

STUDY WITH ME!

스터디 채팅 프로그램

4조

우효진, 강유진, 배광민, 강대준, 김강규, 박규영

CONTENTS

01. 프로젝트 시나리오

- 기획동기
- 벤치마킹

02. 프로젝트 기획

- 브레인스토밍
- 사업성
- 요구분석
- 프로토 타입
- 개발 환경

03. 프로젝트 설계

- 유스케이스 다이어그램
- 시퀀스 다이어그램
- DB 구조

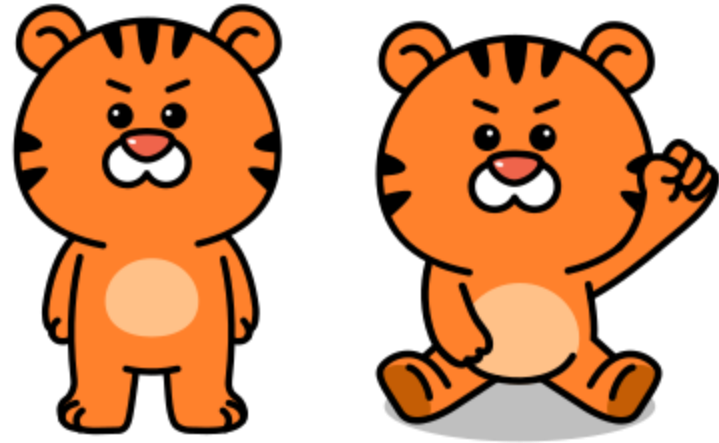
04. 프로젝트 UI 설계

05. 프로젝트 구현

- 시연 영상
- 중요 소스코드 기술

06. 프로젝트 후기

팀원 소개 및 역할

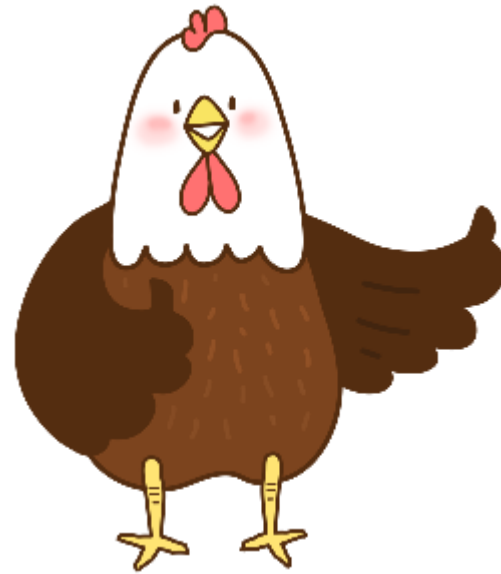


강유진

우효진

서버 & 클라이언트

- 오류사항(채팅창 목록 유지, 채팅기록 저장, 파일보내기, 강퇴버튼)들 수정
- 유저들이 참가했던 스터디 채팅방 필터 개발(내 방만)
- 스터디 채팅방 검색 버튼 이벤트, 프레임 개발
- 방장이 채팅방에 입장했을때 사용자 프레임 새로고침 개발



김강규

버튼 이벤트

- 버튼기능 구현 프레임에서 textField의값을 가져와서 버튼
입력시 프로토콜 + line(내용) 을 서버로 보내기 기능 개발



박규영

클라이언트

- 비밀번호 찾기 페이지 개발
- 채팅방 입장시 입장전에 있었던 채팅내용을 불러오기 개발
- 대기방에서 방 필터 개발

팀원 소개 및 역할



배광
민

프레임 설계

- 전체 프레임 개발
- 버튼 이벤트 개발
- 서버 & 클라이언트 오류 수정 및 개발
- MVC 패턴으로 변경

PM(프로그램 매니저)

- 초기 프로토타입 프로그램 설계
- 화면공유기능 구현
- 추가기능 조사=>구현
- 불필요한 작업 최소화
- DB설계



강대준

01. 프로젝트 시나리오

-기획동기

학원을 다니든 혹은 독학으로 공부를 하던, 스터디가 필요하다고 느껴서 스터디 그룹을 만들어야겠다는 생각이 들었다.

우리가 너무나도 당연하게 쓰는 카카오톡 어플을 써보았지만 카카오톡 어플은 그룹별로 나눌 수가 없어 여러 명의 사람들이 있을 경우 방이 헛갈릴 경우가 너무 많다고 느꼈다.

01. 프로젝트 시나리오

-기획동기

프로젝트의 이름은 study with me라는 이름이다.

study with me라는 이름에 맞게 스터디 그룹을 만들어 찾고 그 안에 스터디 그룹들을 분류하여 채팅방을 만들어 부족한 점을 공부하고 공유하여 알아듣기 쉽게 서로 대화해 가며 채팅하는 프로그램을 만들었다.

다른 채팅 프로그램들과는 다르게 자유게시판을 통 모든 사용자가 공유할 수 있는 공간을 따로 만들어 두었고, 개인정보 수정을 통해 편의성을 높였다.

01. 프로젝트 시나리오

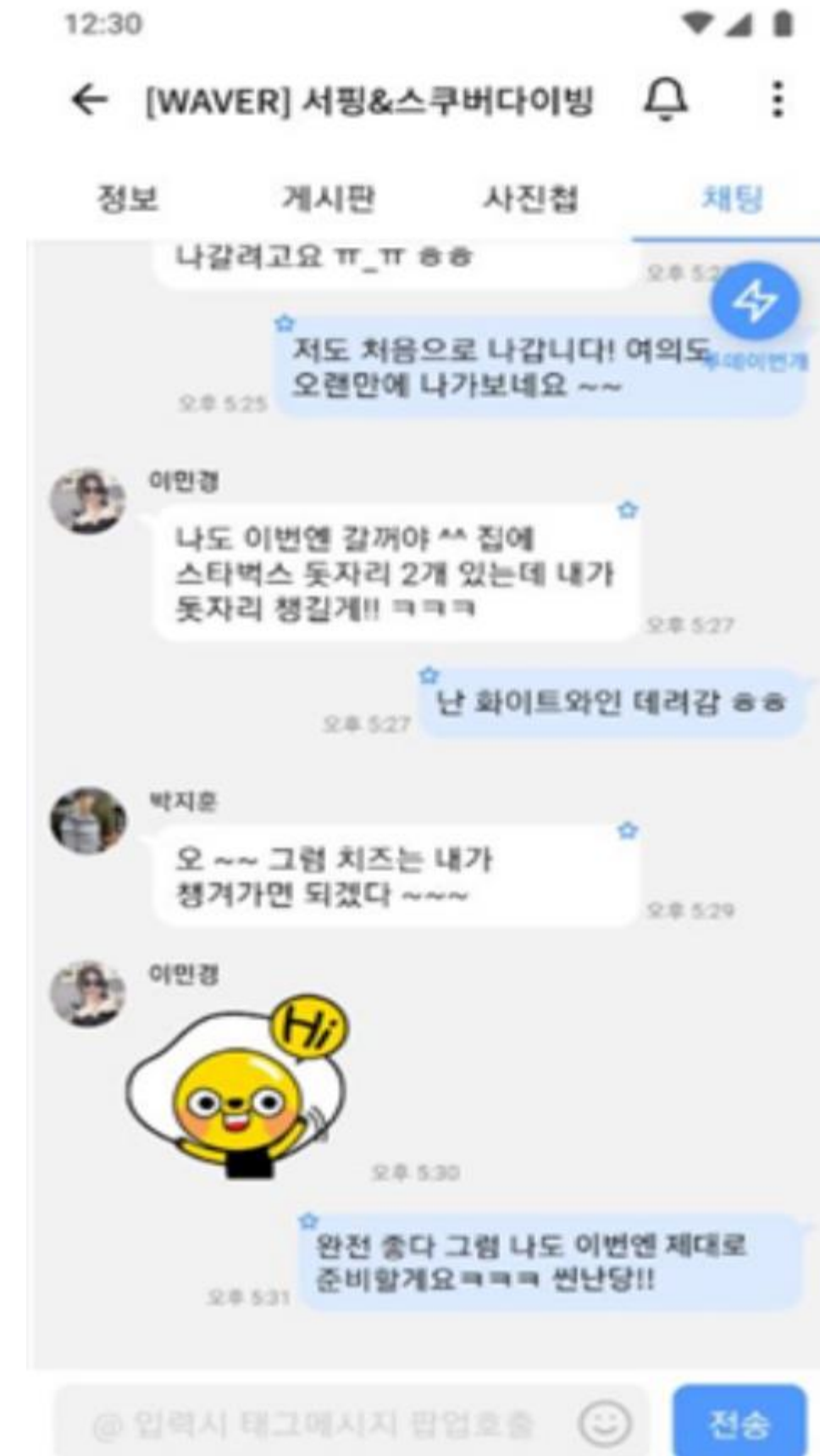
-벤치마킹

[우리동네 소모임] 어플

[우리동네 소모임 채팅방] -> [채팅방 내부]

- 채팅방에 보냈던 링크나 정보들을 따로 모아두는 정보칸이나 여러 가지 글을 게시하거나 약속을 정할 수 있는 투표를 할 수 있는 게시판이 있음

- 채팅방에서 보냈던 사진들을 모아두는 사진첩 기능이 있음



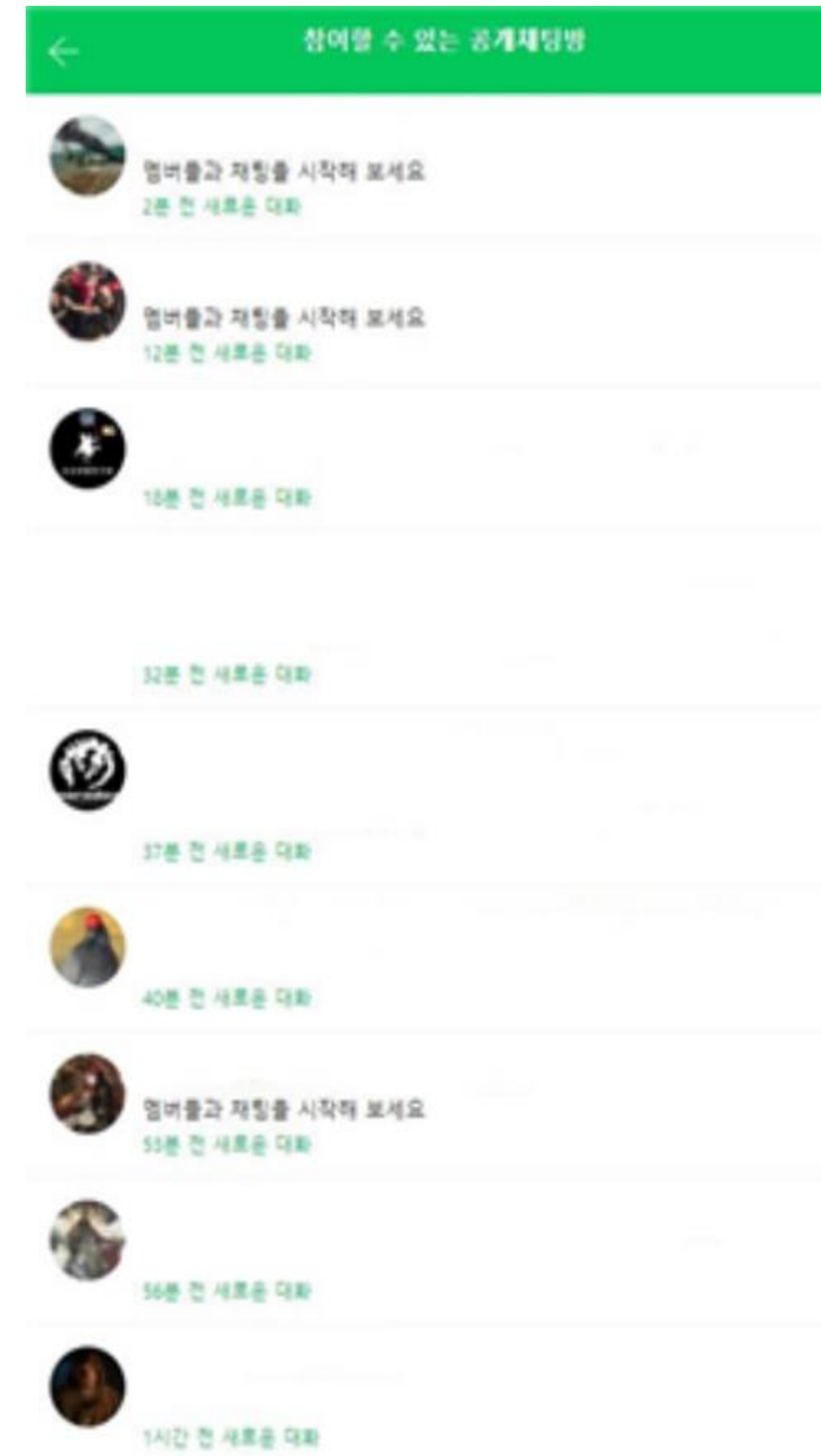
01. 프로젝트 시나리오

-벤치마킹

[네이버 카페]

[네이버카페 오픈채팅방] -> [채팅방 리스트 표시]

- 채팅방을 만들거나 지웠을 때 보여지는 채팅방 리스트들이 보여짐
- 네이버 카페 오픈 채팅방들은 방을 만들거나 지웠을 때 채팅방 리스트에 추가되거나 삭제됨



