

# EECS 22L: Instant Messaging App Alpha Release

Prepared by: Mina Moghadam, Mihnea Chirila, Hamid Nejatollahi, Bo Tsai,  
Prof. Quoc-Viet Dang and Prof. Rainer Dömer

February 28, 2018

The third deliverable in the chat project is the alpha version of the Instant Messaging program for the user, as well as the corresponding alpha version of the program source code and documentation. While certain shortcomings are expected for an alpha version (e.g. command line interface, account management is not fully working or hard coded, it sometimes crashes etc.), the deliverables should give a good impression of the state of the implementation and provide an early "preview" of what the final release will be.

For this submission, the client and server modules have the greatest weightage. A significant implementation of these two modules is expected for alpha release.

The credit will be awarded in the following way: 30% of the credit will be given for the packaging (including file hierarchy), the rest will be given on the content of each file and the performance of the program. For full credit (100%), your package should have the following minimum file hierarchy:

**User/Customer Package** (Chat\_Alpha.tar.gz)

- **README** // author, version, date, general instructions ...
- **COPYRIGHT** // author and copyright
- **INSTALL** // installation instructions
- **bin/** // binary directory
  - *chat* // the executable program
- **doc/** // documentation directory
  - *Chat\_UserManual.pdf*

**Source Code Package** (Chat\_Alpha\_src.tar.gz)

- **README** // author, version, date, general instructions, ...
- **COPYRIGHT** // author and copyright
- **INSTALL** // installation instructions, just let the user type `make` is enough!
- **Makefile** // top-level, tool-specific Makefile, should support: `make`, `make test`, `make test-gui`, `make test-comm`, `make clean`, `make tar` (compress the source code package)
- **bin/** // binary directory, when the user types `make`, the executable or its symbolic link should be generated here, as well as logs
- **doc/** // documentation directory
  - *Chat\_UserManual.pdf*

- *Chat\_SoftwareSpec.pdf*
- **src/** // source directory, with all source code, test files and Makefile
  - *Makefile* // lower-level Makefile for sources
  - *chat\_server.c* // maybe other modules and header files: *chat\_client.c*, *config.c*, *types.h*
  - *test\_comm.c* // check the communication between client and server for different types of requests
  - *etc.*

Notes:

**README** example: This is a alpha version chat software, to install, please type "make", for more detailed instructions, refer to the user manual.

**INSTALL** example: Type `tar -xvzf Chat_Alpha_src.tar.gz`, then `cd Chat_Alpha_src`, then `make`.

The above is fine. Keep **README, INSTALL, COPYRIGHT** brief. Guide the user to refer to the user manual or the developers to software spec for more details.

You can add other folders to your package if necessary, such as GUI libraries, resources (images/fonts/maps), etc.