**COSC 4333 Group 6 Project**

**11/29/2021**

**David Williams**

**Kristy Stallings**

**Sebastian Salois**

**Introduction**

The objective of system is a multithreaded FTP server that enables client-server communication through message-oriented structure. FTP server will handle the files and multiple requests coming in from the client. The server will allow four operations: uploading a file, download a file, deleting a file, and renaming a file. Also, it will accept multiple concurrent operations for the multithreaded FTP server. The client is the end user submitting requests for files to submit or receive from the server.

**Use Cases and Diagrams**

**List of use cases for the system:**

* Client can upload a file to the server
* Client can download a file from the server
* Client can delete a file in the server
* Client can rename a file in the server

**Basic course of events**

|  |  |
| --- | --- |
| **Client** | **Server** |
| Client accesses the server | Server displays choices for operation |
| Client choice is selected | Server request file name for operation |
| Client input is completed | Server receives command, validates information, and executes operation |
| Client receives response from server | Server waits for next command |

**Implementation**

The program was written in C language, and it runs on a UNIX system. The program is compiled by running the command gcc -o projectName projectName.c -pthread, and the program was tested running on a UNIX system with the command ./projectName [port]. Sequence diagram to show the client interaction with the server to complete operations for the FTP server.

Diagram

Description automatically generated

The FTP server handles multiple concurrent operations by creating a thread for the client, and assigning a socket for the execution to wait for the client connection over the TCP network. Then, it will supply a message to the client after the connection is received. After, the client submits the operations to be completed the current connection is closed, and it goes back to wait for the next connection.