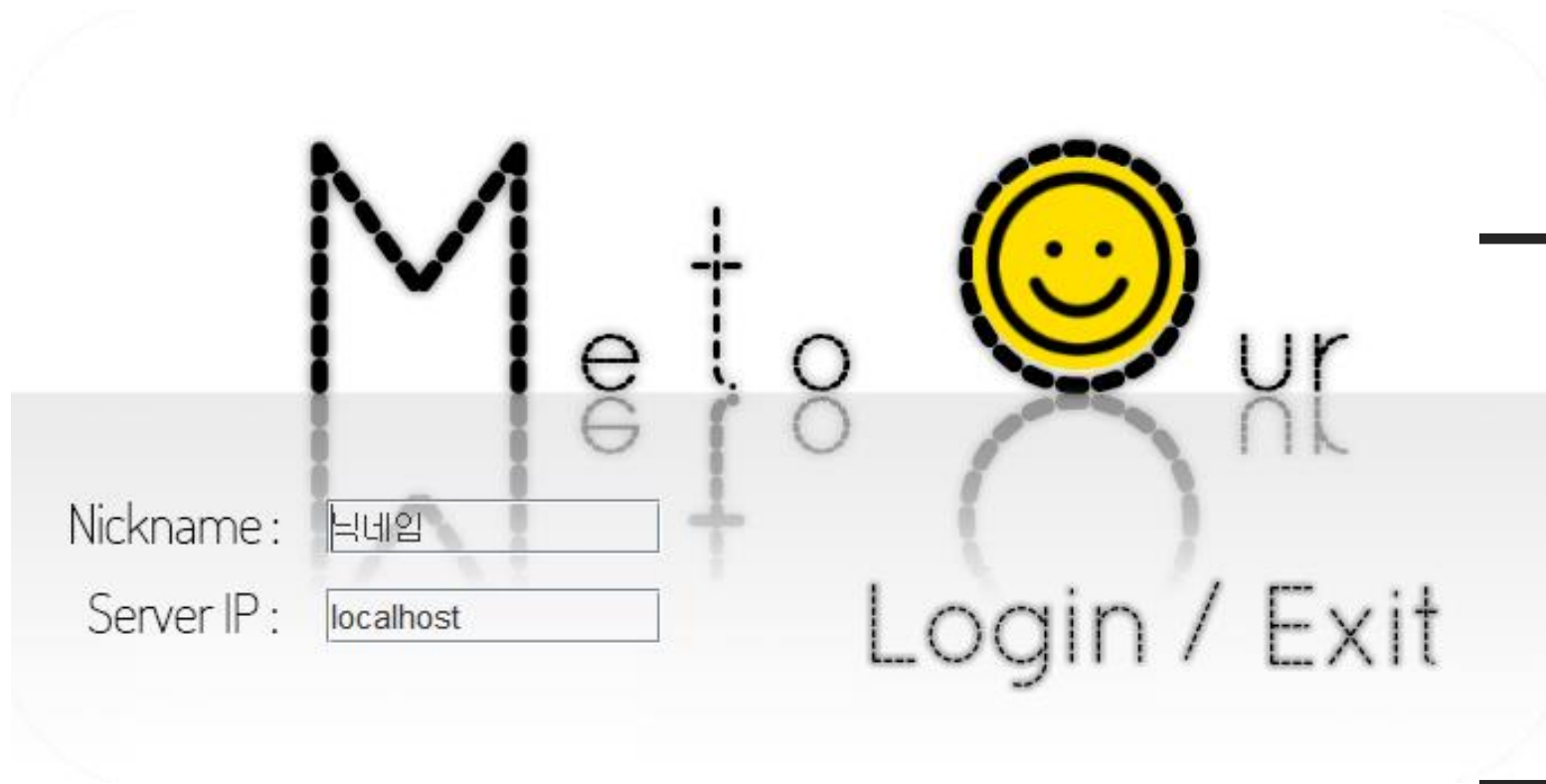
