**INTERNSHIP WINDOWS PHONE APPLICATION**

**Creation Date: Spring 2014**

**Class: CS396 Mobile App Development**

**Instructor: Pete Tucker**

**Group Members: Kelsey Glynn, Thomas Glasser**

*Current status of project is also available at:* [*https://github.com/tglasser15/InternshipApp*](https://github.com/tglasser15/InternshipApp) *for viewing.*

# Description of Application:

Overview:

The *Internship Windows Phone Application* will serve its purpose as a tool and or resource in helping students search for internships. This mobile app, although as of now is only accessible through the Windows Phone, will also in the future allow students to access links to those scholarships. This scope of which these internships are pulled from is very limited, and as of now are only through posted internships through Whitworth and Whitworth affiliated sites (Whitworth’s main website, Twitter, Facebook, and so forth). I will be working with Career Services to how we can get a better look at how this app can be more advantageous to students.

Requirement Specifications:

The core of the Internship App is its integration with API servers. What this looks like is that internships that will be posted on a variety of Whitworth’s websites (as listed above) will be able to be seen on the app. For example, let’s say Career Services posted a new internship about Biology in Spokane, WA on Twitter. The app, with internet connection, will be able to sync with Twitter and access those posts (or tweets) by username. Although the API integration is only limited to Twitter as of now, the goal is that as the end of the semester approaches, more API servers will be integrated into the project allowing for students to pull from multiple resources, including outside of Whitworth’s scope (this includes pulling form websites such as InternMatch and other internship websites). The hope is to maximize the experiences that students will receive.

Here are some of the specific features of the app**: Internship searching, bookmarks, and saved searches**. With the internship searching, students will be able to search for internships either by a general search, by field of interest, and/or location. In addition, on the same page, it will allow to user to click on recent internships posted. The bookmarks features will allows users to bookmark internships that seem interesting to them so that they will be able to come back to them later. The saved searches feature will allow users to come back to searches that they previously have used. These three features are at the core of the app.

The application is interesting because most importantly, it will provide a tool and resource for students to use. In other words, it will be app that can be put to good use and is a practical application. In addition, after talking with students, it is sometimes hard to find good applications or websites to look for internships. Thus, why not have a mobile on your phone where you can pull from multiple sources and see all the information there, all-in-one? The application is also interesting because of the API integration – it’s something different and will be helpful for future use in building apps.

The core features listed above are the main features and goal features of this app and will be the only ones implemented so far. Thus, there are no features as of now that we are not going to implement.

Thus, the following are the basic requirements of the app:

* Provide **easy navigation** so that the user knows where to go and how to backtrack between pages.
* Provide a **Login** page, with the ability to create a new username and password – this allows the user to have their own personal login credentials to the app.
* Have all three core features implemented: **Internship searching, bookmarks, and saved searches**.
  + Allow users to search for internships on a few filter criteria
  + Allow users to see recent internships posted
  + Allow user to bookmark internships of interest
  + Allow users to save search items so that they may come back for later use
* Have a **nice layout** of the interface so that the app is more appealing.

System Requirements:

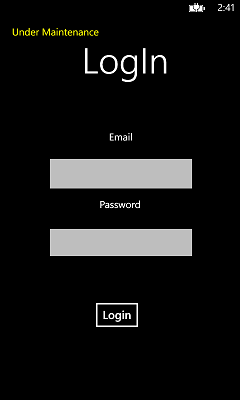
This mobile app will make use of both supporting architectures, API servers and database systems. With the API server, the app will be integrated with a variety of API servers in order to provide a large pull or resources so that users will potentially have the maximized experience of the app and getting the most out of the resources. As of now, the only API server used is Twitter. Currently, the app is pulling tweets based on username, thus when that username posts new tweets, the app is able to already handle how it manages those tweets. For the database system, this will be used to hold current information on users: 1) The database system will hold personal information about the user, such as their username and password, 2) the database system will be able to hold bookmarks based on that username, 3) the database system will be able to hold previous, saved searches based on that username. Thus, the database system will serve as a reference point to authorize users or create new ones.

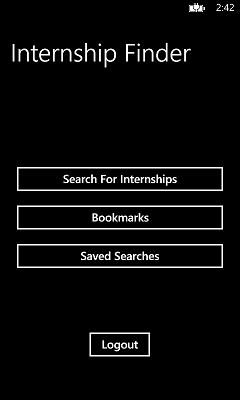
The project requires the following software in order to use:

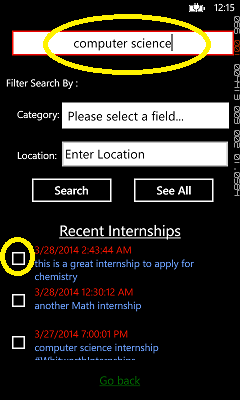
* Visual Studio 2012
* Visual Studio 2012 SDK
* Windows 8.1
* Windows Software Development Kit (SDK) for Windows 8.1

