20-1 사이버보안기초프로젝트(01)

1팀 2차 중간점검

1971060 김서현
1971063 김윤서
1971066 김채연
1971090 한근영

※PPT 슬라이드 26부터 시작됩니다※

20-1 사이버보안기초프로젝트(01)

1팀 프로젝트 제안서

1971060 김서현
1971063 김윤서
1971066 김채연
1971090 한근영

프로젝트명 : **<겹공강을 찾아서>**

회의, 식사 등을 목적으로 약속을 잡을 때, 상대방과 일정이 없는 시간을 비교하여 조율함에 번거로움이 있다. 특히, 여러 명이 모이는 경우 비교에 큰 어려움이 발생한다. 따라서, 이 문제를 해결해주는 프로그램을 제작하는 프로젝트를 진행하고자 한다.

본 프로그램은 만남 가능한 시간을 찾아주어 약속을 잡는데 도움을 준다. 참석 대상들이 시간표를 입력하면 사용자들의 '**공강' 시간을 비교**하여 모두가 참석 가능하거나 가장 많은 사람들이 **참석 가능한 시간을 알려준다**. 또한, **참석 여부 확인**, **예약 문자 전송 기능**을 추가하여 프로그램 내에서 약속을 최종적으로 정할 수 있도록 한다.

팀 구성원	담당 기능
김서현(대 표)	시간표 입력 및 출력
김윤서(조원 1)	약속 시간 예약 전송
김채연(조원 2)	전반적인 구조설계, 약속 참석 여부 확인
한근영(조원 3)	겹공강 분석 및 출력

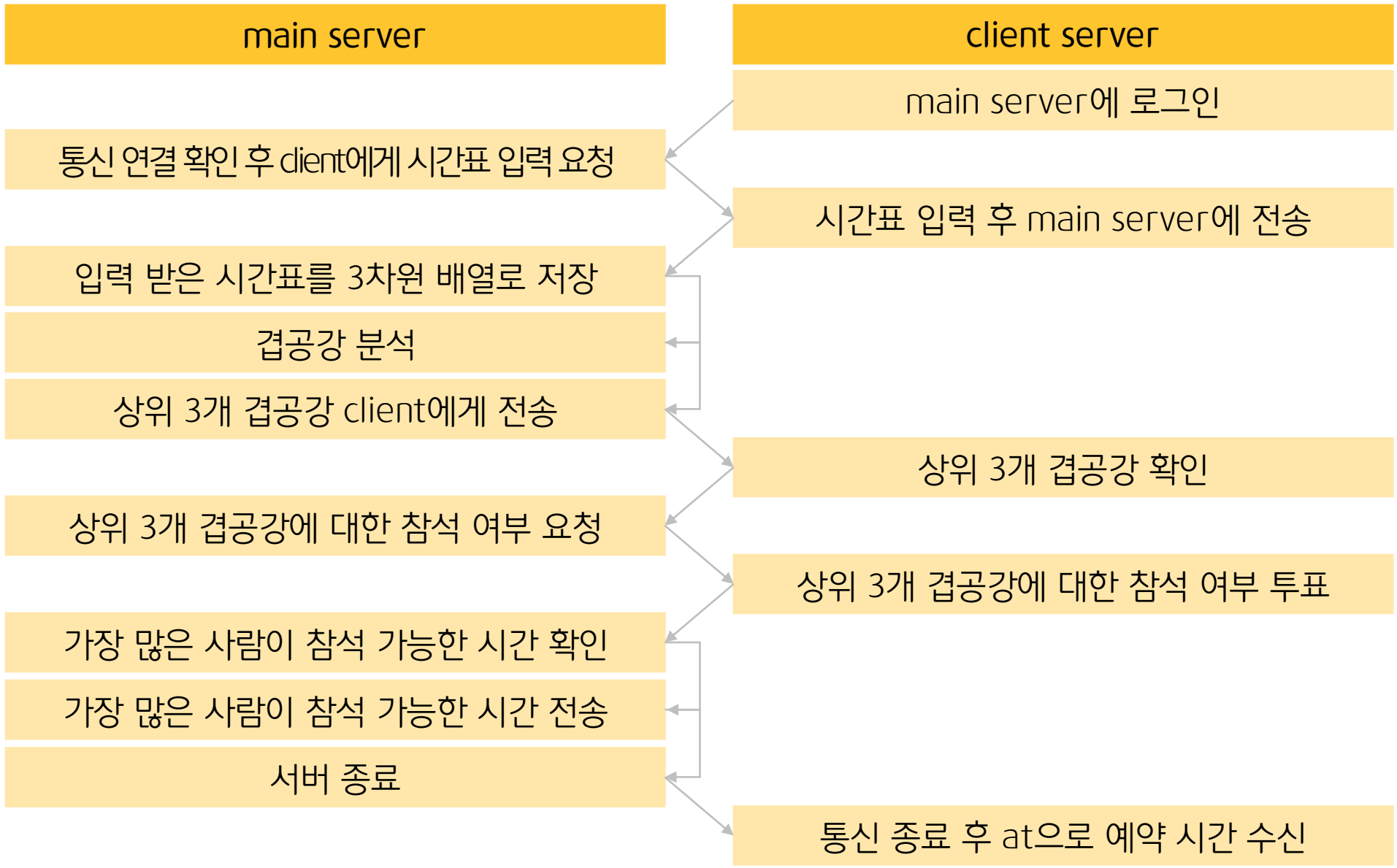
- 모든 코딩 및 디버깅에 조원들이 동일한 비율로 참여하되 각자 가장 집중할 기능을 '담당 기능'이라 칭하여 분배함
- 보고서, 발표 자료 준비 필요 시 이후 역할 분배 예정

주요 기능 | 기능 별 구현 방식

기능	특징
내 시간표 입력 및 출력	• 사용자들의 시간표를 입력 받은 후 출력
겹공강 분석 및 출력	• 사용자들의 겹공강 시간을 파악 후, 해당 시간에 공강인 사람들의 이름을 출력 • 가장 많은 사람이 참석 가능한 시간 상위 3개를 출력
약속 참석 여부 확인	• 가장 많은 사람이 참석 가능한 시간 상위 3개에 대하여 참석 여부 확인
약속 시간 예약 전송	• 참석 여부에 따라 결정된 약속 시간을 사용자들에게 각각 알림

기능	특징
내 시간표 입력 및 출력	for문을 이용하여 사용자의 시간표 정보(요일, 시간)을 입력 받고 저장한 후 배열을 이용하여 시간표 출력
겹공강 분석 및 출력	3차원 배열을 이용하여 모든 사용자들의 공강 시간을 반영(해당 시간에 공강인 사람 이름을 표시)한 시간표 출력 후 for문을 이용하여 가장 많은 사람이 공강인 시간 중 상위 3개와 해당 사용자명을 출력
약속 참석 여부 확인	가장 많은 사람이 공강인 시간 중 상위 3개에 대하여 모든 사용자들에게 참석 여부를 확인 받은 후 for문(참석자 수를 count하고 이름을 저장)을 이용하여 결과를 출력
약속 시간 예약 전송	참석을 하기로 한 사용자들에 한하여 at를 사용하여 약속 시간을 약속 당일 자정에 메일로 예약 전송함

