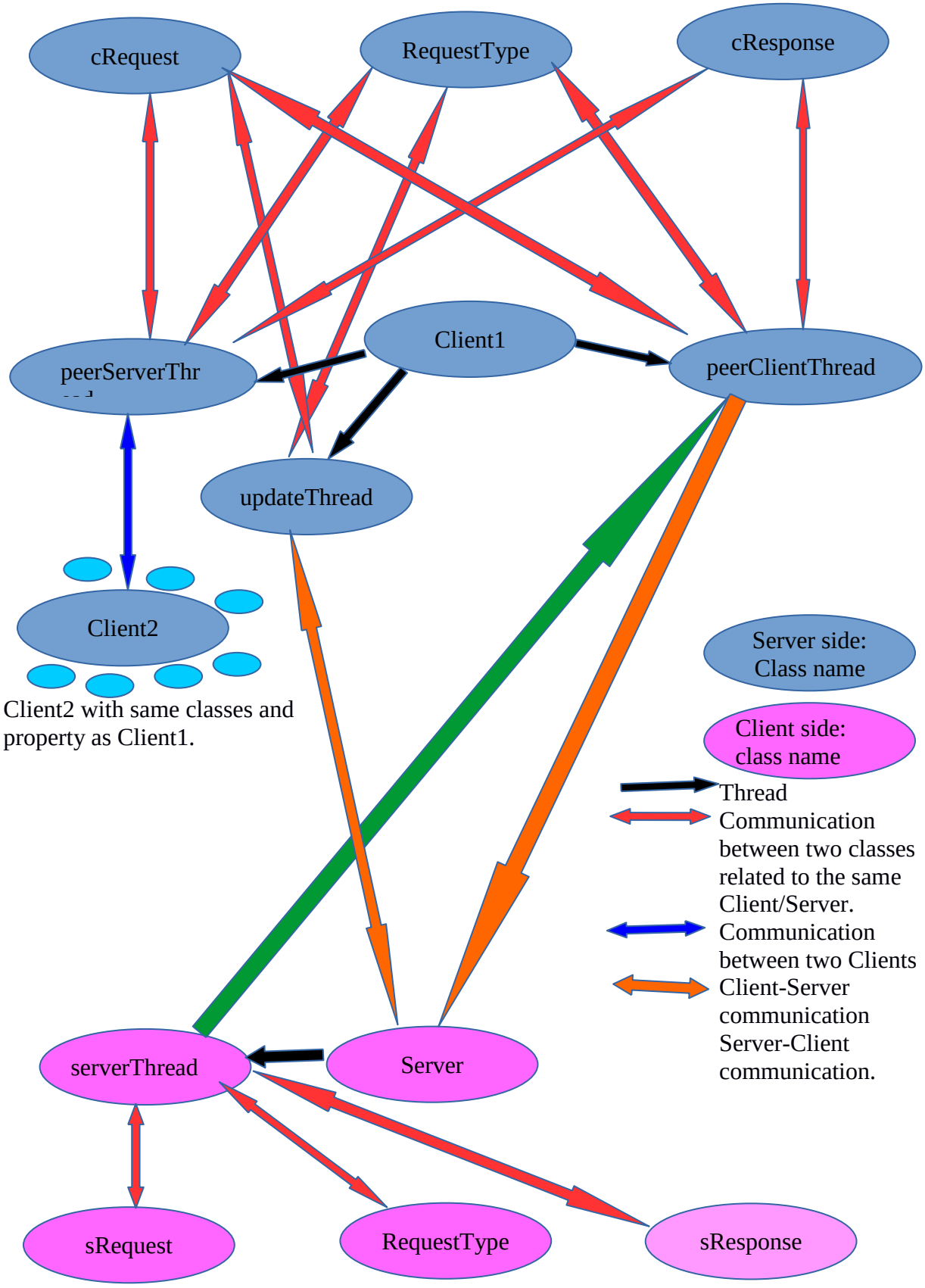
Our Peer to Peer File Sharing System Design.



Flow logic: The above diagram demostrate the flow logic

- First we make the server side active by running the server side program. It waits for client request to process the request type and response with the required data. This is done by **server.java** class which invoke **ServerThread.java** class.
- We run client side to take the request of the user at the client side to either register files to the server, unregister files from the server, receive all the files from a client, request PeerList from the server related to the filename, request a particular file ,don't request file, request an update at the server logs for any change at any client this update is done by **UpdateThread.java** class . This is done by **PeerClientThread.java** class invoke by the **Client.java** class.
- **PeerServerThread.java** class receive the request from the server related to the file requested and access the file from a client which has the file related to the request.
- **Response.java, RequestType.java, Request.java :** This classes are accessible by Client-side as well as server-side. These classes contain functions for proper execution and control of the file transfer or any other type of communication between two peers.

Server side :

sRequest.java : Implements Serializable java package, have following functions :-

- public final InetAddress getIpAddr() : Returns IP-Address InetAddress object which is set by void setIpAddr(InetAddress ipAddr) function.
- void setIpAddr(InetAddress ipAddr) : Set InetAddress object IP-Address using the passed parameter ipAddr.
- int getPort() : Returns the port number.
- void setPort(int port) : Sets the port number to the port value passed to the function.
- String getHostName() : Return HostName as string value.
- void setHostName(String hostName) : Set HostName value with the parameter string value hostName.
- String getFileName() : Return the name of the filename set by void setFileName(String fileName) function.
- RequestType getRequestType() : Return RequestType enum value identifying the type of request made by peers.
- void setRequestType(RequestType requestType) : Set the RequestType enum value identifying the type of request made by peers.
- boolean isDownload() : Return boolean value related to whether Peer wants to download the requested file.
- void setDownload(boolean download) : Set boolean value for download variable.
- ArrayList<String> getFilenames() : Return ArrayList of all the filenames for processing
- void setfilenames(ArrayList<String> filenames) : Set filenames into a ArrayList.

