

CS 350 – Computer/Human Interaction

Fall 2006 – Final Group Project: A Chat System

Out: October 20

Mid-project Report Due: November 10

Presentation Due: November 29

Technical Report and Final Implementation due: December 12

The final group project consists of four parts: a mid-project report, an implementation, a presentation demonstrating the final product, and a technical report describing the project. To reiterate the project overview handout, this group project is worth 40% of your final course grade and is broken down as follows:

5%	Group project mid-project report
10%	Group project written report
10%	Group project presentation
15%	Group project implementation

Group Work

For this project, you will work in groups (two groups of 3 and two groups of 4--no exceptions). Teams are expected to meet regularly and team members are expected to contribute in substantially equal measures. Each team member will receive the same grade unless extenuating circumstances are brought to the attention of the instructor. To help identify and prevent such circumstances, a mid-project report and peer evaluation forms for both the mid-project report and the final report will be required.

The Programming Task

Your software company is creating a chat system for your target users. This will allow users to communicate with other users elsewhere in the world, almost instantaneously. A user types a message into the chat interface, and the message is sent to the other user(s), who can then reply.

Your task is to write the first fully working version of this piece of software. The features which should be supported by this version of the program are:

- Each user must choose a login name that must be entered into the program before the user is allowed to send messages.
- Each message that is sent will carry identification of who sent that message.
- When a user successfully sends a message, the message should appear in another part of the interface as feedback. I.e., the chat window.

- Error messages should occur if important information is omitted, or the user does the wrong thing. However, the interface also can prevent the wrong thing from happening in some cases.
- Your interface must support one-to-one communication (like AOL Instant Messenger) **and** many-to-many communication (a chat room). This may be done with the same interface or two different interfaces that are related.

To make your task simpler there are a few features which do not necessarily have to be implemented:

- The chat program must work, but it can use files to transfer the information. This is not a networks course.
- You do not need to check that login names are unique, or keep a database of login names. This is not a database course.
- There is no need to delete old messages, but you may choose to do so if you want. Or else you may choose to display only the last 10 messages, for instance. Do whatever you think makes the interface most usable.
- Messages do not have to be received automatically by the program. You can make the user press a Listen button, for instance. However, working out how to make messages appear instantly without the user explicitly doing anything would add to the usability of the interface.

You may write your system using any of the following tools:

- HTML and CGI using Perl scripts
- Visual Basic .NET
- Java using the Swing graphics library
- Any combination of the above (dual systems)

Some other notes:

- Everything you put in the interface must be implemented. No useless buttons, please. Unimplemented interface options will be considered negative features of the project.
- Your system will be assessed first on the user interface and the usability considerations, and secondly on the features. A system that provides few features but an excellent interface with corresponding excellent report may well score better than a system that has many features, but a poor interface and/or poor report.
- You'll have to demonstrate the program in KC-267 with at least two chatters on different machines (one projected from the instructor's station), so make sure it compiles and runs in that environment.

Report Logistics

Each group is to produce a 10-20 page technical report. It must include a cover sheet that states the names of all members of your group, describe briefly the contributions of each team member, and contain the following declaration signed and dated by all members of the group:

In accordance with the University of Evansville Honor Code, we attest that the project we are submitting is completely our own work, and that we have not received nor given any unauthorized aid on this project.

The report should consist of three sections: a requirement analysis, a design rationale, and a usability assessment.

Requirements Analysis-Choose the Target Users

You are to choose a specific set of target users who will use your program. Design your interface with those users in mind. The more carefully you think about the needs of that set of users, and the more your interface meets those needs, the better grade you will receive.

Here are some suggestions about sets of users you might choose:

- Primary-school students
- High-school students for remote learning
- Computer-illiterate people
- Doctors and patients in remote locations
- Journalists traveling the world
- Music bands and their fans
- Antarctic explorers and their sponsors
- Visually-impaired users

Other sets of users may be chosen, but they must be cleared with the instructor first.

Brainstorm to develop a root concept as described in Chapter 2. This will have a strong influence on the rest of the project, so think carefully about this. Develop a list of stakeholders and at least two hypothetical stakeholder descriptions that introduce users with different characteristics and personal contexts. Develop least two problem scenarios and accompanying problem claims analysis (at least one positive and one negative feature).

Design Rationale

Explore and summarize relevant metaphors and information technologies at all three levels - activity, information, and interaction. Trace the elaboration of **one** of the problem scenarios from above through

all three phases producing scenarios and accompanying claims analysis (at least one positive and one negative feature) as described in Chapter 3, 4, and 5. Sketches and screen-shots should accompany your scenarios as appropriate.

Usability Assessment

The usability assessment section should include:

- A usability specification (subtasks with performance and satisfaction parameters) related to the elaborated design scenarios from above. Look at the virtual science fair or other case studies for models. Remember the usability outcome levels (i.e., worst, planned, and best) for each subtask are based on your own experiences using the system or on use of related systems.
- A proposal for evaluation test of this specification. This should include a discussion of what measures you would collect, how you would collect them, and example forms such as user reaction surveys. See the case studies for examples.
- A brief discussion of the advantages and problems with using the tool(s) you chose for the project **from the user's point of view**. I.e., how did the tool help or hinder your ability to build the interface you designed.

References

There should be a list of reference to any resources (other than the textbook) used in completing the project.

Mid-project report

On November 10, a mid-project report must be submitted to the instructor by 4:30pm. This report should include the following items:

- Choice and description of target users
- At least 3 problem scenarios and accompanying problem claims
- Progress on developing activity, information, and interaction scenarios. I.e., how far has the group gotten in the SBD approach.
- Choice of implementation technology
- Progress on implementation. I.e., what parts of the project have been implemented, and what parts are left to be implemented.
- One peer evaluation form from each group member. These forms may be submitted to the instructor independent of the rest of the progress report. Individuals failing to submit a peer evaluation form will be assessed a 10% penalty on the mid-project report score.

Presentation Logistics

All groups are expected to be prepared to make a 15-20 minute presentation on November 29 to the entire class. However, some groups will present on December 1. The following topics should be covered in your presentation:

- "Background" of target users
- Discussion of the design decisions via scenarios
- Demonstration of the product
- Discussion of a usability specification and proposal of evaluation test

The presentation may be made by one or more persons in the group, but all group members must be present to answer questions regarding their project. Those absent without excuse on November 29 will be penalized 10% of the presentation score.

Final project submission

Final project submissions are due December 12 by 4:45pm. (Time of the scheduled final exam period for this class.)

HTML/CGI/Perl submissions must be installed on cserver under one of the group member's webspace to be graded. Please make sure your index page clearly indicates which link the instructor is to follow.

Entire VB projects should be zipped and arrangements made to transfer a copy to the instructor either via USB stick or network file transfer.

For Java projects, all source files should be either tarred or zipped and emailed as an attachment to the instructor.

Hand in the following items:

- Cover sheet as described above, instructions for where to find HTML/CGI/Perl projects or how to install Java or VB projects, and a list of any changes you have made to your interface since your presentation.
- A technical report as described above.
- Hardcopy printouts of your *well-commented* source code. (Not including Designer-generated code for VB projects.)
- One peer evaluation form from each group member. These forms may be submitted to the instructor independent of the rest of the final report. Individuals failing to submit a peer evaluation form will be assessed a 10% penalty on the report score.