통신 과정 | 부가 설명



통신 과정 | **부가 설명**

1. [client] 여러 명의 사용자가 서버에 동시에 로그인
2. [main -> client] 시간표 입력 질문 및 형식 전송
3. [client] 시간표 입력
4. [client -> main] 입력 받은 시간표 전송
5. [main] 입력 받은 시간표를 3차원 배열로 저장
6. [main] 3차원 배열로 저장한 시간표들을 비교하여 겹치는 공강을 분석
7. [main] 겹공강인 사람이 많은 시간대 상위 3개 분석
8. [main -> client] 분석 결과 전송
9. [client] 겹공강인 사람이 많은 시간대 상위 3개 확인
10. [client] 겹공강인 사람이 많은 시간대 상위 3개에 대한 참/불참 투표
11. [client -> main] 참/불참 투표 결과 전송
12. [main] 사용자로부터 받은 참석 여부를 분석해서 가장 많은 참석자에 따른 약속시간 확정
13. [main -> client] 최종 약속시간 전송
14. [main] 서버 종료
15. [main -> client] 통신 종료 후
16. [client] at으로 최종 약속 시간 예약 수신
17. [client] 서버 종료

>> **소켓 프로그래밍** 활용
>> client ↔ main 통신에 **TCP/IP 프로토콜** 사용

프로그래밍 언어 | Tool / Command

C (Shell programming 추가 예정)

프로그래밍 언어

Tool / Command

gcc, make, vi, cp, sort, at

5월 6월

일	월	화	수	목	금	토
					1	2 비상 회의 (제안서 작성)
3	4	5	6 교수님과 화상미팅	7	8	9 비상 회의 (제안서 마무리)
10 최종 제안서 제출 마감일	11 정기 회의 (설계 시작)	12	13	14 (예정)교수님과 화상미팅	15	16
17 설계 마무리	18 정기 회의 (코딩 준비)	19 코딩 시작	20	21 (예정)교수님과 화상미팅	22	23
24	25 정기 회의 (코딩 피드백)	26 프로젝트 중간 발표 준비	27	28 프로젝트 중간 발표	29	30 비상 회의 (코딩 피드백)
31						

- 5/11 ~ 5/17: 설계 기간(프로그램의 구조를 설계하며 효율적이고 현실적인 방향 탐구)
- 5/18 ~ 6/2: 전반적인 코딩 기간(각 기능에 대한 코딩을 중점적으로)

5월 | 6월

일	월	화	수	목	금	토
	1 정기 회의 (코딩 마무리)	2 코딩 완료	3 실사용 후 문제점 분석	4 디버깅 시작	5	6
7	8 비상 회의 (디버깅 피드백)	9	10	11	12	13 비상 회의 (디버깅 피드백)
14 디버깅 완료	15 최종 코딩 완료	16 프로젝트 최종 발표 준비	17	18 프로젝트 최종 발표	19	20
21	22	23	24	25	26	27
28	29	30				

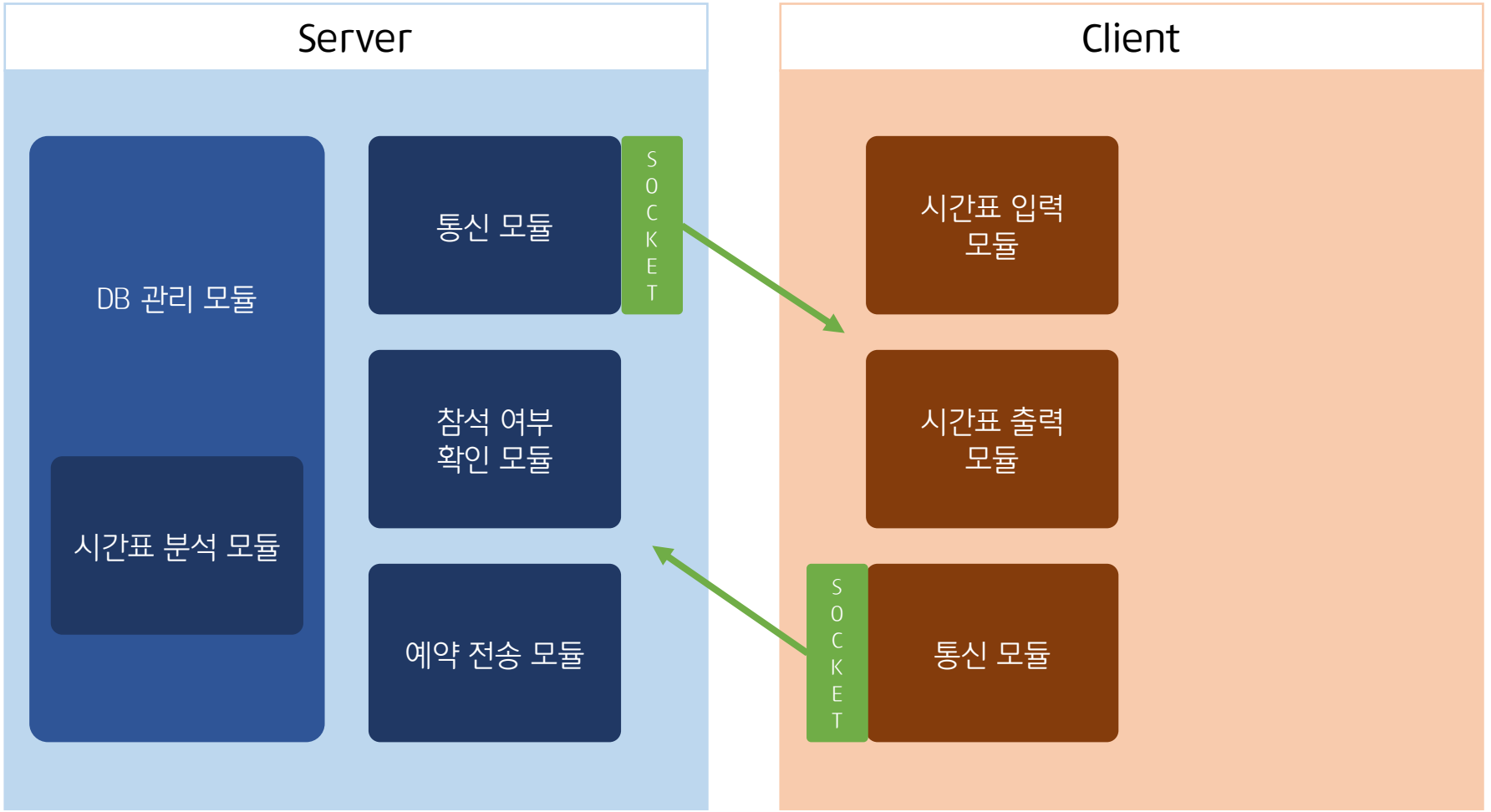
- 6/3 ~ 6/14: 디버깅 기간(가능들의 원활한 연결과 안정적인 통신을 중점적으로)
- 6/15: 코딩 최종 마무리(주석 검토를 중점적으로)

20-1 사이버보안기초프로젝트(01)

