CIS 3515 Assignment 7

Instructions:

Over the next series of labs we will create a Web Browser app, enabling a user to enter a URL, and displaying the specified website in a WebView. Your application will be built using fragments, with each fragment having a specific function.

1. Familiarize yourself with the WebView widget (reference: http://developer.android.com/reference/android/webkit/WebView.html)

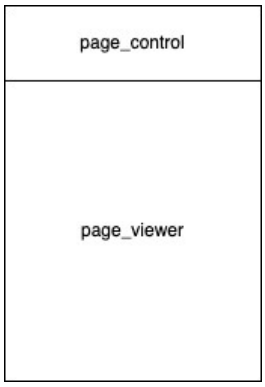
2. Add the following permission request line to your AndroidManifest.xml as a child of the <manifest> tag as follows:

   ```
   <manifest xmlns:android="..." package="...">

      <uses-permission android:name="android.permission.INTERNET"/>
   ```

   This will allow your application to access the internet (it's a web browser after all).

3. Rename your main activity to **BrowserActivity**, and design its layout to contain two containers (FrameLayouts): **page_control** and **page_viewer** (illustrated below).



4. Create two fragment classes, **PageControlFragment**, and **PageViewerFragment** with the following features:

   1. PageControlFragment

      1. This fragment should contain an EditText along with 3 ImageButtons.

         - The EditText will allow the user to enter a URL

         - The ImageButtons should be *Go* (load the entered URL), *Next* (go to the next page), and *Back* (go to the previous page).

        - The ImageButtons will use images instead of text to tell the user how they are used. You can find images online or create your own.

    2. PageViewerFragment

        1. This fragment should contain a WebView

            - The WebView will display webpages when requested

5. When your activity launches, add an instance of PageControlFragment to the page_control container, and an instance of PageViewerFragment to the page_viewer container

6. When the user clicks the Go button, load the URL from the EditText into the WebView object using its ***loadURL()*** method.

7. When the user clicks the Next or Back ImageButtons, your webview should go to the next or previously visited page respectively.

8. When a user clicks on a link from a displayed webpage, the EditText used to accept the URL should be updated to show the URL of the page currently being viewed (the link that was clicked).

9. When the user rotates the device, or changes the device configuration in some way that forces an activity restart, your activity should maintain the previously attached fragments instead of creating new ones. Additionally, the webpage that was previously being viewed in the WebView should also be displayed again, along with the WebViews history (next and previous pages). You are **not allowed** to disable activity restart.

10. WebVeiews throw a MalformedURLException if you attempt to load a URL that is not well formed (for example, does not contain the http:// at the beginning). If the user enters an incomplete url (such as temple.edu instead of http://temple.edu), your application should correct the URL before loading to ensure no exceptions are thrown. You **must not** demand that the user enters the URL correctly. Instead, your application must make the correction automatically.

Upload your project's GitHub link to Canvas.

Considerations and Hints:

- As described in the Guide, giving a WebViewClient object to your WebView will allow your WebView to accept page navigation requests (when the user clicks a link on a displayed website). In addition, Subclassing the WebViewClient will give you the ability to complete some of the requirements of this assignment.

- You can enable JavaScript in your browser by making a call to:

```
webView.getSettings().setJavaScriptEnabled(true);
```

Rubric

Each fragment type contains specified views (no views in Activity)	10%
Browser is able to load an entered website, and Next and Back navigation functions	10%
UrlEditText shows updated website URL when a link is clicked	20%
Fragments retained when activity restarts (instead of creating new ones each time)	20%
WebView reloads currently viewed page when activity restarts	10%
WebView maintains its history when activity restarts	10%
Malformed URLs are corrected before loading	20%