**Meeting Minutes**

**Date:** February 7, 2014

**Start Time:** 1:00pm

**End Time:** 4:00pm

**Members Present:** Drew Aaron, Michael Beaver, Chad Farley, and Travis Hunt

**Members Absent:** Clay Boren and Andrew Hamilton

**Topics** **Discussed**

* Software Requirements Specification
* Client Questions
* High-level Architectural Overview

**Decisions and Actions Taken**

Part of the team met to discuss the software requirements specification document, develop client questions, and draft an initial high-level architectural overview of the backend design. The team currently has nearly enough information to complete the first draft of the specifications document; however, more information is required from the client. Thus, the team will be addressing these concerns in the next round of client questions.

The team is faced with the problem of how to organize and design the backend. The team has opted to make the backend as modular as possible and to represent each component (or module) of the backend as high-level class objects. Details will be determined later.

In the future, the team will need to ask the client whether or not she wants to handle memory allocation with respect to best and worst case scenarios. Also, the team will need to address the problem of speed versus memory. The team needs to create error detection documentation. This needs to detail the errors, designate error codes, list the outputs of the errors, and detail how to recover from errors.

Travis helped Drew set up and connect to the Visual Studio Team Foundation Server repository service.

**Supplementary Information**

**Questions to ask the client:**

1. Who are the users of this software? Can you describe them?

2. How are users going to be using this software? What is the purpose?

3. What deliverables are required? By what time and date are they to be delivered?

4. Would you like the ability to set and run to breakpoints? Or would you rather the debugger be forced to step over each line, starting from the beginning?

5. What specific debugger functionality is required?

6. Should the text editor force characters to uppercase, or should it allow lowercase as well?

7. Would you rather the software run quickly with less memory or run slowly with more memory?

8. What is an acceptable rate of error or failure?

9. How would you prefer the software be licensed? Should rights be relinquished once development has concluded? Should the software be completely free and open source for *anyone* to modify and maintain?

10. Do you have any “security” concerns for this software?

11. What is the maintenance plan for this software? Will the team be relinquished of maintenance responsibilities upon deliverance of the software?

12. How do you want the software to be delivered to the users? Should it be portable, packaged in an installer file, or should there be options for either option?