1팀 1차 중간보고

1971060 김서현
1971063 김윤서
1971066 김채연
1971090 한근영

시스템 구조



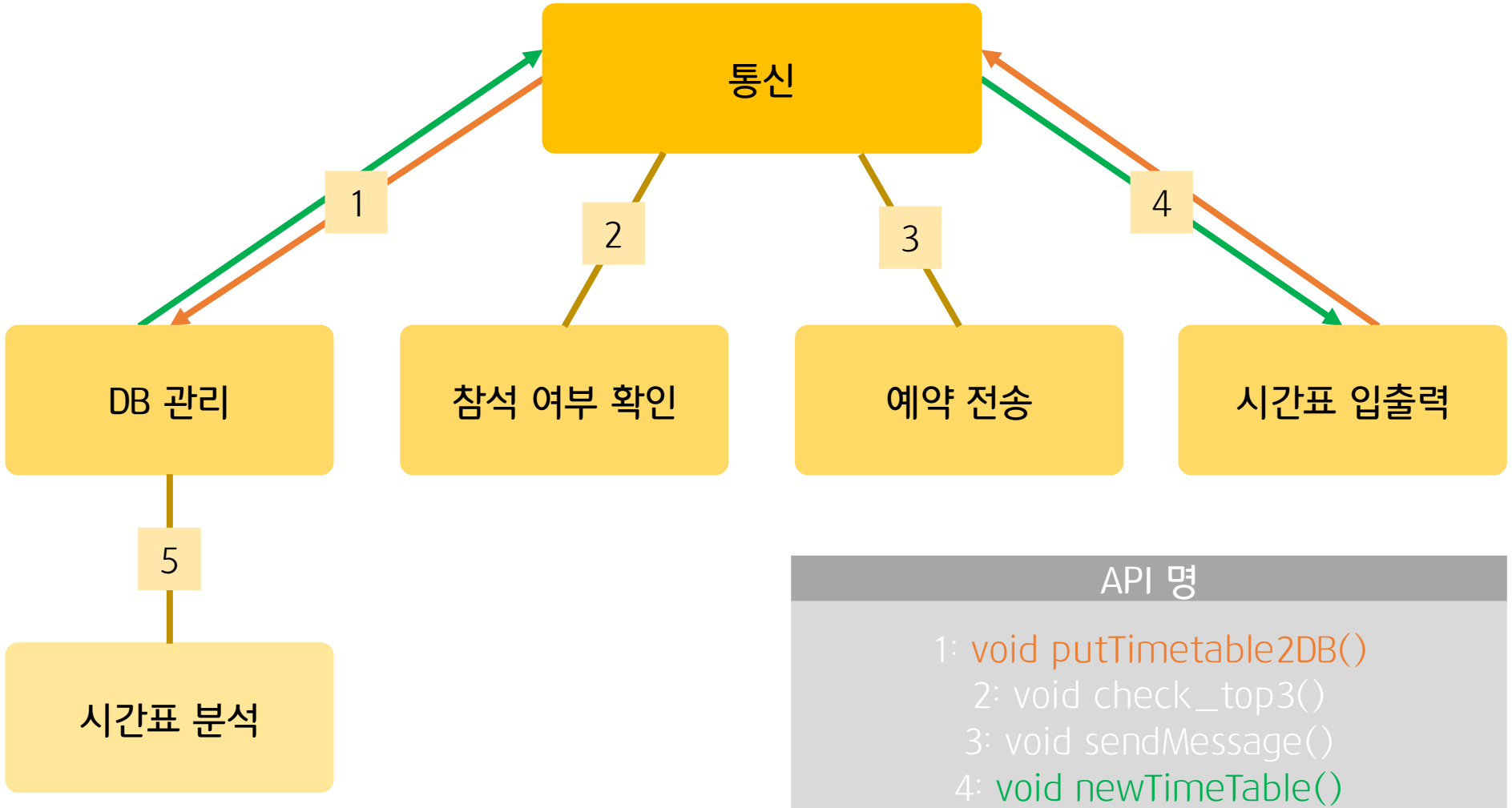
1 설계 내용

- 시스템 구조
- 모듈 간 인터페이스
- 작동 시나리오 점검

2 진행 현황

- 코딩 현황
- 개발 환경 구축
- 향후 계획

모듈 | 모듈 간 인터페이스



API 명

1: void putTimetable2DB()
2: void check_top3()
3: void sendMessage()
4: void newTimeTable()
5: void getResult()

1. void putTimetable2DB(): Client의 "시간표 입력" 모듈로부터 데이터 받아와 (통신 모듈 거쳐) Server의 DB관리 모듈에 저장
2. void check_top3(): "시간표분석" 모듈에서 top3정보 "참석여부"모듈로 이동
3. void sendMessage(): "시간표분석" 모듈에서 확정된 약속시간을 (통신모듈을 거쳐) "예약전송" 모듈로 전달
4. void newTimeTable(): "시간표분석" 모듈의 데이터를 (통신모듈 거쳐) "시간표출력" 모듈로 전달-<공강 인원 시간표>를 출력
5. void getResult(): "DB"모듈의 데이터를 "시간표분석" 모듈이 받아서 시간표 분석