01. 프로젝트 시나리오

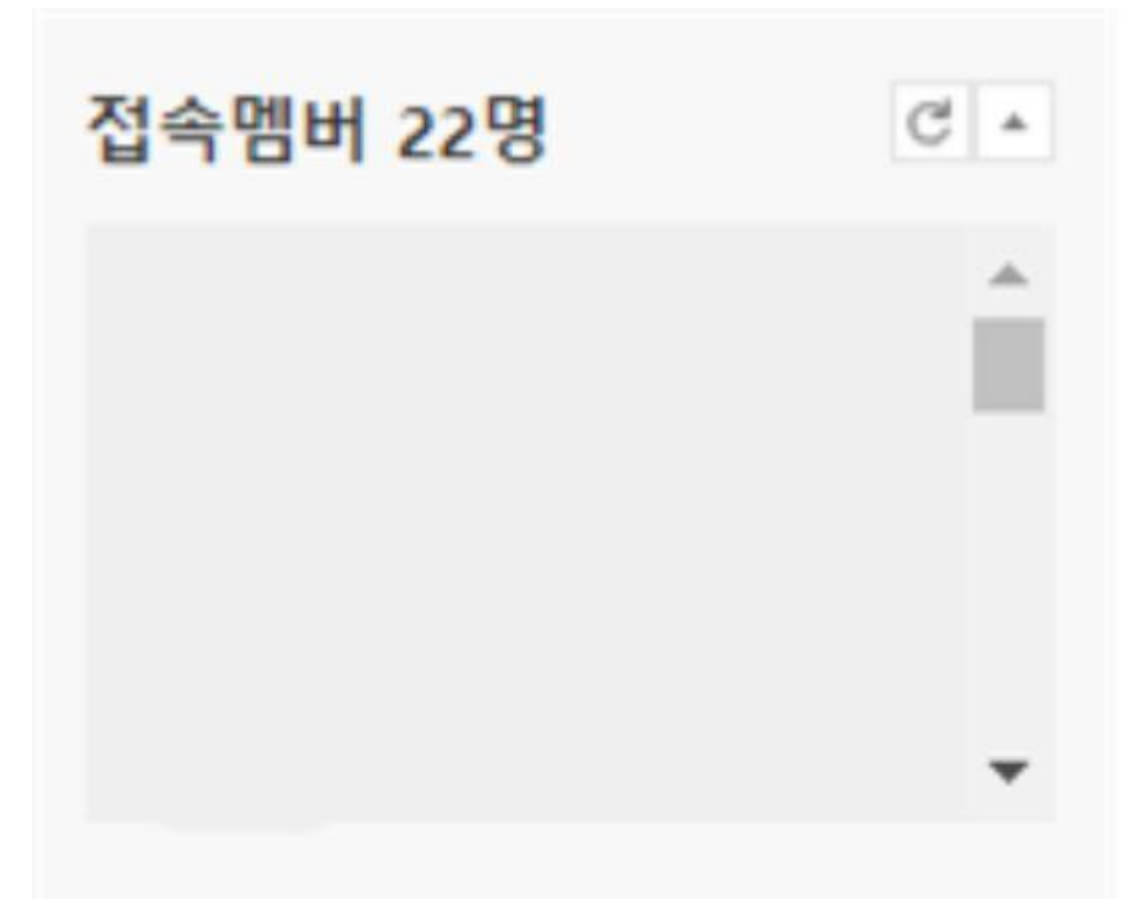
-벤치마킹

[네이버 카페]

[네이버카페 현재 접속자수 표시]

-> [대기방 접속유저 표시 / 채팅방 접속유저 표시]

- 네이버 카페에는 해당 카페에 현재 접속되어있는 카페 멤버가 표시됨



02. 프로젝트 기획

- 브레인스토밍



02. 프로젝트 기획

-사업성

해당 프로젝트로 제작한 프로그램에 대한 미래가능성이 있는
사업성을 작성하였습니다.

1 광고비

프로그램상에 광고칸을
추가하여서 광고를 이용한
수입을 창출하여 부가적인
수입을 창출 가능

2 화질개선

추가금액 결제시 화면 공유 중
기본 화질이 아닌 고화질 기능
등 추가적인 기능 추가로
인한 수입 창출 가능

3 광고제거

광고를 제거하고 싶어하는
사용자분들을 위한 광고제
거 기능을 추가하여 수입
창출 가능

02. 프로젝트 기획

-요구분석

서비스	필요 기능	기능 설명
회원가입	이메일 회원가입	메일 인증 절차 필요, 아이디,비밀번호, 생년월일, 핸드폰번호 기입
	이메일 인증	회원가입시 입력된 메일주소로 인증번호 발송
	아이디 중복	DB에 저장된 아이디와 중복된 아이디가 있는지 검사
로그인	로그인 성공	대기방에서 스터디 채팅방 생성 및 방 입장 가능
대기방	대기방 채팅	대기자 채팅방에서 대기자들과 채팅
	대기방 인원확인	대기실 인원과 현재 대기방에 있는 사용자 확인
	대기자 친구추가	대기자 목록에서 원하는 사용자를 친구추가

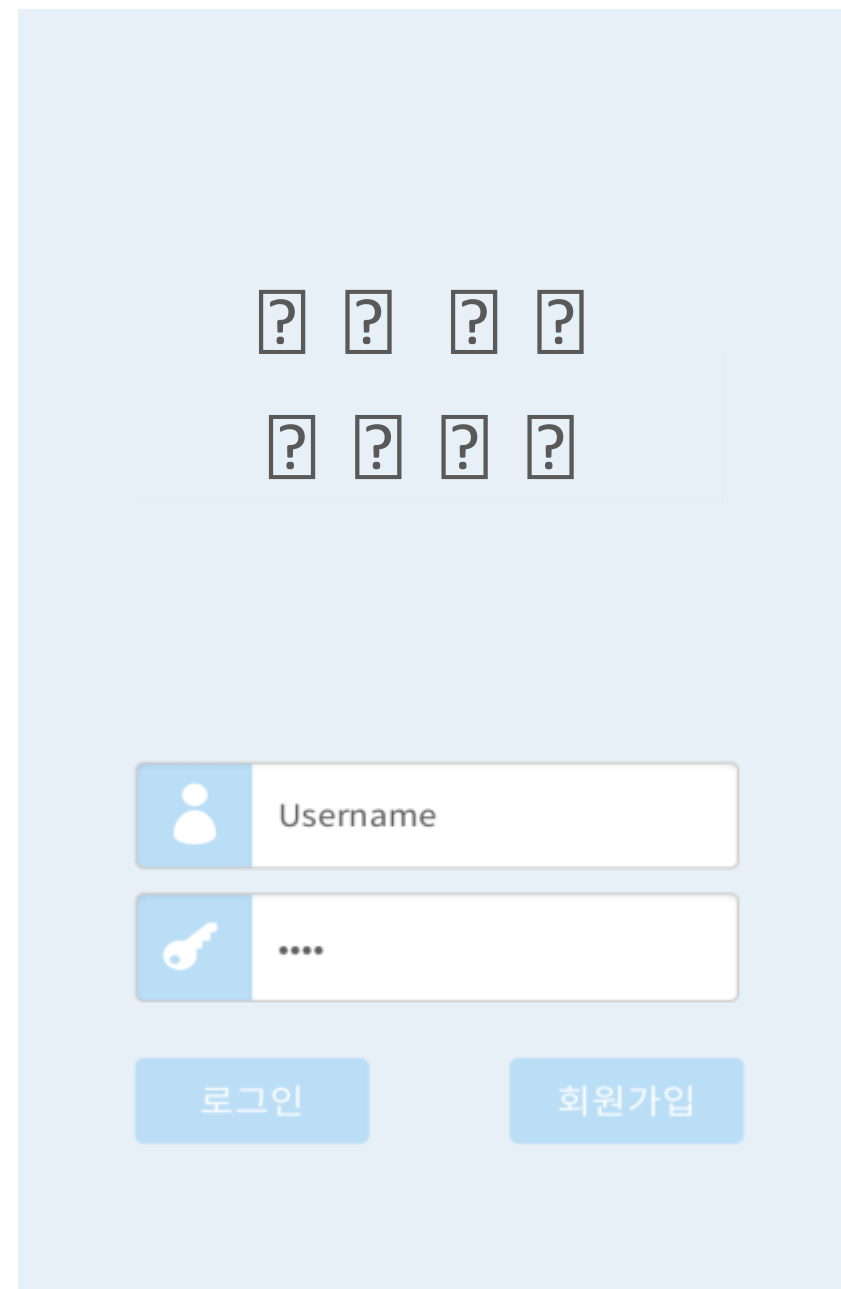
02. 프로젝트 기획

-요구분석

서비스	필요 기능	기능 설명
스터디 채팅방	방만들기	방 만들기를 통해 스터디 채팅방을 생성
	방검색	자신이 원하는 스터디 채팅방을 찾을 수 있게 원하는 방이름 검색
	채팅방 인원	스터디 채팅방에 입장한 사용자 아이디 확인
	채팅방 폴더	스터디 채팅방 안에서 입장한 사람들은 서로 파일을 보내거나 다운
	채팅내용 저장	채팅의 내용이 저장이되어 방을 나갔다 다시 들어와도 채팅내역 존재
	화면 공유	스터디 채팅방 안에 있는 사람들에게 화면공유
게시판	자유게시판	누구든 자유롭게 게시판을 확인 가능하고 게시글을 작성

02. 프로젝트 기획

-프로토타입



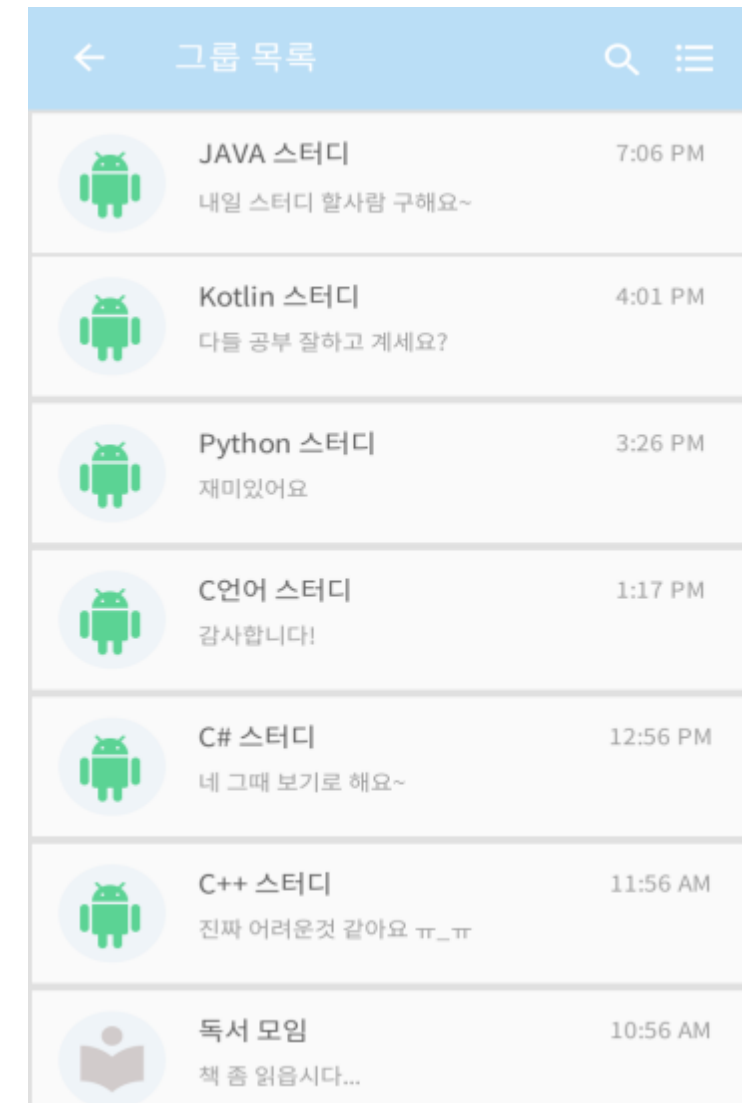
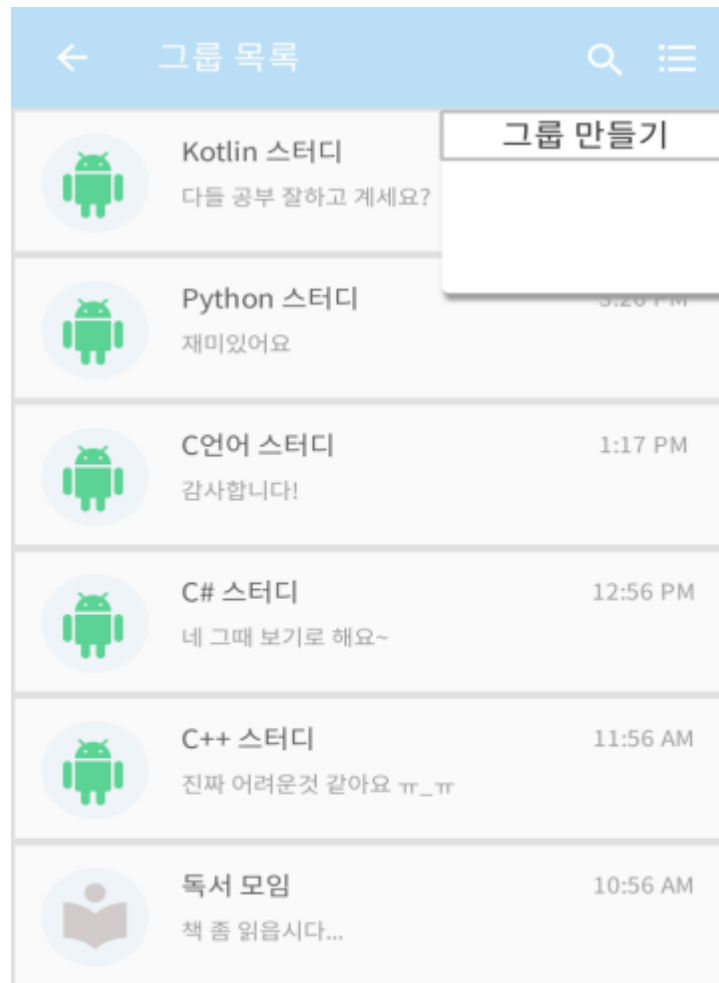
1. 로그인 및 회원가입 메인화면



