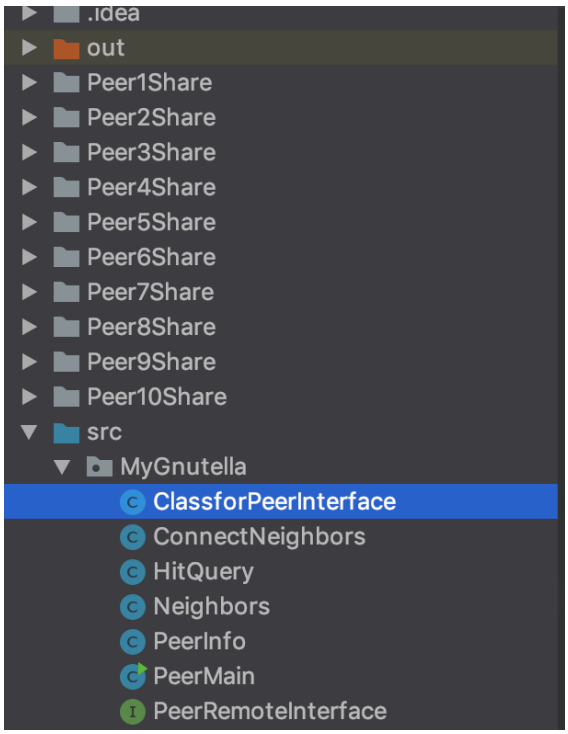
Design Document

1. Introduction

This is a simplified peer-to-peer file sharing system which is similar to Gnutella. There are 10 peers which act as both clients and servers. As clients, they could send to search specified files. As servers, they could receive queries and search in their local storage. Meanwhile, peers broadcast the queries to all of their neighbors.

2. Project

Project is done in IntelliJ IDEA. In package MyGnutella, there are several classes are defined along with one remote interface that is used in RMI process.



ClassforPeerInterface: core methods are implemented here that is used for obtaining the file, and handling queries. Basically, methods in the PeerRemoteInterface are implemented here.

ConnectNeighbors: this class is responsible for connecting the neighbors with each peer.

HitQuery: simply there are two arraylists are defined here to store the info of peers when the hit query is handled.

Neighbors: basic properties for neighbors are declared here.

PeerInfo: this class is used for maintaining the peer detailed info

PeerMain: this is the core class to run the peer as server along with its UI and handle all the actual process of files.

PeerRemoteInterface: the remote interface to use for RMI.

Config file

```
# PeerMain details
peerid.1.ip=127.0.0.1
peerid.1.port=5100
peerid.2.ip=127.0.0.1
peerid.2.port=5200
peerid.3.ip=127.0.0.1
peerid.3.port=5300
peerid.4.ip=127.0.0.1
peerid.4.port=5400
peerid.5.ip=127.0.0.1
peerid.5.port=5500
peerid.6.ip=127.0.0.1
peerid.6.port=5600
peerid.7.ip=127.0.0.1
peerid.7.port=5700
peerid.8.ip=127.0.0.1
peerid.8.port=5800
peerid.9.ip=127.0.0.1
peerid.9.port=5900
peerid.10.ip = 127.0.0.1
peerid.10.port = 6000


# Topology for neighbors:

#all-to-all connection:

peerid.1.neighbors=peerid.2,peerid.3,peerid.4,peerid.5,peerid.6,peerid.7,peerid.8,peerid.9,peerid.10
peerid.2.neighbors=peerid.1,peerid.10,peerid.3,peerid.4,peerid.5,peerid.6,peerid.7,peerid.8,peerid.9
peerid.3.neighbors=peerid.1,peerid.2,peerid.10,peerid.4,peerid.5,peerid.6,peerid.7,peerid.8,peerid.9
peerid.4.neighbors=peerid.1,peerid.2,peerid.3,peerid.10,peerid.5,peerid.6,peerid.7,peerid.8,peerid.9
peerid.5.neighbors=peerid.1,peerid.2,peerid.3,peerid.4,peerid.10,peerid.6,peerid.7,peerid.8,peerid.9
peerid.6.neighbors=peerid.1,peerid.2,peerid.3,peerid.4,peerid.5,peerid.10,peerid.7,peerid.8,peerid.9
peerid.7.neighbors=peerid.1,peerid.2,peerid.3,peerid.4,peerid.5,peerid.6,peerid.10,peerid.8,peerid.9
peerid.8.neighbors=peerid.1,peerid.2,peerid.3,peerid.4,peerid.5,peerid.6,peerid.7,peerid.10,peerid.9
peerid.9.neighbors=peerid.1,peerid.2,peerid.3,peerid.4,peerid.5,peerid.6,peerid.7,peerid.8,peerid.10
peerid.10.neighbors=peerid.1,peerid.2,peerid.3,peerid.4,peerid.5,peerid.6,peerid.7,peerid.8,peerid.9
```

```
peerid.10.neighbors=peerid.1,peerid.2,peerid.3,peerid.

#linear connection
#peerid.1.neighbors=peerid.2
#peerid.2.neighbors=peerid.1,peerid.3
#peerid.3.neighbors=peerid.2,peerid.4
#peerid.4.neighbors=peerid.3,peerid.5
#peerid.5.neighbors=peerid.4,peerid.6
#peerid.6.neighbors=peerid.5,peerid.7
#peerid.7.neighbors=peerid.6,peerid.8
#peerid.8.neighbors=peerid.7,peerid.9
#peerid.9.neighbors=peerid.8,peerid.10
#peerid.10.neighbors=peerid.9
```

Two different topologies could be realized based on two different configuration.