CSCI 3194 (Professional, Ethics, and Design Seminar), Spring 2004

Design Project

1 Project description

1.1 Overview

Most of you have had the experience of "collaborative programming" using a single computer — two or more people clustered around a machine, with the ability to do the following.

- Edit, compile, and run code (sharing a single copy of the source code), and view the results.
- Communicate with each other — verbally and by drawing pictures on a paper or a whiteboard.

This has many advantages in all phases of program design and implementation, including debugging.

Your mission for this course is to design an environment that supports this kind of collaborative programming among people who are *not* all clustered around a single machine — i.e., an environment for distributed interactive collaborative programming.

You have some flexibility in deciding exactly what functionality to provide; that's part of the design problem. In a real-world situation, you would have a "customer" whose needs you are trying to meet, and part of the design problem might be to turn a vague initial problem description into something more specific. For this class, I will play the role of customer, so part of the analysis phase of the project might be asking me questions to help you clarify requirements. Class time will be available for this purpose, or you may ask me in office hours or via e-mail (being sure to coordinate with other members of your group.) Given the nature of this project, it probably makes sense to also view yourselves and your classmates as potential customers/users and try to come up with a design you think will appeal to this group.

1.2 Constraints

Customers typically also impose constraints (how much a solution can cost, what hardware/software platforms it must run on, etc.). For this project, the following constraints apply.

- Your solution should be as cross-platform and portable as possible — e.g., no Windows-only or Linux-only solutions.
- Your solution should not require spending money — e.g., if you use existing products/programs, they must be public-domain / free.

1.3 Project phases

A project such as this one can be broken down into two phases, (1) analysis and (2) implementation design. In the first phase, you analyze the problem and come up with specific requirements that a solution/implementation should meet. During this phase you will probably want to research other programs/products that solve similar problems. In the second phase, you design an implementation that meets these requirements and produce a prototype implementation. Notice that the prototype implementation is mainly intended to show how your solution looks to users, so you do not have to completely implement parts that don't show.

2 Deliverables

The following table lists "deliverables" on which you will be graded. Items with a "Y" in the column labeled "Group?" are to be turned in by group leaders (and should represent the combined efforts of the members of the group); other items are to be turned in by individual students (and should represent individual efforts). Notice that the points add up to the 75 points specified in the syllabus for the project.

What	*Group?*	*Due*	*Points*
Midterm report	Y	March 3	15
Midterm group evaluation	N	March 3	5
Presentation	Y	April 21	15
Final report	Y	April 28	20
Prototype code	Y	April 28	15
Final group evaluation	N	April 28	5

2.1 Midterm report

By midterm, each group should have completed at least a preliminary analysis of the problem, including use cases. Your midterm report will describe this analysis and the use cases, including UML use-case diagrams if you choose.

(The point of asking for a midterm report is twofold: First, it provides an extra incentive to not procrastinate. Second, it will allow me to provide some feedback as to whether you are generally on track.)

2.2 Midterm group evaluation

Each student will also evaluate the performance/contribution of members of his/her group. I will use this information in determining grades. Forms will be provided.

2.3 Presentation

Each group will present its analysis of the problem, its proposed solution, and a prototype implementation.

2.4 Final report

Your final report should be an expanded version of your midterm report, incorporating any changes you have made and adding a section describing your implementation design and prototype.

2.5 Prototype code

You are not required to come up with a complete implementation of your design; this course is meant to be about design more than implementation. However, before starting a complete implementation it can be extremely helpful to develop a prototype that shows your "customer" what you have in mind, to be sure that what you plan to implement really meets his/her needs. Thus, you will develop a prototype implementation you can demonstrate.

2.6 Final group evaluation

See "Midterm group evaluation."