CS 5551 Project Deployment

Adam Carter
Class ID: 9, Project Group ID: 1
CS 5591 – Spring 2015

I. Deployment Artifacts

Provided here are the URLs for all final document artifacts:

Final GitHub Repository:
https://github.com/apshaiTerp/cs-5551-final

Project Video:
https://www.youtube.com/watch?v=jVqatAzhE_o

CS 5551 Project Management

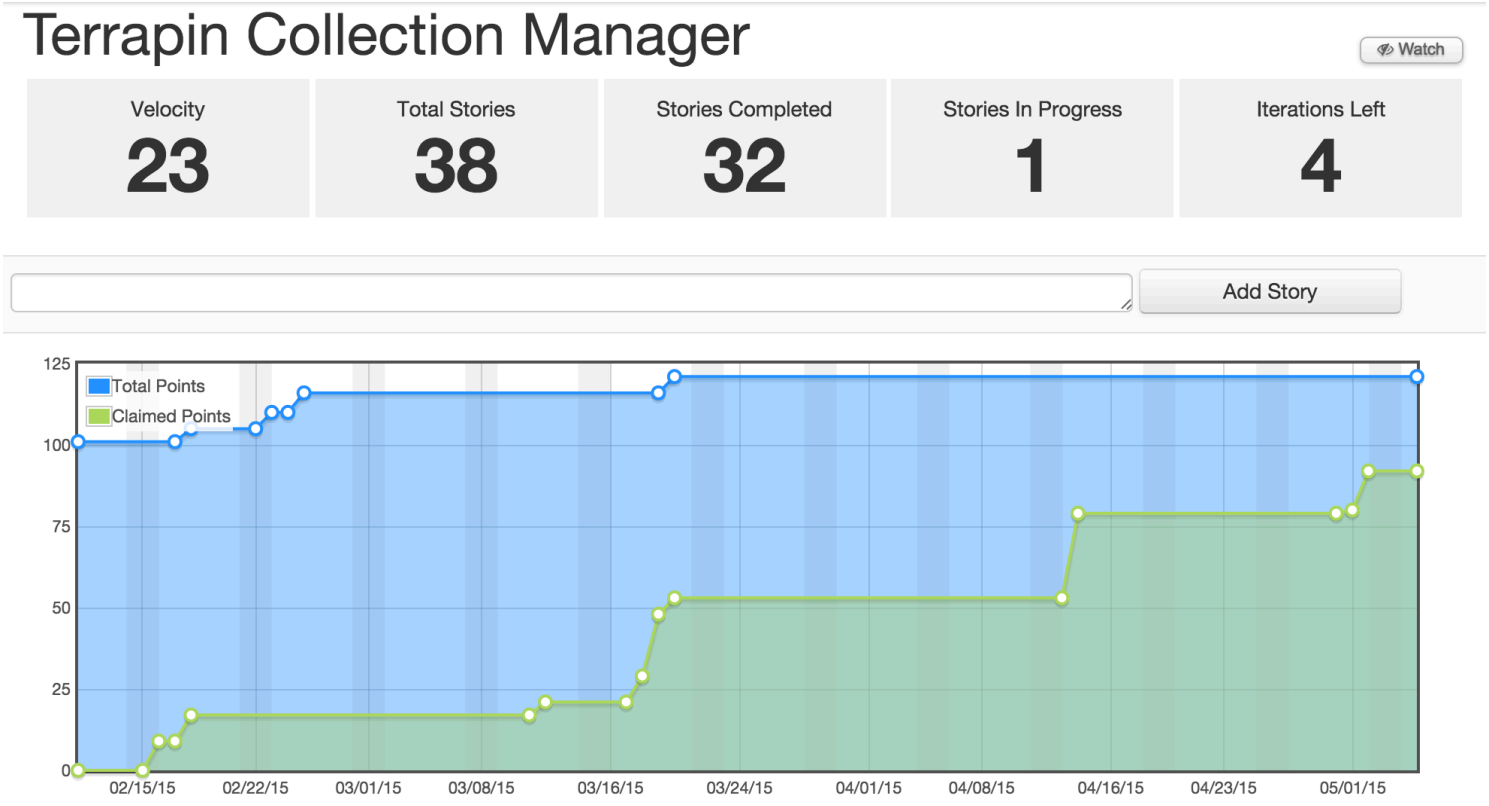
Adam Carter
Class ID: 9, Project Group ID: 1
CS 5591 – Spring 2015

I. Project Management Summary

This project spanned the course of three months and four iterations within those three months. It was performed using Agile development methodologies. In practice, the agile approach served only as a means of tracking progress, since I worked on the project by myself and there was no coordination of tasks among team members.

