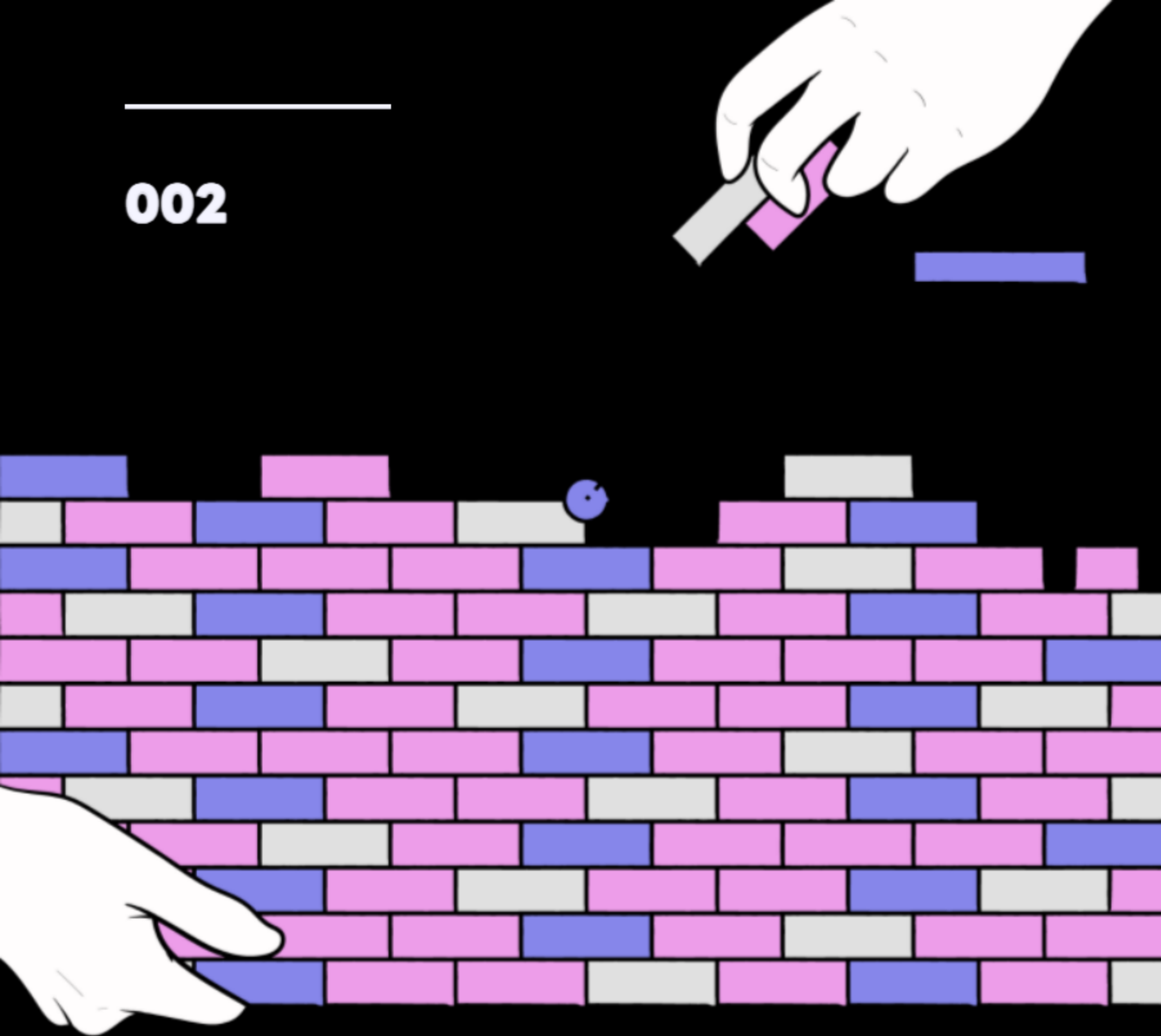


MBTIng

MBTI를 기반으로한 맞춤형 소개팅 사이트

담당자 기준서 김민제 김민중 김선아 정윤석 정현두

002



Consumer Survey Analysis and Implications

Table of Contents

목차

| | |
|----|---------------------|
| 01 | 프로젝트 주제 및 기획 (전체) → |
| 02 | 자신이 맡은 파트 명시 → |
| 03 | UML, ERD 등 → |
| 04 | 기능 구현 및 설명 → |
| 05 | 사용 기술 및 개발 환경 명시 → |
| 06 | 개인 프로젝트 기여도 명시 → |

01. 프로젝트 주제 및 기획

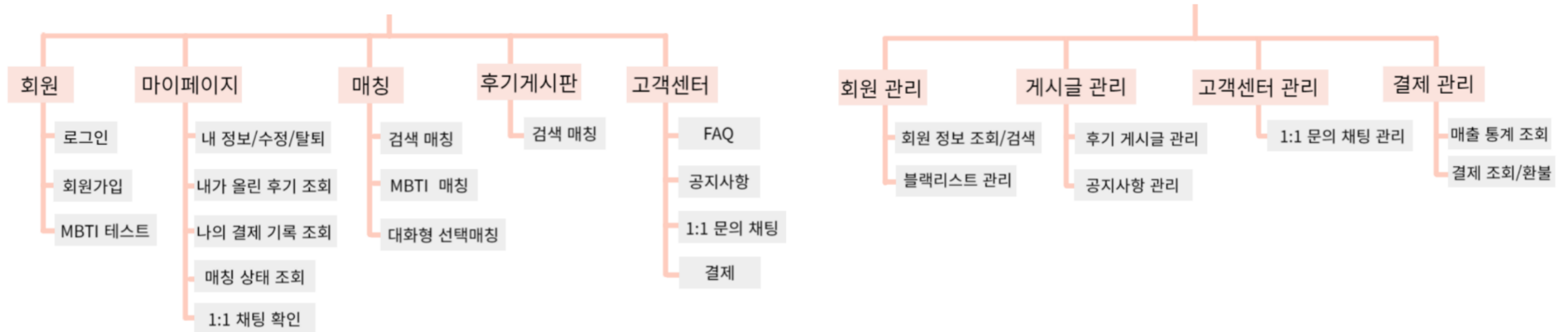
배경

소개팅 사이트는 많지만, 최근 현대인의 첫 인상을 좌우하는 근간이 된 MBTI를 바탕으로 소개팅을 진행하는 사이트를 만들고자 함.

