6.005 Project 2 Team Contract

Jeffrey Chan, Charles Liu, Evan Thomas

I. Goals

What are the goals of the team?

The team has three main goals in mind. The first goal is to create a functional, reliable, well-designed, and well-tested IM client that is easy and fun to use. The second goal is to achieve an A on the assignment through good design, implementation, and testing, and by spending enough time and effort on the project. The final goal is to have fun with the assignment, to make sure that everyone has a positive experience (with the project and with each other) applying the things that we learned in creating a potentially impressive program.

What are your personal goals for this assignment?

All team members seconded the team goals of getting an A on the assignment and showing off a functional IM client by the end of the project.

What kind of obstacles might you encounter in reaching your goals?

Our obstacles may come in many forms. The first obstacle is in design. We are tasked with breaking down decently complex program into manageable, testable parts. To ensure that things go as smoothly as possible, our first meeting or two will be focused on a discussion of design issues. We hope to break the program down into modular components that can be individually designed and implemented, and when we all agree on these components, start to attack the design of each one. We hope that every team member will contribute all of their ideas, since it never hurts to have more ideas. Then we will hopefully arrive at one decision through discussion and comparing pros and cons. If we cannot reach a unanimous decision, the majority will attempt to convince the other team member. If the decision only matters for a single member's tasks and not the overall integration of the program, we may continue with the decision, or we can ask our TA for advice.

In the implementation stage, we may encounter obstacles that were not thought of in the design stage. If the issue is minor, the implementer should feel free to make a decision as long as the overall design patterns and the integration of the program is not affected. Of course, everyone should feel free to bring up issues encountered in the team meetings and ask the entire team to brainstorm possible solutions. Finally, implementation and testing require time, so we hope to set deadlines for smaller chunks of work that will lead to a successful implementation of the program and will allow for time for testing and code review.

What happens if all you decide you want to get an A grade, but because of time constraints, one person decides that a B will be acceptable?

We will hold a meeting in which everyone will express their thoughts about the current state of the work. If the consensus is that we still want to go for the A, then we hope that the one team member will carry their weight. If not, see below.

Is it acceptable for one or two team members to do more work than the others in order to get the team an A?

In general, it is not acceptable for one team member to require that other team members "carry" them for the project to get done. That said, unexpected things do happen and for minor things ("hey, this just happened, can you change these few lines in my code") we are willing to help each other out. Otherwise, we will hold each other to agreed-upon deadlines, and will give gentle reminders as they approach. Although we hope that we will never have to deal with these issues, we will deal with one team member not carrying his weight internally first. If that doesn't work, we will meet with the TA, and if a resolution is not found soon enough that it impacts the final project, we will attach a note to the final submission that indicates the work done per person versus the initial agreements.

II. Meeting Norms

Do you have a preference for when meetings will be held? Do you have a preference for where they should be held?

Meetings will preferably be held at ZBT during the evenings, such as after dinner, or a mutually convenient time. ZBT is a mutually convenient location and we can set aside large amounts of time to talk and look over each other's code without being constrained with each other's schedules beyond lecture time.

How will you use the in class time?

We will use in class time for one of two things. Either we will work on individual things, anticipating a meeting in the evening, or we will meet if we feel that 50 minutes is enough to complete what we need to talk about.

How often do you think the team will need to meet outside of class? How long do you anticipate meetings will be?

We hope to meet at least once every other day, at the very least to check up on everyone, make sure that everyone's work is okay, and see if we need to adjust the division of work or deadlines. The first meetings will be longer since we need to make our design decisions, and in the future if we need to get together to review some code or revise our design, the meetings may take longer. We aim for meetings to be less than an hour unless we need to collaborate and review/integrate code, in which case it will probably take longer.

Will it be okay for team members to eat during meetings?

Sure!

How will you record and distribute the minutes and action lists produced by each meeting?

