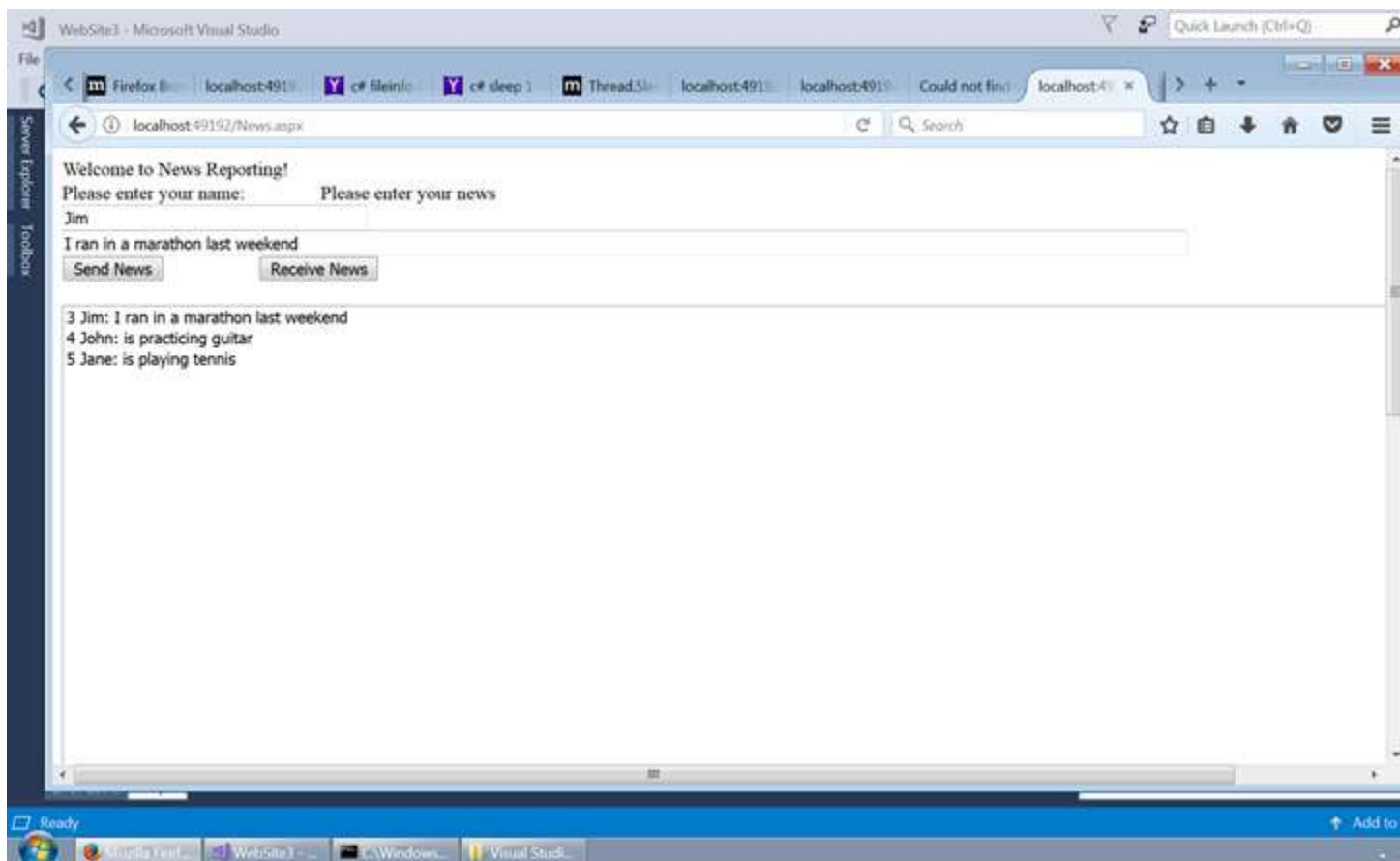


Homework 1: Creating a News Application in .NET

In this homework you will write a program, perform some testing on it, and write up your results as a paper. The paper should have the following sections:

- Introduction: Describe in your words (not mine) what the project is supposed to do.
- Method: What you implemented in code from a features perspective.
- Results and Analysis: The results you encountered and why you think they occurred.
- Conclusion: A one or two paragraph summary of your findings.