# Developmental Process:

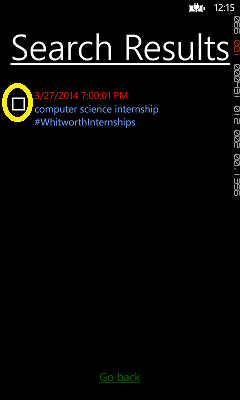
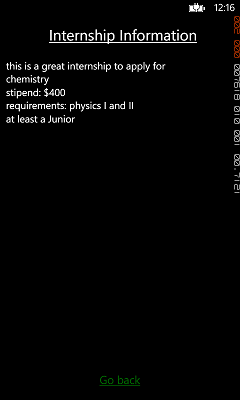
Detailed discussion of application:

The Internship App will serve its purpose in meeting student need with that need being having a successful app where internship information can be gathered in one general place. The layout of the application is such:

1. The user will have their own unique login credentials or the ability to create a new one. The login credentials will be checked against some sort of database system that will manage this information accordingly.
2. The proceeding menu will allows the user to make use of the core features as noted above. This will serve as the main navigation page.



1. Upon clicking the “Search for Internships”, the user will be given the following image on the right. As you can see, the user will be able to filter their searches or do a general search of internships. In addition, the user will be able to look at recent internships posted.



1. *Left:*  Results based on the searches will be given, allowing the user to click on the internships they wish to view
2. *Right:* The resulting click will view information on the internship.

The other features not displayed are the bookmarks page and saved searches page. Again, the bookmarks page will allow the user to view bookmarked internships while the saved searches page will allows the user to view previous, saved searches so that they can easily find internships.

Another feature of the app will be of course to have the ability to remove “expired” internships. What is great about the API integration is that once the user’s that the app is pulling from removes the posted internships, the app will not read in those internships thus already “removing” the internships. However for others, which many internship websites still contain expired internships, the app will be able to remove expired internships based on their date and time given by those posts.

The app requires internet access, either wirelessly or plugged-in. Because of this, the app will also make use in downloading the internship information before hand and/or saving the internships that have been bookmarked allowing the user to only view the internships that they saved.

Target device:

The target device will be the Windows Phone, version 8.1. Because the app requires integration of an API server, and with the C# programming language being one of the programming languages we know best, we thought it’d be easier and appropriate to use something we know best while at the same time trying something new and different. In addition, the Windows Phone, version 8.1 will provide a nice layout to our interface and design and with Visual Studios, the easy-to-use tools makes it easier to implement the features we wish to implement. The hope is that in the future, this app will be readily available for Android and iPhone users, providing even a broader network and use to students that have these devices.

Alternate designs and algorithms:

An alternate design would be for distributing some of the features in different places. For example, recent internships are posted on the same page as the search page. This may be helpful to have as another core feature on the main page when the user just wants to view a longer list of recent internships. In addition, the design will be able to handle multi-tasking, so that the user does not have to depend on only buttons on a certain page, which the app currently does. For example, the “logout” button is only on the main app page. It would be nice if the user was able to logout at any point during the runtime of the app rather than just on that main page.

To add to the alternate design, with the main app page, it would be great to make all the buttons a tabular event. Currently, I have three pages set that can be navigated from the main app page. However, it’d be more efficient if all that information can be viewed on that single page and this I believe can be handled by using tabs to navigate instead. This way, the user would be able to look at all their bookmarks and saved searches without actually having to open up a new page.

