

0.1 General

Candidates will be provided a branch built out with the general structure of the exam with a stripped binary of the reference implementation's server.

```
/
├── bin
│   └── example_server
├── build
├── include
├── test
│   ├── server
│   └── client
└── src
```

They can use the provided server binary implementation to test against their client, and then build their server to be able to work with that client. Their implementations should use the `/test/server` directory as their server's home directory and `/test/client` for their client's home directory. Once the exam is complete the code reviewers can run

the client's server and test their client, as well as the provided python grading script against it to test the general functionality of the provided products.

0.2 Requirements to check:

Authentication:

- Cannot authenticate with invalid username/password
- Can authenticate with valid username/password

Permissions/Users:

- New users can be successfully created
- Duplicate users cannot be created
- Each additional user that successfully authenticates gets a unique Session ID
- New users can only be created at the same permission level of the creating user or below
- Existing users can be successfully deleted
- Non-existing users cannot be successfully deleted
- Only users with `CREATE_ADMIN` privileges can delete users

Files:

- Users with `CREATE_R` privileges can successfully perform `get_file` and `list_remote_directory`
- Users with `CREATE_R` privileges can NOT successfully perform `put_file`, `rename_file`, `make_remote_directory`
- Users with `CREATE_RW` privileges can successfully perform `CREATE_R` and `CREATE_ADMIN` functions and `put_file`, `rename_file`, `make_remote_directory`
- Files are protected while being accessed by a client

Additional:

- Expired Session IDs cannot be used to perform privileged server actions
- Can support multiple users at once
- Followed instructions for coding standards