We will have an email thread going on that contains the action items and notes from each meeting. If that is insufficient we will move to a shared GoogleDoc.

III. Work Norms

Returnin and Project:

The team will meet Monday night for as long as necessary to finish the first milestone. Then team members are free to work on returnin things until the due date Wednesday. The team will allow for another day free of project deadlines and meeting commitments for team members to work on the returnin for the final two problem sets.

How much time per week do you anticipate it will take to make the project successful?

We anticipate around 10 hours of individual work per week, plus time spent in team meetings. Of course, this depends on individual coding speed. We also hope to make our design such that it is as easy to implement as possible.

How will work be distributed?

We will try to divide the project into modules logically based on the program's requirements, then come to a consensus on the relative amount of work in each module. We will first proceed by volunteer so everyone can work to their strengths, and if not all parts are covered, we will see if we can come to an agreement internally. If that does not work then we will resort to random assignment.

How will deadlines be set?

Deadlines will be set through a combination of the person responsible telling everyone when he can finish, and when the team needs the code by. Ultimately, we would like a high quality product and it would waste everyone's time to have to revisit large amounts of code, so we will try our best to let the person who is working on each piece set a deadline, as long as it is reasonable and does not conflict with the progress of the overall project.

How will you decide who should do which tasks?

As discussed above, we will first use a volunteer system and allow everyone to play to their strengths. If that doesn't work, we will try to assign to people who have completed first or have done the least, and if all else fails use a random system.

Where will you record who is responsible for which tasks?

We will use the same protocol as our meeting notes: an email thread and a GoogleDoc if necessary.

What will happen if someone does not follow through on a commitment (e.g. missing a deadline, not showing up to meetings)?

For deadlines, we will give gentle reminders and push the teammate more, especially if it is interfering with the overall progress. We will try to proceed with what we have, and expect that teammate to get his work done as soon as possible. For meetings, we will try include everyone's opinion but that obviously cannot be done if someone isn't there. The remaining two teammates will do their best at making a decision for the team if it needs to be done right away. If the problem becomes serious and cannot be internally resolved, we will talk to the TA and follow procedures for uneven work.

How will the work be reviewed?

In general, we will try to follow the following format: one person does the implementation, runs some basic tests and submits it to the team. A different person will write detailed tests, and report the results. We will convene as a team to review the code, and discuss what needs to be improved, if any, and give it back to the original writer of the code to fix up what needs to be fixed. So tests will be written by someone who did not write the original code, and the implementation will be reviewed by everyone sitting together looking at the code.

What happens if people have different opinions on the quality of the work?

We hope to avoid this issue as much as possible by agreeing beforehand on the design and specs. Then during code review, if someone has an issue on the quality, they can bring up an example that causes the code to fail, in which case the original person who wrote the code must fix the issue. If the issue is more about the style of coding, we will discuss as a team what is the best way to proceed, and talk to our TA if we cannot come to an agreement.

What will you do if one or more team members are not doing their share of the work?

See the last question in Part I. We will try to resolve things internally, then meet with a TA and finally submit the project with a note attached if all else fails.

How will you deal with different work habits of individual team members (e.g., some people like to get assignments done as early as possible; others like to work under the pressure of a deadline)?

We understand that different people have different work schedules, so we will try to set deadlines to accommodate them as much as possible. Other than meetings, when everyone should be at one place, team members should feel free to work on their own schedules.

IV. Decision Making

Do you need consensus (100% approval of all team members) before making a decision?

We would like 100% consensus on decisions, especially the decisions that have wide-reaching consequences for the overall program. If there is a 2/3 majority but the third team member feels that it is completely wrong, we will speak to the TA, but we hope it will not get to this point.

What will you do if one of you fixates on a particular idea?

Like above. If it is a good idea, then by all means we will pursue it. However, if the majority feels otherwise (too time-consuming, not enough flexibility, etc.) we will try to convince the remaining team member, and bring it to the TA as a last resort.