CS 488 Draft SRS Guidance

As I mentioned yesterday in the briefings, our immediate goal is to compile a draft Software Requirements and Specification (SRS) document for your project. Although it should be complete from your perspective, I consider it a draft because I'll review it and meet with you to revise it into the "final" form. The final form ideally wouldn't change over the remainder of the two quarters, but such a thing never happens in software engineering. You will have to document any substantive changes. Keep this original version intact, and copy it to be the living version. At the end, you need to show that you satisfied all the requirements and also account for any differences between the versions. We'll discuss the form of the final deliverable next quarter.

There is no particular form for this document because every project is different. Research SRS formats and choose whatever you feel is appropriate. It has to account for everything you know about the project, including the softer aspects like background on the problem domain and the expected end users. Absolutely it needs the requirements (what needs to be satisfied) and corresponding specifications (how the requirements will be done). In some case, your client may be forcing certain specs on you. In others, you may have more than one possible approach to consider. This is fine and typical. You'll need time to prototype proofs of concept, etc. before you fully commit. But you need to know what your likely options are now. They may change, but hope not, and if they do, you'll have to account for the changes.

Formal use cases will be required at some point, but it may be too early. If you have them, include them; if not, get on eliciting them. Again, format doesn't matter. If you want to use UML, fine.

From the specifications you can do some basic planning of resources. Assuming the specs are complete, correct, and immutable (bad assumptions x 3), then it would be possible to estimate how long each would take, as well as the dependencies on which order they need to be accomplished in , and then who's assigned to do them. I will expect you to keep me informed in the status reports and briefings about these aspects, but realistically no timeline ever proceeds as planned, so expect a lot of fluidity. You're free to use any project-management tools you want. I don't care about the tools, but I will want to see the timeline and resource allocations in your briefings. I don't think it's appropriate to include it in your draft SRS because it's expected to change immediately, but your final should have your best estimate.

The draft SRS is due the Monday (27 Oct.) before our next briefing (29 Oct.) We do meet that Monday as a class for a presentation by FAST Enterprises. It's about an hour, starting at 2. Attendance is required. They hired several of our 488/490 students last time, and they've become a good source of entry-level jobs. (Which is partially why we're running these resume workshops to get you prepared. There's yet another one Thursday, 23 Oct.)

The final SRS will be due some time after the briefings. I can't set the exact date until I see how things turn out. From that point onward, you're working on the project. The SRS is your roadmap. The status reports and briefings are your connection to your boss and client. Make sure they stay consistent and in sync, and be prepared to account for any substantive deviations.

Everyone has had a software-engineering course. Refer back to it for guidance on how to proceed. That class (regardless of who taught it) was simplified with a controlled toy project with a known outcome that you could actually achieve. These 488 projects are real, and it's quite easy to bite off more than you can chew. We need to know what's going on all the time and then to adjust accordingly.