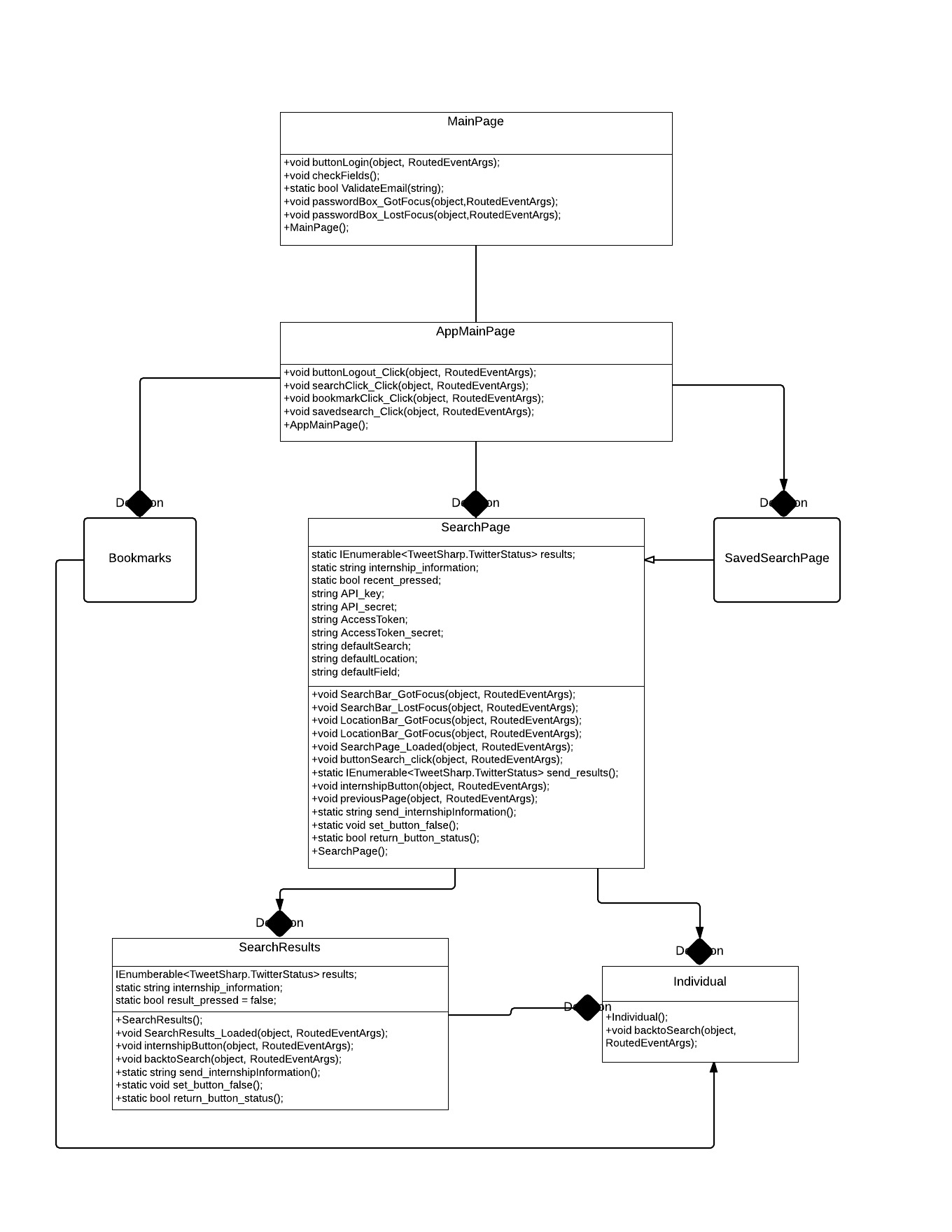
In terms of algorithms, the only algorithm being used is the algorithm to pull tweets off of a user’s profile. An alternate algorithm would be for the tweets to be pulled from multiple user profiles as possible so that it’s not just limited to one user. This would again allow users to pull from more resources.

Effects of Carry Principle:

* Personal – The login in page was a result of the personal aspect of the carry principle. The app will provide a means of allowing users to have their own unique username and password while being able to access their own personalized bookmarks and saved searches. An additional feature may be to add settings to which the user can access to change color theme, text color, etc.
* Always connected - Since this app requires internet connection, the user must always be connected to some internet source. However, since this always isn’t the case, the app will be able to previously download a set of internships and or save the internships that have been bookmarked so that the user can at least access the internships they saved.
* Small – The buttons will be large and text large so that the user can reach all features with the use of one finger and view the information without zooming in.

Schedule:

* **Project Design: Mar 31**
  + Completed
* Week of April 7th:
  + Speak with Career Services about internships
  + Implement the bookmarks and saved searches features
* Week of April 14th: Get the project ready for demoing beta version
  + Implement more API servers
* **Project Prototype/Beta: Apr 16**
  + Already completed – have working buttons and features already implemented with design
* Week of April 21:
  + Implement login credentials and authorization
* Week of April 28 – May 2:
  + Work on parsing for links and opening a new browser upon click
  + Work on dealing with expired internships
* Week of May 5: Fixing bugs and errors (if time, provide a settings feature)
* Finals Week: Write-Up and presentation
* **Final Deliverable May 16**

****

ADT/UML Diagrams

# Class Concepts:

Concepts from class applied

The carry principle was a concept that was applied to this project. As we were creating this mobile application, we had to think of ways of how the app would affect a wide range of users and the carry principle provided a standard list that our app can meet in order to be successfully. For instance, we used the principle of small, personal and always connected as the current capabilities that our app possesses.

Another concept applied was that since our target device is the Windows Phone devices, we used information gathered about C# libraries and made a large use of those libraries in our project.

Concepts outside of class that were applied

A concept outside of class that was applied was the use of API integration with mobile app projects. We have never heard of this so this project will be a new and different experience. How this specific API integration in our project works is that it first checks for an internet connection then authorizes specific API keys and access tokens in order to pull from feeds off of twitter based on username. These are placed in a holder variable that contains all of the tweets.

# References:

For obtaining the twitter feeds by username, I have used the following resource’s example code as the basis for the project:

[**http://code.msdn.microsoft.com/wpapps/Twitter-Sample-using-f36bab75**](http://code.msdn.microsoft.com/wpapps/Twitter-Sample-using-f36bab75)

# Acknowledgements:

**This project has been completed at Whitworth University, which has provided the materials and tools to complete the project. Thanks to Pete Tucker for helping with the project.**