구현 완료 | 구현 예정

1 사용자가 접속해서 자신의 이름을 입력

2 사용자 자신의 시간표 입력

3 사용자 자신의 시간표 출력

단계 별 구체적 작동 과정

1 사용자 화면에는 '접속되었습니다. '라는 문구가 뜨고
서버에는 '등록하겠습니다 [접속자 수 n]' 문구가 뜬다

2 사용자는 'A요일 class B 수업 여부 : '라는 문구가 뜨면 해당 질문에 대한 답변을
수업이 있으면 'o', 없으면 'x'로 입력한다

3 지정한 시간표 양식에 맞춰서 수업이 있으면 'o', 수업이 없으면 'x'가 출력된다

구현 완료 | 구현 예정

4 겹공강 분석 및 결과 출력

5 참석 여부 확인

6 예약 문자 발송

7 접속 종료

단계 별 구체적 작동 과정

4 메인 서버에서 사용자들의 시간표 분석 후,
시간대 별 참석 가능 인원 수를 시간표 양식에 맞게 출력

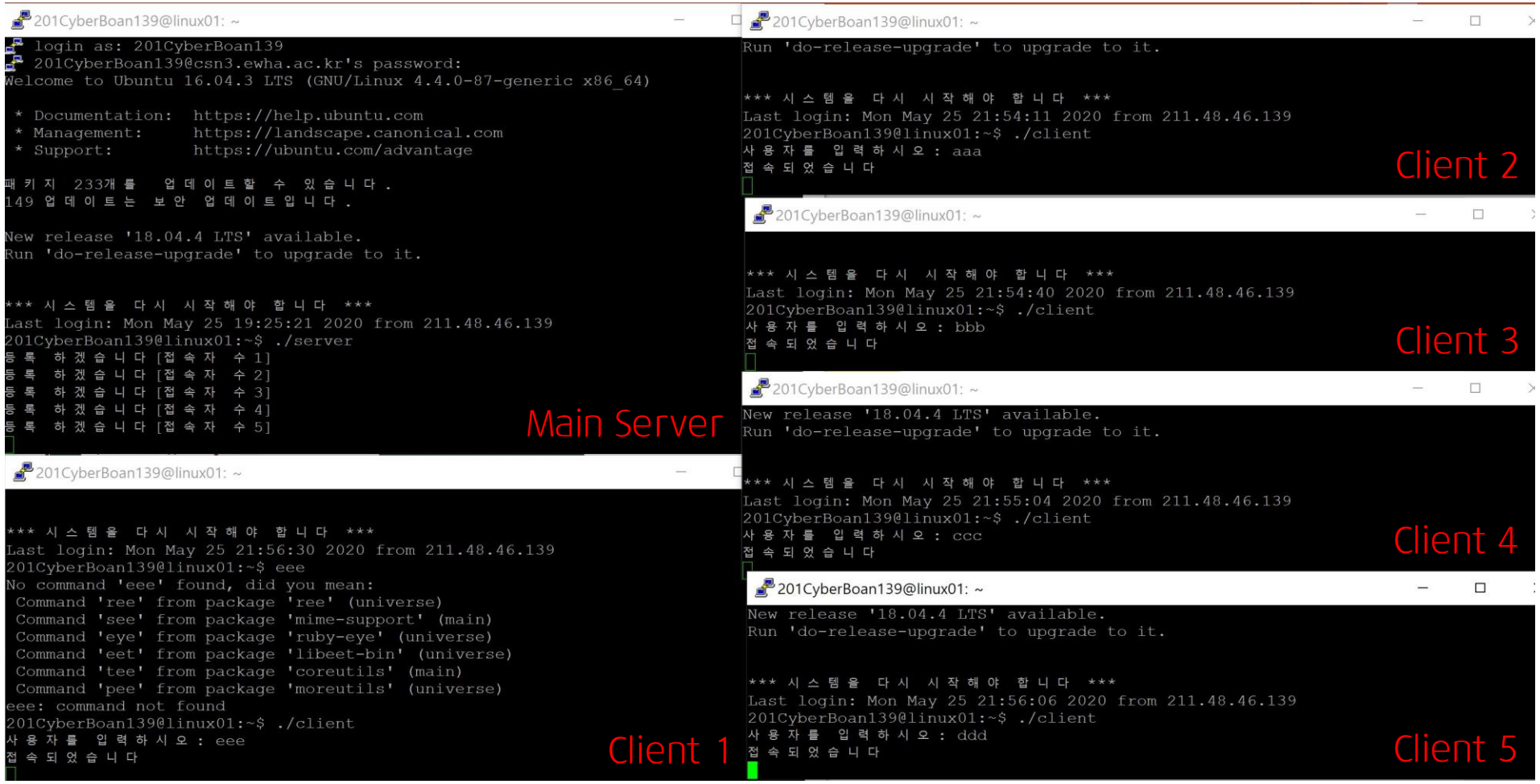
5 시간대 가장 많은 인원이 참석할 수 있는 상위 3개의 시간을 출력 후,
사용자들에게 참석 여부를 투표 받음

6 참석 여부에 따라 가장 많은 인원이 참석 가능한 시간을 약속 시간으로 지정하여
해당 시간을 사용자들에게 예약 문자로 전송

7 서버에는 '클라이언트 해체' 문구 출력

1 설계 내용
- 시스템 구조
- 모듈 간 인터페이스
- 작동 시나리오 점검

2 진행 현황
- 코딩 현황
- 개발 환경 구축
- 향후 계획



다중 접속 및 양방향 통신

메인 서버-클라이언트 연결 & 클라이언트 동시 접속(5명까지)

```c
if((ssock=socket(AF_INET,SOCK_STREAM,IPPROTO_TCP))<0)
{
        perror("socket error:");
        exit(1);
}

setsockopt(ssock,SOL_SOCKET,SO_REUSEADDR,&optval,sizeof(optval));

memset(add_num,0,sizeof(add_num));
memset(&server_addr,0,sizeof(server_addr));
server_addr.sin_family=AF_INET;
server_addr.sin_addr.s_addr=htonl(INADDR_ANY);
server_addr.sin_port=htons(40000);

int ch=bind(ssock,(struct sockaddr*)&server_addr,sizeof(server_addr));
if(ch<0)
{
        perror("bind error:");
        exit(1);
}

ch=listen(ssock,5);
if(ch<0)
{
        perror("listen error:");
        exit(1);
}
```

```c
if((csock=accept(ssock,(struct sockaddr*)&client_addr,&clen))<0)
{
        perror("accept error:");
        exit(1);
}
```

```c
data_len=read(fd,(struct message*)&read_message,sizeof(read_message));
if(data_len>0)
{
        writeMessage((void*)&read_message,(void*)add_num,fd,maxfd);

}
```

Main Server 주요 코드
통신 모듈 부분

```c
void writeMessage(void*client_message,void*num,int basefd, int maxfd)
{
        int index;
        struct message*cl_message;
        struct add_num *index_num;
//      char all[]="ALL";

        cl_message=(struct message*)client_message;
        index_num=(struct add_num*)num;

        if(strcmp(cl_message->sbuf,"123")==0)
        {
                printf("등 록  하 겠 습 니 다 [접 속 자  수 %d]\n",++C_Num);
                for(index=0;index<USER;index++)
                {
                        if(((index_num+index)->anum)==basefd)
                        {
                                strcpy((index_num+index)->name,(cl_message->us
                        }
                }

                write(basefd,greet,sizeof(greet));
        }
        else
        {
                printf("message from client:%s",(char*)(cl_message->sbuf));
        }
}
```

```c
if((ssock = socket(AF_INET, SOCK_STREAM, 0))<0){
        perror("socket error: ");
        exit(1);
}
memset(&server_addr, 0, sizeof(server_addr));
server_addr.sin_family = AF_INET;
server_addr.sin_addr.s_addr = inet_addr("203.255.176.225");
server_addr.sin_port = htons(40000);

clen = sizeof(server_addr);

printf("사 용 자 를  입 력 하 시 오 : ");
fgets(name, MAXBUF, stdin);


*(name+(strlen(name)-1))='\0';

if(connect(ssock, (struct sockaddr *)&server_addr, clen)<0){
        perror("connect error: ");
        return -1;
}
memset(&write_message, 0, sizeof(write_message));
strcpy(write_message.user, name);
strcpy(write_message.sbuf, "123");
if(write(ssock, (struct message*)&write_message, sizeof(write_message))<0){
        perror("write error: ");
        exit(1);
}

FD_ZERO(&read_fds);
FD_SET(ssock, &read_fds);
FD_SET(0, &read_fds);
```

(좌)Main Server 주요 코드와 (우)Client 주요 코드
통신 모듈 부분

```
printf("자신의 시간표를 입력해주세요 : ");
printf("해당 시간에 수업이 있으면 o(소문자) 입력\n");
printf("해당 시간에 수업이 없으면 x(소문자) 입력\n\n");
printf("%s %s 수업 여부 :",weekdays[0],classes[0]);

for(int i=0;i<5;i++)
{
        for(int j=0;j<7;j++)
        {
                while(1)
                {
                        scanf("%c",&check,1);
                        schedule[j][i]=check;

                        if(check=='o'||check=='x') break;
                        else printf("%s %s 수업 여부 : ",weekdays[i],classes[j]);
                }
        }
        printf("\n");
}

printf("       ");

for(int j=0;j<7;j++)
{
        printf("%s",classes[j]);
        printf("  ");
}
printf("\n");

for(int i=0;i<5;i++)
{
        printf("%s\t",weekdays[i]);
        for(int j=0;j<7;j++)
        {
                printf("%c\t",schedule[j][i]);
        }
        printf("\n");
}
```