To summarize the project metrics, 38 stories were created. Of those, 32 stories were completed, one story was left in progress at the completion of this project, and five stories were not begun and left in the backlog. Because of the lack of a need to coordinate work effort, I did not actively update the ScrumDo project at regular intervals, so the burndown charts do not actively reflect regular progress during the iteration.

The ScrumDo Agile Planning history site for this project can be accessed here:
https://www.scrumdo.com/projects/project/terrapin-collection-manager/summary



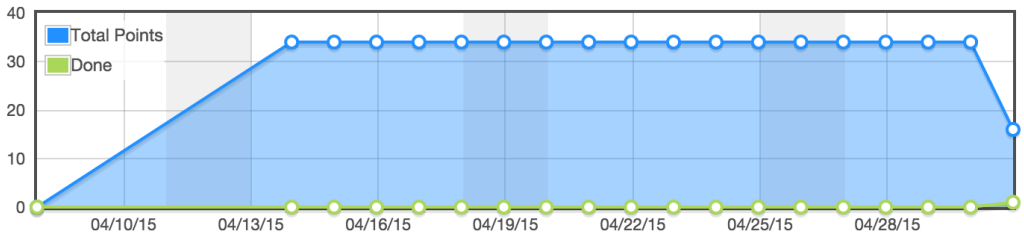
Overall Project Burnup Chart

Fourth Iteration

Apr 08, 2015 - May 01, 2015

13
Points Completed

0 Starting Points
34 Max Points

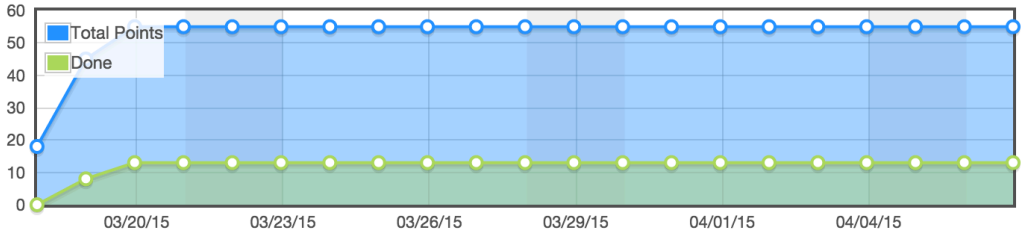


Third Iteration

Mar 18, 2015 - Apr 07, 2015

39
Points Completed

18 Starting Points
55 Max Points

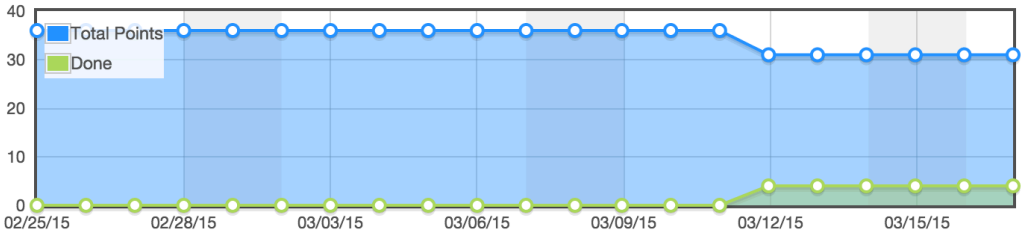


Second Iteration

Feb 25, 2015 - Mar 17, 2015

12
Points Completed

36 Starting Points
36 Max Points

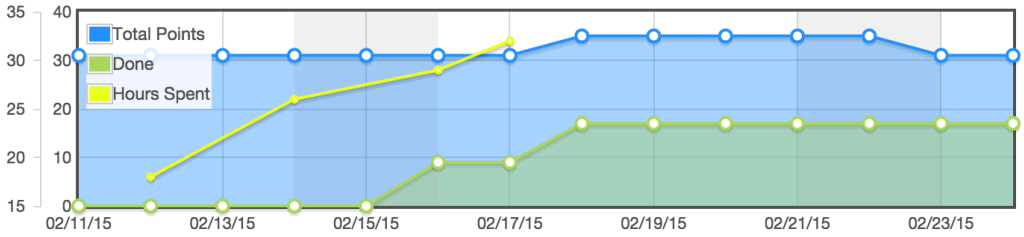


First Iteration

Feb 11, 2015 - Feb 24, 2015

28
Points Completed

31 Starting Points
35 Max Points



Iteration Burnup Charts for each iteration

I think the charts are a fairly accurate reflection of the work performed. I experienced a number of delays during the second iteration working through some of the data parsing tasks. I did not assign story points for some of the last iteration stories because I was more focused on completing the stories themselves than maintaining the storyboard.

CS 5551 Final Project Evaluation

Adam Carter
Class ID: 9, Project Group ID: 1
CS 5591 – Spring 2015

I. Introduction

The purpose of this report is to review the project as a whole at project completion. When I began this semester, I had no experience in web development at all, so many of the concepts and technologies were new to me. I also made the decision for several reasons to work independently, which put more demands on my time than perhaps other groups experienced. I chose a passion project, which brings with it personal expectations of quality and usability, and approached this project as a change to build a platform and learn skills that I would expand upon as I continue to grow and improve this system.

I general, I feel very impressed with the work I completed. I'll discuss some of the specifics in later sections of this document, but given where I was to begin the semester, I know that I learned a lot about web design, and had to learn quickly to be able to implement a rather ambitious feature set. All told, I created about 25k LOC as part of this project, which is a significantly larger project than anything I have ever taken on outside of work.

I think the sentiment that best describes the state of my system is showing promise for future development. I feel like I've implemented most of the core functionality required to make the system viable, but there are many more features I will want to add to make it into a system that both looks and performs like something I would want to use regularly.

II. Comparison to Original Plan

In my initial project plan, I proposed the following set of system requirements:

