**Collaborative Latex Document** <https://www.overleaf.com/4292655hhfmgx#/12728627/>

What about user roles? Should we include this as a non-functional (i think) requirement? i.e., admin, project owner, user where an admin oversees all activity on the system and a project owner is the creator of a project who has administrative project privileges over users of that respective project. ~Austin

Requirements:

1. Capable of supporting editing, compilation, and execution of Java programs. Programs to be composed as projects and support multiple source code files across multiple directories within a shared top-level directory.
2. Ability to import/export AND/OR function on projects/source code stored in the ordinary local file system.
3. Syntax coloring; visual indication to see who coded what.
4. Multi-user, up to >= 32 users can share an IDE session.
5. Shared sessions should allow people to move around their view (read-only, at least) independently, and to quickly jump to where other users are looking.
6. Users can text chat to individuals, transient shared session members, persistent project member lists, and all logged in users. User profiles should include persistent data including project ownerships and memberships, friends, e-mail, profile image.
   1. For the user profiles I think it would be good to have achievements, like certain number of lines of code have been written, and have been in a certain number of reviews, etc.
7. Users should have easy access (background?) awareness information of other users, especially friends and members of shared projects.
8. Resource defense strategy that includes not subjecting any sQuire server to becoming unresponsive due to a runaway program, and not allowing any sQuire server to give up shell access via an executing program in sQuire.
9. Indication of who coded what could be background color, underline color, sidebar color, shade/fill pattern, or icon/avatar. Since multiple people might edit a given line over its history, support for past history or anyhow multiple persons in this indication is strongly recommended.
10. Teams should decide whether write access to shared editing should be turn-based or simultaneous
11. Teams should decide if voice or video is essential. Voice, if supported, might be restricted as to number of listeners and/or number of simultaneous transmitters. Video, if supported, might be restricted as to number of viewers and/or number of simultaneous.
12. Ability to up or down vote comments on code.
13. Chat widget that displays people who are currently online, as well as other people working on other files in the project.