

# ASSIGNMENT ONE

## Socket programming project (Networks)

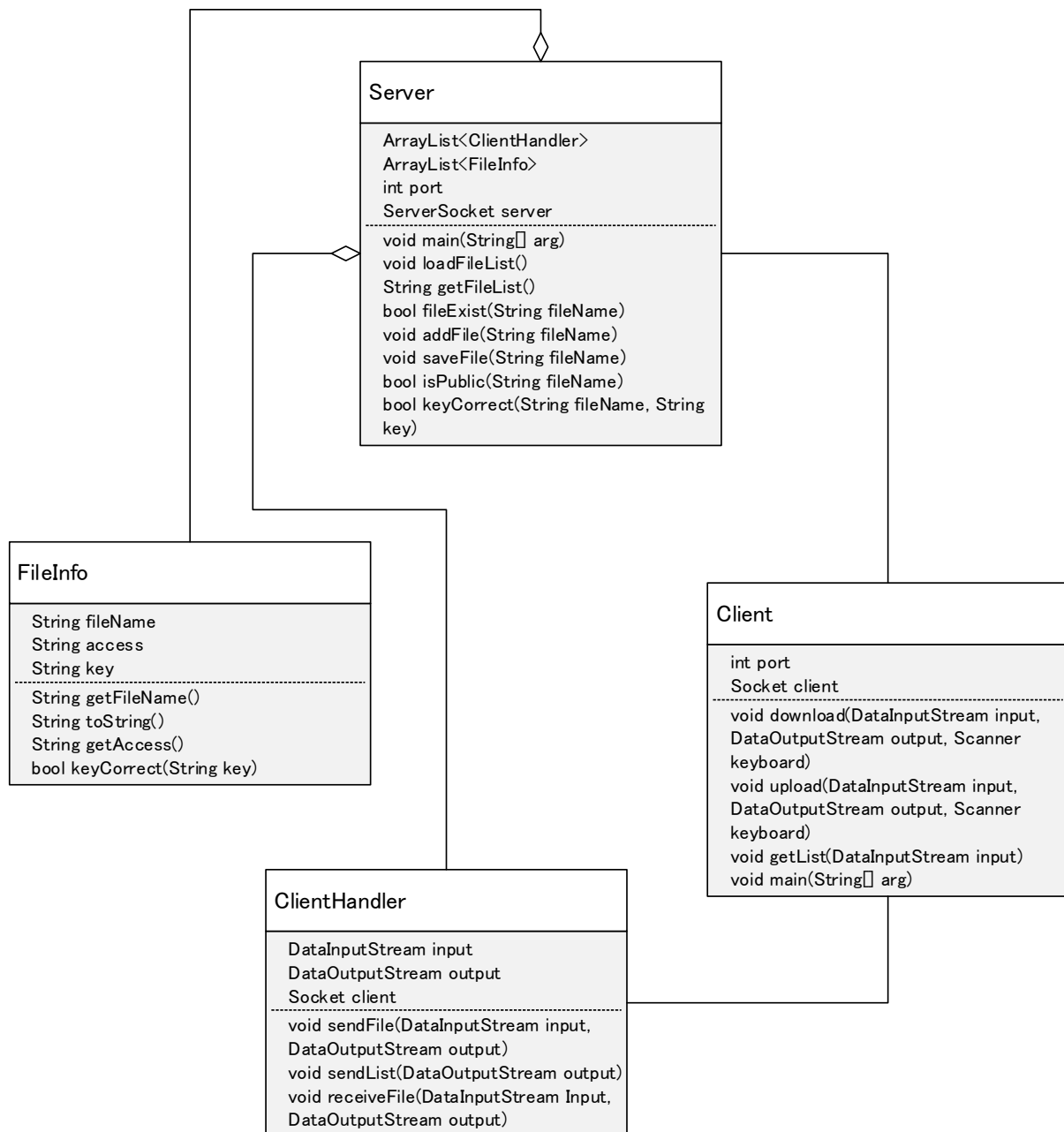
### Introduction

The client-server model is a distributed communication framework of network processes among service requestors and providers. Service providers are called servers and service requestors are called clients. A server provides services to clients that exchange information with it. This was the first model used to develop networked applications.

