project name :- Network File Sharing Server And Client

**Team members :- J . Ameesha**

**B. Supriya**

**M. Haritha**

**High-Level Design**

**Components**

**Server:**

**Responsibilities:**

Listen for incoming client connections.

Receive file requests from clients.

Send requested files to clients.

Handle multiple clients (optional for scalability).

**Client:**

**Responsibilities:**

Connect to the server.

Send file requests to the server.

Receive and save files from the server.

**Communication Protocol**

**Protocol :-**

**Request:** Client sends a request message containing the file name.

**Response:** Server responds by sending the file in chunks.

**Error Handling:** If the file is not found, the server sends an error message.

Workflow

**Client Requests File:**

The client sends a file request message (file name) to the server.

Server Handles Request:

The server receives the request, opens the requested file, and sends it to the client in chunks.

**Client Receives File:**

The client receives the file data in chunks and writes it to a file on disk.

Low-Level Design

Server :-

**Socket Creation:**

Create a socket using socket.socket().

**Bind to Port:**

Bind the socket to a specific port and IP address using socket.bind().

**Listen for Connections:**

Set the socket to listen for incoming connections using socket.listen().

**Accept Connection:**

Accept a client connection using socket.accept().

**Receive File Request:**

Receive the file name request from the client using socket.recv().

**Open and Send File:**

Open the requested file and send it in chunks using file.read() and socket.sendall().

**Error Handling:**

If the file is not found, send an error message to the client.

**Close Connections:**

Close the file and client connection using file.close() and socket.close().

**Client :-**

**1. Socket Creation:**

Create a socket using socket.socket().

**2. Connect to Server:**

Connect the socket to the server using socket.connect().

**3. Send File Request:**

Send the file name request to the server using socket.sendall().

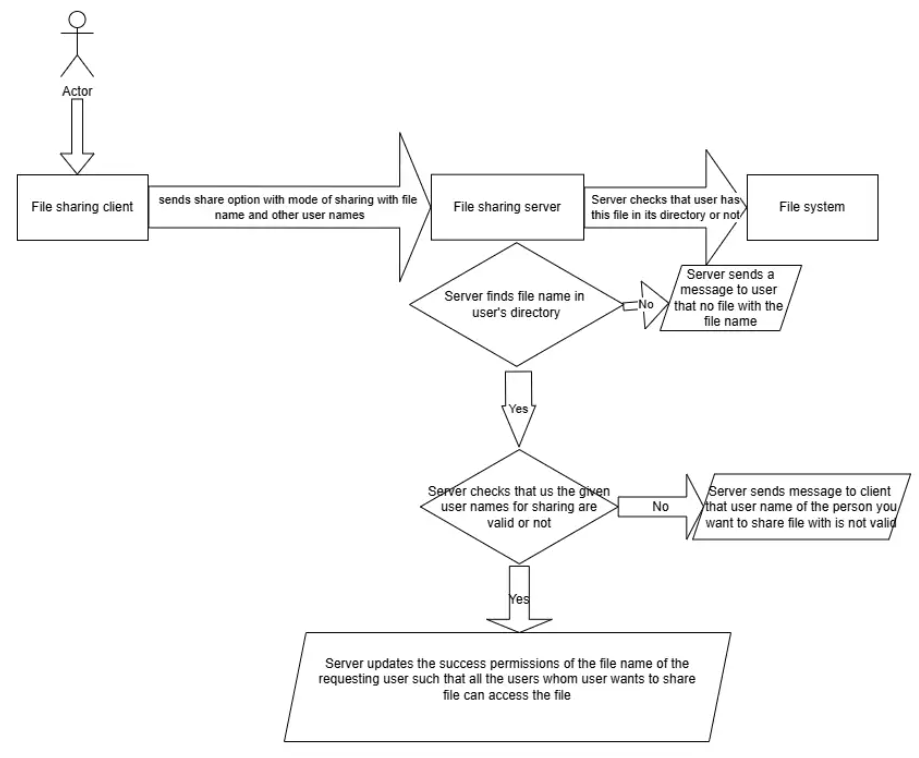
**4. Receive and Save File:**

Receive the file data in chunks and write it to a file on disk using file.write() and socket.recv().

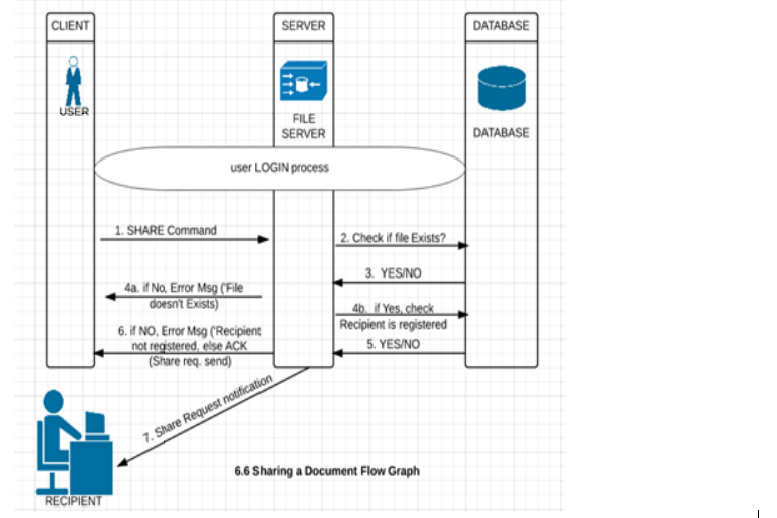
**5. Close Connection:**

Close the socket connection using socket.close().

**FLOWCHART :-**

****

**Sharing a Document Flow Graph :-**

****