Hi Ben, Raymod and David,

Please see some requested changes. Sorry for the brevity, let me know if you need any additional information, thanks.

Section 2:
"Currently, when the users connecting to the service exceeds a certain limit, their becomes a delay between the server response and a client’s actions."

The issue is larger than a delay beween server response and client actions. In particular, the Java Tomcat Servlet's lack a method of asynchronously communicating connection information with in object memory. The purpose of the main purpose of the project is to resolve this distributed system problem through the use of a shared memory cache with a write through or write back database back-end for redundancy.

Section 4:
"Implement database to track connection history of a single user."
This issue affects multiple users in a distributed system.

Section 5:
"We must configure Apache Guacamole to scale in such a way that a single server running an operating system can handle multiple client, or user, connections with minimal loss to latency or a connection crash."
This should read something like:
"We must configure Apache Guacamole to scale in such a way that ~~a single~~ multiple Java Tomcat Servlets and the AngularJS clients running in a distributed system scales. This will consist of multiple client, or user, connections connecting to several Java Tomcat Servlets scales with minimal loss to latency or a connection crash."

"We need only work on the scaling part of the system unless deemed otherwise by the team mentor."

Section 7:
Please clarify "A server must be able to run multiple operating systems without a drop in performance" Where are these operating systems running that you speak about?

This makes sense to me, but your documentation should be clearer.

Section 8.1:
"We assume that our server and client setup for testing on our own machines mimic the intended functionality for our mentor."

What is my intended functionality? We've meet several times you should be able to clarify these points in this technical documentation...

Setion 8.2:
"The version of Apache Guacamole that we pull from the docker container is quite out of date."
Please change to:

"The public DockerHub version of Apache Guacamole is out of date with some newer distribution packages. Therefore, we'll have to customize the Docker Container according to the steps provided by Michael Barkdoll to use the latest source code for the docker image."

Please remove:
"Our mentor’s wife is having a baby, so he will be busy with family things and may not always be available."
I've consistently responded in a reasonable time frame perhaps even more than other mentors might be available. Please avoid last minute waterfall development.

Section 9:
"Our team may not finish work on the event driven architecture part of the project. We aim to lay the groundwork, but depending on how much time we have left to complete this section, it may not get done."

Seems reasonable, please stay proactive on it though throughout your next stages of development so at the very least you can present how it should work in a practical manner with your final project poster/presentation. This way if the work isn't completed by your project it is something that I can pick up from in a later senior project. As discussed though, I think it is reasonable to complete this feature request considering how complicated a senior project is supposed to be and the fact that the shared memory cache should be relatively easy to implement.

FR-02 Connection processing
Should likely reference that the issue is related to connection groups in Apache Guacamole.

13. Use cases
"1. User connects to Guac server with username and password"
Technically, the user first connects to the apache guacamole-client docker container which has a AngularJS client interacting the a Java Tomcat Servlet.

"3. Different user at different web client connects to server and creates operating system instance"
Grossly inaccurate. An admin user will create a connection group with multiple servers that clients can connect to within the connection group. End-users connect to the connection group through I believe a balancing connection group that load balances the connections. However, the Balancing connection group lacks shared object memory across multiple tomcat servlets. This is where your work comes in.. 🙂