In this assignment we had to design and implement a Client-Server file sharing application. Our application performs the basic functions of a Client-Server file sharing application.

## Application Description

### Class Diagram



## Features

### In the Client class:

The class file has three main features:

#### ❑ Upload

This function has 2 inputs, filename and access, both of which are specified by the user. The Client accepts input from the user. It creates input and output streams, which reads the file and sends the file to a specific destination, in this case it would be the Server. The purpose of the upload function is to allow the user to save files onto the server.

#### ❑ Download

The application allows users to upload files onto the server, it will only be appropriate to allow them to download files from the server as well. The download function has the purpose of retrieving files from the Server, if the file is not present an error message is sent. This function accepts output from the Server which contains the details and file content and stores it in a location accessible to Client.

#### ❑ Get List

With the `getList` function a user is able to request a list of available files on the server.

#### In the Server class:

The Server class creates a `ClientHandler` object. This object is responsible for communicating with a particular client.

#### In the ClientHandler class:

This class represents the Server in the Client-Server model, for each individual client which communicates with the server. Main features includes:

#### ❑ RecieveFile

When a client uploads a file it needs to be saved in the Server. This function receives the file from the Client and saves it in a destination where all Server files are stored.

#### ❑ SendFile

When a client requests to download a file from the server there has to be a method in place to send the file. This method receives the filename, checks to see if the file is in the server and send it to Client. If it is not in the server, it displays a 404 file not found error message.

#### ❑ GetFileList

This function sends the client a list of files available on the server.

#### Additional features

The program allows uses to specific the file they are going to upload, whether they want it to be public(can be downloaded by anyone) or private(only clients who can provide the correct key will be able to download the file).

If the client chooses the file to be private, the client should provide a key to make sure only clients who can provide the key will be able to download the file

## Protocol Design and Specification

### Protocol Specification

Client	ClientHandler
MESSAGE: UPLOAD	
Prompts the user to input the name of the file they want to upload. Checks that the file path entered by the user is correct and that the file does exist on the user's machine. Catches <code>FileNotFoundException</code> and prompts the user to re-enter a valid file path until a file is found. Then prompts the user to specify the access protocol for the file they wish to upload (either 'public' or 'private'). Makes a call to the <code>upload()</code>	Makes a call to its <code>receiveFile()</code> method. Returns "File Received" message to the user once the file is written to the server

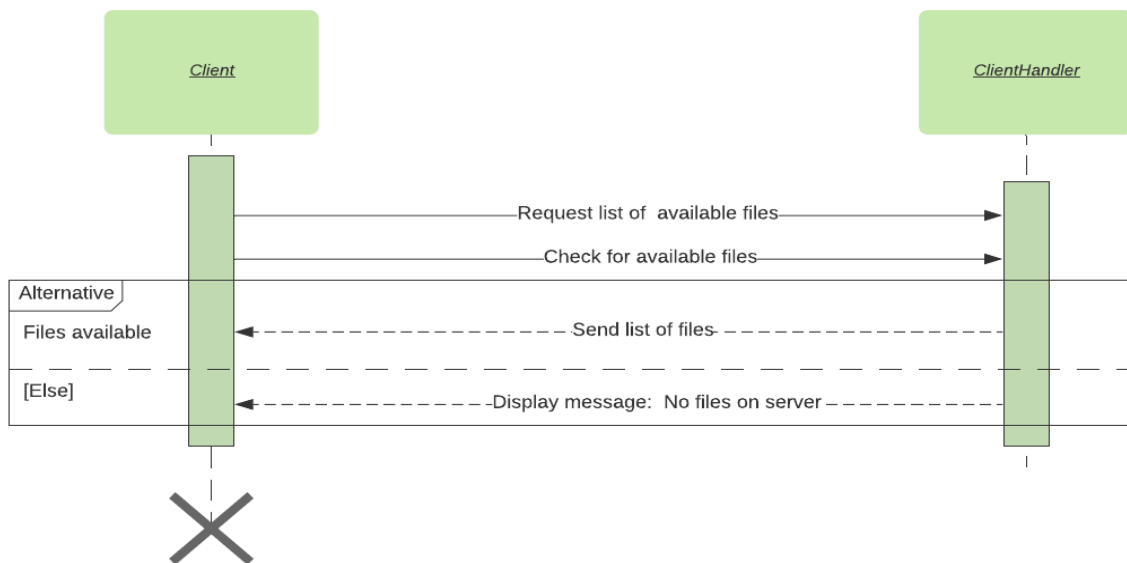
method passing in the file name and access protocol as two String parameters	
MESSAGE: DOWNLOAD	
Prompts the user to enter the name of the file they want to download from the server. Passes that filename into the download() method as a String parameter	First checks if the file exists and if it does then it makes a call to its sendFile() method which sends the file back to the client. A confirmation message is displayed confirming that the file was successfully sent to the client as well as a message telling the user when the transfer is complete
MESSAGE: GETLIST	
Returns a list of files on the server in the form of a String.	Makes a call to its sendList() method which returns a String back to the client