사용자 입장

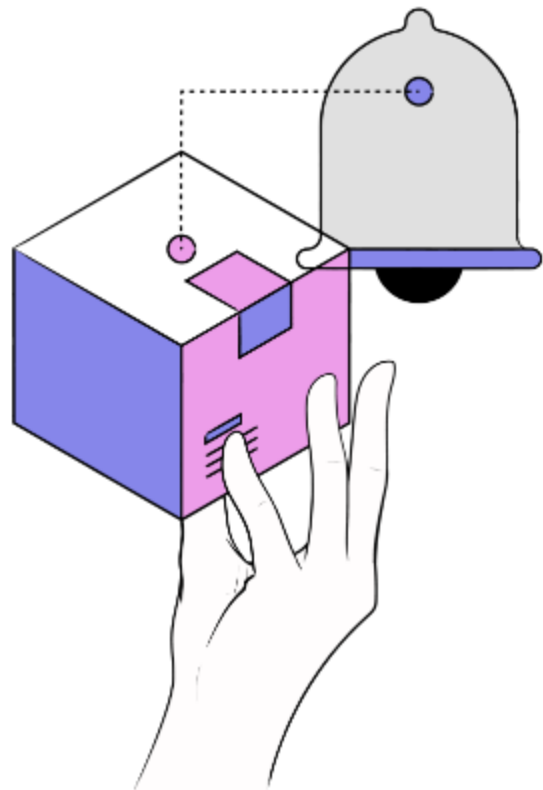
기간 : 2023년 11월 28일 ~ 2023년 12월 20일

관리자 입장



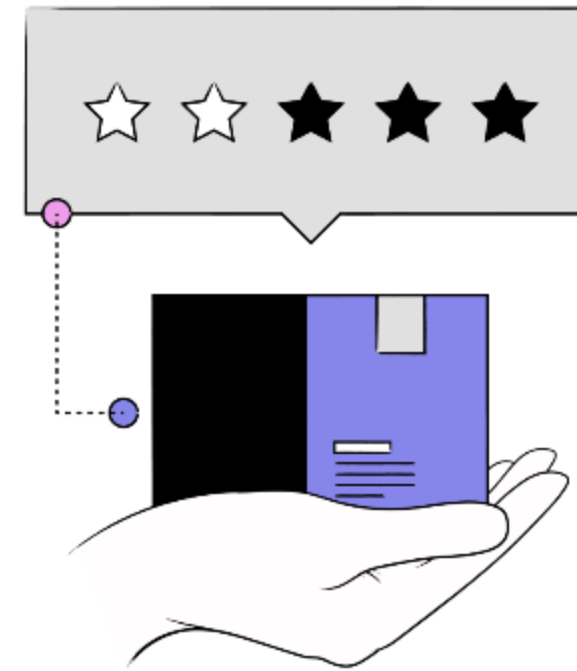
02. 자신이 맡은 파트 명시

전체 파트



- 관리자
- 회원 (회원 가입, 로그인, 마이페이지)
- 매칭 (MBTI 설문지, 결과에 따른 매칭)
- 로그인 (카카오 로그인, 일반 로그인)
- 후기게시판
- 고객센터 (결제 API)
- 채팅 (1:1 채팅, 대화 가이드 기능)
- 메인화면