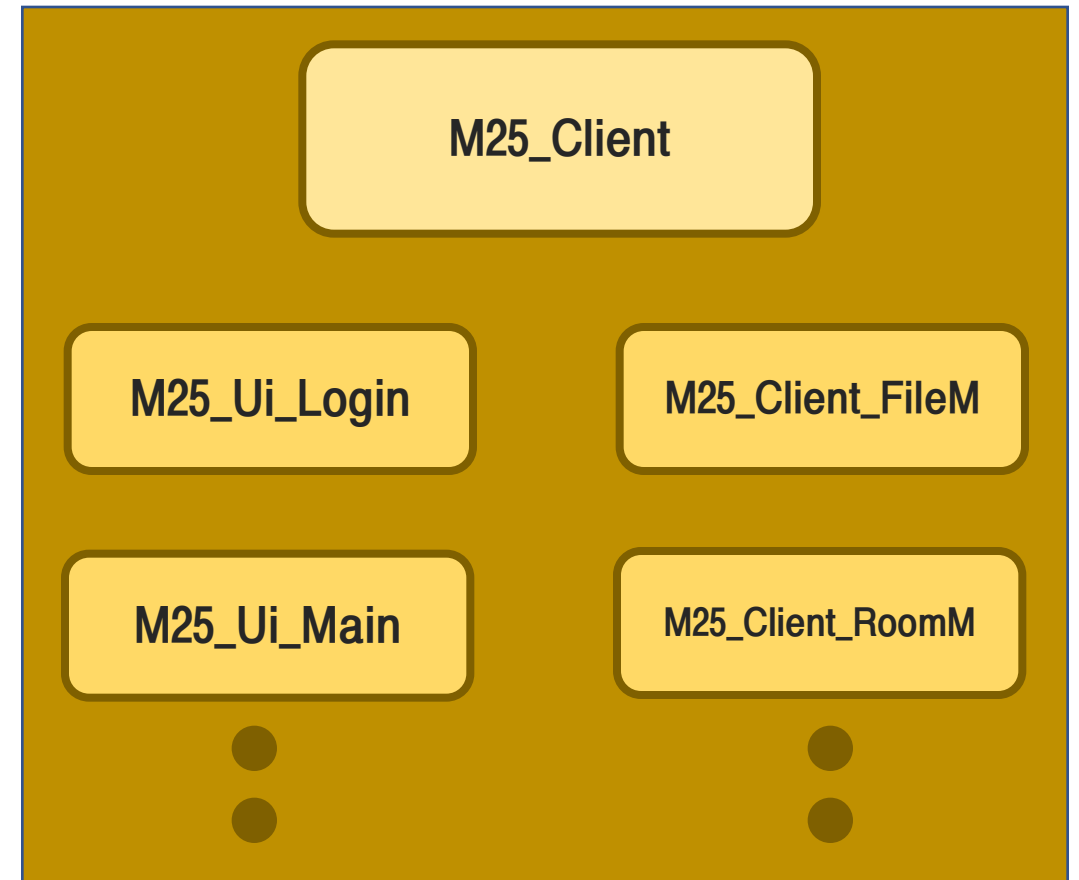