2. 로그인 후 보이는
그룹(모임)목록 화면

02. 프로젝트 기획

-프로토타입



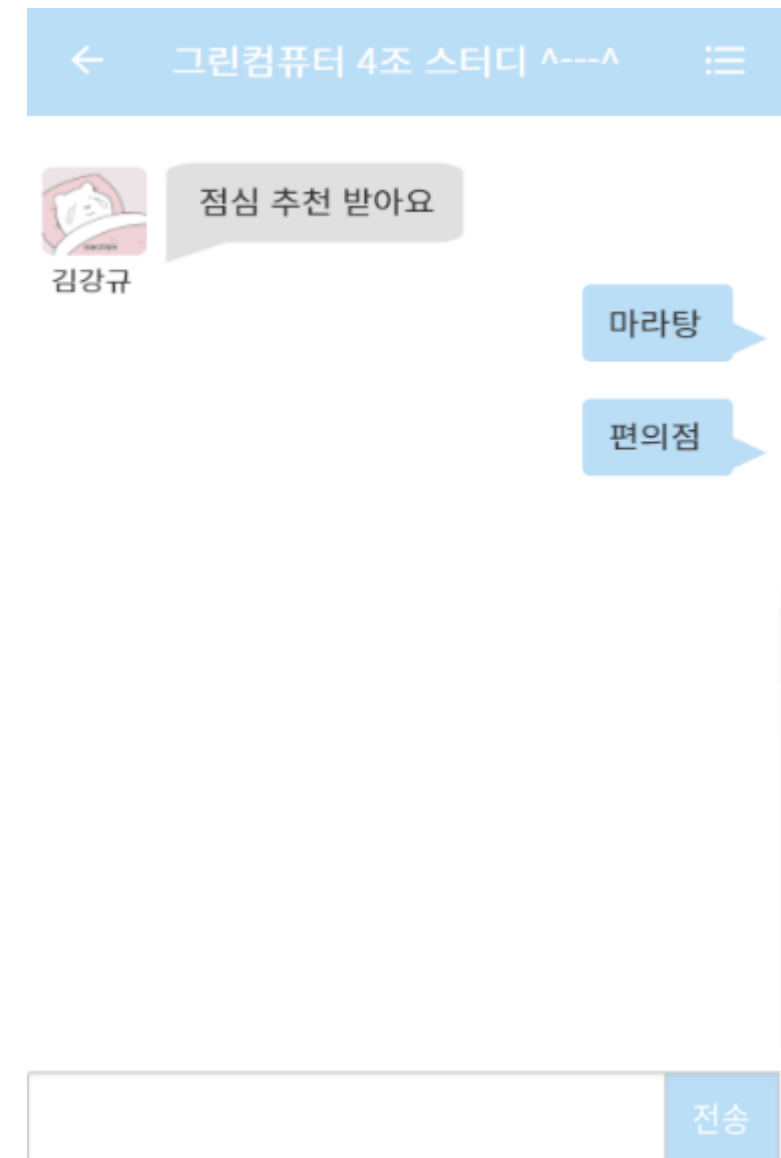
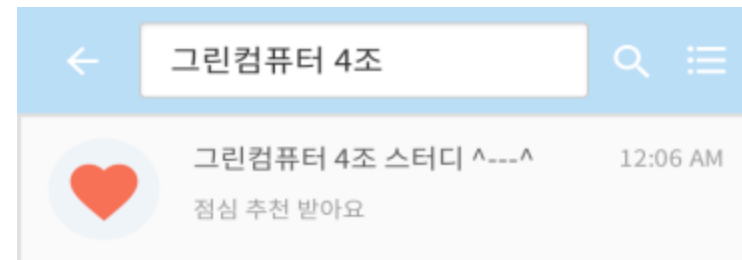
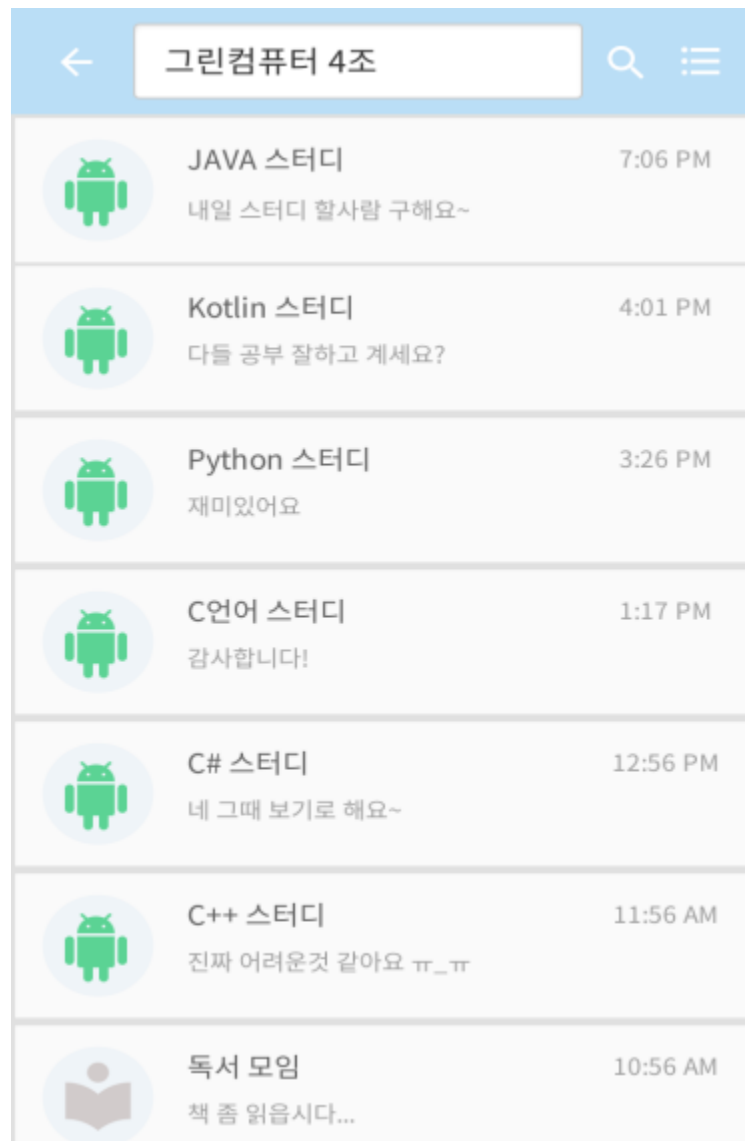
3. 상단 메뉴버튼 클릭시
그룹 만들기 목록 화면

4. 그룹 만들기 팝업 화면

5. 작성한 그룹이 만들어진 화면

02. 프로젝트 기획

-프로토타입



6. 상단의 검색 아이콘 버튼을
클릭시 검색창이 활성화 되는 화면

7. 검색어와 맞는
그룹이 검색된 화면

8. 그룹 채팅방 입장시 채팅 화면

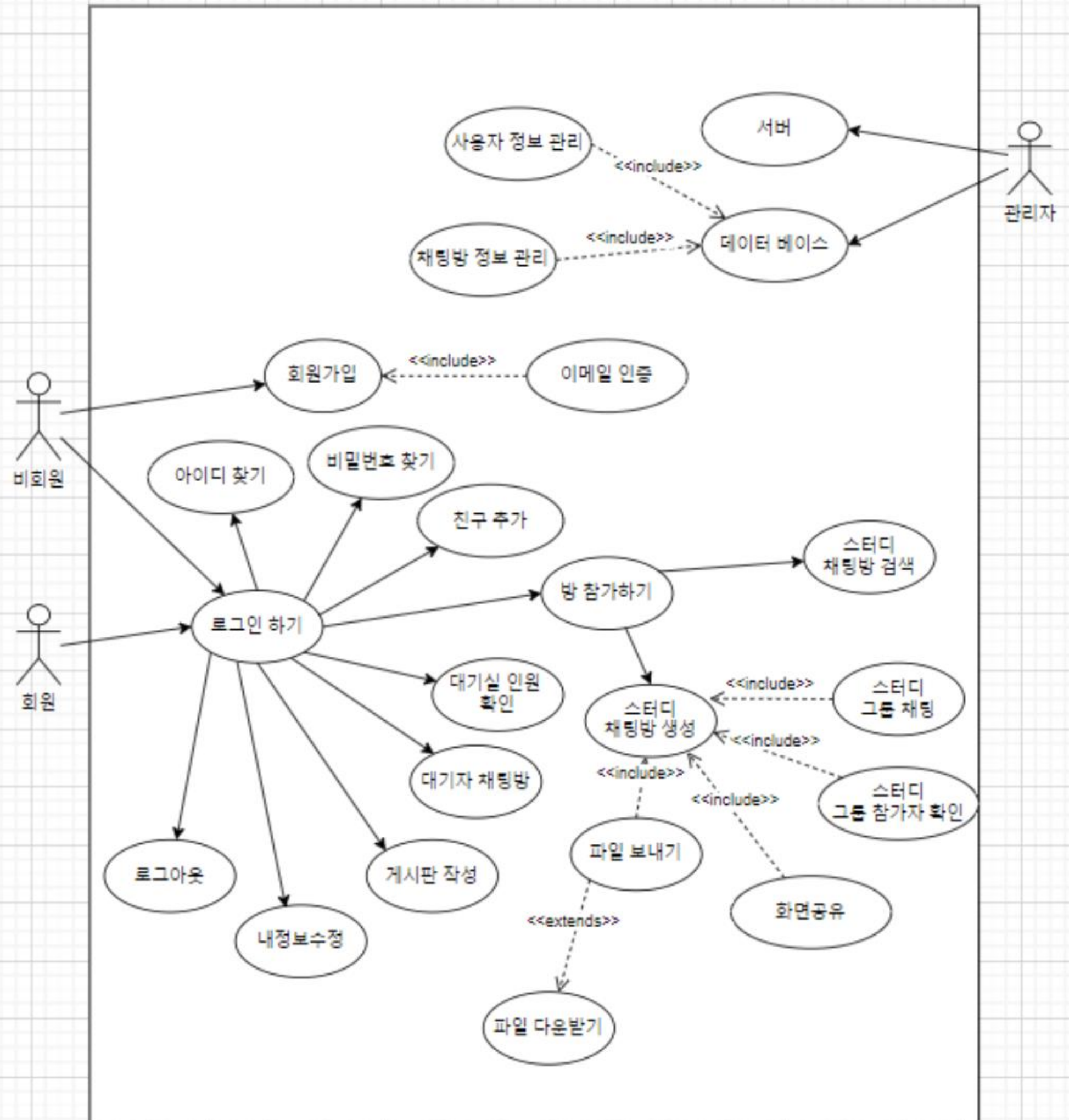
02. 프로젝트 기획

-개발환경

작업	개발환경
Github 저장소 및 협업설정	Github Desktop,Github
DB설계	MariaDB
서버 설계/제작	Eclipse,socket
클라이언트 설계/제작	Eclipse,socket
GUI 설계/제작	Eclipse,Swing
DB연결,연동확인	Eclipse,MariaDB
클라이언트와 GUI연결	Eclipse