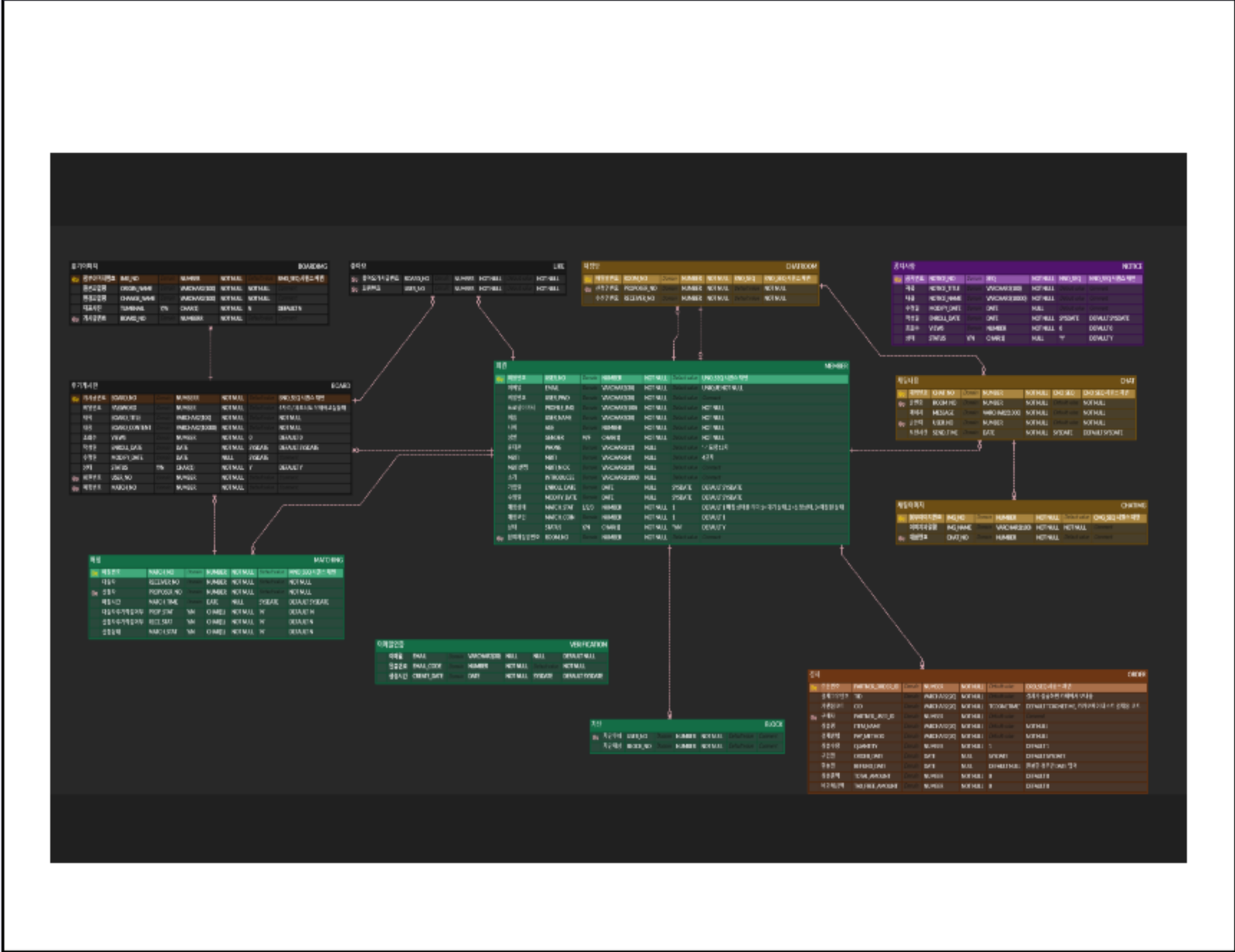
내가 맡은 파트



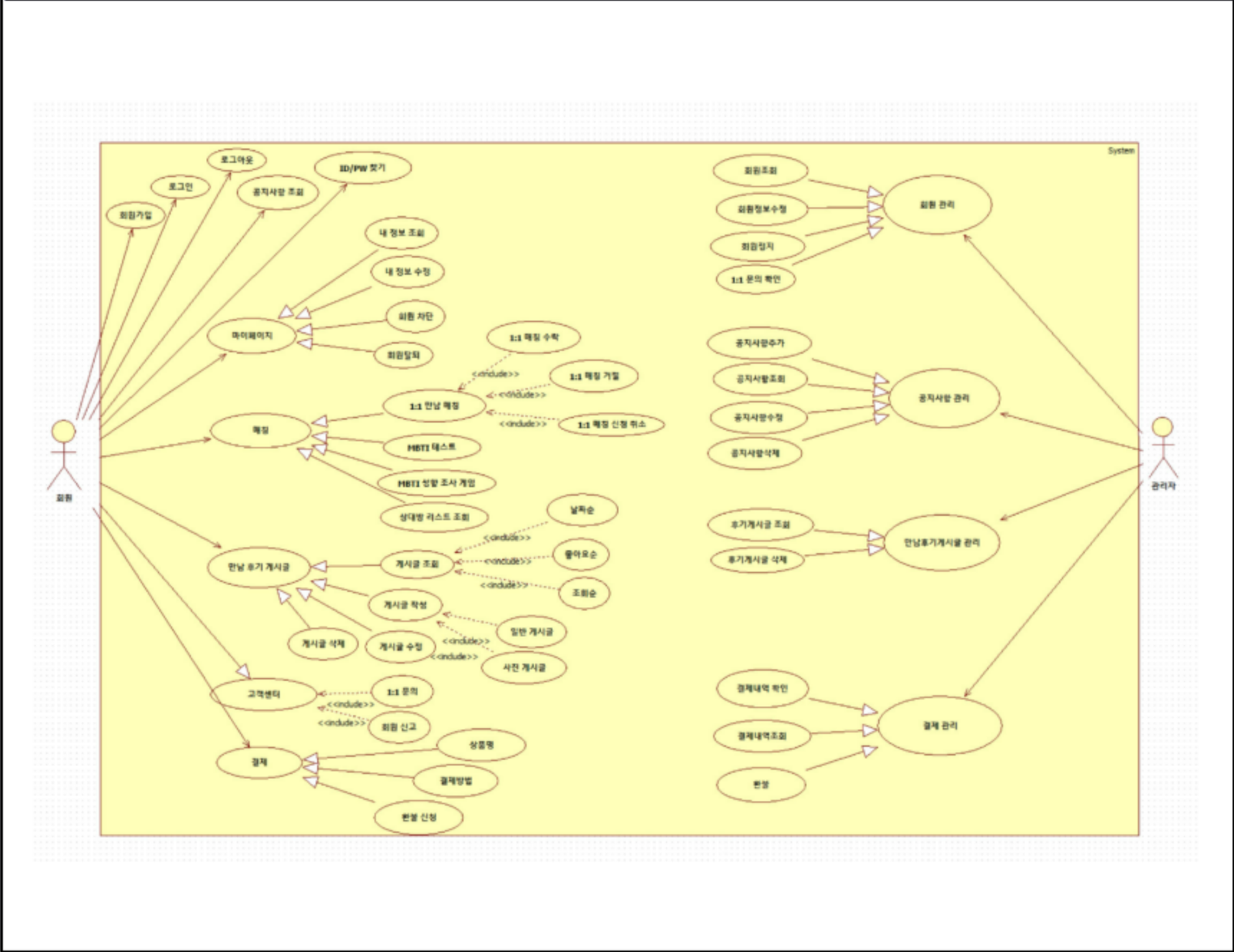
- 메인 화면
 - 가장 신청을 많이 받은 회원
Best 4 추천
 - 내 이상형 기반 추천
 - 채팅 API
- 카카오 로그인 (API)
- 소켓 채팅 기능 (1:1 채팅)
 - 채팅 가이드 기능
 - 원하는 사용자와의 1:1 채팅

