Functionality Testing

1. Unit Testing (ServerConnectionTest.java)

We created unit tests for each session method to ensure that server responses returned are as expected.

- Account Creation
- Change master password
- Account deletion
- Credential addition
- Credential removal
- Credential retrieval
- Credential updates
- Account authentication

2. Testing on Terminal

We tested each shell command with various cases.

Comands: login, register, add, get, creds, delete, change, exit, logout, unregister

- login
    - Empty username, empty password, non-existent username, wrong password
    - Username that contains “../”, “..\” or escape characters (e.g. “\t”)
    - Login again without logging out
- register
    - Register an account that already exists
    - Register multiple accounts in a row
- add
    - Credentials that already exists
- get all, get username, creds username, creds all
    - No credential exists
- exit, logout
    - Not logged in
- unregister
    - Not logged in
    - Incorrect master password (prompts for confirmation)
- delete
    - No credential exists
- change
    - No credential exists

We programmed most of the project after discussing a general design for the client and server modules and agreeing on how we would send communications between the two. Most of the code was written in a group where we could ask questions or bounce ideas off one another. Our test suite covers the full set of server functionality; the client code is all tested by hand.

3. Security Auditing

All portions of our code were audited for security by someone other than the one who authored the code. These audits found several vulnerabilities, including the ciphertext switching integrity concern and a vulnerability where a username containing “..” could be used to access the server-side stored credential files of a different user.