수업이 있으면 o, 수업이 없으면 x로 입력

```
201CyberBoan139@linux01:~$ ./client
사용자를 입력하시오: aaa
접속되었습니다

자신의 시간표를 입력해주세요 : 해당 시간에 수업이 있으면 o(소문자) 입력
해당 시간에 수업이 없으면 x(소문자) 입력

Mon class1 수업 여부 :Mon class1 수업 여부 : o
Mon class2 수업 여부 : x
Mon class3 수업 여부 : o
Mon class4 수업 여부 : o
Mon class5 수업 여부 : o
Mon class6 수업 여부 : x
Mon class7 수업 여부 : x

Tue class1 수업 여부 : o
Tue class2 수업 여부 : o
Tue class3 수업 여부 : x
Tue class4 수업 여부 : x
Tue class5 수업 여부 : x
Tue class6 수업 여부 : o
Tue class7 수업 여부 : o

Wed class1 수업 여부 : o
Wed class2 수업 여부 : x
Wed class3 수업 여부 : x
Wed class4 수업 여부 : o
Wed class5 수업 여부 : x
Wed class6 수업 여부 : x
Wed class7 수업 여부 : x

Thu class1 수업 여부 : o
Thu class2 수업 여부 : o
Thu class3 수업 여부 : o
Thu class4 수업 여부 : x
Thu class5 수업 여부 : o
Thu class6 수업 여부 : x
Thu class7 수업 여부 : o

Fri class1 수업 여부 : x
Fri class2 수업 여부 : x
Fri class3 수업 여부 : o
Fri class4 수업 여부 : x
Fri class5 수업 여부 : o
Fri class6 수업 여부 : x
Fri class7 수업 여부 : x
```

출력 결과

	class1	class2	class3	class4	class5	class6	class7
Mon	o	x	o	o	o	x	x
Tue	o	o	x	x	x	o	o
Wed	o	x	x	o	x	x	x
Thu	o	o	o	x	o	x	o
Fri	x	x	o	x	o	x	x

시간표 입력 및 출력 및 공강 분석

	class1	class2	class3	class4	class5	class6	class7
Mon	o	o	o	o	o	x	x
Tue	o	x	x	x	x	o	x
Wed	o	x	x	o	o	o	x
Thu	x	o	o	x	x	o	o
Fri	x	x	o	o	o	o	x

Client 1

	class1	class2	class3	class4	class5	class6	class7
Mon	o	x	x	x	x	o	o
Tue	o	o	o	x	o	x	x
Wed	x	o	o	o	o	x	x
Thu	o	x	o	x	o	o	o
Fri	x	o	x	o	x	x	x

Client 2

	class1	class2	class3	class4	class5	class6	class7
Mon	o	x	x	x	o	x	o
Tue	x	o	x	o	o	o	x
Wed	x	o	x	o	x	o	x
Thu	o	x	o	x	x	o	o
Fri	x	x	o	o	o	o	o

Client 3

	class1	class2	class3	class4	class5	class6	class7
Mon	0	2	2	2	1	2	1
Tue	1	1	2	2	1	1	3
Wed	2	1	2	0	1	1	3
Thu	1	2	0	3	2	0	0
Fri	3	2	1	0	1	1	2

출력 결과

겹공강 분석

1 설계 내용
- 시스템 구조
- 모듈 간 인터페이스
- 작동 시나리오 점검

2 진행 현황
- 코딩 현황
- 개발 환경 구축
- 향후 계획

개발 환경

- Putty에서 C언어로 모든 코딩 파일을 만들었음
- 메인 서버와 관련된 부분은 server.c파일에, 클라이언트와 관련된 부분은 client.c파일에 vi편집기를 이용해서 구현
- c파일을 gcc를 이용해서 각각 server와 client라는 컴파일 파일로 만들었고
- 하나의 server와 여러 개의 client를 동시에 실행시켜서 양방향 통신 및 데이터 주고 받는 부분을 확인
- 팀원들 각자가 만든 파일들을 tmp 디렉토리에 저장하여 함께 공유하고 수정해 나가는 방식으로 개발

1 설계 내용

- 시스템 구조
- 모듈 간 인터페이스
- 작동 시나리오 점검

2 진행 현황

- 코딩 현황
- 개발 환경 구축
- 향후 계획

향후 계획

기능
내 시간표 입력 및 출력
겹공강 분석 및 출력
약속 참석 여부 확인
약속 시간 예약 전송

0단계	설계 -모듈/인터페이스 정의 및 시스템 구축
1단계	통신 모듈 구축 - 사용자 이름 입력 받기(접속 확인용)* - 양방향 통신 - 다중 접속 확인
2단계	시간표 입/출력 모듈 구축 -시간표 o/x로 입력 받기 -입력 받은 시간표를 가시적으로 출력 -시간표 공강을 분석하여 해당 시간에 공강인 사람 수 출력*
3단계	참석 여부 확인 모듈 구축 -top3출력 후 참석 여부 사용자로부터 받기 -사용자의 대답을 분석하여 최종 공강 시간 정하기
4단계	예약 모듈 구축 -at을 사용하여 구축(예정)
추가 구현	-시간표 공강 분석 결과 출력에 사람 수 뿐만 아니라 해당되는 사용자의 이름까지 출력 -채팅 기능

20-1 사이버보안기초프로젝트(01)

1팀 2차 중간점검

1971060 김서현
1971063 김윤서
1971066 김채연
1971090 한근영

기능	특징
내 시간표 입력 및 출력	• 사용자들의 시간표를 입력 받은 후 확인을 위해 출력 • 추가 가능) 출력 결과가 틀릴 경우, 출력이 옳을 때까지 입력을 받을 수 있도록 모든 단계를 완료 후 가능하면 구현할 예정
겹공강 분석 및 출력	• 사용자들의 겹공강 시간을 파악 후, 해당 시간에 공강인 사람의 수를 시간표 양식에 맞춰서 출력
약속 시간 투표 및 확정	• CASE 1) 가장 많은 사람이 참석 가능한 시간을 출력 • CASE 2) 가장 많은 사람이 참석 가능한 시간이 다수일 경우, 해당 시간을 전부 출력 후 투표를 받아서 약속 시간 선정
약속 시간 예약 전송	• 최종 결정된 약속 시간을 사용자들에게 예상 참석 인원 수와 함께 모든 사용자들에게 각각 알림

1
프로그램 접속
및 이름 입력

2
시간표 입력

3
입력 시간표 확인

단계 별 구체적 시나리오 설명

1 프로그램에 접속하면 '사용자를 입력하시오 : '라는 문구가 뜬다.
이름을 입력하면 프로그램에 접속이 되고, '접속 되었습니다.'라는 문구가 뜬다.