03. ERD , UML

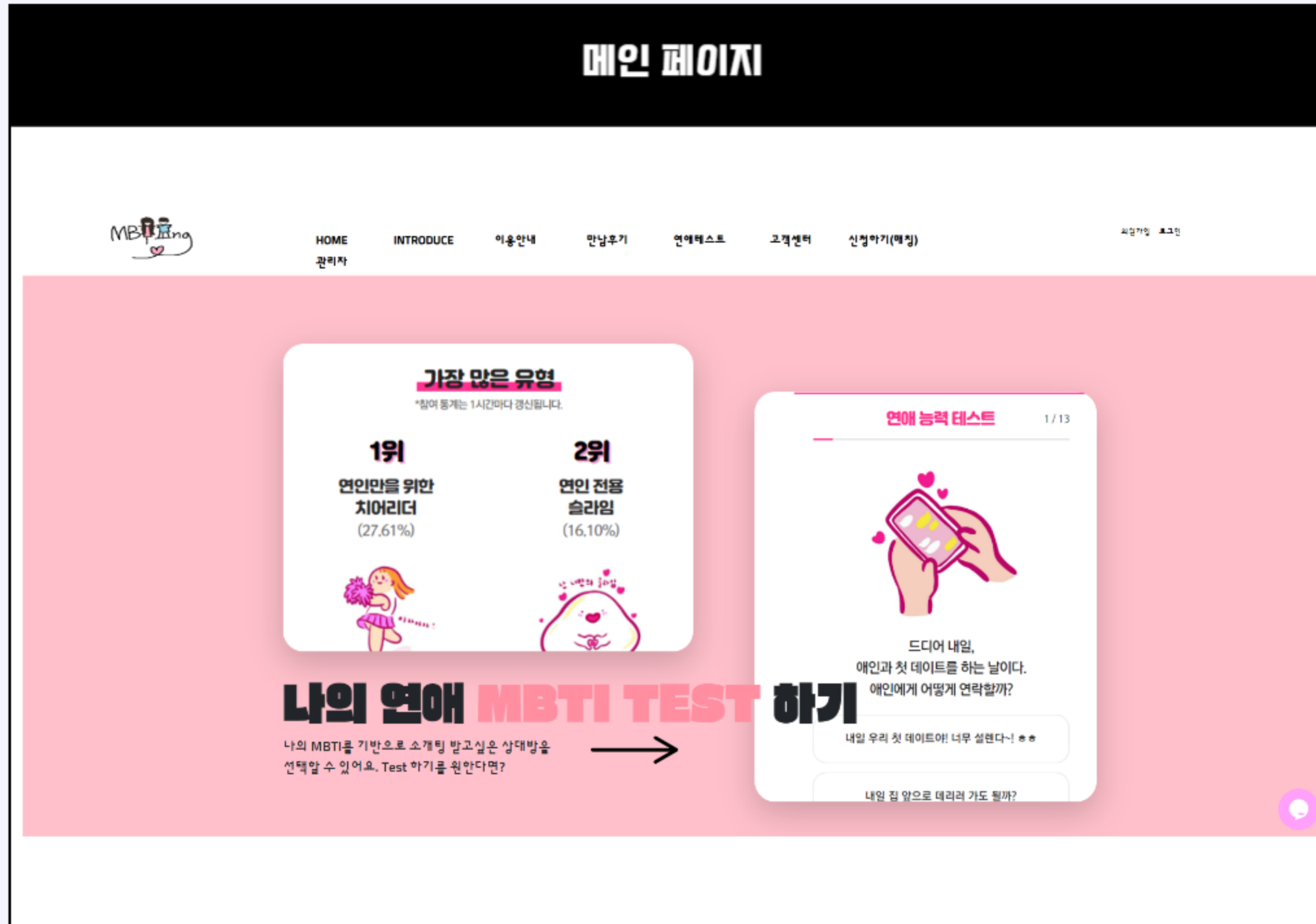
ERD



| |
|-----|
| UML |
|-----|



04. 기능 구현 및 설명

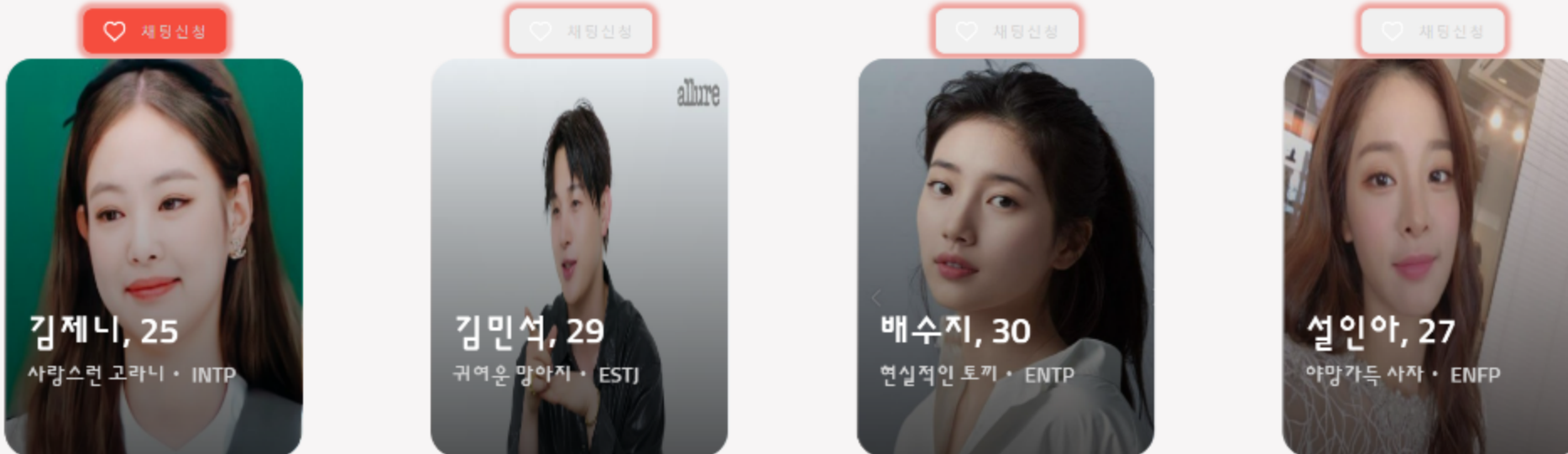


1. 메뉴바는 마우스를 올리면 클릭하고자 하는 글씨만 변하도록 지정함.
2. MBTI TEST하기를 누르면 뒷 배경에 나오는 연애 테스트로 넘어가도록 a 태그 설정
3. 화면은 자동으로 넘어가도록 swipe API를 사용함