### Protocol Constraints

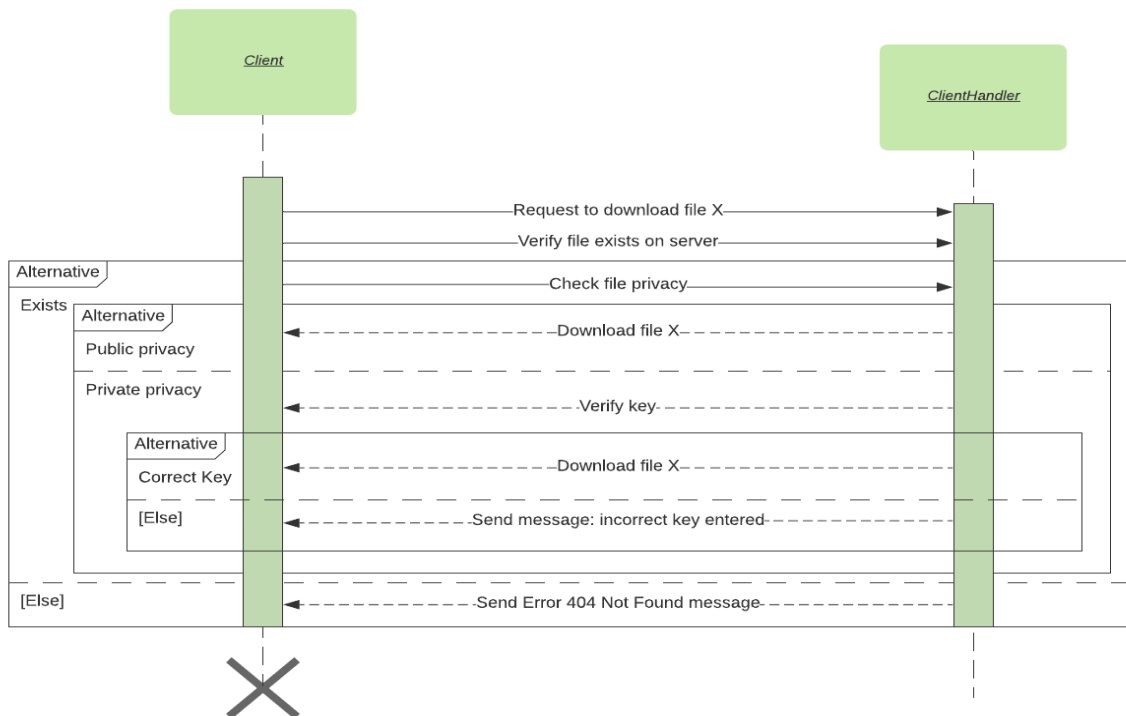
Client can choose the file they upload to be private or public. If private, they should provide a key, and in that case, no one without the key can download the file.

