# Patrick Brossard - Week 3 Status Report

I’ve been progressing well along my project plan. I have not quite completed the requirements documentation like I had hoped for this week, but I have completed the database schema which I thought I would do next week. The use case details documents have been more time consuming than I envisioned. Here’s my current status:

|  |  |
| --- | --- |
| Week | Work Item |
| 1 | ~~Concept Draft~~ |
| 2 | ~~Research Proposal~~  ~~Technology Selection~~  Stakeholder Letter (signature still outstanding) |
| 3 | Requirements Documentation (Use cases complete, use case details in progress)  ~~Hosting and Network Documentation~~  ~~Status Report~~ |
| 4 | ~~Database Schema~~  Start Software Architecture  Start UI Wireframes  Status Report |
| 5 | Software Architecture, continued  UI Wireframes, continued  Prototypes  Status Report |
| 6 | Complete Software Architecture  Complete UI Wireframes  Prototypes, continued  Status Report |
| 7 | Complete Prototypes  Finalize Presentation  Reflection Paper  Status Report |
| 8 | Presentation  Finalize Project deliverables  Final Submission |

## Major Milestones

The technology selection was a major decision which needed to be made. I have decided to build on a Microsoft technology stack. All the various considerations will be in my final report, but the decision came down to Microsoft/C#/ASP.Net versus Ruby on Rails versus Clojure/Java. My relative familiarity with the Microsoft stack, as well as a new (and legal!) option to avoid software licensing fees finally pushed me in that direction.

Completing the basic requirements for the site was also a major step. I had to constrain my scope quite a bit to make the design fit within what I could get done. At least there is room for lots of improvement in the future! I’ve been jotting down some notes on future development which I will also include with the project.

Once the basic requirements were done, I found that flowed easily into building out the database schema. I banged out the database design and implementation in a few hours. Database design is nice in that once the design is done, the implementation is as well! At least in terms of tables, there will probably still be some stored procedures to create.

The network design was fairly straightforward as well. Full discussion will be in the paper, but I have decided to go with Amazon’s “Elastic Compute Cloud”. After I did research on a few platforms, I found its combination of features to price were superior.

## Research Activities

In terms of requirements gathering, most of the “research” was just me putting down on paper what I felt the site should do. I did do some research into a better tool for requirements capture. I’ve settled on ArgoUML (<http://argouml.tigris.org/>) as I have always hated the UML functionality in Visio. After having used it a bit, I’m lukewarm. It’s better than Visio, but it has its own wrinkles.

I did quite a bit of research on hosting. This was mostly in the form of checking out pricing and specifications of service providers. Lots of web browsing.

I have finished reading the excellent Domain-Driven Design Quickly book (<http://www.infoq.com/minibooks/domain-driven-design-quickly>). I think I have a pretty good grasp on the approach and documentation I need to build.

Finally, I have started downloading and compiling some ASP.Net MVC “reference” type architectures – applications which don’t really do much other than model good design principles or a particular approach to architecture. The ones I will focus on for the purposes of the project are:

The Sharp Architecture: <http://sharparchitecture.net/>

Nerd Dinner: <http://nerddinner.codeplex.com/>

Mvc Music Store: <http://mvcmusicstore.codeplex.com/>