# Technology Plan

# Submitted By: Christopher Newby

# Class: CS 350

# Instructor: Mark Seaman

# Date: 10/16/17

## Framework Selection

Select 3 Most Important Frameworks

Research them and list the vital issues with each

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Popularity** | **Future** | **Community** | **Expertise** |
| ASP (.NET) | Very high, most popular .NET framework. Has the most resources and widely used base of .NET frameworks. | Expansive, has potential for expansion as web services continue to grow. Has nearly ubiquitous presence on web. | Large amount of community support available via MSDN, stack overflow, etc. The proprietary nature of the .NET framework means it has less support than some open source frameworks, but plenty of resources available for consumption by novice users. | Less than 30 hours with ASP, but more than 700 hours with .NET framework and languages. Have had limited experience using ASP during internship at Encompass. Have used .NET for several projects of my own and for many school assignments. |
| SpringMVC (Java) | Most popular java framework. Has been around for more than 10 years. Geared for web apps, app configurations and security. | Has many applications as the use of java continues to grow. Java has many open source IDE’s and other tools available, meaning its popularity will continue to grow as web continues to grow in popularity. | Not as deep a pool of community support as python or .NET, but does have a large amount of documentation and resources available for use. Multiple free IDE’s are available for Java. | No time with Spring MVC, but have at least 300 hours with Java language. Have used MVC framework in .NET framework sporadically so I at least know what MVC is. |
| Django (Python) | One of the most popular python frameworks. Has massive potential for using python to develop web applications. | Python has shown significant growth over the past 10 years, and will most likely continue to grow as Django and other frameworks continue to make the | Deep support community. Open source means it has a lot of non-proprietary libraries and community support from various sources. The community support on this framework is probably the most helpful of the 3 listed frameworks. | No time with Django, but have 100 hours with Python language. My expertise with Python is mostly based around algorithms and data structures used in different graph search and sorting algorithms, and optimizing record search algorithms. |
|  |  |  |  |  |

## Tools to Use

Select the Tools that you will use for the Social Networking application.

* **IDE** – I will use Visual Studio Enterprise 2017 for this app, connected to SQL Express 2016.
* **Version Control** – I will most likely use Github for version control, but may rely on frequent external hard drive backups.
* **Programming language** – C#, HTML, CSS, possibly some Javascript and/or JQuery.
* **Code Libraries** - .NET library, Javascript, JQuery.
* **Design Tools** – Pencil, Sketch.io, Visio, and SQL Server Management Studio.
* **Hosting** – Azure or Github hosting.
* **Testing Framework** – Will use unit testing and system testing capabilities in Visual Studio 2017.

## Cost of Learning

Assess your experience level by counting hours you have spent using each tool or technology.

1. ASP – Have only spent about 20-25 hours with ASP, but have around 700 hours with the .NET languages. I would consider myself a novice with ASP, and leaning toward, but not quite proficient with the .NET library. I estimate needing 20 more hours with ASP to be able to use it with relative ease.
2. SpringMVC – I have zero experience with SpringMVC, but have approximately 300 hours using Java with Eclipse Neon. My experience with SpringMVC is beginner at best, but competent with Java. I would need at least 20 hours to learn SpringMVC, plus around another 15 hours to learn how Java fits into the framework.
3. Django – I have zero experience with Django, but have roughly 100 hours using Python with the IDLE compiler. My experience with Django is beginner, but just enough experience with Python to consider myself at the lower rung of competent. I estimate a minimum of 30 hours working with Django to learn how to use it properly, plus around another 20 hours to learn the context and use of Python within the framework’s structure.

For each technology that you have spent less than 100 hours with, add 16 hours to the project cost estimate.

1. ASP – 20 more hours of use + 16 hours = 36 hours cost of use.
2. SpringMVC – 35 hours of needed experience + 16 hours = 51 hours cost of use.
3. Django – 50 hours of needed experience + 16 hours = 66 hours cost of use.

Total the number of hours you will need to learn the tools.

1. ASP – Time required to learn tools is minimal compared to other listed tools. Another 36 hours of consistent use with this tool would be enough time needed to learn enough to be able to adequately use ASP. My knowledge of .NET gives me an advantage in this tool, and my overall cost of learning with this tool is minimized.
2. SpringMVC – Estimated time to reasonable knowledge of use is 51 hours, with possibly another 10-20 to moderately increased productivity. Refamiliarizing myself with Java would come quickly but would be a part of the increased productivity cost.
3. Django – I have found several tutorials with Django that seem to be very helpful, so I will estimate that I would need about 30 total hours of time spend on Django before I could adequately use this tool. Python is not my first language, so the additional time needed is to familiarize myself with the additional knowledge of the language itself needed to further my use capabilities, and in the context of its use in this framework.

Maximum tools cost = 7 tools \* 20 hrs/tool = 140 hours

This is the startup cost that must be paid before you can produce value for the project.