## Sequence Diagram

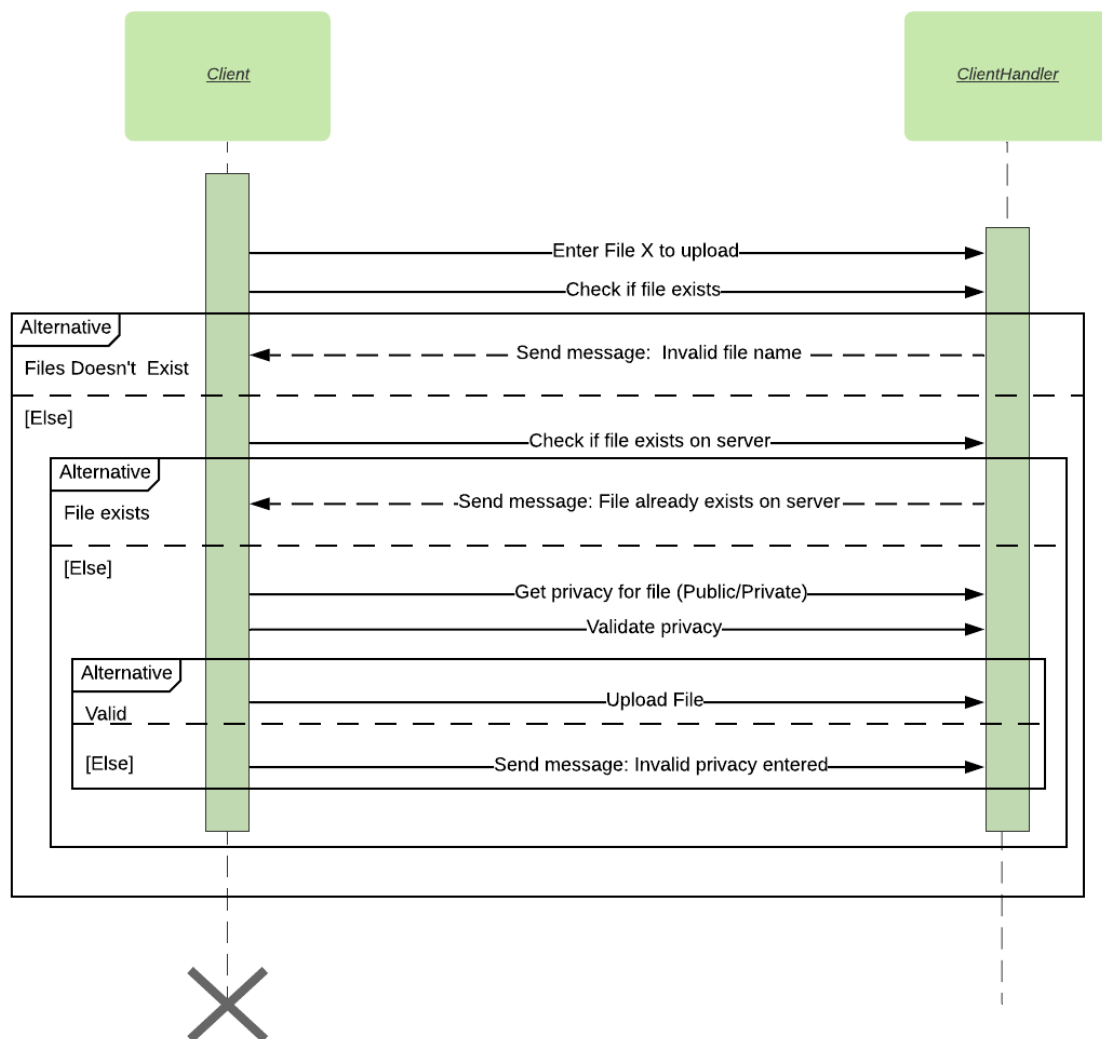
### Sequence Diagram for Querying List of Available Files



### Sequence Diagram for Downloading File X



### Sequence Diagram for Uploading File X



## Discussion

TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are the two major transport protocols. They both work at the transport layer in the TCP/IP model, but have different usages. In our file sharing application we made use of TCP sockets. TCP is connection oriented – once a connection is established, data can be sent bidirectional. UDP is a simpler, connectionless Internet protocol. Multiple messages are sent as packets in chunks using UDP.

TCP may be more heavyweight compared to the lightweight UDP but it has many advantages. It is good for error detection and retransmits packets if a sender doesn't get acknowledgement after a period of time.