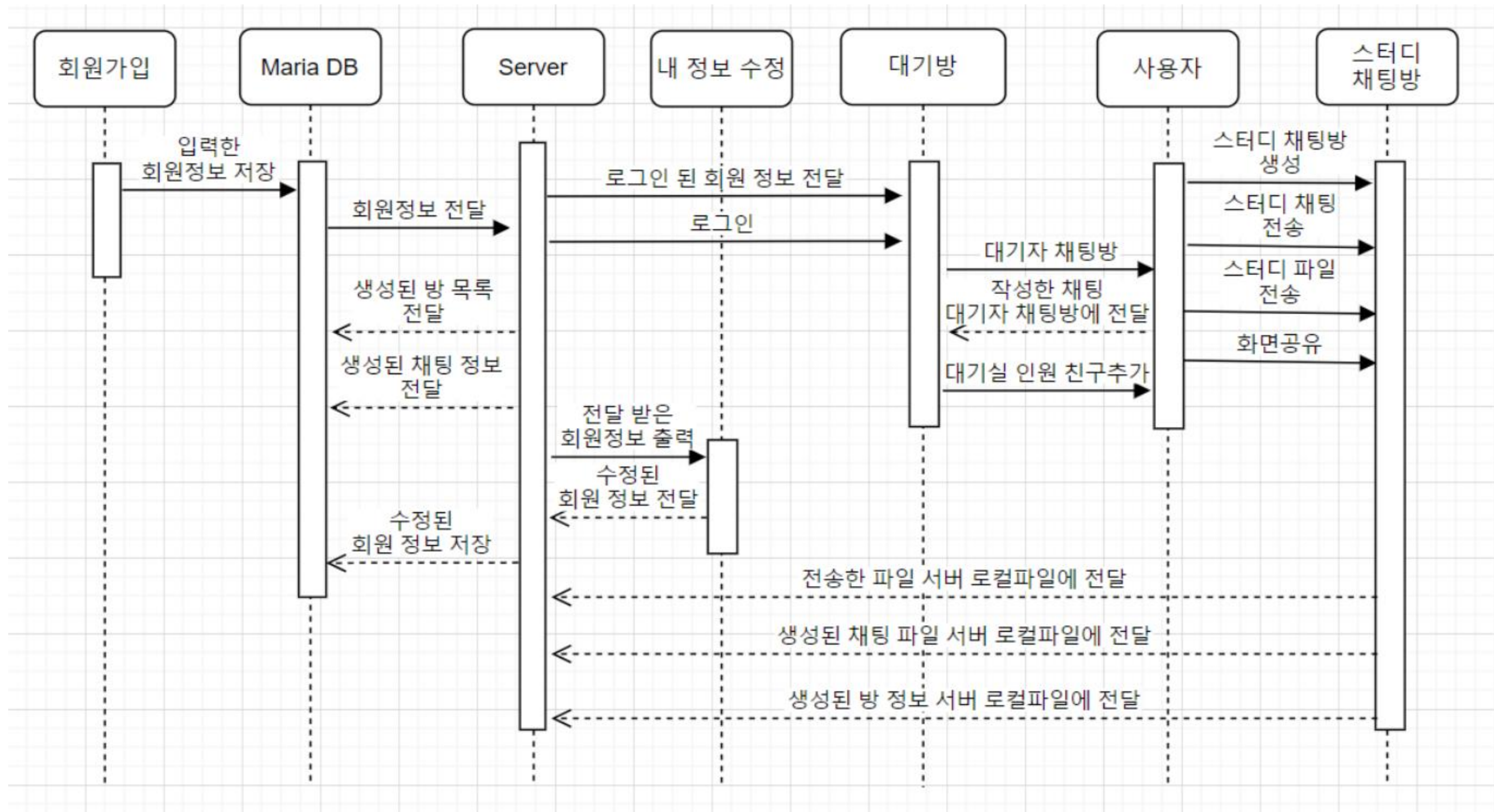
03. 프로젝트 설계

-유스케이스 다이어그램



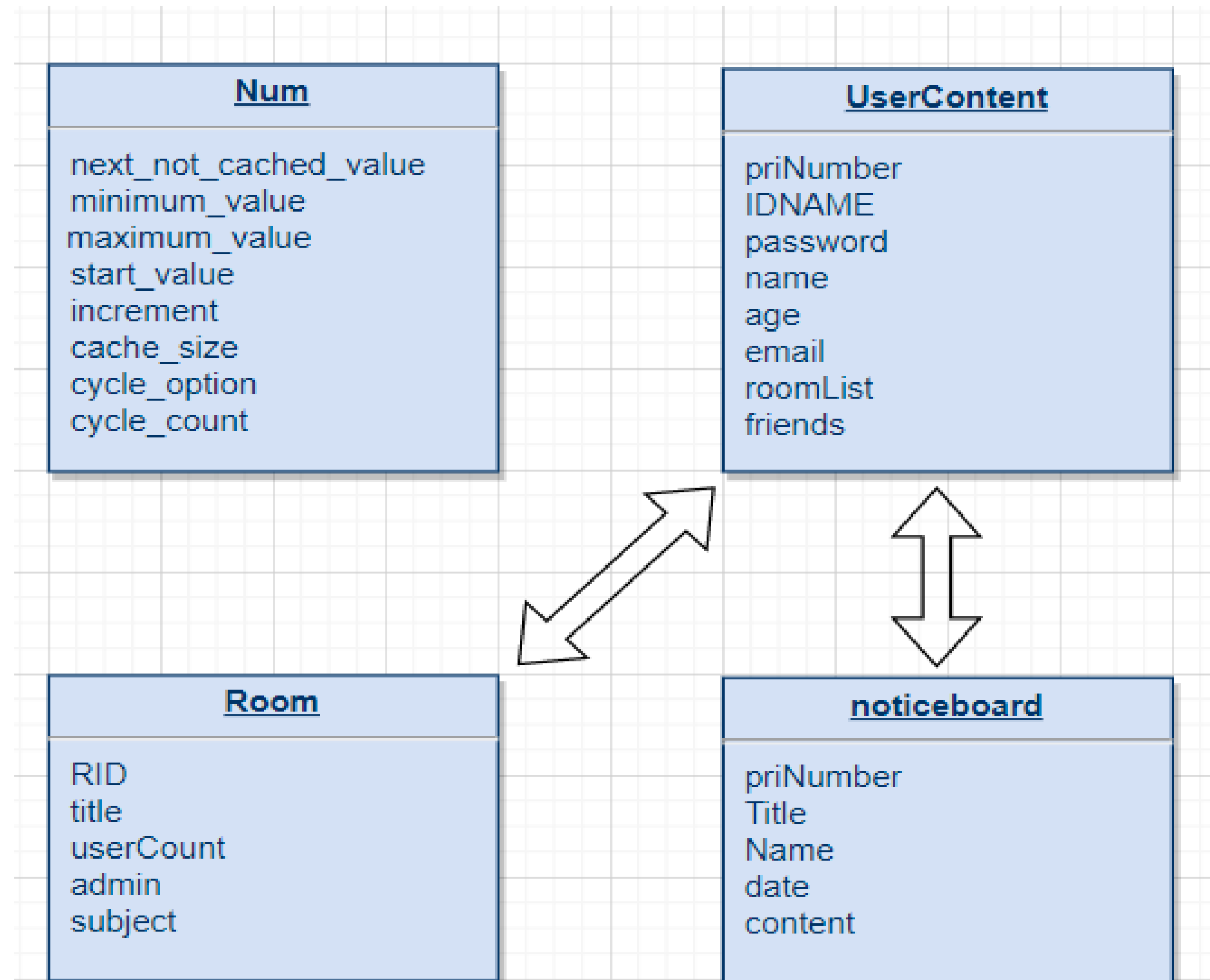
03. 프로젝트 설계

- 시퀀스 다이어그램



03. 프로젝트 설계

- DB 구조



04. 프로젝트 UI 설계

The main screen features a light blue background with the title "STUDY WITH ME!" in large, bold, white letters with a black outline. Below the title, there are two buttons: "로그인 하기" (Login) and "회원 가입" (Sign Up). At the bottom right, there is a "종료" (End) button.

메인화면

The sign-up screen features a light blue background with the title "STUDY WITH ME!". Below the title, there are several input fields and buttons: "이름:" (Name), "아이디:" (ID) with a "중복" (Duplicate) button, "비밀번호:" (Password), "생년월일" (Date of Birth) with dropdowns for year (1999), month (01), and day (06), "Email" with a text field, "@ naver.com" dropdown, and "전송" (Send) button. Below these, there is an "email 인증" (Email Verification) section with a text field and an "인증" (Verify) button. At the bottom, there are "취소" (Cancel) and "확인" (Confirm) buttons.

회원가입 화면

The login screen features a light blue background with the title "STUDY WITH ME!". Below the title, there are input fields for "아이디:" (ID) and "비밀번호:" (Password). Below these, there is a "로그인" (Login) button. At the bottom, there are two buttons: "아이디 찾기" (Find ID) and "비밀번호 찾기" (Find Password). At the bottom right, there is a "종료" (End) button.

로그인 화면

04. 프로젝트 UI 설계

친구 방 참가 자유 게시판 스타디윗미!

내 정보 친구

STUDY WITH ME!

이름 : 우효진 >>

아이디... : woohyojin >>

패스워드 : ***** >>

이메일 : woohyojin12@nav... >>

생일 : 19990106 >>

로그아웃 회원탈퇴

내정보 확인 화면

STUDY WITH ME!

1998 년 03 월 26 일

취소 수정

내정보 수정 화면

친구 방 참가 자유 게시판 스타디윗미!

내 정보 친구

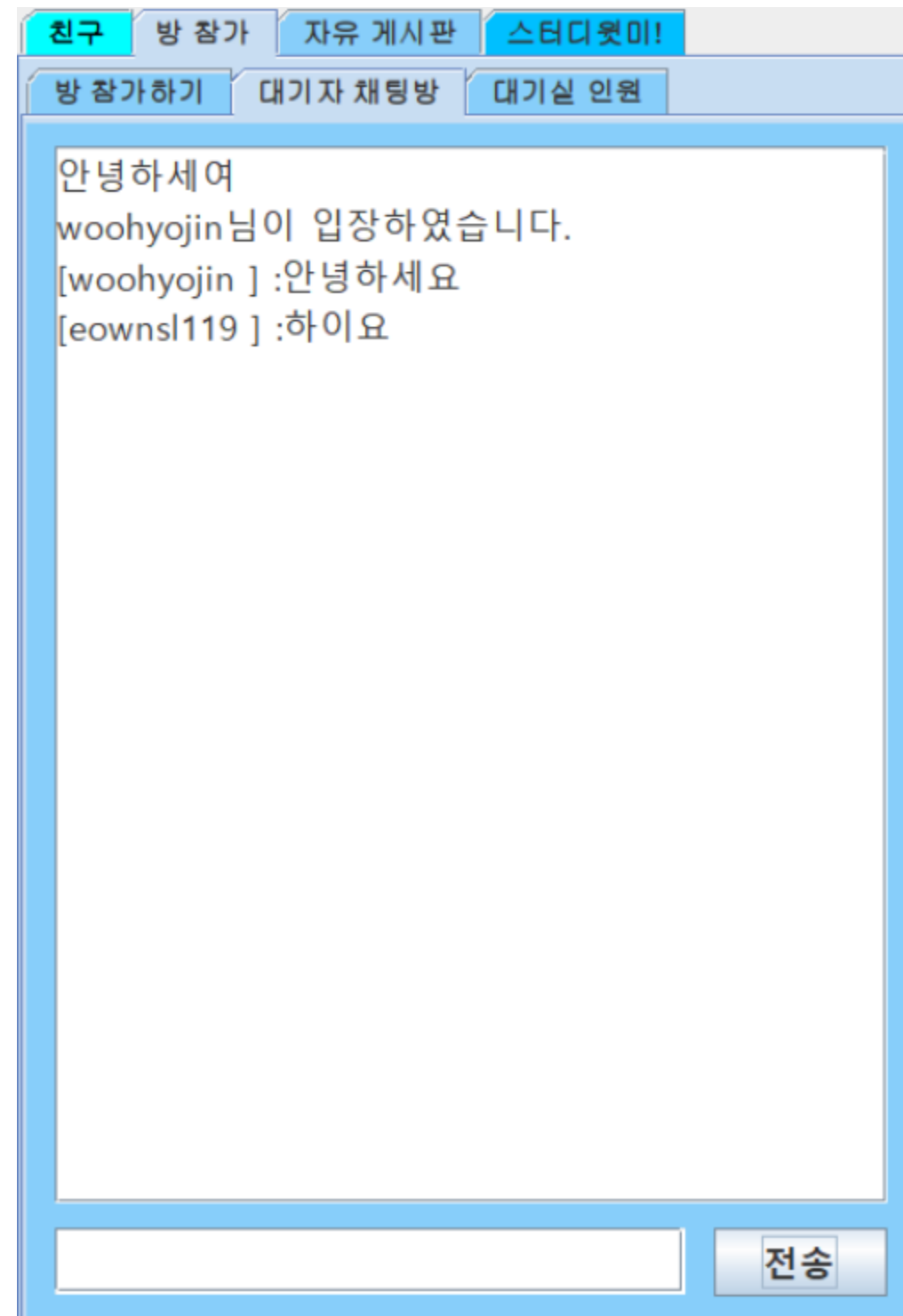
STUDY WITH ME!

아이디:	Lirodek0	on	친구 삭제
아이디:	eownsl119	on	친구 삭제
아이디:	dbwls9675	on	친구 삭제

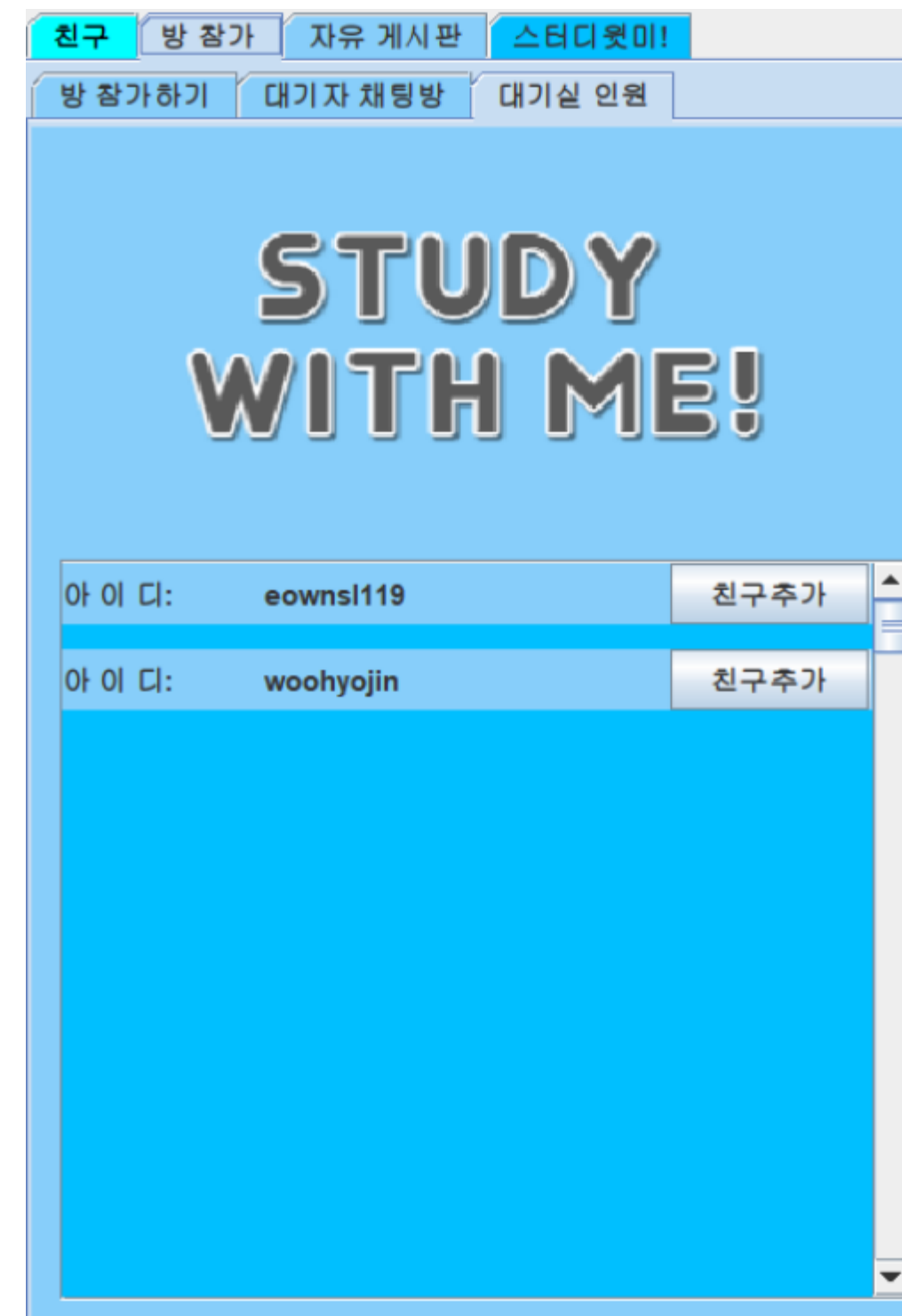
친구 추가 된 친구창 화면

친구 추가 된 친구창 화면

04. 프로젝트 UI 설계



대기자 채팅방 화면



대기실 인원 친구추가 화면

04. 프로젝트 UI 설계

The screenshot shows a web application interface for a forum. At the top, there are four tabs: '친구' (Friends), '방 참가' (Join Room), '자유 게시판' (Free Board), and '스터디넷미!' (StudyNetMe!). The '자유 게시판' tab is selected. Below the tabs, there is a '작성' (Write) button. The main content area displays a list of two forum posts. Each post shows its number, title, author, and creation date, followed by an '입장' (View) button.

