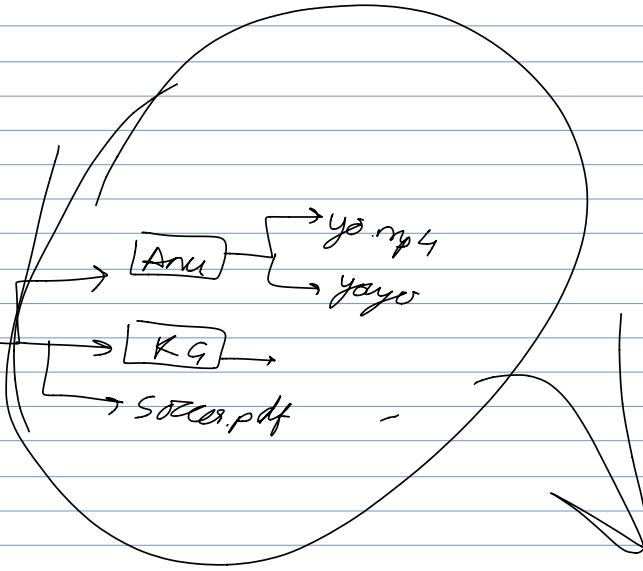
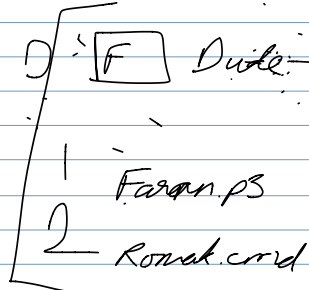


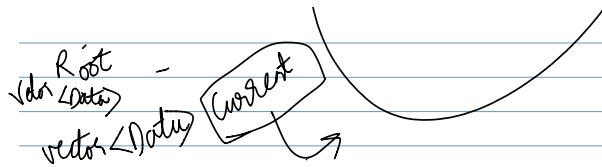
Filechanges Mainfiles

Data

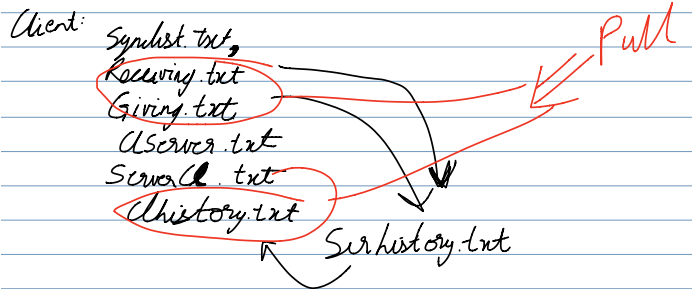
int x  
string name  
std::vector<Data>

x=0 ⇒ file  
x=1 ⇒ folder





Refresh Sync List



Data = Name,  
vector  
vector<Data>

WriteData(Data, N)

```

s1 = ""
for elem in vcl:
    s1 += elem.name
    s1 += elem.type ; if(elem.type) => WriteData(elem.data, N + elem.name);
Write(s1, N + name)

```

ReadData(Location)

```

ans = {}
open f
for lines:
    if lines.type == 1:
        ans.push_back(ReadData(Location + " " + lines.name))
    else:
        ans.push_back(Data(lines.type, lines.name))
return Data(k, l, ans)

```

- 1) Open Login GUI.
  - 2) Give button pressed with key
    - a) if login,
      - Send data to server
      - Receive confirmation.
      - If Yes:
        - Open files UI
        - Receive Server Files
        - " Shared P'
        - " Receiving files
- Sync.
- Give instruction list

Sync:

- Receive Server Files
- " Shared P'
- " Receiving files

Sync.

→ Give instruction list

↳ Execute instruction list  
Refresh serverfiles  
Refresh clientfiles.

↳ Execute instruction list  
Refresh serverfiles  
Refresh clientfiles.

Share files:

↳ Open UI for username, preferences,  
Send message to network  
↳ Add file to giving  
↳ Add file to receiving.

Logout:

Open Message  
Close socket

Add to sync list

• Add to sync list

Keep online only

• Remove from sync list  
• Delete file from client.

Refresh Button:

- Data Writer Read  
- Get Delta of files on disk  
- Sync.

New User:

Get details  
Send to server

• Add to userbase  
• Write Userbase  
• Create folder

Make initialisation files.

Client:

Make initialisation Files

Add file to DD:

• Path → DD folder.  
• Add to sync list  
• Sync.

Server: