Use Case: Client Writes File to Server

Summary: User starts up the client, selects "Write" operation, inputs the name of the file to be written, selects mode, Write operation begins, Client notifies user when write is complete.

Actors:
    Primary Actor: Client
    Secondary Actor: Server

Precondition: The Server is already up and running, and ready to read from the Client.

Main Sequence:

    1. The user starts up the Client.
    2. A dialog box will appear, prompting the user to input the desired file transfer operation. The user inputs "Write". (Include use case **Client Inputs Invalid Operation**)
    3. The user is then prompted to input the name of the file to be written.
    4. The user is then prompted to input the desired mode.
    5. Once the Client receives all the required information from the user, the write operation begins.
    6. Once the operation is complete, the Client will notify the user that the write operation has successfully completed, then the Client will terminate.

Non Functional Requirements:

    a) Usability Requirement:
        The Client should tell the user when the write operation is complete.

Postcondition: The file has been successfully sent to the Server.

Use Case: Client Reads File From Server

Summary: User starts up the client, selects "Read" operation, inputs the name of the file to be read, selects mode, Read operation begins, Client notifies user when read is complete.

Actors:
    Primary Actor: Client
    Secondary Actor: Server

Precondition: The Server is already up and running, and is ready to write to the Client.

Main Sequence:
    1. The user starts up the Client.
    2. A dialog box will appear, prompting the user to input the desired file transfer operation. The user inputs "Read". (Include use case **Client Inputs Invalid Operation**)
    3. The user is then prompted to input the name of the file to be read.
    4. The user is then prompted to input the desired mode.
    5. Once the Client receives all the required information from the user, the read operation begins.
    6. Once the operation is complete, the Client will notify the user that the read operation has successfully completed, then the Client will terminate.

Non Functional Requirements:

    a) Usability Requirement:
        The Client should tell the user when the write operation is complete.

Postcondition: The file has been successfully received by the Client.

Use Case: Client Inputs Invalid Operation

Summary: User inputs wrong invalid operation during setup, the program aborts.

Actors:
      Primary Actor: User
      Secondary Actor: Client

Precondition: The user has input an invalid operation.

Main Sequence:
      1. The user inputs an invalid operation that is not recognized by the system
      (not "Read" or "Write").
      2. The Client notifies the user of that error.

Non Functional Requirements:

      a) Usability Requirement:
         The Client should notify the user if the chosen operation is incorrect, and
         terminate.

Postcondition: The Client aborts.

Use Case: Server Operator Initializes Server

Summary: Server operator starts up the Server, Server listens for any incoming read/write requests.

Actors:
    Primary Actor: Server

Precondition: The Server is not running.

Main Sequence:
    1. The server operator launches the server.
    2. The Server starts up, and starts listening for any incoming Client read/write requests on port 69.

Non Functional Requirements:

    a) Interface Requirement:
       The Server should utilise port 69, as per TFTP specifications.

Postcondition: The Server is now ready handle read/write requests from Clients.

Use Case: Server Operator Requests Server Shutdown.

Summary: Server operator requests Server to shutdown, Server completes current requests and aborts.

Actors:
      Primary Actor: Server

Precondition: The Server is running and listening for Client requests.

Main Sequence:
      1. The server operator requests the Server to shutdown by typing "quit" into the dialog box running in the background.
      2. The Server completes any transfers currently in progress.
      3. The Server stops accepting requests, and shuts down.

Non Functional Requirements:

      a) Operation Requirement:
          The server operator should be able to shutdown the Server at any time.
      b) Interface Requirement:
          The Server should complete all transfers in progress before shutting down.

Postcondition: The Server has shut down and is not accepting any requests from Clients.