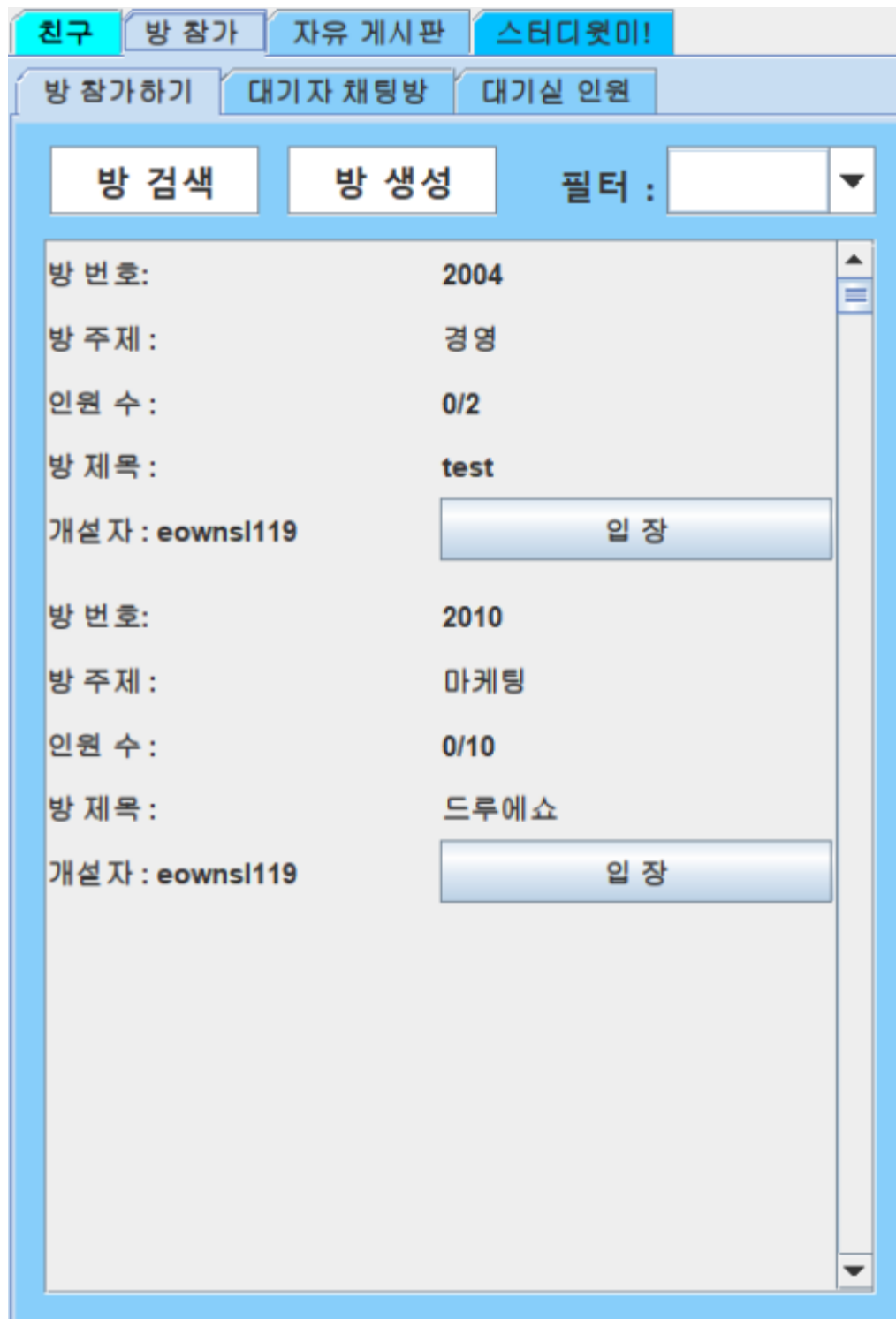
번호	제목	작성자	작성일
11	req	eownsl119	2022-02-20
23	화면 설계	woohyojin	2022-02-20

게시판 리스트 화면

The screenshot shows a web application interface for creating a new forum post. The page has a light blue background. At the top, there is a '제목' (Title) label and a text input field containing '화면설계'. Below this, there is a '내용' (Content) label and a large text area containing '화면설계 test'. At the bottom of the page, there are two buttons: '취소' (Cancel) on the left and '작성' (Write) on the right.

게시판 작성화면

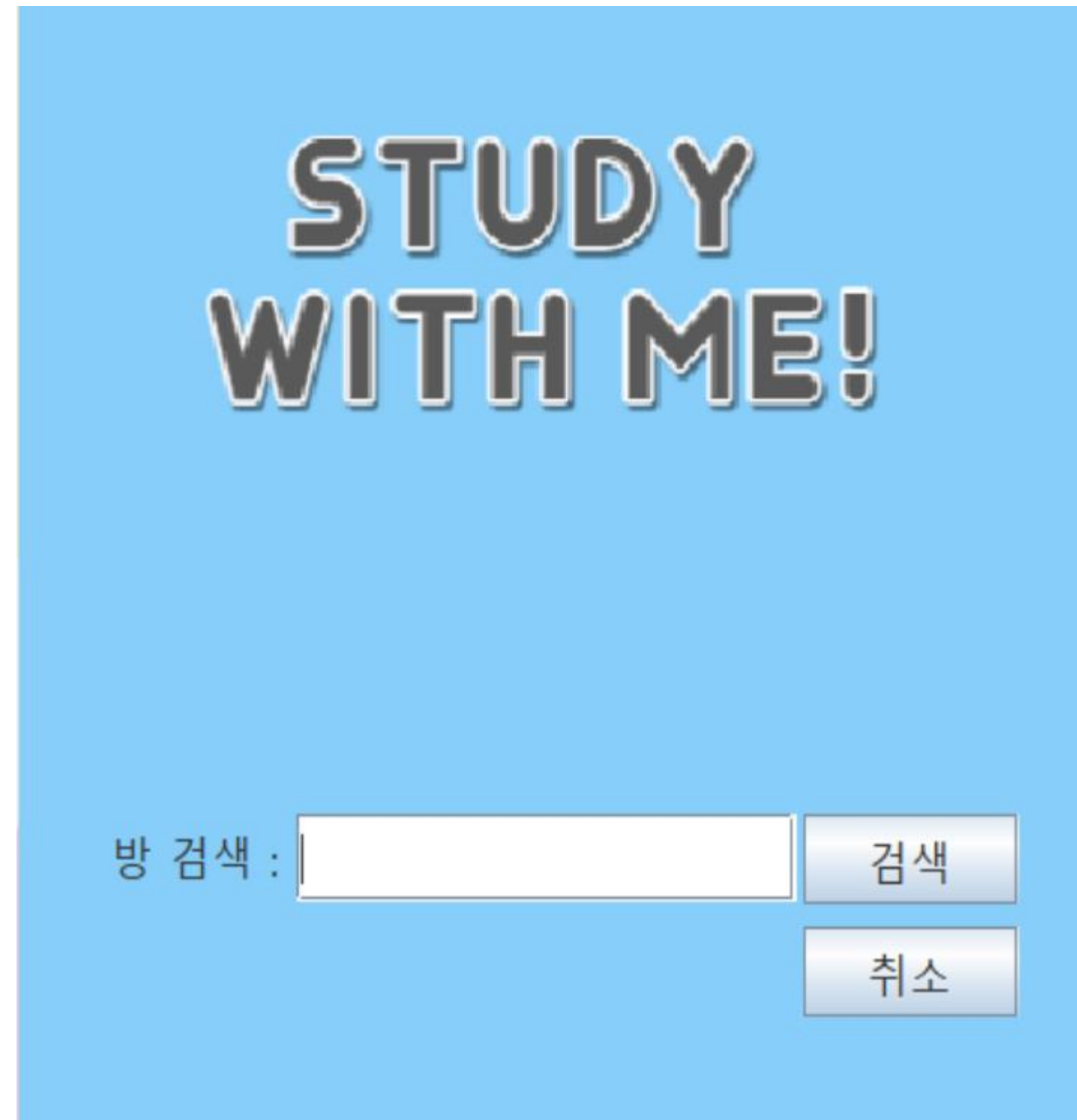
04. 프로젝트 UI 설계



The interface shows a list of rooms with the following details:

방 번호:	방 주제:	인원 수:	방 제목:	개설자:	입장
2004	경영	0/2	test	eownsl119	입장
2010	마케팅	0/10	드루에쇼	eownsl119	입장

방 목록 화면

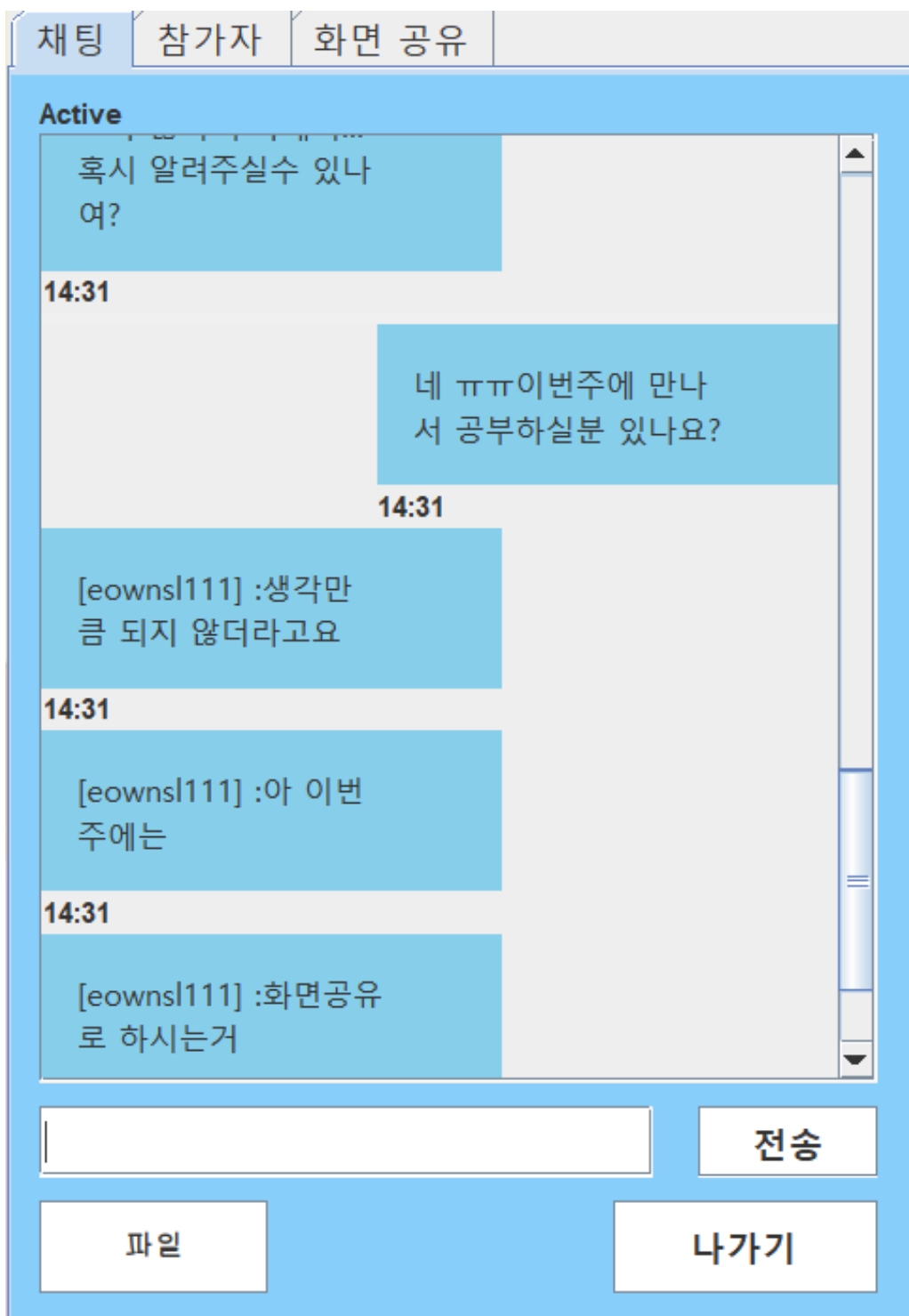


The interface features a large blue background with the text "STUDY WITH ME!" at the top. Below it, there is a search bar and two buttons:

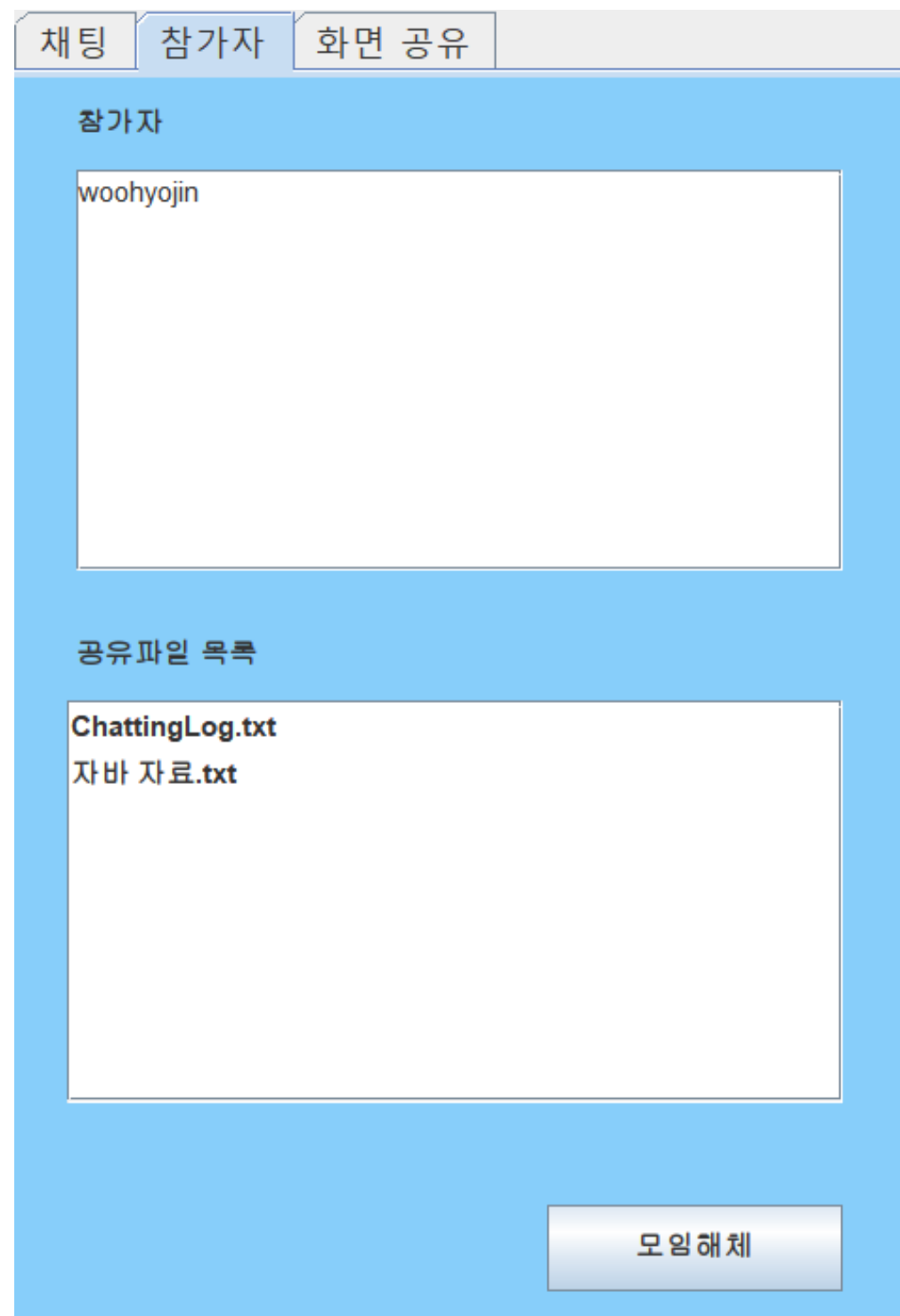
방 검색 :