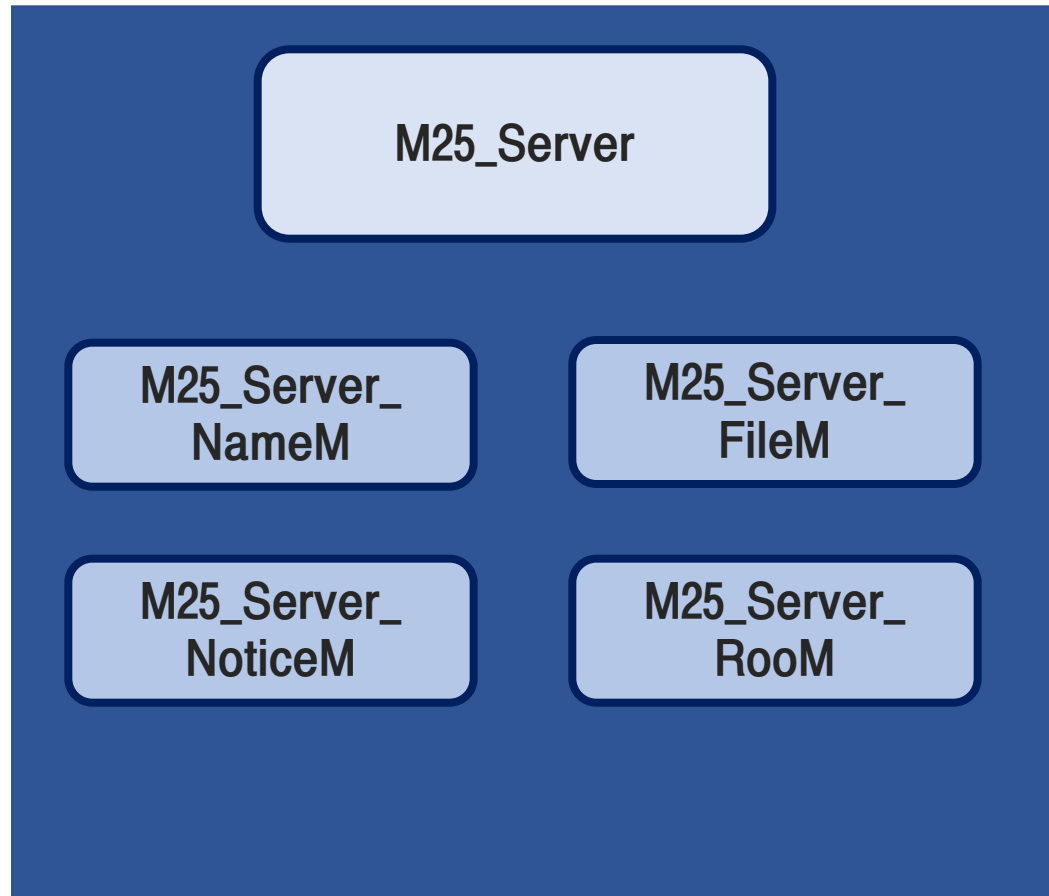

