

CS 350 - Computer/Human Interaction

Spring 2017 – C# Prototype

Out: Thursday, March 1, 2018

Due: Tuesday, March 27, 2018

A C# Message Board Application

While a CGI web-based message board is interesting, your software company has decided that they would like to see a prototype C# message board Internet application.

The Task

- Your task is to create a prototype interface for this C# message board Internet application. Note again that unlike a blogging system, a message board just displays the messages it receives in chronological order. That is, replies to messages are not displayed together with the original message. As before, each message on the board will have at least the date, the sender's name, a subject line, and a message body.
- The prototype is to demonstrate the feasibility of such a product, and mainly is to demonstrate the interface. You should create both an interface for posting messages and an interface for viewing messages. You do not need to provide a screen for logging in, though you may.
- You should provide feedback when posting messages, and appropriate error messages if vital information is omitted when posting a new message. You will have to decide on how messages are displayed. Unlike a web-based system, where you get automatic scrolling from the web browser, you probably do not want to display all messages in one long list. You may wish to provide a catalog of all messages, or may want to display the most recent messages with the ability to go back to previous messages. However you display the messages, you should be able to distinguish one message from another.
- As with the web-based message board project, each message should be stored in a separate file. The format of this file is up to you, but you should think about how it will be read back in for displaying.
- Since this is not a networking class, the prototype does not need to actually communicate with other instances via the Internet. A simple way to allow communication between two running prototypes is to write messages files into the same directory (e.g., login into your ACENET account on two different machines and write to a network drive), and look into that directory periodically to see if any new messages have appeared. (Note: a reference implementation on how do this will be provided sometime after spring break. In the meantime, just write the files into the AppData area and provide a button and click handler that will read and display the data.) Write your code so that the location of the common directory can be changed easily.

What to submit

Create a single **PDF** document containing the following:

- The following statement, **signed and dated** by the submitter:

In accordance with the University of Evansville Honor Code, I attest that the project I am submitting is completely my own work, and that I have not received nor given any

unauthorized aid on this project.

- Short instructions of how a user views messages and how a user posts messages with your interface. You should assume the reader is a first-time, novice user who has not necessarily used similar interfaces before.

Put the PDF document in the top level of your project folder. Zip up your entire project folder and **submit the zipfile to the submission system** at submission.evansville.edu. Reminder: Your username is your ACENET username with "-cs350" appended to it. Your initial password is your student ID number starting with the 0.

Grading

Prototypes will be graded in the following manner:

- 25 points - prototype demonstrates an interface that meets the above specification
- 10 points - aesthetics and usability; 6-7 points for average, 8-9 points for above average, 10 points for exceptional
- 5 points - readability of instructions