04. 기능 구현 및 설명

메인 페이지

현재 가장 매칭 신청을 많이 받은 Top 4



```
<!-- top 회원 정보를 불러오기 위한 쿼리문 -->
<select id="selectTopMemberList" resultMap="memberResultSet" >
  SELECT ROWNUM, A.*
  FROM (
    SELECT profile_img, age, mbti, user_name, mbti_nick, email
    FROM (
      SELECT M.*, RANK() OVER (ORDER BY COALESCE(matching_count, 0) DESC) AS rnk
      FROM MEMBER M
      LEFT JOIN (
        SELECT RECEIVER_NO, COUNT(*) AS matching_count
        FROM MATCHING
        GROUP BY RECEIVER_NO
      ) C ON M.user_no = C.RECEIVER_NO
      WHERE STATUS = 'Y'
      AND MBTI is not null
    )
    WHERE rnk BETWEEN 1 AND 4
  ) A
  WHERE ROWNUM BETWEEN 1 AND 4
</select>
```

1. match 테이블을 이용해서 가장 많은 매칭 신청을 받은 회원을 대상으로 main에 랜덤 추천함.
=> 딱 4명만 선정해야 하기 때문에 rank()를 사용하고, rownum을 사용하기 위해 서브쿼리를 이용.
2. 채팅 신청을 클릭하면 모달 창으로 회원에게 신청된 상태를 알리고 수락이 되어야만 채팅이 가능함.

04. 기능 구현 및 설명

메인 페이지

인기 게시물



1. 후기 게시판 중에서도 가장 많은 좋아요를 받은 회원을 대상으로 메인에 top 5를 지정해서 보여주도록함.

2. 실제 DB에 있는 후기 내용을 노출 시키고, 대표 사진을 폴라로이드 형식으로 지정해서 나열함.

```

<!-- mouse wheel을 사용하기 위한 script 코드 입니당. -->
<script>
    window.addEventListener("wheel", function(e){
        e.preventDefault();
    },{passive : false});

    var mHtml = $("html");
    var page = 1;

    mHtml.animate({scrollTop : 0},10);

    $(window).on("wheel", function(e) {
        if(mHtml.is(":animated")) return;
        if(e.originalEvent.deltaY > 0) {
            if(page == 4) return;
            page++;
        } else if(e.originalEvent.deltaY < 0) {
            if(page == 1) return;
            page--;
        }
        var posTop =(page-1) * $(window).height();
        mHtml.animate({scrollTop : posTop});
    })
</script>

```

=> 모든 화면은 마우스 휠에
맞게 화면이 짝 차도록
width와 height를
100% 주고 jquery를
사용하여 동적인 움직임.

04. 기능 구현 및 설명

카카오 로그인

김민제님 환영합니다



마이페이지 로그아웃

MBKing

user04@naver.com

.....

로그인

또는

kakao로 로그인

비밀번호를 잊으셨나요?

계정이 없으신가요? 가입하기

kakao

tnsenqn1118@naver.com

TIP 카카오계정에 있다면 메일 아이디만 입력해 보세요.

.....

