

CSCI 428/628 - Advanced Web Programming - Assignment: Picture Down Loading

There is a text file at <http://www.seasite.niu.edu/csci628/picuris.txt> that contains urls of jpeg picture files. Here are a couple of examples from the file:

<http://www.seasite.niu.edu/picdb/phil/00112s.jpg>
<http://www.seasite.niu.edu/picdb/phil/00113s.jpg>

All of the picture filenames have the letter 's' at the end because this denotes a 'small' picture - a thumbnail. Each file name ending with 's' has a corresponding filename without the 's' that is the name of the big picture. So at <http://www.seasite.niu.edu/picdb/phil/> there is a file named 00112.jpg and another named 00113.jpg.

Write a phone app that will read the picuris file and download each of the small picture files named there. Display the thumbnail pictures using an Image component and have these Image components be stored in a List<Image>. When a user clicks on a thumbnail, fetch the larger picture stream and display the large picture in an Image component on a new page.

One strategy:

Use a BackgroundWorker to start a WebClient. The WebClient's job is to read the picuris.txt file and to save each url in a List<string>. When the WebClient has finished, have the BackgroundWorker start a second BackgroundWorker. BackgroundWorker 2 can iterate through the List<string> that contains the urls and create a custom Picture object for each one. The custom Picture object can store the name of the larger jpeg as well as a Stream that is the thumbnail. Save these Picture objects in a List<Picture>. BackgroundWorker 2 will start a WebClient to go read the Stream associated with the thumbnail image and save it in the custom Picture object. When BackgroundWorker 2 finishes, it can iterate through the List<Picture> and for each one, create a BitmapImage, save that in the Picture object and assign the previously saved Stream as the Bitmap's source. When done, assign the List<Picture> to the DataContext for the page.

Clicking a thumbnail should navigate to a new page where a BackgroundWorker and a WebClient working together can fetch the larger jpeg stream and display it in another Image.

For iOS

Display the images in collection and modal views. The first view shows all images in the collection view layout. The second view is a modal view, showing a user-selected image from the collection view as discussed in class.