EECS 22L: Instant Messaging App Beta Release

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

March 7, 2018

The third deliverable in the chat project is the beta version of the Instant Messaging program for the user, as well as the corresponding beta version of the program source code and documentation. While certain shortcomings are expected for a beta version (e.g. account management is not fully working or it is hard coded, it sometimes crashes, the extra features are not implemented etc.), the deliverables should implement the **core** functionalities of your application.

Thus, for this submission, the minimum required features are: GUI interface, basic account and contact list management and interface, and fully working client to client communication.

NOTE: 10 points will be deducted from the current assignment for all the teams that have not submitted their team photos and information by this assignment's deadline.

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_Beta.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_Beta_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 beta version chat software, to install, please type ”make”, for more detailed instructions, refer to the user manual.
INSTALL example: Type `tar -xvzf Chat_Beta_src.tar.gz`, then `cd Chat_Beta_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.