방 검색 화면

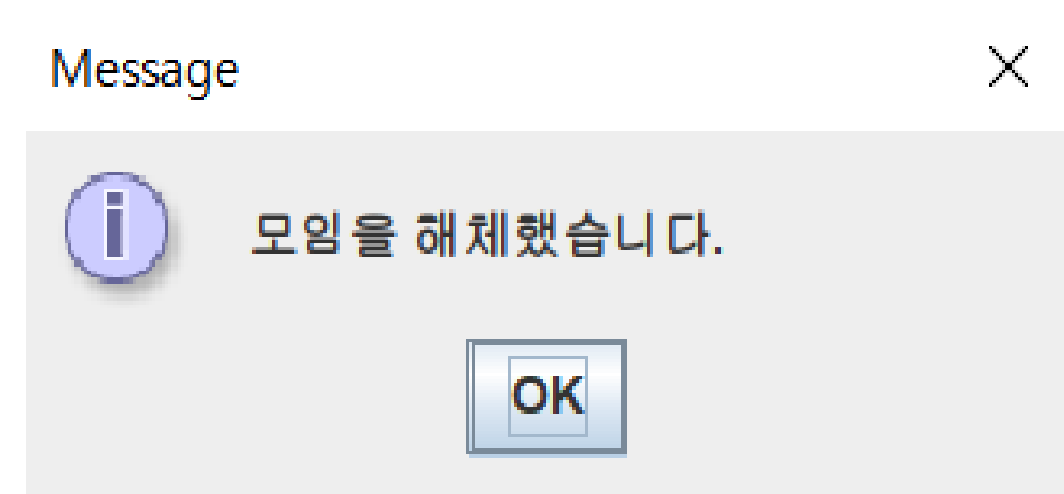
04. 프로젝트 UI 설계



스터디 채팅방 화면

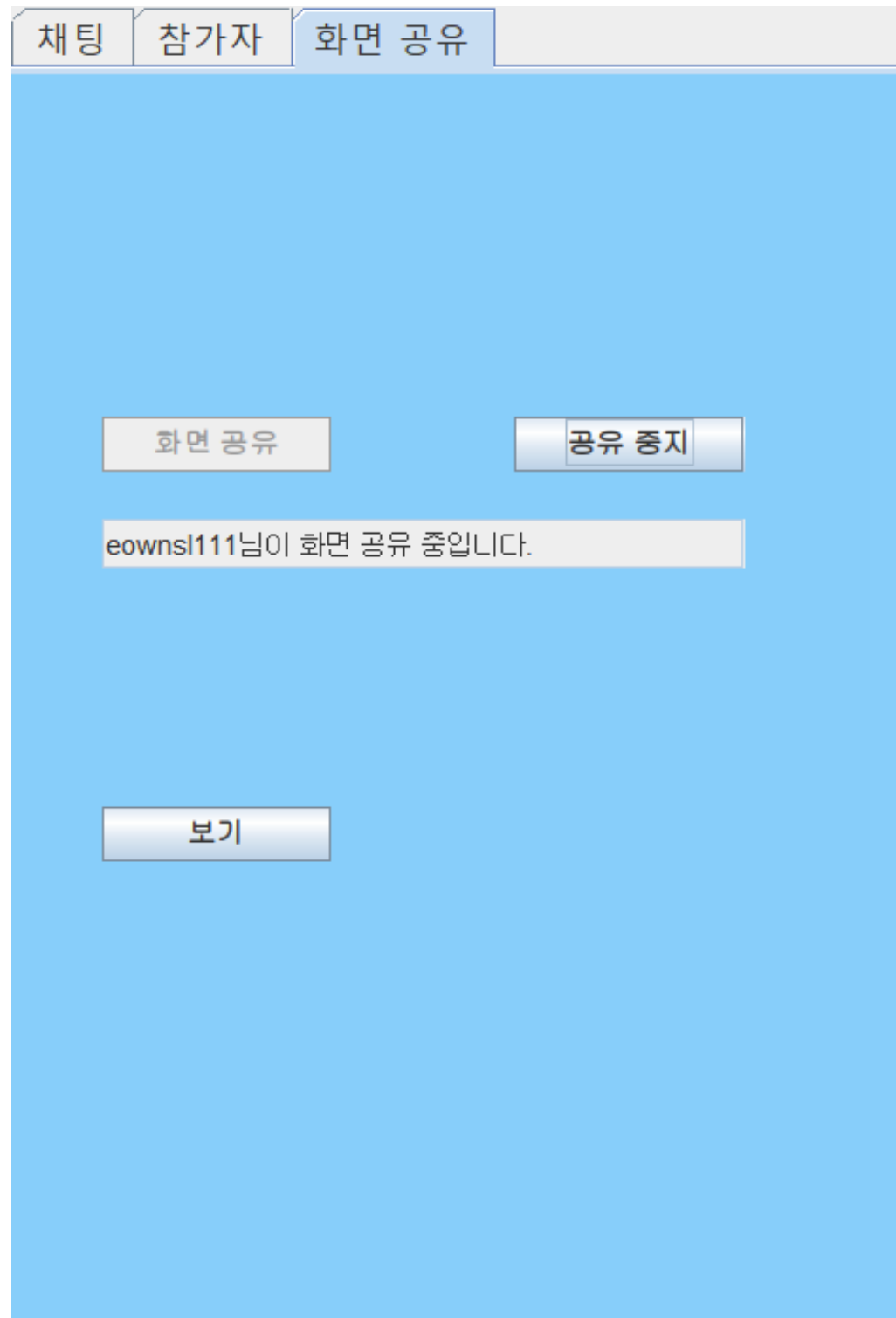


참가자 목록 및 파일 목록 화면

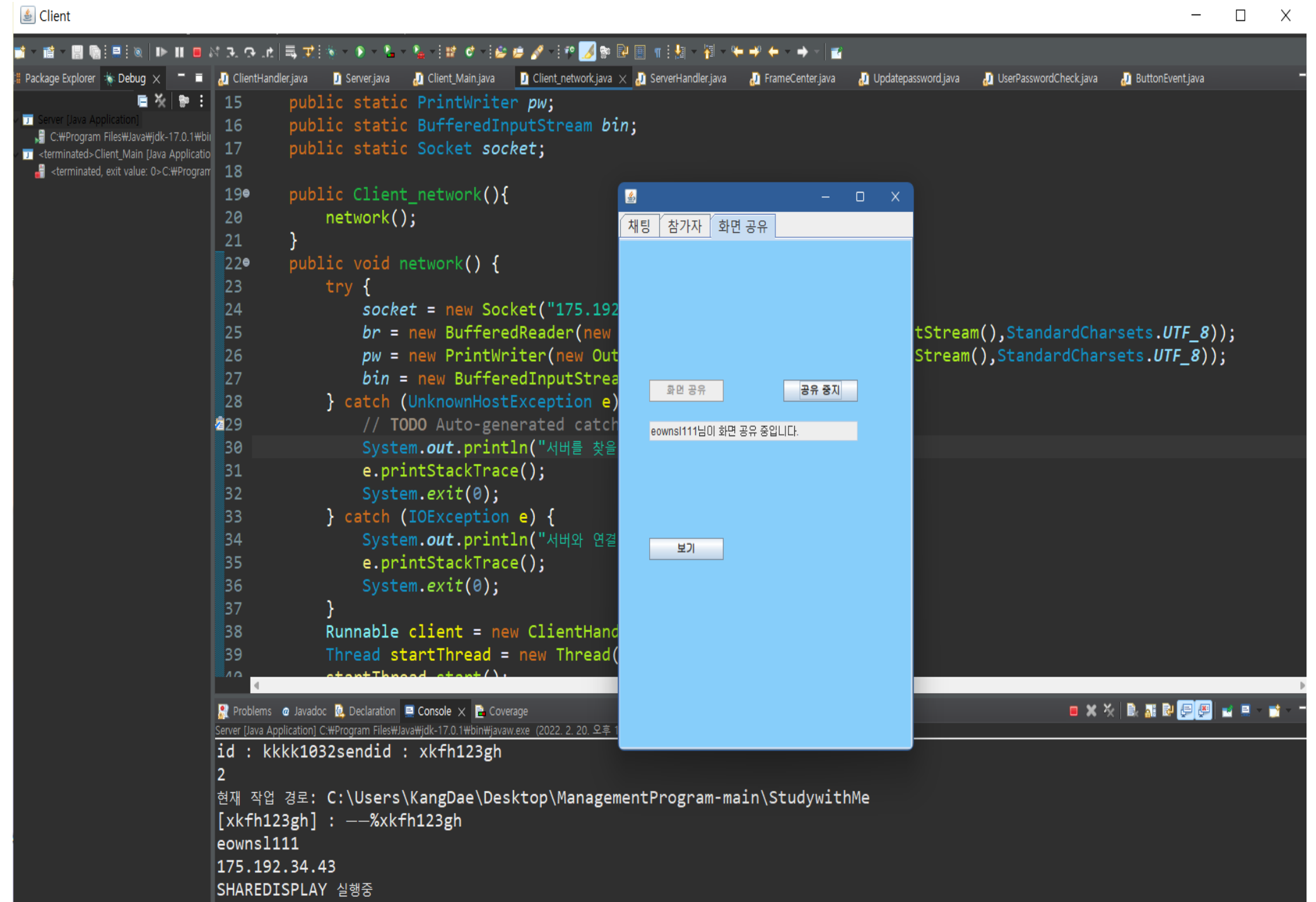


모임해체 버튼 클릭시 팝업창

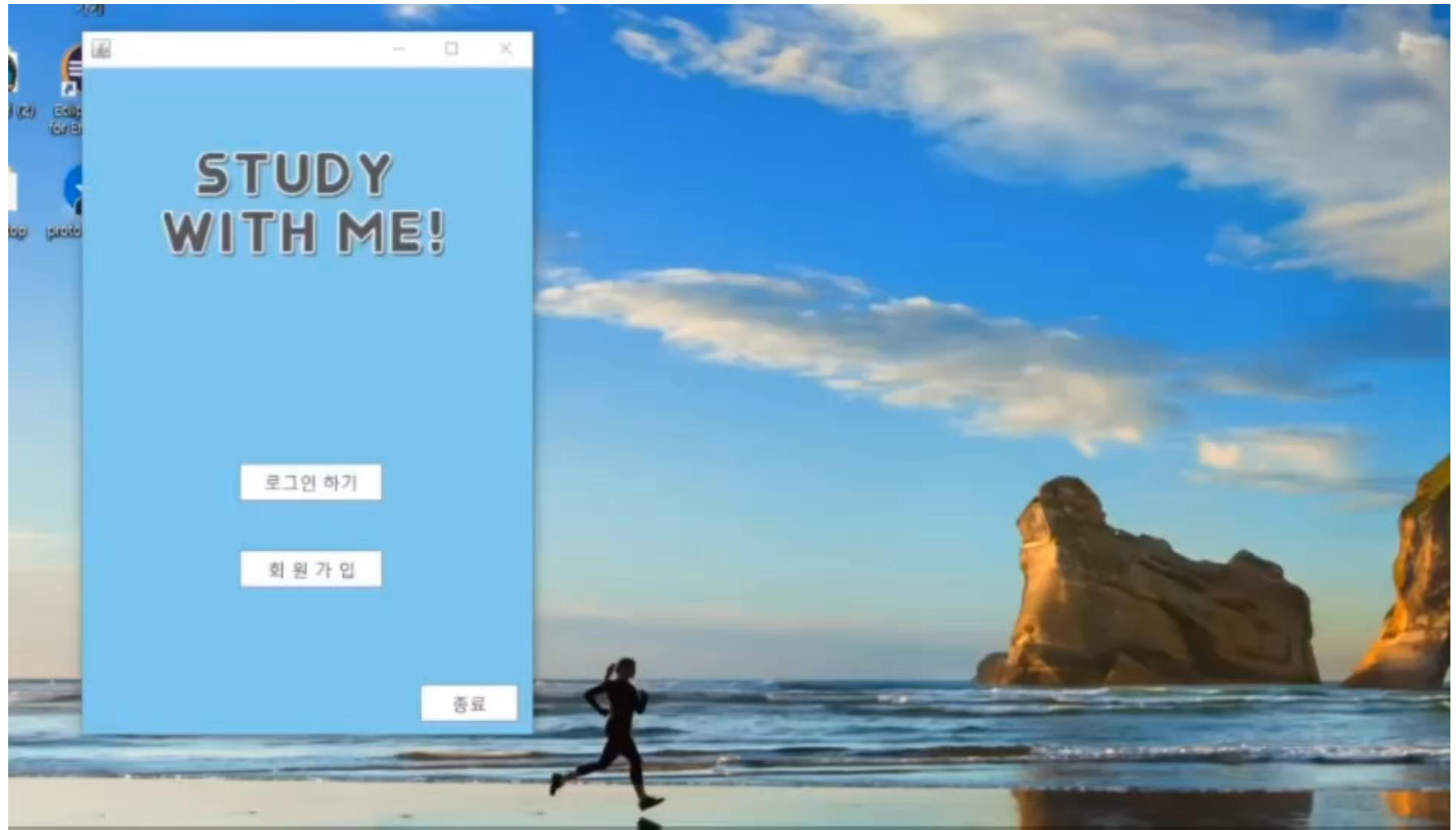
04. 프로젝트 UI 설계



화면 공유 화면



보기 버튼 클릭시 보여지는 공유된 화면



05. 프로젝트 구현

-중요소스코드

STUDY WITH ME!

