

Church of CVN: Demo 1

bl2719, mwg2126, nr2645, rtt2114

1. The date and time at which you already completed this demo, and briefly describe any challenges that arose during the demo.

Our first demo occurred on 11/1/18 at 8:30 pm, EST. We demonstrated the following (challenges *italicized*):

- code-base: running a server and demonstrating how multiple clients may connect to the server and therefore communicate with each other by sending messages back and forth. Any number of messages may be sent and are received by all connected clients.
- Junit and integration tests: our current test suite looks at common errors occurring during socket setup (port ranges, blocked ports or hosts), TCP setup (port ranges and the creation of multiple hosts), and message passing (long messages, short messages, UTF-8 style issues like emoji) between clients. Each of these functions is associated with a class in the test folder so that our IDE (i.e., IntelliJ) and maven may run the tests easily.
- travisCI integration (post-commit): our code-base is linked to GitHub for version control and travisCI for continuous integration. Each time a file changes in either the main (used for demo) or develop (used for internal development) branches, the travisCI server runs a full maven build—clean, validate, compile, verify—resulting in either a build success or failure. This information is then linked to our GitHub readMe.
- precommit was demonstrated through the PRECOMMIT bash script found on the repository's main file structure. The file catches style errors prior to upload to git (*demonstrating our pre-commit on a new clone of the git required re-copying our template into its correct path and took an extra minute or two*).
- Static analysis was also demonstrated in the .html file located in /target/site/pmd.html. In the target file, maven surefire reports of our tests may also be found.

2. The specific user stories and conditions of satisfaction that were demonstrated, with an explanation of any changes since your revised proposal.

We fulfilled user story 1 (from the revised project proposal). All of the conditions of satisfaction were met. We understood the main focus of this first sprint to be ensuring best-practice software principles, and therefore pushed our effort toward setting up maven, travisCI, github, trello, PMD, and google's style guideline. Please note, on the original project proposal we included user accounts as a condition of satisfaction under the first user story. Thinking later that this was a larger-scope feature, we moved the attribute into a singular user story for the revised version, and therefore did not implement this (old) condition of satisfaction. User accounts makes much more sense when paired with a history-type function, which should be rolling out in the next sprint.

3. A brief discussion of your CI mechanisms, including which technology you used.

Our GitHub is linked to TravisCI which uses a .yml file to run maven's clean, validate, compile, and verify functions. Additionally, our POM file from maven has a google-style-checker and PMD analysis built in to the compile phase.

4. A link to the github repository where your entire codebase resides. Tag the revisions that were shown in the demo.

<https://github.com/LooseScruz/ada> (Revision tagged as Demo 1)