# DESIGN

## 1. Authentication

- **a) Overview:**
    - The system will verify the identity of each user attempting to access our system by implementing a login system. This will take the form of usernames and (salted and hashed) passwords stored on a local csv file.

.

- **b) Intended implementation:**
    - Complete

- **c) Current implementation:**
    - Generating passwords:
        - The *secrets* module in python is used to generate a unique salt for each password, which is concatenated with the password to create a salted password
        - This salted password is then hashed using the SHA-256 hashing algorithm
    - Storing passwords
        - Passwords are stored in their corresponding record within the userinfo.csv file, in the format *username, userID, (salted and hashed) password, salt*
    - Authenticating passwords
        - When a user inputs a password, a salt is generated with the secrets module and concatenated with the user input to create a salted user input
        - The salted user input is then hashed using the SHA-256 hashing algorithm
        - The (salted and hashed) user input is compared with the (salted and hashed) stored password
        - If both values match, the user is authenticated and can access the system
    - Password policies
        - When creating a password, user passwords are required to adhere to a set of length and complexity requirements (e.g. minimum length, number of special characters, numbers, capital letters, etc.)
    - Verifying user email
        - When a user creates an account, they are prompted to enter their username
        - A 6-digit PIN is sent to their email and the user is prompted by the system to enter their PIN
        - If the entered PIN and the sent PIN match the email is verified and the user record is created
    - Deleting records
        - Prompt user for password before deleting record by username
    - Password recovery

- If a user forgets their password, they can enter a PIN automatically sent to their registered email to reset their password

## 2. Authorization

- **a) Overview**
  - The system will determine and enforce the actions allowed for an authenticated user. It ensures users can only perform actions they're permitted to, such as viewing or uploading photos.

- **b) Intended implementation:**
  - File uploads:
    - We will check the file extension of each uploaded file and return an exception if the file is not a png or jpg
  - Password recovery:
    - Users will be able to recover their passwords via a recovery email link

- **c) Current implementation:**
  - Confidentiality tiers:
    - Tiers of confidentiality will be implemented to ensure that normal users do not have the same action privileges as administrators
    - Within each user record in userinfo.csv, a new column will be created titled "access level", defining the actions available to the user depending on their access level
    - The access level will be an integer representation of its sensitivity (e.g. 1: top secret, 2: confidential, etc.)
  - File uploads:
    - Users can currently upload any file they wish to the network
    - For testing, we have only been sending jpg files

## 3. Audit

- **a) Overview:**
  - The system will maintain detailed logs of system access and activities, such as login attempts, file uploads, and downloads.

- **b) Intended implementation:**
  - Tracking user logins
    - Timestamps and user IDs will be recorded each time a user logs on/logs off the system (both successful and unsuccessful attempts)
  - Tracking user actions on network
    - User activity will be tracked by labeling user actions with the following:
      - User ID
      - User action (e.g. send, receive, change password, etc.)

        - Action ID
            - This will also assist in preventing replay attacks, in that an attacker cannot simply replay a message without the user seeing that the action ID is repeated/not in consecutive order with previous action IDs
    - Storing audit information
        - Audit information will be encrypted and stored on a separate database
        - Encryption will be used to transfer and store audit data, as in 1b
- **c) Current implementation:**
    - Tracking user logins
        - Timestamps and user IDs will be recorded each time a user logs on/logs off the system (both successful and unsuccessful attempts)
    - Tracks when admin/superadmins create user/admins
    - Only admin and superadmin can view the audit logs

## 4. Confidentiality

- **a) Overview:**
    - The system will ensure that only the administrator and the user themselves should be able to access that user's login data and audit logs
    - The system will ensure that the file's contents can only be read by the recipient.

- **b) Intended implementation:**
    - Files stored on disk that have private information should be encrypted.

- **c) Current implementation:**
    - Ensuring user privacy:
        - User cannot view, edit, delete, or replay messages sent across the network between other users (i.e. Mallory in A4)
    - Every chunk of data is encrypted using TLS when being sent over the network.

## 5. Integrity

- **a) Overview:**
    - The system will employ integrity checks to ensure that a file is not corrupted in transit or replaced with harmful content

- **b) Current implementation**
    - TLS
        - Message Integrity Check: This method utilizes a MAC from the data and a secret key to confirm authenticity before transmission.
        - Encryption: Encrypts both the data and the MAC for secure transmission.
        - Verification at Receiver: Decrypts the data upon receipt and recalculates the MAC to ensure no changes occurred during transit.