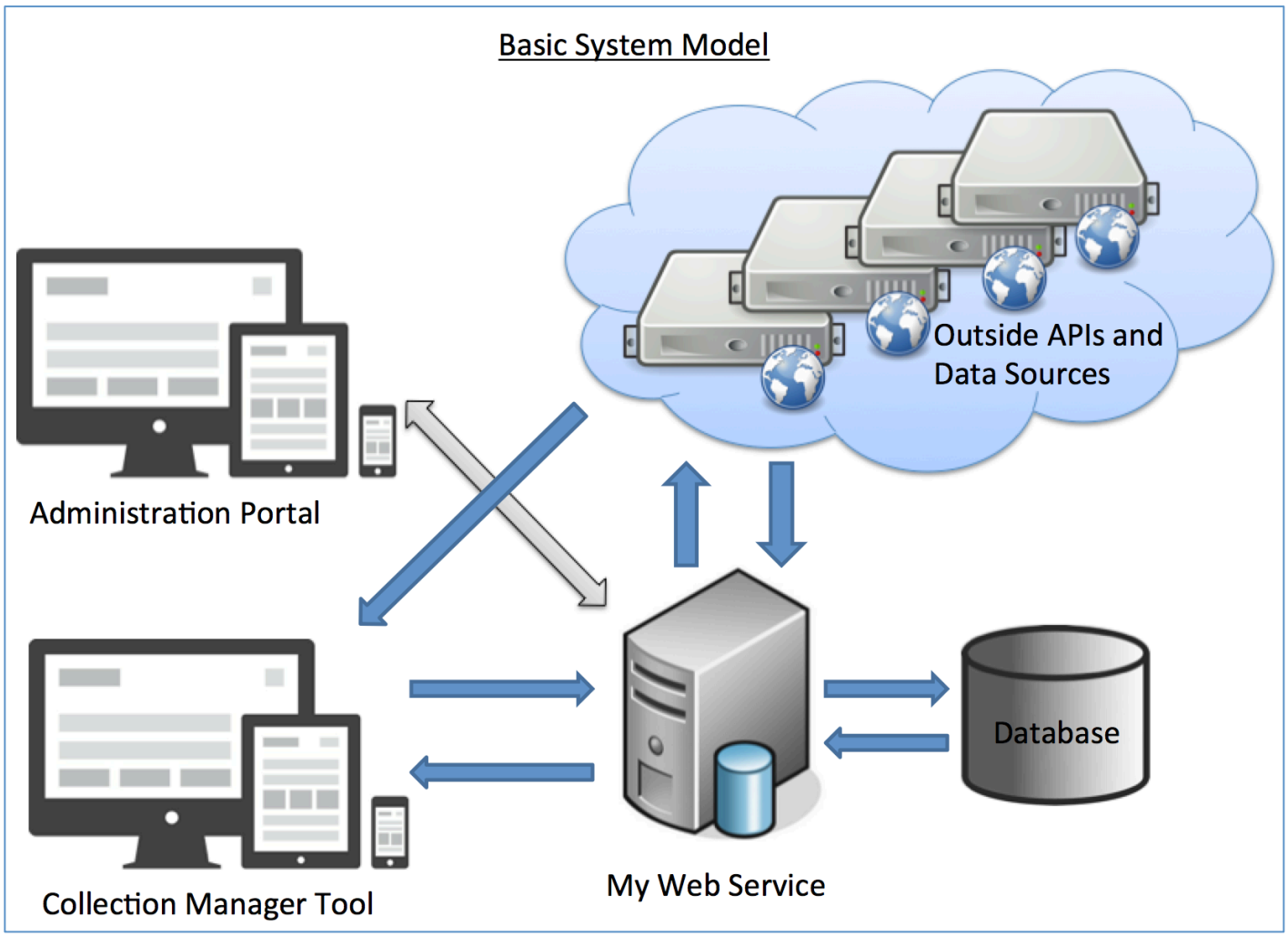
Requirements

i. The system shall allow users to create personal accounts to access the system.
ii. The system shall allow users to build a personal game collection.
iii. The system shall allow users to select any game from the BoardGameGeek index to add to their collection.
iv. The system shall allow users to search their collection for games based on common criteria.
v. The system shall allow users to access retailer information for games available for purchase at their respective sites.
vi. The system shall allow superusers to map board game entries to retail offerings when unable to be mapped automatically.
vii. The system shall allow users to build a list of games they desire to purchase soon.
viii. The system shall allow users to add or view supplemental game content such as game manuals and related video content.
ix. The system shall allow users to edit subjective game definitions like game weight.
x. The system shall be able to distinguish between Base, Expansion, and Collectible games.
xi. The system shall be able to detect new offerings from potential retail providers.
xii. The system shall allow users to be notified of selected system events.

