**Networked File Sharing Server and Client**

1. **High-Level Design (HLD)**

**1. Objectives and Requirements:**

* **Objective:** Create a networked file-sharing application with server-client architecture, enabling file transfer, listing, uploading, authentication, and encryption.
* **Requirements:**
  + **Functional:** File listing, file transfer, file upload, authentication, encryption.
  + **Non-Functional:** Performance, security, reliability, scalability.

**2. Architecture:**

* **Client-Server Model:**
  + **Client:** Sends requests to the server for file operations.
  + **Server:** Responds to client requests and performs file operations.

**3. Components:**

* **Client Component:**
  + User Interface: Command-line interface for interaction.
  + Network Interface: Establish connection with the server.
  + File Handling: Request and receive files from the server, send files to the server.
  + Security: Handle authentication and encryption.
* **Server Component:**
  + Network Interface: Accept connections from clients.
  + Authentication: Validate client credentials.
  + File Operations: List, send, and receive files.
  + Security: Handle encryption and decryption.

**4. Data Flow:**

* **Client to Server:**
  + Authentication data.
  + File operation requests (list, get, upload).
* **Server to Client:**
  + Authentication response.
  + List of files.
  + File data.

**5. Technology Stack:**

* **Programming Language:** C++
* **Libraries:** Standard C++ libraries for file I/O, networking (sockets), and encryption.

1. **Low-Level Design (LLD)**

**1. Detailed Class/Module Design:**

**Client:**

* **ClientSocket:**
  + Methods: connectToServer(), sendData(), receiveData().
* **ClientAuth:**
  + Methods: sendPassword(), receiveAuthResponse().
* **FileHandler:**
  + Methods: listFiles(), downloadFile(), uploadFile().
* **Encryption:**
  + Methods: encryptData(), decryptData().

**Server:**

* **ServerSocket:**
  + Methods: bindSocket(), listenForConnections(), acceptClient().
* **ServerAuth:**
  + Methods: validatePassword().
* **FileHandler:**
  + Methods: sendFileList(), sendFile(), receiveFile().
* **Encryption:**
  + Methods: encryptData(), decryptData().

**2. Sequence Diagrams:**

* **Client Authentication:**
  1. Client sends password to Server.
  2. Server validates password.
  3. Server sends authentication response.
* **File Listing:**
  1. Client requests file list.
  2. Server retrieves file list.
  3. Server sends file list to Client.
* **File Download:**
  1. Client requests file download.
  2. Server reads file, encrypts data.
  3. Server sends encrypted data to Client.
  4. Client receives and decrypts data.
* **File Upload:**
  1. Client requests file upload.
  2. Client reads file, encrypts data.
  3. Client sends encrypted data to Server.
  4. Server receives and decrypts data.