[Message To Our] Project



김수성
김송인
변진수
이동준
이시현

Project – Diagram



Detail – Server

M25_Server

: 메인메소드를 가지고 하위 서버들의 객체를 생성하여 총괄관리
서버의 IP, 서버포트번호, 자주쓰는 단어들 정리.

M25_Server_
NameM

클라이언트
접속 연결,
닉네임 체크,
로그아웃 체크

M25_Server_
NoticeM

공지글
작성, 수정, 삭제,
자료구조 관리

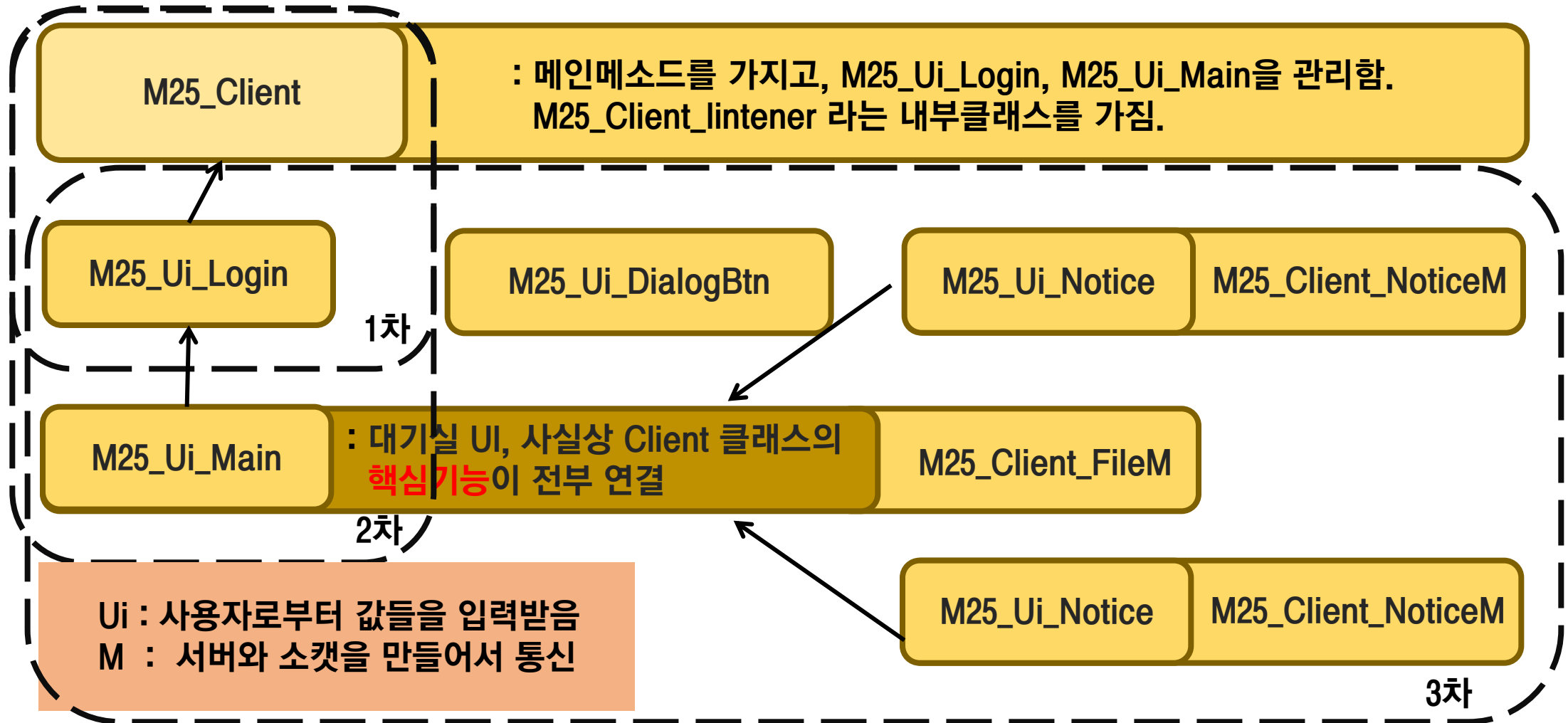
M25_Server_
FileM

파일
업로드, 다운로드
자료구조 관리

M25_Server_
RoomM

다대다
채팅방
관리

Detail - Client



Core Code – Server

M25_Server

```
public static void main(String[] args){
    M25_Server me = new M25_Server(serverPort);
    me.start();
    me.fileM.start();
    me.noticeM.start();
    me.roomM.start();
} // main end
```

```
void sendMessage(String msg, String subject) {
```

```
void sendList(String msg) {
    StringBuffer sendM = new StringBuffer();
    String tmp = "";
    if(msg.equals(ONLINE_LIST)){
```

```
public class M25_Server extends Thread{

    ServerSocket mainS;
    Socket clientS;
    int serM_port;

    static int peopleCnt;
    static HashMap<String, M25_Server_NameM> NICK_M; // Nick-Sock
    static int serverPort = 4000;
    static int serverFileP = 4001;
    static int serverFileP2 = 4002; // 사실상 안씀
    static int serverNotitcP = 4003;
    static int serverRoomP = 4004;

    static final String SOCK_CLOSE = "##exit##";
    static final String LIST_EMPTY = "##비었음##";
    static final String ONLINE_LIST = "온라인";
    static final String FILE_LIST = "파일";
    static final String NOTICE_LIST = "공지";

    static final String DEFAULT_FILE_LIST = "./bin/file_list.bin";
    static final String DEFAULT_NOTICE_LIST = "./bin/notice_list.bin";

    M25_Server_FileM fileM;
    M25_Server_NoticeM noticeM;
    M25_Server_RoomM roomM;
```

