Game

Design Review for Books

March 27, 2005

Version 1.1

Summary

The main issues this review addresses are security, synchronization, specificity in regards to algorithms and hardware being used in addition to the system scalability to a large number of consumers.

Our recommendations are to give more thought to securing the users’ information as well as implementing a time-out feature. Synchronization is of high priority for an interactive system as this one so the consumer/producer problem should be discussed in more detail. Efficient algorithms and reliable hardware are also high priority considering the large amount of users on the system at any given time. Lastly, there needs to be more focus on more rigorous testing of the scalability of the system with the estimated 50,000 consumers.

Questions

1) Inconsistencies

Although there were no inconsistencies found throughout the document, there is one consistent detail which is left unexplained. There is no reason given as to why the number of 50,000 was chosen as the target amount of consumers logged into the system as well as if the system can be scaled to such a number.

2) Omissions

The document does not mention how the Books project will address two issues that are key to any online commerce interface: security and synchronization. Although the system is not responsible for the financial transactions, it will nevertheless be storing plenty of private information about the users - such as their contact information, sales history, etc. It is unclear how this information will be protected from spammers and other individuals with malicious intent. The document does not mention what will prevent malicious parties from accessing password files or the database itself.

Even if assuming that the personal data is stored very safely, the issue of session-timeout is not addressed. Systems like Ebay and Amazon will prompt the user for their password after a certain period of inactivity. The Books system does not define an idle-time period, and therefore, allows for third parties to access a user’s account if he or she is away from his or her computer for an extended amount of time.

Furthermore, if this system expects to service multiple clients simultaneously, it must be able to do so without race conditions. The document omits a description of how atomic operations and sequential integrity will be ensured. For instance, what will prevent two users from buying the same book at the same time? If two requests arrive concurrently, what will decide which client 'wins' the sale? Although the class diagrams are very extensive, they do not provide explanations regarding which functions must happen sequentially and/or atomically.
Lastly, the testing plan seems a little vague - a system of this size will probably need automatic testing in addition to manual testing. Trial runs by several people may not surface all the errors that can be generated by 50,000 customers.

3) Unclear Issues

Aside from the omissions described above, the document is very clear and concise. The class diagrams illustrate class functions, components, and dependencies in a comprehensive, programmer-friendly manner. However, there is no mention of any special algorithms that may be essential to the system's efficiency. More specifically, the system may use any number of searching and sorting routines. Even the most knowledgeable programmer would be unsure which routines the team had in mind from the document alone. It essential to have efficient algorithms in mind in the document since the system will have to service up to 50,000 users and system performance will rely heavily on search and sort speeds.

4) Technical Errors

The document's lack of mentioning synchronization issues may lead to some technical errors. Other errors would address how the user interacts with the system in certain features. From the site map, it would be more comprehensible to the user if (s)he could get to the login screen without having to view the registration page first. Additionally, it would perhaps be more convenient if the user could add a book to the database without first having to search for it.

5) Testing Issues

It seems that the group has put a lot of thought into their testing strategy. The time estimates are appropriate and the team has chosen an interesting strategy for writing the code. They have chosen to have everyone program units for all the different layers of the system so each member can understand the full inner-workings. In the long run, this will make every team member more effective at debugging the system, but this approach may also make testing slower. A CVS repository and good communication will be vital if the teammates are to simultaneously write to the same units.

6) Environment Assumptions

There are no external, off-the-shelf software components that need to be bought; the design document is very thorough in detailing the software elements of the system. However, there seems to be little information as to what hardware this system will be hosted on. The requirements state that the system has to support 50,000 users, but the design document lacks a description of the hardware that is to support such traffic.

7) Realistic Planning

The development plan seems very reasonable. It's obvious a great deal of thought has gone into architecting the system to be modular, and thus extensible. This removes many potential bottlenecks because subsystems act independently. Once the core functionality is in
place, additional features can be easily added, even ones beyond the current scope of the design document. Also, because the site is purely data driven, a fake, test database could be used to test features to which no interface that adds or generates data has been created. These design features lend well to incremental development, as it is suggested in the document.

The Books team has sent considerable time detailing the various components of the system. It seems that all the core functionality and many of the peripheral components have been thought through. The database scheme has also been provided as well. However, there was little documentation included along side their UML diagrams or written descriptions. Besides function names, there is little indication of how the various components interact at a fine granularity.

There were no time estimates provided, other than a final project completion data for the end of the semester.

The accompanying use cases and diagrams clearly illustrate a subset of the systems navigation. These were especially useful to understand the system interface and features for someone who is not familiar with the design or project.