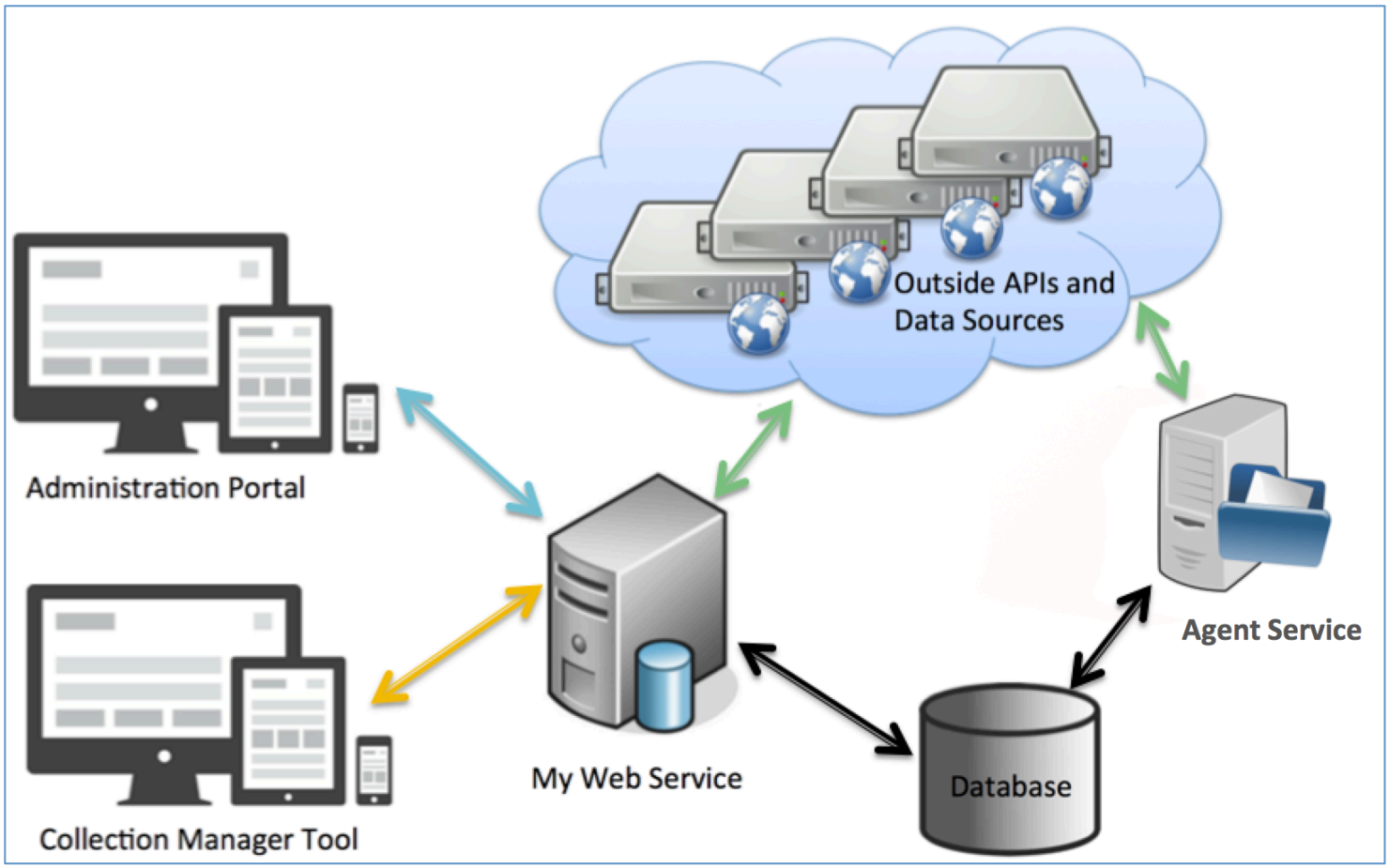
From this list of twelve requirements, ten were implemented completely, with one partially complete (Requirement VIII) and one not yet begun (XII).

I think with those exceptions, I was able to realize the initial goals of the system as I designed them.

I also feel like my initial project plan and story planning were pretty consistent through the project, with the exception of the emergence of the background agent processing system. This was the only really new element in the system architecture. I'll post the architecture images here to show the difference between the plan architecture and the architecture as implemented:



Original Model from Project Plan



Final System Architecture Diagram

III. Skills Learned

I came into this semester with a high amount of Java experience, but no web development experience. I've used traditional Relational Databases for a long time, but decided for this project I wanted to dig in and learn how to use a NoSQL database. I was familiar with REST architecture principles, but had never worked on such an architecture before, so all of these elements were new.

One benefit I had in this particular case is my pre-existing domain knowledge in the Board Game domain. It allowed me insights into the data process and known elements, which greatly helped in system modeling.

Over the course of the semester, I feel like I reached a level of proficiency with web development technologies including HTML5, JavaScript, BootStrap, and jQuery. One of the biggest things I had to learn that I feel proud of is the extensive use of dynamic HTML generation required to get meaningful data on my sites. I also worked hard to create a Responsive design that works well in both traditional browsers and mobile browsers. I also feel like I learned to a much greater degree how to interact with REST calls, and feel much more

comfortable with my understanding of when to implement a PUT vs a POST and other semantic differences inherent to the REST architecture style.

