DESIGN DOCUMENT

Psuedocode for SERVER.cpp:

```
int main()
{

//First it checks the command line arguments

if(# of args = 1)
      then we return error
else if(# of args = 2)
      set portnum to 8080 by default
      parse hostname and set it as the address
else if(# of args = 3)
      set portnum to the one given in the arguments
      parse hostname and set it to address
else if( # of args > 3)
      return error


//Then we create the file socket descriptor

if not created succesfully(meaning socket() returns 0) then we return an
error

//Then since the server was created succesfully we enter a while loop so
the server remains on
while(1)
{
      //First we parse in the first token of the header which contains
either PUT or GET

      if(request is a PUT)
      {
                  //Then we parse in the filename and the content length
from the header
                  //check if filename meets requirements
                  for( i =0; i<strlen(filename); i++)
                  {
                              if(a character in the filename does not meet
the requirements)
                              {
                                    send a 400 Bad request
                                    close the socket
                                    break from the loop;
                                    set fileval = 1// this is a flag so
that when we exit out the loop we can continue in the while loop
                              }
                  }
                  if(fileval == 1)
                  {
                              continue;
                              //that way the rest of the code in the while
loop does not get executed and the server does not crash
```

```
                }
                //Then we check the length of filename
                else if( filename is not 27 chars long)
                {
                            send a 400 Bad Request header
                            close socket
                            continue;//so that server doesnt execute the
rest of code
                }
                //Then we open the file and get file descriptor
                fd = open()

                if(file was created/opened succesfully)
                {
                            read in the data from the client
                            write to the file
                            close the file
                }
                else
                {
                            if(it was a permission error)
                                  return a 403 forbidden

                }

      }
      else if( request is GET)
      {
                //First, check the filename requirements
                //This part of the code is the same as the first part in
PUT

                //Then, open the file so we can read from it
                fd = open();

                if fd > 0 (succesful)
                {
                            if(fstat() != -1) if fstat is -1 then there
was an error
                            {
                                  char * bufferhead = calloc the size of
the file
                                  char * contentlength = will hold the
string "Content-Length: <cont length>\r\n\r\n"

                                  send the headers in order so that
client can read the file data


                            }
                            else
                            {
                                  get the errno string so we can see what
the error was
```

```
                                        if(File not found error)
                                        {
                                                send a 404 not found messae
                                                close socket
                                        }
                                        else//it was a permission error
                                        {
                                                send a 403 forbidden message
                                                close socket
                                        }
                                }

                        }
        }
        else//wasnt a get or put request
        {
                        return a 500 Internal Server Error
        }
}
return 0;

}
```