방 제목 :

방 주제 :

허용인원 :

방 생성시

방 생성

취소

Title, Subject, Usercount

▶ 버튼 이벤트 부분

```
String line = "";
line = title + "%" + userCount + "%" + subject;
pw.println(Protocol.ROOMMAKE + "|" + line);
```

▶ 서버 부분

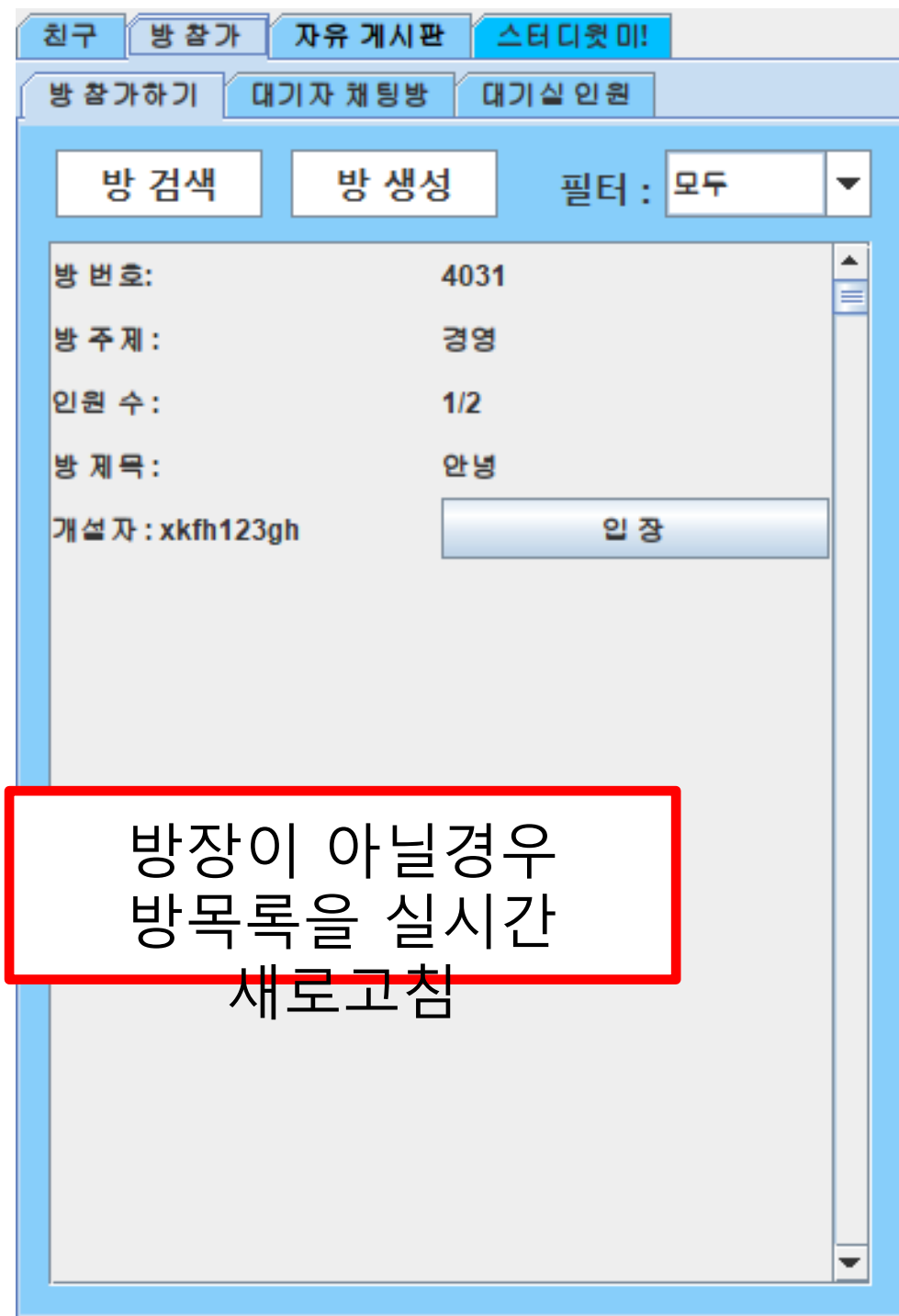
```
String roomListMessage = "";
// 룸리스트 정보
for (int i = 0; i < roomtotalList.size(); i++) {
    roomListMessage += (roomtotalList.get(i).getrID() + "%" + roomtotalList.get(i).getTitle() + "%"
        + roomtotalList.get(i).getUserCount() + "%" + roomtotalList.get(i).getMasterName() + "%"
        + roomtotalList.get(i).getSubject() + "%" + roomtotalList.get(i).roomInUserList.size()
        + "%" + "-");
}

System.out.println(roomListMessage);

for (int i = 0; i < waitUserList.size(); i++) { // 대기자수
    // 방만든사람에게는 바로 채팅화면으로
    // 대기자ID랑 방만든사람ID가 같을때
    if (waitUserList.get(i).user.getIdName().compareTo(tempRoom.getMasterName()) == 0) {
        waitUserList.get(i).pw.println(Protocol.ROOMMAKE_OK1 + "|" + tempRoom.getMasterName());
        // 방 만든사람에게 대기방 새로고침
        waitUserList.get(i).pw.println(Protocol.ROOMMAKE_OK + "|" + roomListMessage);
        waitUserList.get(i).pw.flush();
    } else { // 다른 대기방사람들에게는 대기방만 새로고침
        waitUserList.get(i).pw.println(Protocol.ROOMMAKE_OK + "|" + roomListMessage);
        waitUserList.get(i).pw.flush();
    }
}
```


05. 프로젝트 구현

-중요소스코드



▶ 클라이언트 부분

```
else if (line[0].compareTo(Protocol.ROOMMAKE_OK1) == 0) // 방만들어짐 (만든 당사자) // 입장

    System.out.println("방장 입장화면 변환");
    frameMakeRoom.frameDown(); // 대기방 화면 끄고
    System.out.println("RoomMaker: " + line[1]);
    frameChattingRoom.vertical.removeAll();
    frameChattingRoom.setVisible(true);
    frameChattingRoom.Chatting_textarea_Inuserlist.setText(line[1]);
```

- 방장일 경우 초기화된 채팅방을 실행시켜줌

```
String roomList[] = line[1].split("-"); // 방 갯수
String roomfirst[] = line[1].split("%");

for (int i = 0; i < roomList.length; i++) {
    System.out.print(roomList[i] + "/");
}

String roomListDetail[]; // 방세부
System.out.println("RoomList size : " + roomList.length);

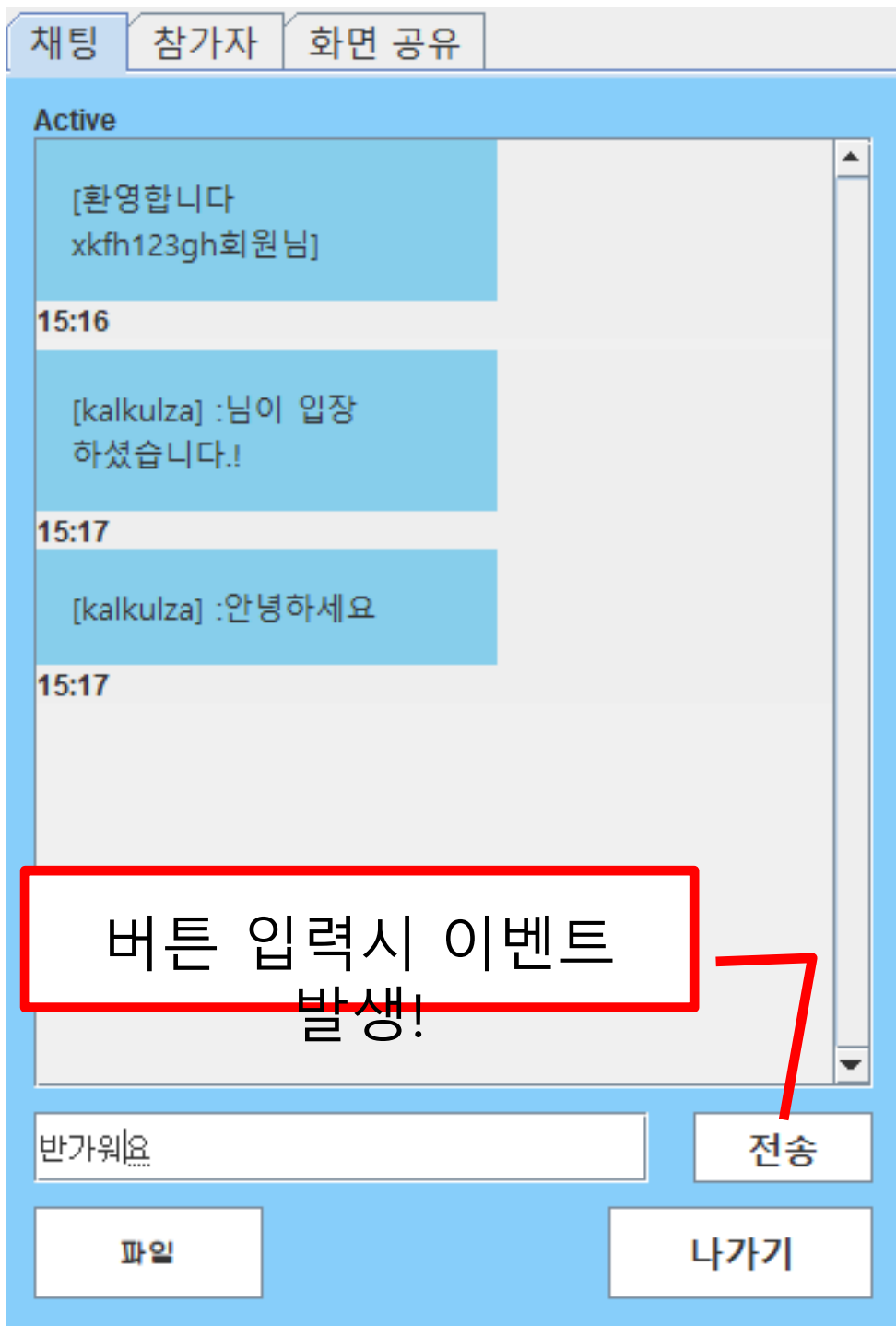
frameCenter.containPanelClear(); // 룸 프레임에 컨테이너를 비워주고
for (int i = 0; i < roomList.length; i++) {

    frameCenter.panelRoomList[i].init(); // 방리스트를 받은거로 다시 생성해주고
    roomListDetail = roomList[i].split("%");
    String userNumber = "";

    userNumber += (roomListDetail[5] + "/" + roomListDetail[2]);
    frameCenter.panelRoomList[i].labelArray[1].setText(roomListDetail[0]); // 방번호
    frameCenter.panelRoomList[i].labelArray[3].setText(roomListDetail[4]); // 방주제
    frameCenter.panelRoomList[i].labelArray[5].setText(userNumber); // 인원수
    frameCenter.panelRoomList[i].labelArray[7].setText(roomListDetail[1]); // 방제목
    frameCenter.panelRoomList[i].labelArray[8].setText("개설자 : " + roomListDetail[3]); // 개설자
```

05. 프로젝트 구현

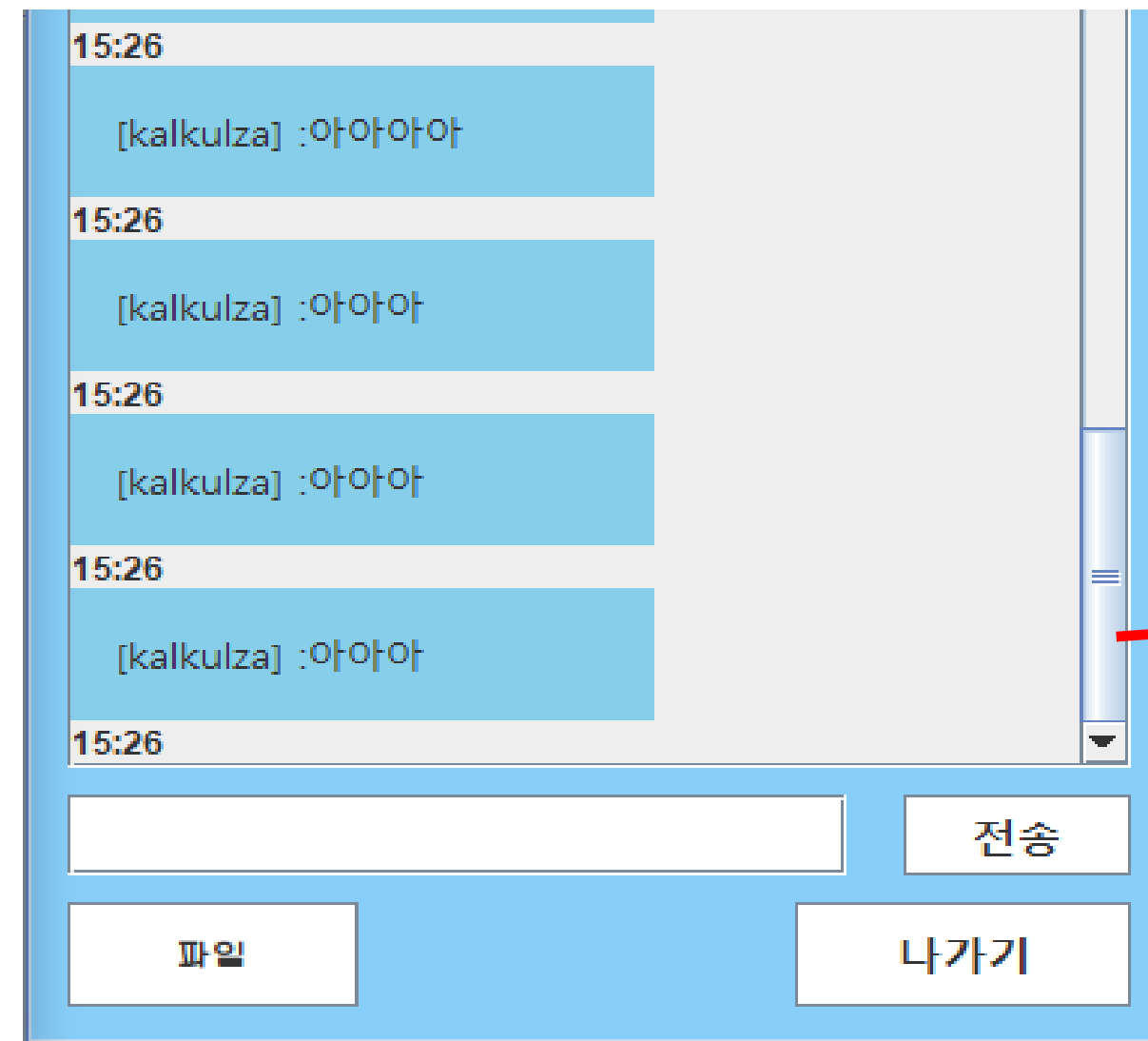
- 중요 소스 코드



▶ 채팅 저장 버튼 이벤트 부분

```
pw.println(Protocol.CHATTINGSENDMESSAGE + "|" + message + "%" + myIdname);  
pw.flush();  
frameChattingRoom.chatting_textField_message.setText("");  
pw.println(Protocol.CHATTINGSCROLLBARDOWN + "|" + "message");  
pw.flush();
```

