[redacted]

Project 4 Report

Architectural Design:

- Mutual authentication is achieved when the user first logins on the client. The client would create a statement that included which client and which user was signing in. It then would sign it with the user's private key and then send over the signed statement, along with the original statement for the server to check. The server would check using the user's public key and if the signed statement content is similar to the statement, then the user is authenticated.
- The main library used for cryptographic functions was pycryptodome, which included all the functions needed for signing, encrypting, and decrypting. Along with those, a random function was used to create a random key for the encryption method of the file. SHA 256 was used to hash it, and RSA was used for the public and private keys. Pkcs1_15 was the signature function used to sign the documents and sign the statement for authentication.
- The user's id and session token are stored in the database. The database's unique identifier was the document id, then it had fields for the document owner, and lists for each access right that was granted to other users, it would also store the expiration time for each user who was granted an access right.

Implementation Details:

- Started with the login method and then started working my way through each. As I worked through each, I would implement the foundational functionality first and then go back later for the cases that weren't accounted for earlier. I used a lot of conditionals to take care of the edge cases especially when the granting access to different users came into play. Then as I did testing on the functionality, for the occasional error I would go back and fix it.
- One part that wasn't implemented was on the logout, the documents in the checkout folder had to be check back in. Other than that, all of the features should have been implemented and tested.
- I made the assumption that the user and client keys would be generated manually.