RequestType.java : enum RegisterType having some values

sResponse.java : Implements Serializable java package, having following functions :-

- String getMessage() : This function return a message related to response.
- void setMessage(String message) : This function set the message to some string value passed as a parameter.
- boolean isSuccess() : this function return true if a communication between Client and server is a successful one else it return false.

- void setSuccess(boolean success) : This function set the success boolean variable equals to the boolean value passed to the function as the parameter confirming successful communication or not.
- ArrayList<Request> getPeerList() : This function returns a string array list with all the peer information in it.
- void setPeerList(ArrayList<Request> peerList) : This function set a string array list with peer information.
- boolean isfileExists() : This function checks whether a particular file exist or not.
- void setfileExits(boolean fileExists) : This function set boolean fileExists variable to true if a particular file exists else it set the variable to false.

Server.java : This class contains the main() function, it generate a server thread for processing different kinds requests from the clients. It calls the run() function which generate server thread and creates ServerThread.java object for processing client request.

ServerThread.java : This class uses ObjectInputStream ad other class objects to process the RequestType of the Client.

- It register the files of a requesting client on the server side when RequestType is request. It also checks redundancy by rejecting the same request from the same client if no changes has been made.
- If the RequestType is RequestPeerList then it uses the filename passed to the server with the request object to find out which all peer has the same file and return the peer list to the client.
- If the RequestType is unregister then this class unregister the files of the related peer which made the unregister request.
- If the RequestType is RequestUpdate then it first unregister the peer making request and the register fresh update on the server side.
- If the RequestType is RequestAllFiles then it return all the files related to the peer's request.
- boolean unregisterClient(int clientPort) : This function is used to unregister Client using the client port number.
- void sendObject(Response response, boolean fromServer) : It bundle the response from the server into an object and then send it to the requesting client.

Client side :

cRequest.java : Implements Serializable java package, have following functions :-

- public final InetAddress getIpAddr() : Returns IP-Address InetAddress object which is set by void setIpAddr(InetAddress ipAddr) function.
- void setIpAddr(InetAddress ipAddr) : Set InetAddress object IP-Address using the passed parameter ipAddr.
- int getPort() : Returns the port number.
- void setPort(int port) : Sets the port number to the port value passed to the function.
- String getHostName() : Return HostName as string value.
- void setHostName(String hostName) : Set HostName value with the parameter string value hostName.
- String getFileName() : Return the name of the filename set by void setFileName(String fileName) function.
- RequestType getRequestType() : Return RequestType enum value identifying the type of request made by peers.

- void setRequestType(RequestType requestType) : Set the RequestType enum value identifying the type of request made by peers.
- boolean isDownload() : Return boolean value related to whether Peer wants to download the requested file.
- void setDownload(boolean download) : Set boolean value for download variable.
- ArrayList<String> getFilenames() : Return ArrayList of all the filenames for processing
- void setfilenames(ArrayList<String> filenames) : Set filenames into a ArrayList.

RequestType.java : enum RegisterType having some values related to the type of request client initiates.

cResponse.java : Implements Serializable java package, having following functions :-

- String getMessage() : This function return a message related to response.
- void setMessage(String message) : This function set the message to some string value passed as a parameter.
- boolean isSuccess() : this function return true if a communication between Client and server is a successful one else it return false.
- void setSuccess(boolean success) : This function set the success boolean variable equals to the boolean value passed to the function as the parameter confirming successful communication or not.
- ArrayList<Request> getPeerList() : This function returns a string array list with all the peer information in it.
- void setPeerList(ArrayList<Request> peerList) : This function set a string array list with peer information.
- boolean isfileExists() : This function checks whether a particular file exist or not.
- void setfileExits(boolean fileExists) : This function set boolean fileExists variable to true if a particular file exists else it set the variable to false.

Client.java : This class has the Client side main() function, which intern class updateThread Class object to keep checking the status of the files present at the Client associated and update automatically when there is a change in the file system. It also create peerClientThread and peerServerThread class object to take and process the client request.

UpdateThread.java : This class invoke it's constructor when it's object is created, a value one second is passed for checking the status of the file at the client side whether it has undergone any change. If it has undergone any change then it update the file status at the server side as well.

PeerClientThread.java : This class take the client requestType and calls related functions :-

- run() : This function accept the client request and call the respective function.s
- Register() : This function register the files in the client-side at the server-side.
- Unregister() : This function unregister the files of the particular client at the server-side.
- RequestFilelist(): This function access all the file at the server.
- RequestPeerlist(): This function access all the peer names and information associated with it from the server.
- SendObject2Server(): This function use object ObjectOutputStream to send related object to the server-side.

- SendObject2Client() :This function use object ObjectOutputStream to send related object to the client-side.\
- receiveFile() : This function receive the file and its content from the client which has the reuested file.

PeerServerThread.java : This class is used to make client act as a server by communicating with other client member in the connected cluster. It has following functions :-

- run() : This function accept the filename requested from requesting client and then process on the file depending upon the requestType parameter.
- DoesFileExists: This function set the boolean variable fileExist to True or False depending upon whether the file requested exists in the client or not.
- SendFile() : This function return the file to the requesting client.
- ReadRequestObject(): This function read the object related to the communication between the requesting client and processing client.

Files :

- **Input_files :** This folder contains all the documented files at the client side and this folder is accessible by the Java classes at the client side
- **Output_files :** This folder contains the requested file after receiving it from other peers.