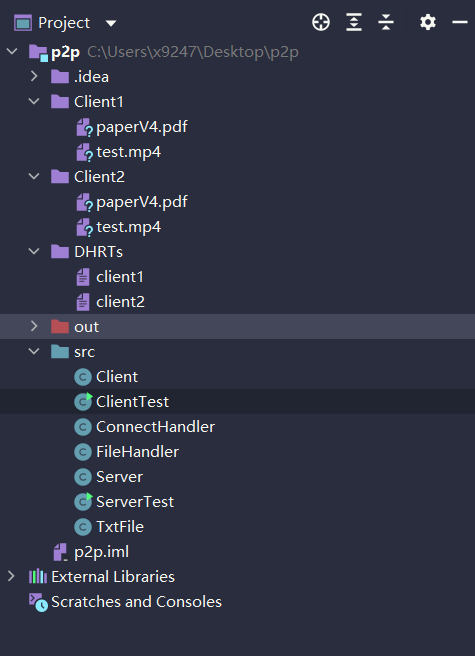
Design document

1. Project Structure and principle

This peer-to-peer project contains 7 classes, and each of them has some specific functions.

There are two test classes—ClientTest and ServerTest. In ServerTest, a server object is created and its start method is called to start the server. In ClientTest, two clients (peers) object are created and initialized in two threads in the method of GUI. The ServerTest should be start before the ClientTest to prevent the situation of client’s lose connection. It’s free and easy for users to add more clients due to the multi-thread is introduced in ClientTest class so that different client object has the same structure.

The Server class and the Client class abstract two main components of the system—server and client. For the server object, it holds three kinds of HashTable, in separate, UHPT, UHRT, and ipPortTable. All of these will be introduced in next several sections. These tables will be created when the server object is created and updated when new client is online, client shares file, or client is offline. They will not be stored in the local disk. The server also holds its port number to build the serverSocket. For the client, it holds its DHRT, IP address and port number, and the IP Address and port number of the server. The DHRT table will be stored in the local disk as txt format and can be upload to the server when the client is online. For the communication process, server will build a serverSocket and wait for client’s connection. After a client connect the server, it will start a new thread called ConnectHandler to handle the communication. Several client threads will start together and connect to the server. After they have been handled by the ConnectHandlers, they can share the files or ask for some files. The share message will be transmitted only from client to the server to upload the tables in server, but for request message, the server will check the tables and give some replies, which includes target client’s IP Address and port number so that these two clients can build socket connection.

In order to build socket connection between two clients, one client should act as server and start a serverSocket to wait for others connect. After a client object was created, a new fileSocket will be built to wait for the connection. If another client connects to the original one, it will start a new thread like ConnectHandler, named FileHandler, which has the job to deal with the connection.

In the end, for TxtFile class, it contains three methods to create, read, and write the txt file.

1. Tables stored in the system
2. UHPT

|  |  |
| --- | --- |
| GUID of client (UUID) | Routing Matric/Distance (int) |

GUID of client is generated by server by using UUID.randomUUID()

Routing Matric/Distance is an integer generated randomly from 1 to 100

1. UHRT

|  |  |
| --- | --- |
| GUID of file/MD5 (String) | List of clients has the file (Arraylist) |

GUID of file/MD5 is generated by MessageDigest class

List of clients has the file contains the GUID of clients

1. ipPortTable

|  |  |
| --- | --- |
| GUID of client (UUID) | IP combined with port number (String) |

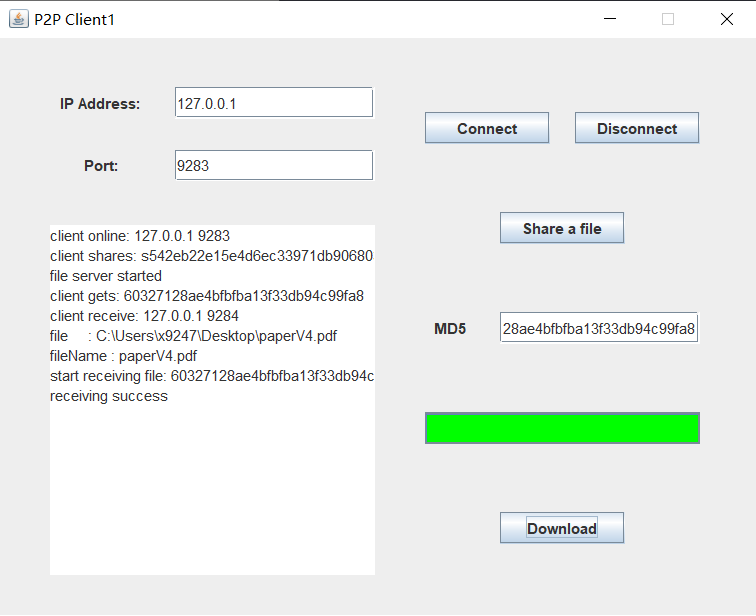
IP combined with port number is a String combined IP Address and port number with port number at the beginning of the String.

1. DHRT

|  |  |
| --- | --- |
| GUID of file/MD5 (String) | Absolute path of the file (String) |

Absolute path of the file is stored in the DHRT in order to search the file easily.

1. GUI



The detailed explanation can be found in video.

1. Performance, Pros and Cons

The system has a high performance in transmission in about 50MB/s due to the transmission takes place locally. For the pros, the system is very reliable because it is built on the TCP protocols, which is simple, stable, and it’s hard to lose the packs. The system has a high useability, in which users only need to know the server IP and port to build the connect and type the MD5 value of a file to download. The file sharing is very convent as well. The system has a high efficiency, which means the file transmission is pretty quick. I can save the time of moving some file from one computer to another. In the end, the system also has good portability. The system can be implemented in different platforms and operating systems, which even can be modified to realize some other kinds of works like communicate or message sending. For the cons, the functionality of the system is constrained. Only for the system implemented by ourselves, it just runs on the local network and its hard to communicate with the devices on the real Internet.

1. Rational of the decision

The multi-thread idea is the core idea in the project. Considering the principle of the p2p system, the server must allow several clients to connect, so that it needs to start corresponding threads for them to handle the information transmission separately. For clients, different clients cannot influent each other, so they also need different threads. Knowing when the multi-thread is need is very important for the project construction and can also protect data safety. Another essential idea is the table. The server and client need several tables to store the data of files and connections to maintain the functionality of the system. It’s very familiar with the concept of database. By consider the tables as another kind of database, it is much more easier to understand relations of the system.