Capstone Journal #1 – Work Log

**02/04/2017-02/05/2017**

10 Hours

Stated planning the system and laying out my schedule and the system lifecycle. Created some draft documents and a project plan.

**02/11/2017-02/12/2017**

5 hours

Moved onto the design phase, wrote some more draft documents. I also started diagramming the system flow and GUI designs.

**02/19/2017**

8 hours

Finalized the system specification and design documents. I will not be able to commit many hours to this project for the next few weeks because other work is taking priority. Finished so far:

* Project plan
* System overview
* System architecture
* Technology choices
* Security
* Data normalization
* Database structure
* User views

**02/22/2017**

10 hours

Thankfully its now reading week and I can catchup with this capstone project. I spent the first half of the week working on my ORSIE project but I will devote the next half solely on getting the web API built.

I bought a server/domain subscription to host my portfolio website earlier, so I may as well use it for this as well. I created a MySQL 5.7 database and passed it to the web server. I than wrote a series of SQL scripts to create the schema and populate the tables with some light sample data.

After reading through MSDN’s documentation and watching some tutorials, I created an ASP.NET Web API 2 project solution. I added classes and functionality for user records.

**02/25/2017-02/28/2017**

20 hours

Most of the back end of the app is complete. I doubt I’ll have the time to complete every controller/model but it will just be a matter of copying code and changing variable names.

Completed:

* Model/controllers for game, codex, and ally records.
* Initial testing
* Basic user token authentication
* Error handling
* HTTP responses

**03/01/2017-03/02/2017**

12 hours

I didn’t finish all that was planned but the web API is good to go. I was going to use my GoDaddy server for the web API as well as the database but apparently CPanel services doesn’t support ASP MVC. I subscribed to Microsoft’s Azure service and published the API to the cloud server. The IP is not static so I must continually update the IP permissions on my SQL server. This is a short-term solution. If I have the spare time, I’ll probably switch to a web server from a different host.

Overall, the API is 90% complete and the database is complete outside of additional test data. I’ve been thinking of shortening the two client apps into one application for the sake of brevity.