☐ 간편로그인 정보 저장

로그인

또는

QR코드 로그인

회원가입

계정 찾기 비밀번호 찾기

이용약관 개인정보처리방침 운영정책 고객센터 공지사항 한국어

Copyright © Kakao Corp. All rights reserved.

```
//카카오 로그인 후 받은 토큰을 받아내기 위한 service
@Override
public HashMap<String, Object> getUserInfo(String access_token) {

    //카카오의 정보를 받을 HashMap을 설정해줌.
    HashMap<String, Object> userInfo = new HashMap<String, Object>();
    String reqURL = "https://kapi.kakao.com/v2/user/me";

    try {
        URL url = new URL(reqURL);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setRequestMethod("POST");

        // 이걸 kakao 공식 문서에 있는 것과 공한을 받아내기 위한 공식적인 문구임!!!
        conn.setRequestProperty("Authorization", "Bearer " + access_token);

        int responseCode = conn.getResponseCode();
        System.out.println("responseCode : " + responseCode);

        BufferedReader br = new BufferedReader(new InputStreamReader(conn.getInputStream()));

        String line = "";
        String result = "";

        while ((line = br.readLine()) != null) {
            result += line;
        }
        System.out.println("response body : " + result);

        JsonParser parser = new JsonParser();
        JsonElement element = parser.parse(result);

        JsonObject properties = element.getAsJsonObject().get("properties").getAsJsonObject();
        JsonObject kakao_account = element.getAsJsonObject().get("kakao_account").getAsJsonObject();

        String nickname = properties.getAsJsonObject().get("nickname").getAsString();
        String profile_image = properties.getAsJsonObject().get("profile_image").getAsString();
        System.out.println("nickname : " + nickname);

        String email = kakao_account.getAsJsonObject().get("email").getAsString();

        System.out.println("email : " + email);

        userInfo.put("nickname", nickname);
        userInfo.put("profile_image", profile_image);
        userInfo.put("email", email);

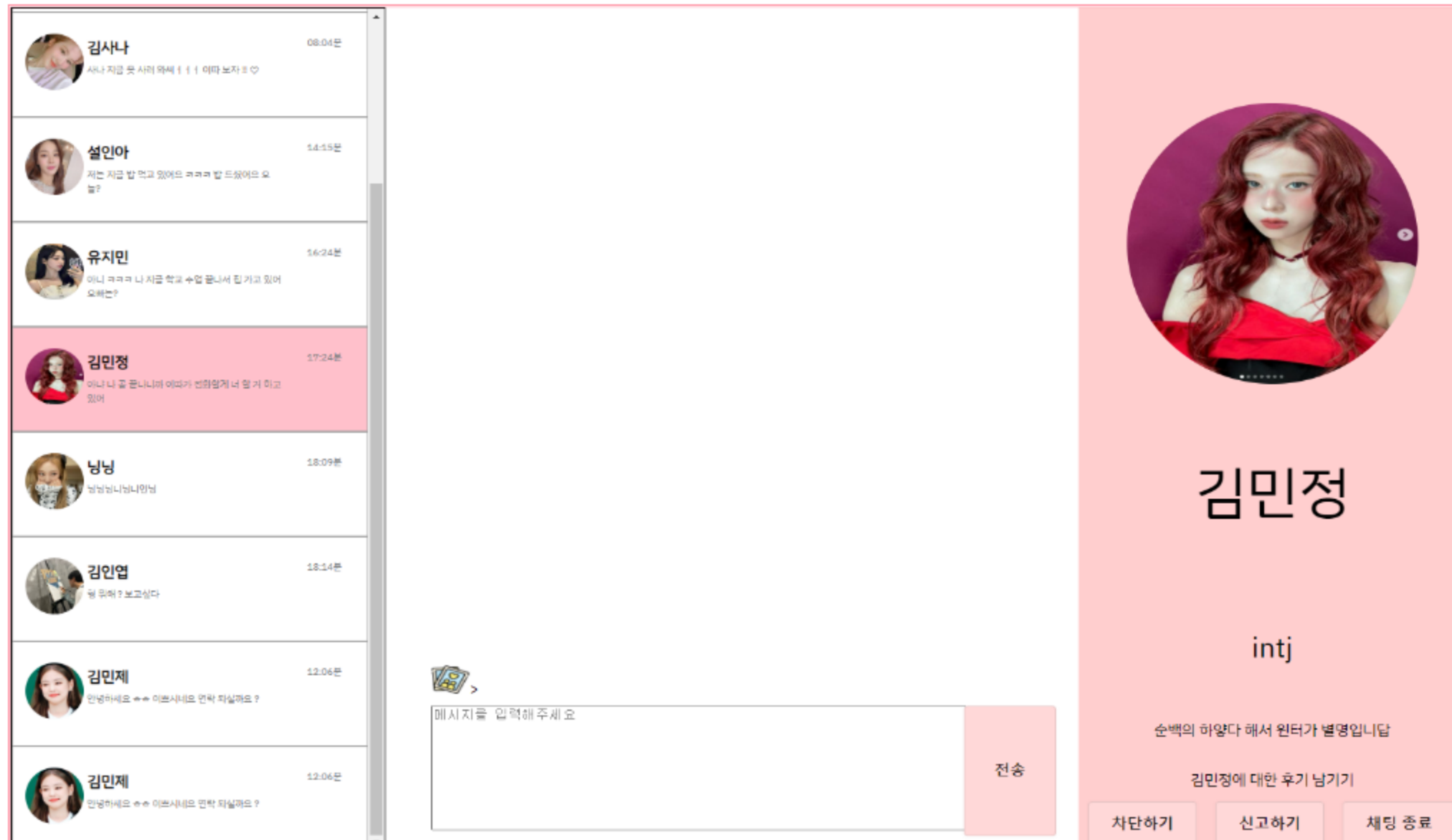
    } catch (IOException e) {
        e.printStackTrace();
    }

    return userInfo;
}
```

1. 카카오로 로그인 하기를 누를 시에 카카오로부터 토큰을 받아서 연결함.
2. 받은 토큰을 통해서 사용자의 정보를 가져오는 위의 코드 내에서 사용자의 정보를 JSON 객체를 사용하여 properties와 kakao_account라는 키값에서 받아옴.
3. 그 정보에 맞게 로그인을 할 시에 사용자의 프로필 사진과 카카오에 지정된 본인의 이름을 화면에 띄워주도록 함.