The descriptions below describe more information on what is needed:



Introduction

Create a News web form application where news can be sent, shared and received.

There are two buttons: Send News, where you can send who you are and what your news is to the centralized repository. There is a Receive News, where you can receive a set of the shared news. Preferably the listBox would never have more than 10 items in it, but would retain the ten most recent

submissions. (This may be accomplished for example, by removing the first entry when adding an 11th entry.) See the figure below to see what the application looks like:

Each line should include a count, the sender's name, and the news item. The count should represent the number of the news event, and may reach high numbers.

We want users to be able to see other users' data. You can try two techniques: determine if web page data can be shared via the listbox and/or via a file.

The purposes of this experiment is to learn through the use of .NET web pages to:

- Determine whether .NET uses processes or threads between users of the same app. Remember that processes do not share a process image, whereas threads do.
- Learn what one user shares with another user. The count variable should be tested as a local variable in a procedure, a form class attribute, or static class attribute. These tests will help to indicate which variables are shared between users, and if each form is a thread or process.
- Program with .NET. For example, also work with files to share information between different web users.
- Determine if synchronization errors may occur between web users through the use of a static count variable. At the time send news is executed, the count is incremented. If you then create a delay of 5+ second (when another news item can be simultaneously submitted), then the news item is saved to a file. How could you ensure each news item has a unique number, with no duplicates and repeats?
- Test a distributed communication. During debug, copy the web page URL and create other web pages to work with.

Method:

Create a web form as shown above. Use a ListBox to create the array.

After you compile and run your code, you may create multiple copies of the same form by opening new browsers and copying the URL from the first browser. The URL may look something like this:
<http://localhost:51765/Default.aspx>.

Results and Analysis:

For this assignment you will turn in code as well as a paper describing your findings. The questions to investigate in the paper include:

1. Are web forms shared between users? Are attributes stored in a Web Form shared between users? Are listbox items shared? Are static or non-static counts shared? Why or why not do you conclude they are shared? Discuss using operating systems concepts to discuss whether different web users are different processes or threads, and why some information may be

shared or not shared, as per internal concepts of storage on stack, data section, separate objects, etc. Also use operating system tools (displaying list of processes or threads) to confirm your hypothesis.

2. Does the use of the static count work, providing always unique numbers? Is it possible for two news items to get the same number? Why or why not?
3. How do you find working with .NET? Where did you find useful information via Google? What is different relative to Java?

The important thing with this assignment is to figure out what works and does not work, and define what may be happening. It is not necessary to get everything correct, but if you do not get something working, tell me why. Good luck!

Submission:

For the code, only include the .cs that shows the code that you wrote. Submit your code to Canvas. The paper should be written with font 11 or 12, single or double spaced. Turn in a paper copy of the document, with a .doc or .pdf version to Canvas as well. (I will be grading off of the papers - the Canvas submission is used as a backup only.)

Hints to Starting the Programming Project:

Getting used to Visual Studio: See the web page: 'C# Description & Labs' for a description of pertinent C# code and the two labs we did in class. The two labs may help you to remember how things work.

It is best to use the dual-boot machines in the computer science lab, set to Windows. You may of course use your own computer: it is possible to download a version of MS Visual Studio. See system administrator Matt McPherson for details (mcphersm@uwp.edu). I do not recommend using the security lab machines for anything except doing the labs.

Start the project as follows:

In MS Visual Studio do a:

File -> new -> Web Site
Set options: Visual C# option; ASP.NET Empty Web Site
Click OK

In the Solution Explorer on the right hand side, right click and select Add->Web Form.
Enter Item Name: News.
Select Design. Create your form with drag and drop icons from the Toolbox (left side).
Enter two Textboxes for the name and news entry.
Add a ListBox for the news display. Selected Edit Items, Add, ListItem, OK.
Select a button for the Send News and Receive News.

Use the labs and 'C# Description and Labs' web site to help you program your code. When programming with a file, you may want to use the @"C:\Users\Student\Documents\Visual Studio 2017\news.txt" file to store news to.