Core Code – Server

M25_Server_FileM

```
public class M25_Server_FileM extends Thread {
    // 업로드      : 업로드      @패스(업로드할 파일이름만)  @pw
    // 다운로드    : 다운로드    @패스(다운로드 파일이름만)
    // 삭제        : 업로드      @패스(삭제할 파일+이름)      @pw

    private M25_Server upper;
    private int portC;
    // private int portF;

    private ServerSocket servSockC;
    // private ServerSocket servSockF;

    // 파일이름 - pw
    HashMap<String, String> file_list; // 파일이름 / pw
}
```

```
class M25_Server_FileM_Runner_C extends Thread{

public void run() {
    isr_c = new InputStreamReader(is_c);
    buffr_c = new BufferedReader(isr_c);

    try {
        String msg = buffr_c.readLine();
        System.out.println(sock.getInetAddress()+"이 보낸 메세지 : 
        command(msg);

    } catch (IOException e) {
        e.printStackTrace();
    }
} // run() end

private void command(String msg) {

    commands = msg.split("@");
    if(commands[0].equals("업로드")){
        saveFile();
    }else if(commands[0].equals("다운로드")){
        sendFile();
    }else if(commands[0].equals("삭제")){
        deleteFile();
    }
} // command() end
}
```

Core Code – Client

M25_Client

```
class M25_Client_lintener extends Thread{
    String msg;
    @SuppressWarnings("deprecation")
    public void run() {
        try {
            while(true){
                // M25_Seaver로 부터 명령 대기중 ...
                msg = buffr.readLine();
                System.out.println("공지의 msg값 -----"+msg);
                String[] tmp = msg.split("리스트@@");
                for (int i = 0; i < tmp.length; i++) {
                    System.out.println("tmp의 값 "+i+" -----"+tmp[i]);
                }
                if(tmp.length>1){
                    String[] tmp2 = tmp[1].split(",");
                    if(tmp[0].equals(ONLINE_LIST)){
                        myMainUi.onlineList.removeAll();
                        for( int i=0; i<tmp2.length; i++ ){
                            if(tmp2[i].trim().equals(people.getNickName())){tmp2[i]
                                myMainUi.onlineList.add(tmp2[i].trim());
                            }
                        }
                    }
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```
static String serverIP ;
static int serverPort = 4000;
static int serverFileP = 4001;
static int serverFileP2 = 4002;
static int serverNotitcP = 4003;
static int serverRoomP = 4004;

static public M25_People    people;

public Socket                mainS;
public OutputStream          mainOs;
private OutputStreamWriter   osw ;

public InputStream            mainIs;
private InputStreamReader    isr ;
private BufferedReader         buffr;

public M25_Ui_Login           myLoginUi;
public M25_Ui_Main             myMainUi;

public static final String SOCK_CLOSE = "##exit##";
static final String LIST_EMPTY = "##비었음##";
static final String ONLINE_LIST = "온라인";
static final String FILE_LIST = "파일";
static final String NOTICE_LIST = "공지";
```

Core Code – Client

M25_Ui_Login

```
static M25_Client actor;           //  
  
public M25_Ui_Login(M25_Client me) {  
    actor = me;
```

M25_UI_Main

```
M25_Ui_chatRoom chatRoom;  
static M25_Client actor;  
static M25_Client_FileM fileM;  
  
public M25_Ui_Main(M25_Client m25_Client) {
```

M25_Ui_Notice

```
if(MorR == "read"){  
    actor = new M25_Client_NoticeM(this, "nickname");  
    actor.start();  
    M25_Notice tmpNotice = new M25_Notice();  
    tmpNotice.setNoticeTilte(readTitle);  
    actor.loadNotice(tmpNotice, "");  
    temp = readTitle;
```

M25_Client_NoticeM

```
public M25_Client_NoticeM(M25_Ui_Notice gui, String nick){  
    this.gui = gui;  
    this.nickName = nick;  
}  
  
public void sendNotice(M25_Notice tmp, String msg){
```


Thing to Add

- 1) 서버에서 파일리스트, 공지글 리스트를 특정주기로 저장.
- 2) 불필요한 static 변수 제거.
- 3) 변수들 이름 통일성 있게 변경.
- 4) 서버 UI 및 각종 로그 기록수단 확보.
- 5) ...



구현하기 전 더 체계적이고 상세한 설계, **스토리라인** 구성.
구성원들 간의 **의견교류**를 통해 **공통된 그림**을 가지게 함.