04. 기능 구현 및 설명

카카오 로그인



```
function getRoomList() {
}

$.ajax({
  url: "chatRoomList.do",
  data: {
    userEmail: "${sessionScope.loginMember.email}"
  },
  dataType: "json",
  async: false, // async : false를 줌으로써 비동기를 동기화 처리 할 수 있다.
  success: function(data) {

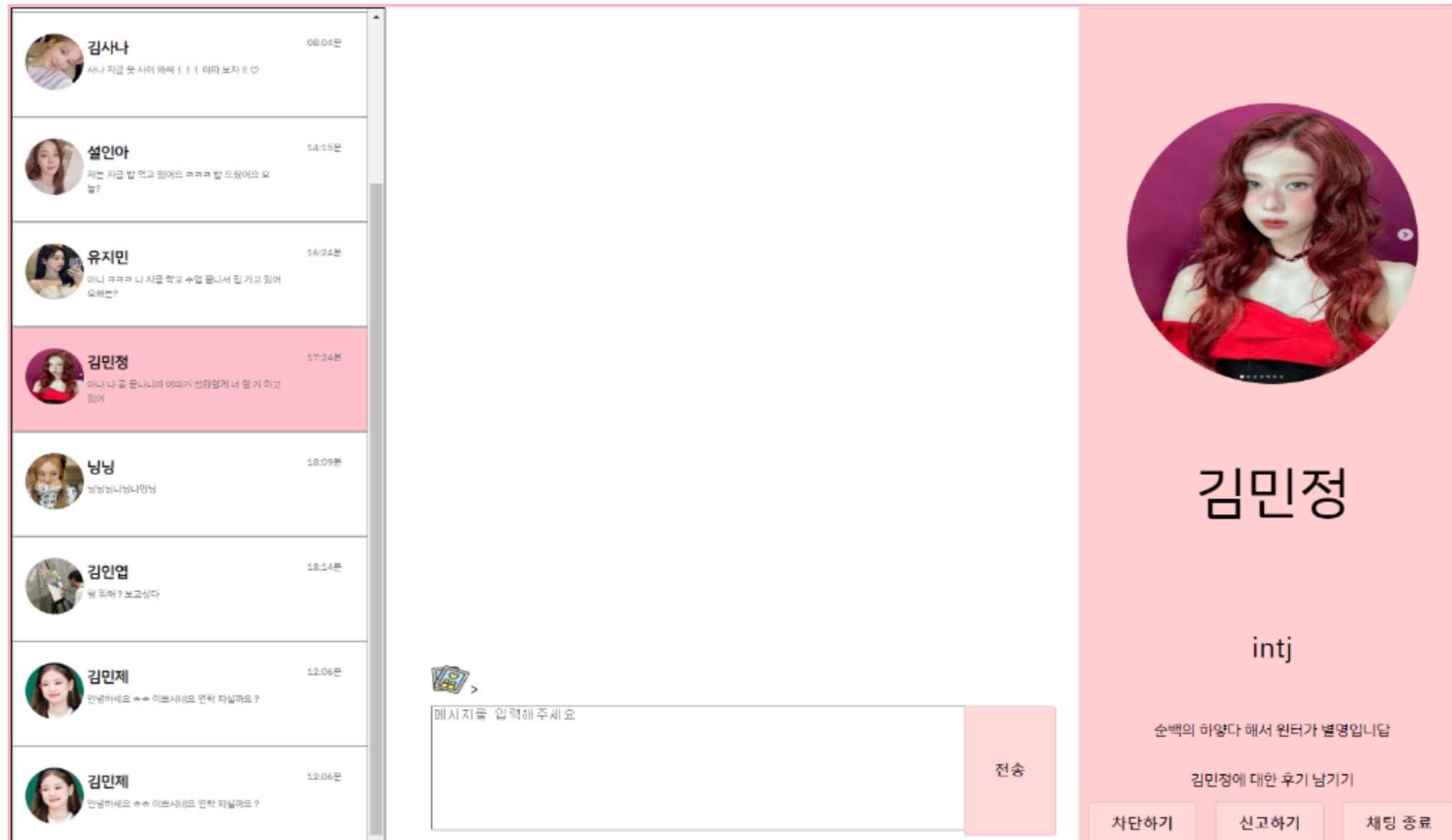
    for(var i in data) {
      //로그인한 상태인 내가 채팅을 건 상태의 채팅 내역이면
      if(data[i].userEmail == "${sessionScope.loginMember.email}") {

      }
    }
  }
});
```

1. 지금까지 채팅을 한 회원의 정보를 불러와서 왼쪽에 list로 추가하도록함.
data로 로그인한 회원의 email을 보내서 기록을 가져오도록 함.

04. 기능 구현 및 설명

카카오 로그인



```
<script>

let roomNo = "${requestScope.roomNo}";
//console.log(roomNo);

let socket;
//연결 실행 시 실행될 함수

$(function(){
    let url = "ws://localhost:8081/mbtting/chat.do";

    socket = new WebSocket(url);

    //연결 성공 시 실행될 함수 onopen
    socket.onopen = function() {
        const data = {
            "roomNo" : roomNo,
            "name" : "${ loginMember.userName }",
            "email" : "${ loginMember.email }",
            "message" : "ENTER-CHAT"
        };
        let jsonData = JSON.stringify(data);
        socket.send(jsonData);
    };

    //연결 종료 시 실행될 함수 onclose
    socket.onclose = function() {
        console.log("서버와 연결이 종료");
    };

    //연결 오류 발생 시 실행될 함수
    socket.onerror = function() {
        console.log("서버 오류남");
    };

    socket.onmessage = function(e) {
        let div = $("

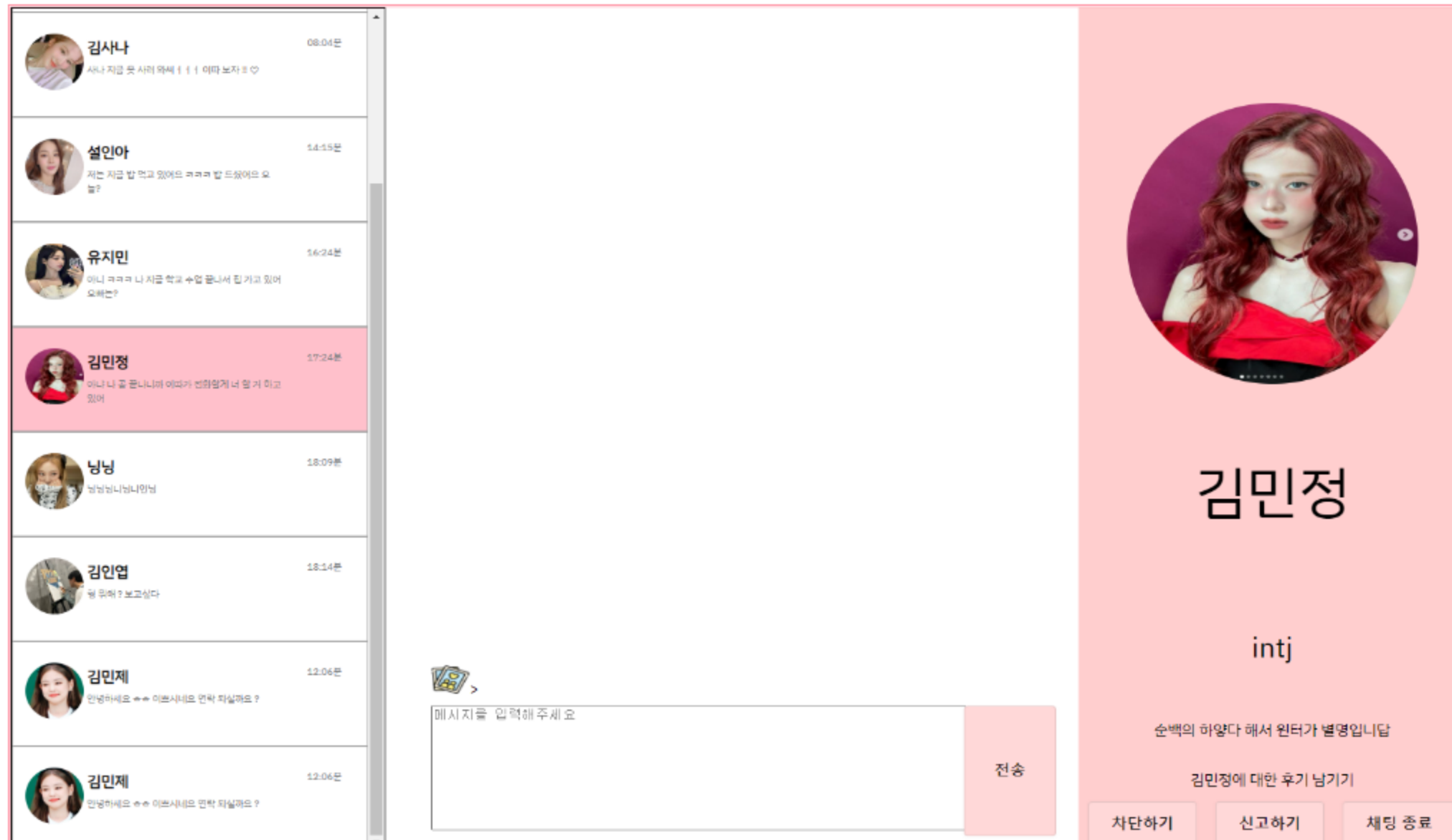
" + e.data + "</div>");
        console.log(div);
        $("#message_wrap1").append(div);
    };
});