2 사용자는 'A요일 class B 수업 여부 : '라는 문구가 뜨면
해당 질문에 대한 답변을 수업이 있으면 'o', 없으면 'x'로 입력한다

3 지정한 시간표 양식에 맞춰서 사용자의 입력에 따라
수업이 있으면 'o', 수업이 없으면 'x'가 출력된다.

4 겹공강 결과 출력

5 참석 시간 결정 및 최종 확인

6 예약 문자 확인

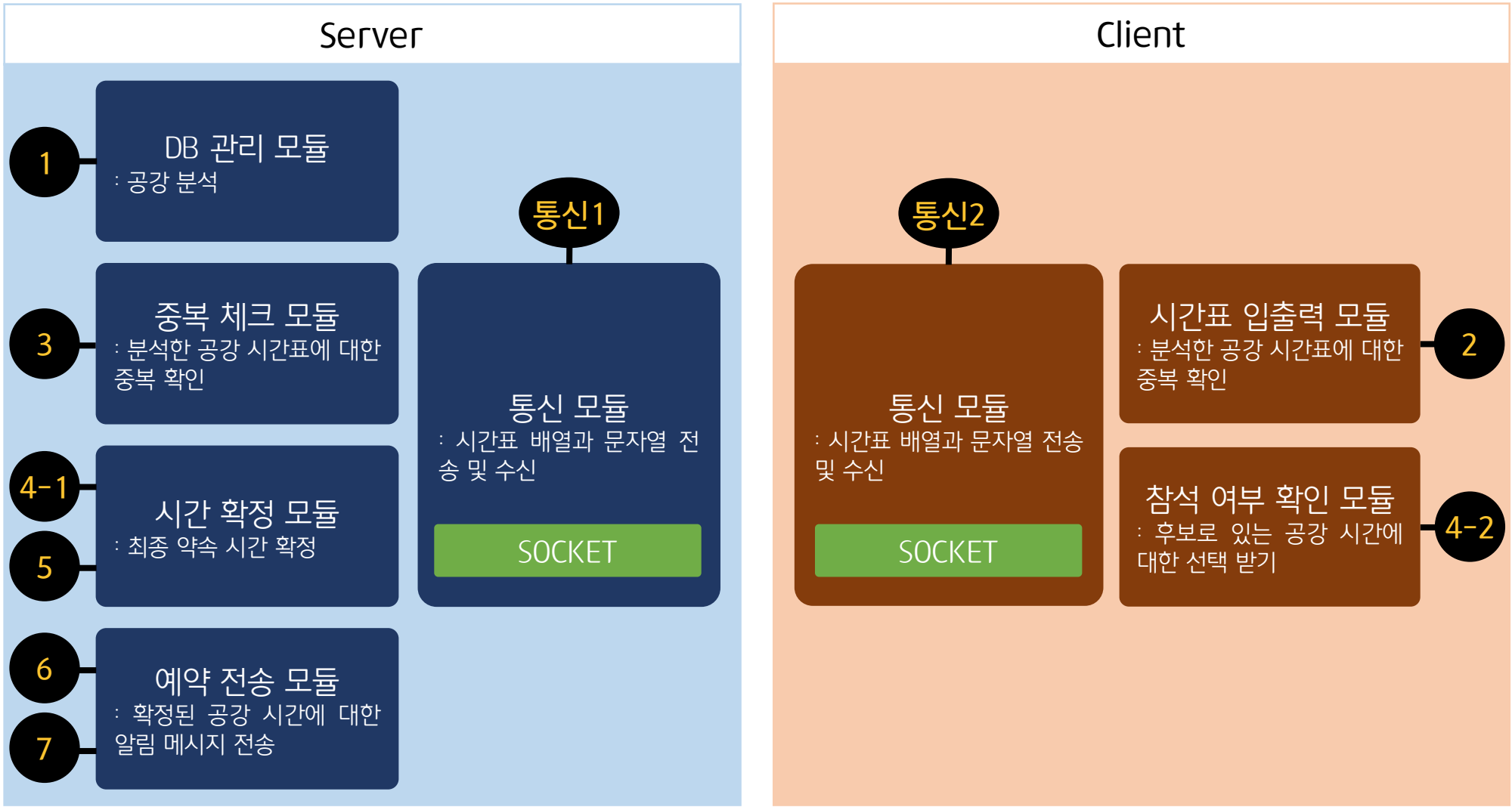
단계 별 구체적 시나리오 설명

4 모든 사용자들이 앞의 과정을 완료한 후,
겹공강 분석 결과가 겹공강 인원에 따라 시간표 양식에 맞춰서 출력된다.

5 CASE 1) 겹공강 인원 수가 가장 많은 시간이 하나일 경우
해당 시간이 약속 시간으로 출력된다.
CASE 2) 겹공강 인원 수가 가장 많은 시간이 다수일 경우
해당 시간들이 출력되면 가장 선호하는 시간을 입력한다. 그 후, 모든 사용자들이 가장 선호하는 시간이 약속시간으로 출력된다.

6 예약된 문자를 전송 받으면 예상 참석 인원 수와 약속 시간을 확인한다.

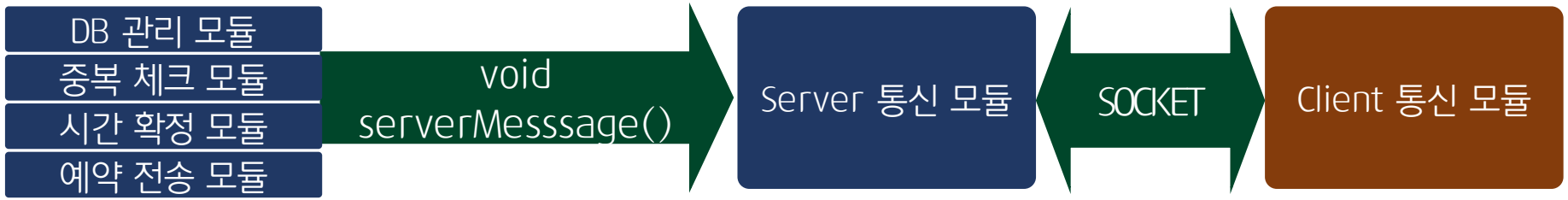
Static Diagram | API reference | Dynamic Diagram



통신1) void serverMessage(void*client_message, void*num, int basefd, int maxfd)

: server쪽에서 제공하는 API로 server 통신 모듈과 client의 통신 모듈 간 소켓 통신을 제공하는 인터페이스이다. 또한 DB 관리/중복체크/시간확정/ 예약전송 모듈에서 클라이언트 쪽으로 전송하기 위한 데이터를 받는 일도 수행한다.

- *void*client_message:* 수신 데이터가 저장된 버퍼
- *void*num:* client를 등록하기 위한 숫자
- *int basefd:* 소켓 번호
- *int maxfd:* 클라이언트 다중 연결 최대 개수



통신2) void clientMessage(void*client_message, void*num, int basefd)

: client쪽에서 제공하는 API로 server 통신 모듈과 client의 통신 모듈 간 소켓 통신을 제공하는 인터페이스이다. 또한 시간표 입출력/참석여부 모듈에서 서버 쪽으로 전송하기 위한 데이터를 받는 일도 수행한다.