- 프로토콜과 함께 보낸 유저의 이름을 담아서 서버로 보내줌



05. 프로젝트 구현

- 중요 소스 코드

방 폴더에 채팅 내역을
저장

```
1 [xkfh123gh] : 안녕%xkfh123gh
2 [kalkulza] : 안녕%kalkulza
3 [xkfh123gh] : 반가워%xkfh123gh
4 [kalkulza] : 반가워%kalkulza
5 [kalkulza] : 아노말루스%kalkulza
6 [kalkulza] : 뭐?%kalkulza
7
```

▶ 채팅 저장 서버 부분

```
for (int i = 0; i < roomUserSize; i++) {
    if (roomtotalList.get(roomtotalList.indexOf(priRoom)).roomInUserList.get(i).user.getIdName()
        .equals(id)) {
        System.out.println(line.length);
        roomtotalList.get(roomtotalList.indexOf(priRoom)).roomInUserList.get(i).pw
            .println(Protocol.CHATTINGSENDMESSAGE_MASTER_OK + "|" + user.getIdName() + "|"
                + message);
        String log = Protocol.CHATTINGSENDMESSAGE_MASTER_OK + "|" + user.getIdName() + "|"
            + message;
        roomtotalList.get(roomtotalList.indexOf(priRoom)).roomInUserList.get(i).pw.flush();
    }
}
```

- 방안의 유저들에게 채팅을 분배해주는 역할

```
fw = new BufferedWriter(new FileWriter(Path, true));
fw.write "[" + user.getIdName() + "] : " + line[1] + "\n");
fw.flush();
```


05. 프로젝트 구현

- 중요 소스 코드



▶ 채팅 저장클라이언트 부분

```
else if (line[0].compareTo(Protocol.CHATTINGSENDMESSAGE_MASTER_OK) == 0) {
    if (line.length == 3) {
        String message = line[2];
        JPanel panel = frameChattingRoom.formatLabel(message);
        JPanel right = new JPanel(new BorderLayout());
        right.add(panel, BorderLayout.LINE_END);
        frameChattingRoom.vertical.add(right);
        frameChattingRoom.chatting_chattingPanel.repaint();
    }
}
else if (line[0].compareTo(Protocol.CHATTINGSENDMESSAGE_OK) == 0) {
    if (line.length == 3) {
        String message = "[" + line[1] + "]" + line[2];
        JPanel p2 = frameChattingRoom.formatLabel(message);
        JPanel left = new JPanel(new BorderLayout());
        left.add(p2, BorderLayout.LINE_START);
        frameChattingRoom.vertical.add(left);
        frameChattingRoom.chatting_chattingPanel.repaint();
    }
}
```

- 서버에서 분배받은 프로토콜에 따라 받은 문자를 JPanel에 담아서 실행

```
public static JPanel formatLabel(String message){
    JPanel chattingPanel = new JPanel();
    chattingPanel.setLayout(new BoxLayout(chattingPanel, BoxLayout.Y_AXIS));

    JLabel chattingLabel = new JLabel("<html><p style = \"width : 100px\">" + message + "</p></html>");
    chattingLabel.setFont(new Font("맑은 고딕", Font.PLAIN, 13));
    chattingLabel.setBackground(new Color(135, 206, 235));
    chattingLabel.setOpaque(true);
    chattingLabel.setBorder(new EmptyBorder(15,15,15,50));

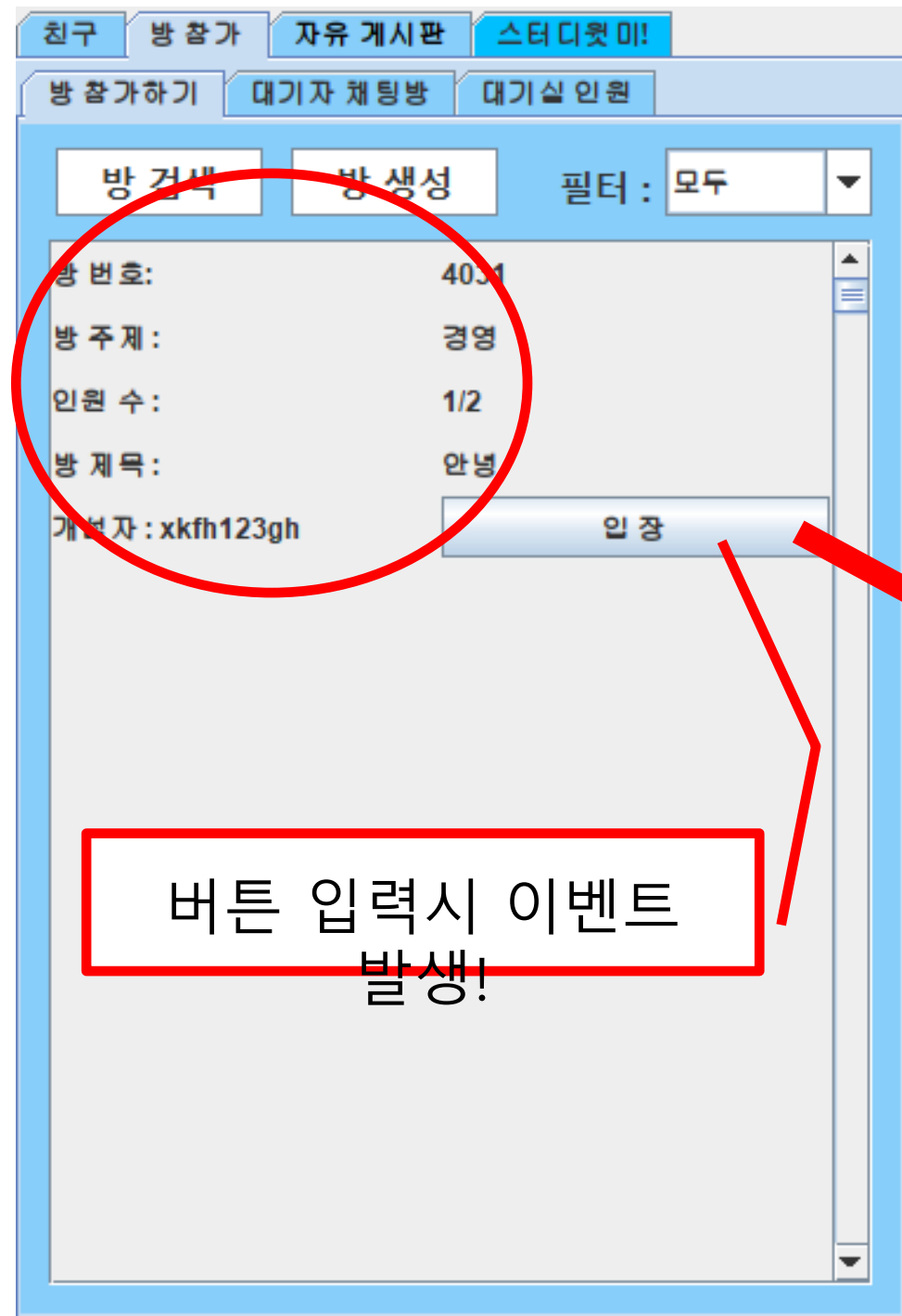
    Calendar cal = Calendar.getInstance();
    SimpleDateFormat sdf = new SimpleDateFormat("HH:mm");

    JLabel timeLabel = new JLabel();
    timeLabel.setText(sdf.format(cal.getTime()));

    chattingPanel.add(chattingLabel);
    chattingPanel.add(timeLabel);
    return chattingPanel;
}
```

05. 프로젝트 구현

- 중요 소스 코드



▶ 채팅 불러오기 버튼 이벤트 부분

```
if(e.getSource() == enterButton) {  
    String count[] = labelArray[5].getText().split("/");  
    if(count[0].compareTo(count[1]) == 0) {  
        JOptionPane.showMessageDialog(this, "인원수 초과로 들어갈 수 없습니다.");  
    } else {  
        String line = "";  
        System.out.println(labelArray[1].getText());  
        line += (Protocol.ENTERROOM + "|" + labelArray[1].getText());  
        pw.println(line);  
        pw.flush();  
    }  
}
```

- Label[1].getText() == 방의 고유번호

```
public static String labelName[] = { "방 번호: ", " ", "방 주제: ", " ", "인원 수: ", " ",  
    "방 제목: ", " ", " "};
```

05. 프로젝트 구현

- 중요 소스 코드

```
String folder = path + "\\roomFolder\\" + priRoom.getrID() + "\\";
System.out.println(folder);
// 폴더명으로 파일객체 생성
File file = new File(folder);

// 폴더라면 폴더가 가진 파일객체를 리스트로 받는다.
File[] list = file.listFiles();

String fileList = "";
// 리스트에서 파일을 하나씩 꺼낸다
for (File f : list) {
    // 파일일 경우만 출력
    if (f.isFile()) {
        fileList += (f.getName() + "%");
    }
    System.out.println();
}
System.out.println("FileList : " + fileList);

for (int i = 0; i < roomtotalList.get(index).roomInUserList.size(); i++) {
    if (!(roomtotalList.get(index).roomInUserList.get(i).user.getIdName() == user.getIdName())) {
        roomtotalList.get(index).roomInUserList.get(i).pw.println(
            Protocol.CHATTINGSENDMESSAGE_OK + "|" + user.getIdName() + "|" + "님이 입장하셨습니다.!!");
        roomtotalList.get(index).roomInUserList.get(i).pw.flush();
    }
}
```

입장시 채팅했던
내역이 저장된
txt파일에서 채팅을
불러옴

▶ 채팅 불러오기 서버부분

```
System.out.println(waitUserList.size());
String thisName = waitUserList.get(waitUserList.indexOf(this)).user.getIdName();
String[] RoomListArr = null;
String sql = "select * from UserContent where IDNAME = '" + this.user.getIdName() + "'";
String updateSql = "update usercontent set RoomList=? where IDNAME=?";
String userpath = path + "\\userFolder\\" + this.user.getIdName() + "\\Roomlist.txt";
```

- 방에 자신의 이름이 없을 경우 DB에 추가

```
fw = new BufferedWriter(new FileWriter(userpath, true));
fw.write(priRoom.getrID() + "%");
fw.flush();
pstmt = conn.prepareStatement(updateSql);
String roomId = Files.readString(Paths.get(userpath));
pstmt.setString(1, roomId);
pstmt.setString(2, user.getIdName());
pstmt.executeUpdate();
```

05. 프로젝트 구현

- 중요 소스 코드

```
17[kalkulza] : 아아아%kalkulza
18[kalkulza] : 아아아%kalkulza
19[xkfh123gh] : 프로그래밍언어중에 %xkfh123gh
20[xkfh123gh] : 어떤 언어가 제일 좋다고 생각하나요?%xkfh123gh
21[kalkulza] : 저는 제가 할줄 아는 언어가 가장 좋다고 생각해요%kalkulza
22[xkfh123gh] : 맞는말이네요%xkfh123gh
23
```

서버에서 가져온
txt 데이터를
필요에 맞게
자르기

▶ 채팅 불러오기 클라이언트부분

```
String LinesTest[] = line[1].split(", ");
String roomMember[] = line[2].split("%");// 룸에 들어온사람들
String LinesTest1 = "";
String lineList = "";

for (int i = 0; i < LinesTest.length; i++) {
    LinesTest1 += (LinesTest[i] + "-\n");
}

for (int i = 0; i < roomMember.length; i++) {
    lineList += (roomMember[i] + "\n");
}

String testNew[] = LinesTest1.replace("[", "").replace("]", "").replace(" ", "").split("-");
LinesTest1 = "";
for (int i = 0; i < testNew.length - 1; i++) {
    testNew[i] = testNew[i].substring(testNew[i].indexOf(":") + 1);
    LinesTest1 += (testNew[i] + "%");
}
```

- 채팅 저장시와 마찬가지로 패널에 표시

```
for (int i = 0; i < messages.length - 1; i++) {
    if (line[3].equals(ids[i])) {

        JPanel panel = frameChattingRoom.formatLabel(messages[i]);
        JPanel right = new JPanel(new BorderLayout());
        right.add(panel, BorderLayout.LINE_END);
        frameChattingRoom.vertical.add(right);
        frameChattingRoom.chatting_chattingPanel.repaint();
    } else {
        String message = "[" + ids[i] + "]" + ":" + messages[i];
        JPanel p2 = frameChattingRoom.formatLabel(message);
        JPanel left = new JPanel(new BorderLayout());
        left.add(p2, BorderLayout.LINE_START);
        frameChattingRoom.vertical.add(left);
        frameChattingRoom.chatting_chattingPanel.repaint();
    }
}
```

06. 프로젝트 후기

처음하는 프로젝트라 GIT 협업 사용에도 어려움이 있었고 또한 어떻게 프로젝트를 시작해야 할지 몰랐었다.

하지만 팀프로젝트를 하는 동안 기존에 알고 있던 코드들과 달리 새로운 코드들을 많이 접할 수 있어서 실력향상과 팀워크에 좋은 경험이 되었다.

조원들과 팀프로젝트를 진행 하는동안 작동되지 않는 기능들을 하나씩 추가 및 수정하고, 기능들이 정상적으로 작동 될 때 마다 뿌듯함을 느껴 서로 부족한 점을 도와주면서 팀 프로젝트에 대한 중요성을 알게되었다.

**지금까지 발표를 시청해주셔서
감사합니다!**