IV. Reflections

I want to begin talking about Agile. When reflecting on the agile approach, I feel like in my particular case it functioned much more as a task list than it really did a project management tool. I have used Agile in a professional setting for nearly five years, so the Agile process is not new to me, but I found the experience of maintaining ScrumDo stories much more of a hassle working as a team of one than a benefit, which is why in later iterations I would only update stories one a week or so, and sometimes not even until just before the iteration cutoff.

One challenge I faced was of my own making. Knowing what it would take to get a system like this up and running, I spent a lot of out of the classroom time implementing my system. Some of that time was spent learning and trying to get up to speed, which I tried to mitigate by focusing on core Java architectural elements while I was working on the class labs to gain exposure in the web development skill set. I believe it's fair to say that most weeks since beginning the project in early February I committed 20 – 25 hours of time, in addition to my work responsibilities and workload from my other class this semester. This meant a lot of late nights, and it probably wasn't until about halfway through the project that I really began to grasp how much time was going to be required to implement this system the way it needed to be done to demonstrate value.

If I had to do it again, I'm not sure what I would change to better manage the workload and burnout. With so much new to learn, I probably should have set more modest expectations in terms of what I set out to accomplish. Even with that, it is hard for me not to also consider how much work I have left to do to improve the system to the point where it's truly stable and usable and bring it to the point I would feel comfortable with public consumption.

One final comment I wish to make regarding this project is the amount of what I would consider to be busy work. Perhaps I found some of the assignments more difficult than other groups because I was working on a team of one, but I found many of the documentation tasks to be particularly tedious. One example here is the heavy reliance on Class Diagrams as a part of documentation. In my professional efforts over the last eleven years, I have never once either begun a project with class diagrams, consulted class diagrams, or been required to generate class diagrams. For this project, outside of my initial proposal, every diagram I created was generated from code. In my personal opinion, most systems large enough to do any meaningful work render such a large web of UML that such diagrams are simply noise. They contain too many objects to be valuable, and very few UML modeling tools can efficiently model or organize a system of more than 20 objects. This was a point I tried to make in my Second Iteration report, including links to the images generated, but because the actual useless image was not included in my document, I lost points on the report. I have since included them, despite the fact that the models are rendered so small as to be illegible. I find state diagrams and other variants of UML to be even more tedious and of no value, which is why I never use them and refused to generate any such thing as part of the system.

I think the overreliance on such forms of documentation actually runs counter to the notion of agile development, forcing a form of waterfall design into the agile mold. Students are being asked to think about design before beginning, which for most students is probably necessary, but perhaps because those are lessons I have already learned, to me they felt much more like busywork than necessary.

Similarly, I understand the intent behind this final report, though I find it the most ludicrous assignment we were asked to do yet. We live in a time, and have been for some time, where the notion of buying software in a box at the store is almost obsolete. My laptop doesn't even have an optical drive anymore, so the idea of creating a CD-ROM and a small tree's worth of paper documentation is ridiculous, especially when the end product is a website and related architecture.

If you really wanted to simulate what modern software release process is like, then require students to generate an online user guide as a wiki. The only paper documentation that ships with products these days is a Quick Start guide.

It is much of this type of work that I found most tedious, because it feels ten years behind where the industry is now. It's an experience I'm sure many students have never had, but the approach feels very dated, when there are better, freely accessible ways of creating the same experience.

V. Conclusion

Because this is a passion project, my ultimate goal is to bring this system to a point where I would feel comfortable having other users interacting and using my tool to help them manage their board game collections. I still feel this to be an underrepresented space in the hobby, but there are still a number of features I want/need to implement before that could become a reality. I am very pleased with the work I was able to do. I implemented a consistent graphical theme that has a comic-book style that works well for this domain. I have the foundation of a capable system that I can easily continue to expand and improve upon. I've removed some of the intimidation behind web development that I felt before attempting this project.

I look forward to continuing to work on this system and help it realize all the goals I outlined in my initial proposal and plan.