- *void*client_message:* 수신 데이터가 저장된 버퍼
- *void*num: server*를 등록하기 위한 숫자
- *int basefd:* 소켓 번호

1) char** DBManage(char* allTimeTable, int C_Num)

: 서버의 DB 관리 모듈에서 제공하는 API로, 통신 모듈로부터 사용자가 입력한 시간표를 받아 2차원 배열로 저장하는 일을 수행한다. 반환 값은 모든 시간표를 저장한 2차원 배열에 대한 이중 포인터이다.

- *char* allTimeTable:* client로부터 받은 시간표를 저장한 배열에 대한 포인터
- *int C_Num:* server에 접속된 client의 수

2) int* IO_timetable(int* timetable_Free)

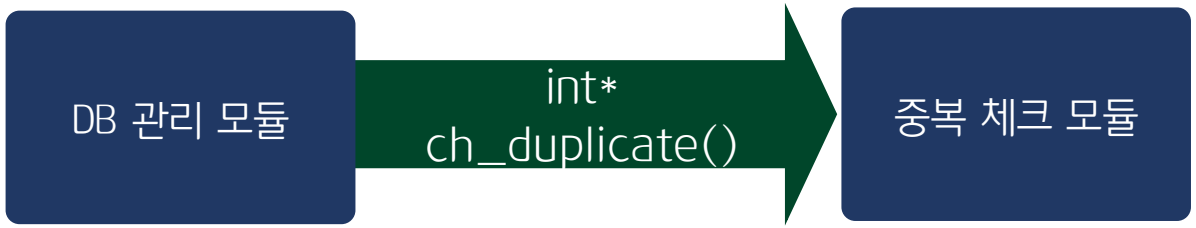
: 시간표 입출력 모듈에서 제공하는 AIPI로, 통신 모듈로부터 DB관리 모듈에서 분석한 겹공강 시간표를 받아오는 역할을 한다.

- *int* timetable_Free:* 겹공강 시간표를 저장한 배열에 대한 포인터

3) int* ch_duplicate(int* timeTable_Free)

: 중복 체크 모듈에서 제공하는 API로, DB관리 모듈로부터 정수 배열의 겹공강 시간표를 인자로 받아 오는 역할을 한다.

- *int* timeTable_Free:* 공강 분석 시간표에 대한 포인터



4-1) struct finalDay no_duplicate(int* num)

: 중복되는 시간이 없는 경우, 중복 체크 모듈로부터 최종 약속 정보를 수신 받는다.

- int* num: 최종 확정된 공강 시간에 대한 정보가 담긴 배열에 대한 포인터

```
struct finalDay {
        int weekday; //최종적으로 결정된 공강 시간의 요일
        int class; //최종적으로 결정된 공강 시간의 교시
}
```

4-2) int selectDay(int* choose)

: 중복되는 시간이 있는 경우, client의 통신 모듈로부터 중복시간에 대한 client의 답변 데이터를 수신 받는다. Client들에게 1차적으로 결정된 중복되는 공강 시간들에 대한 의견을 물어보고 각각의 클라이언트들에 대한 답변을 반환한다.

- *int* choose:* 1차적으로 정해진 중복되는 공강 시간들을 저장한 정수형 1차원 배열에 대한 포인터

5) struct finalDay result(int* selectDay)

: 시간확정 모듈에서 정의하는 API로 통신 모듈에서 server의 시간 확정 모듈로 참석여부 투표 받은 결과를 수신 받는다. 최종적으로 공강 시간을 결정하여 구조체인 finalDay로 반환한다.

- *int* selectDay:* 중복되는 공강 시간들에 대한 client들의 답변들을 저장한 정수형 배열에 대한 포인터

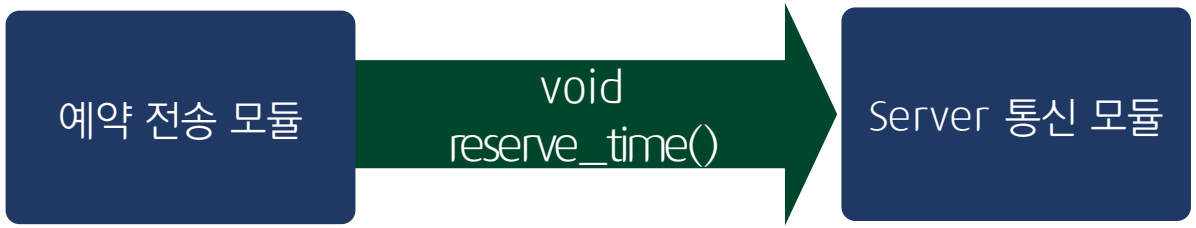
6) int* inputFinalTime(struct finalDay f)

: 시간 확정 모듈에서 최종적으로 확정한 시간에 대한 정보를 예약 전송 모듈로 수신 받을 수 있도록 하는 api로 최종적인 시간 및 참여 인원수 등에 대한 최종 데이터를 반환한다.

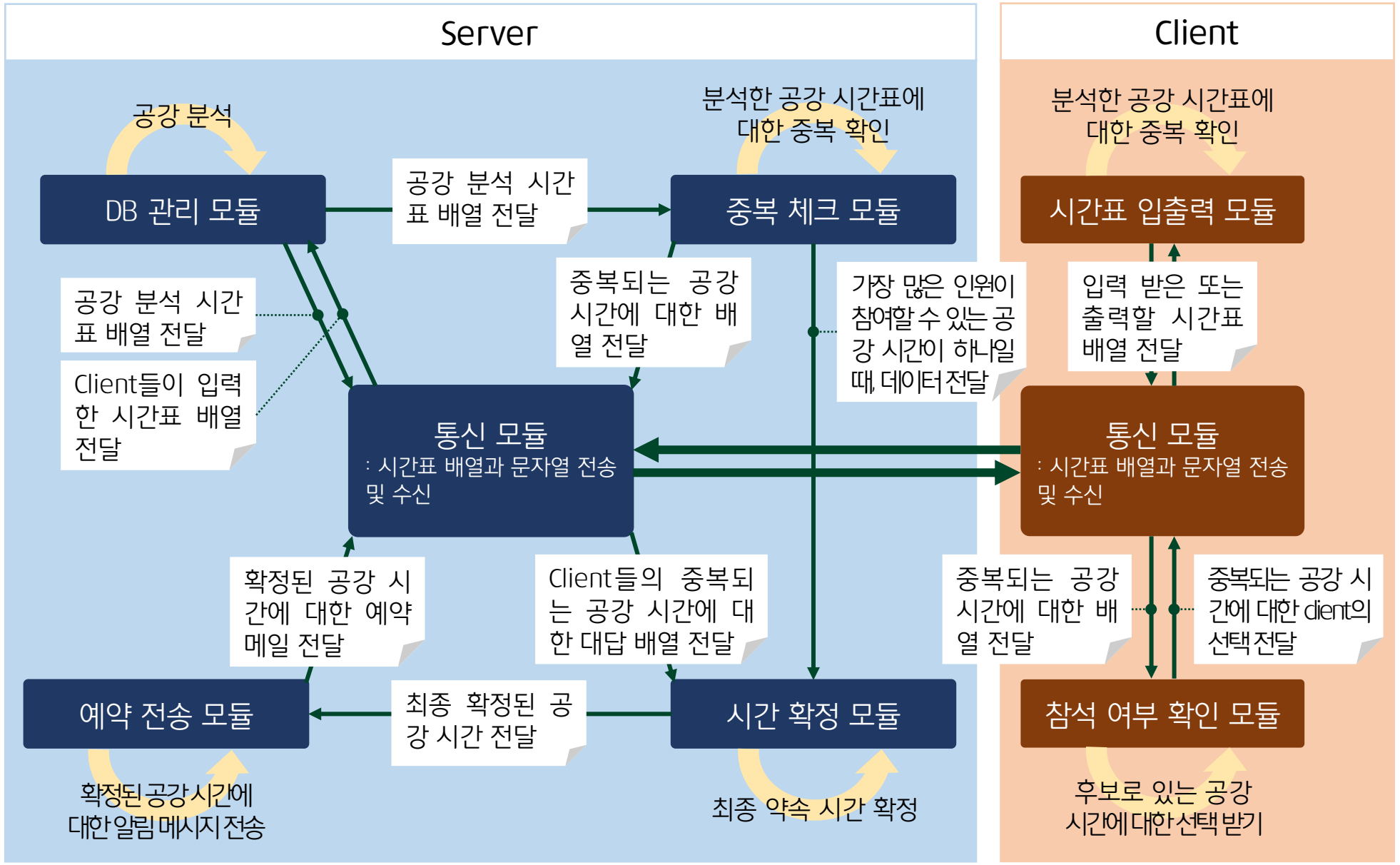
- Int* inputFinalTime: 예약메시지에 포함할 최종 데이터에 대한 정수형 배열 포인터