```

1. 채팅 시에는 socket이 실행되도록 하고, 방번호는 전역 변수로 지정함. 방번호, 이름, 이메일, 메시지를 넘기면서, DB에는 실제 사용자의 메시지가 실시간으로 insert되도록 함.

04. 기능 구현 및 설명

카카오 로그인



```
// * 2-1 추가 된 것이 내가 보낸 것인지, 상대방이 보낸 것인지 확인하기
function CheckLR(data) {
  // email이 loginSession의 email과 다르면 왼쪽, 같으면 오른쪽
  const LR = (data.email !== `${sessionScope.loginMember.email}`) ? "Left" : "Right";
  // 메시지 추가
  //console.log(LR);
  appendMessageTag(LR, data.email, data.message, data.name);
}

// * 3 메시지 태그 append
function appendMessageTag(LR_className, email, message, name) {

  const chatLi = createMessageTag(LR_className, email, message, name);

  //console.log(chatLi);
  $('div.chatDiv').append(chatLi);

  // 스크롤바 아래 고정
  $('div.chatDiv').scrollTop($('div.chatDiv').prop('scrollHeight'));
}

// * 4 메시지 태그 생성
function createMessageTag(LR_className, email, message, name) {

  //console.log(LR_className + email + message + name);
  // 형식 가져오기
  let chatLi = $('div.chatDiv ul li').clone();

  console.log(chatLi);

  chatLi.find('.chat').addClass(LR_className); // left : right 클래스 추가
  // find() : chatLi의 하위 요소 찾기
  chatLi.find('.sender div').text(name); // 이름 추가
  chatLi.find('.chat p').text(message); // 메시지 추가
  chatLi.find('.chat p').addClass("message");
  chatLi.find('.sender div').addClass(LR_className);

  console.log(chatLi);
  return chatLi;
};
</script>
```

1. 화면에는 실시간으로 내가 보낸 메시지라면 오른쪽, 상대방이라면 왼쪽으로 붙여주기 위해서 data를 검사하는 것을 사용함.

05. 사용 기술 및 개발환경

사용 기술 및 개발환경

[사용기술 및 개발환경]
운영체제
: Windows OS

사용 언어

: Front - javascript, JQuery, HTML5, CSS3, AJAX
: Back JAVA 11, Spring Regacy, JSP

FrameWork/ Library
: jquery, Bootstrap4

DB
: Oracle 11g EE

Tool
: STS 4.23.0, Visual Studio Code, Oracle SQL Developer 23.1.0,
Kakao Oven, Star UML 5.0, ERD Cloud , Pigma

WAS
: Apache Tomcat 9.0.80

Collaboration
: Slack

API
: twak.to 채팅 API, Swiper API, kakap API

06. 팀 내 기여도

| 전체 파트 | 내가 맡은 파트 |
|--|--|
| <ul style="list-style-type: none"> • 관리자 • 회원 (회원 가입, 로그인, 마이페이지) • 매칭 (MBTI 설문지, 결과에 따른 매칭) • 로그인 (카카오 로그인, 일반 로그인) • 후기게시판 • 고객센터 (결제 API) • 채팅 (1:1 채팅, 대화 가이드 기능) • 메인화면 | <ul style="list-style-type: none"> • 메인 화면 <ul style="list-style-type: none"> - 가장 신청을 많이 받은 회원 Best 4 추천 - 내 이상형 기반 추천 - 채팅 API • 카카오 로그인 (API) • 소켓 채팅 기능 (1:1 채팅) <ul style="list-style-type: none"> - 채팅 가이드 기능 - 원하는 사용자와의 1:1 채팅 |

**=> 기준으로 저희 사이트의 정체성이 되는
채팅 기능을 맡고, 메인 화면을 맡아서
30%는 된다고 생각합니다.**

NBTIng



Thank you