7) void reserve_time(int* finalTime)

: 예약전송 모듈에서 정의하는 API이다. 예약 전송 데이터(시간, 참여 인원수 등)를 통신 모듈로 보내준다. 최종적으로 정해진 시간에 대한 예약 전송을 클라이언트들에게 보낼 수 있도록 한다.



Static Diagram | API reference | **Dynamic Diagram**



※ 표현 '중복되는 공강 시간'이란 가장 많은 사람이 참석 가능한 공강 시간이 여럿일 때 이 시간들을 의미하며, 이중 하나를 고르는 것을 '선택'이라고 표현

전체 | 모듈

목표 코딩 60% 도달

Server

DB 관리 모듈
: 공강 분석

중복 체크 모듈
: 분석한 공강 시간표에 대한 중복 확인

시간 확정 모듈
: 최종 약속 시간 확정

예약 전송 모듈
: 확정된 공강 시간에 대한 알림 메시지 전송

통신 모듈
: 시간표 배열과 문자열 전송 및 수신

Client

통신 모듈
: 시간표 배열과 문자열 전송 및 수신

시간표 입출력 모듈
: 분석한 공강 시간표에 대한 중복 확인

참석 여부 확인 모듈
: 후보로 있는 공강 시간에 대한 선택 받기

모듈 별 목표 코딩 도달 현황

모듈 별 코딩의 난이도가 다르기 때문에 각 모듈을 전체 코딩 진행 현황 계산 시 같은 비율로 계산하지 않음

Server

- DB 관리 모듈
  : 공강 분석
  100%
- 중복 체크 모듈
  : 분석한 공강 시간표에 대한 중복 확인
  40%
- 시간 확정 모듈
  : 최종 약속 시간 확정
  0%
- 예약 전송 모듈
  : 확정된 공강 시간에 대한 알림 메시지 전송
  0%
- 통신 모듈
  : 시간표 배열과 문자열 전송 및 수신
  70%

Client

- 통신 모듈
  : 시간표 배열과 문자열 전송 및 수신
  70%
- 시간표 입출력 모듈
  : 분석한 공강 시간표에 대한 중복 확인
  100%
- 참석 여부 확인 모듈
  : 후보로 있는 공강 시간에 대한 선택 받기
  0%

실행순서	case	체크리스트	결과
1. 접속 및 사용자 정보 입력	이름 입력 후	“사용자의 이름을 입력하세요”가 뜨는가?	0
		“접속되었습니다” 안내 메시지가 뜨는가?	0
2. 시간표 입력	시간표 입력 시작	시간표 입력 질문이 잘 뜨는가?	0
	0,X로 입력	시간표를 월 1교시부터 금 7교시까지 0,X 로 정상적으로 입력받는가?	0
	그 외 입력	사용자가 0,X 로 정상적으로 입력할 때까지 다시 입력 받는가?	0
3. 시간표 확인	입력 끝난 후	입력한 시간표가 정리된 형태로 뜨는가?	0
4. 겹공강 시간표 출력	공강 분석 후	겹공강 분석 후, 시간별로 공강인 인원수를 표시하는 시간표가 오류 없이 잘 뜨는가?	△

※ ‘△’는 Visual Studio에서는 출력이 되지만 PuTTY에서는 출력이 되지 않는 부분임

실행순서	case	체크리스트	결과
5. 중복 체크	n명이 가능한 시간이 하나일 경우 (중복이 없는 경우)	투표를 생략하고, 시간 확정 단계로 넘어가는가?	△
	n명이 가능한 시간이 여러 개일 경우 (중복이 있을 경우)	공강 후보들이 번호순으로 뜨는가?	△
		공강 후보 중 사용자가 선택한 대로 답변을 입력 받는가?	
		투표 결과가 반반일 경우, 안내 메시지와 함께 프로그램이 랜덤으로 시간을 확정해 주는가?	
6. 시간 확정	투표 끝난 후	투표 결과에 따라 공강 시간을 정상적으로 확정하는가?	
	최종 약속시간 출력	최종 확정된 공강 시간이 출력되는가?	
7. 예약 전송		확정한 시간에 맞춰 메시지가 수신되는가?	
		메시지 내용에 [확정한 시간, 참석가능한 인원 수] 정보가 표시되는가?	
		참석 불가능한 클라이언트에게도 정상적으로 메시지가 전송되는가?	

※ '△'는 Visual Studio에서는 출력이 되지만 PuTTY에서는 출력이 되지 않는 부분임

기능
내 시간표 입력 및 출력
겹공강 분석 및 출력
약속 시간 투표 및 확정
약속 시간 예약 전송

0단계	설계 -모듈/인터페이스 정의 및 시스템 구축
1단계	통신 모듈 구축 - 사용자 이름 입력 받기(접속 확인용)* - 양방향 통신 - 다중 접속 확인 완료
2단계	시간표 입/출력 모듈 구축 -시간표 o/x로 입력 받기 -입력 받은 시간표를 가시적으로 출력 -시간표 공강을 분석하여 해당 시간에 공강인 사람 수 출력*
3단계	참석 여부 확인 모듈 구축 -top3출력 후 참석 여부 사용자로부터 받기 -사용자의 대답을 분석하여 최종 공강 시간 정하기
4단계	예약 모듈 구축 -at을 사용하여 구축(예정)
추가 구현	- 시간표 공강 분석 결과 출력에 사람 수 뿐만 아니라 해당되는 사용자의 이름까지 출력 - 채팅 기능 - 시간표 잘못 입력 시 바르게 다시 입력 가능한 확인 및 재입력 기능